www.zte.com.cn/magazine/English

ISSN 1673-5188
CODEN ZCTOAK

ZTE COMMUNICATIONS

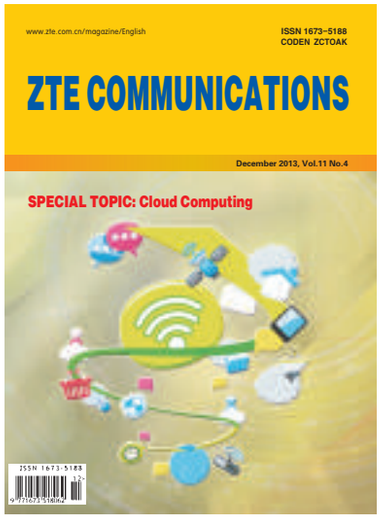
December 2013, Vol.11 No.4

SPECIAL TOPIC: Cloud Computing

ISSN 1673-5188
12>
9 771673 518062

ZTE中兴 ZTE COMMUNICATIONS
VOLUME 11 NUMBER 4 DECEMBER 2013

CONTENTS



Special Topic

Cloud Computing

CONTENTS

ZTE COMMUNICATIONS
Vol. 11 No.4 (Issue 40)
Quarterly
First English Issue Published in 2003

Supervised by:
Anhui Science and Technology Department

Sponsored by:
ZTE Corporation and Anhui Science and Technology Information Research Institute

Editorial Correspondence:
Add: 12F Kaixuan Building,
329 Jinzhai Road,
HeFei 230061, P. R. China
Tel: +86-551-65533356
Fax: +86-551-65850139
Email: magazine@zte.com.cn

Published and Circulated (Home and Abroad) by:
Editorial Office of
ZTE Communications

Printed by:
Hefei Zhongjian Color Printing Company

Publication Date:
December 25, 2013

Publication Licenses:

ISSN 1673-5188
CN 34-1294/ TN

Advertising License:
皖合工商广字0058号

Annual Subscription Rate:
RMB 80

Research Papers

Roundup

Cloud Computing
Hong Cai

Cloud Computing

▶ Hong Cai

Dr. Hong Cai is CTO of Cloud Computing at ZTE Cloud & IT Business Unit. He works with different teams to define the roadmap for the ZTE Cloud & IT product family. Before joining ZTE in 2012, he worked at IBM for nearly 15 years. There, he focused on cloud computing and service computing. He is also very active in academia, serving on the editorial boards of IJWSR, IJCAC and IEEE Transactions on Service Computing. He has served approximately 20 IEEE international conferences as workshop chair or program committee member. He has published more than 50 journal or conference papers and co-authored the book Services Computing. He has filed more than 10 patents and given numerous keynote speeches and public talks at conferences and universities. In 1997, he received his PhD degree from Tsinghua University, China.

In the last five years, great progress has been made in cloud computing, especially in virtualization, standardization, and automation. This has resulted in numerous cloud services, such as Infrastructure as a Service, Platform as a Service, and Software as a Service. Many cloud technologies have matured and have been commercialized. However, issues such as information security, mobility, energy efficiency, and infrastructure optimization are becoming more serious. The key causes of these issues are increased scale of data centers, convergence of IT and CT, increased user concern about information security, and increased opex instead of capex.

For this special issue of *ZTE Communications*, researchers from industry and academia were called to submit articles detailing the latest progress on cloud computing.

The first paper, "Software-Defined Data Center," by Ali et al., gives an overview of key technologies and standardization of SDDC as well as challenges associated with it. The paper points out that a unified control plane allows rich resource abstractions to enable orchestration purpose fit systems and/or providing programmable infrastructures to enable dynamic optimization in response to business requirements."

In their paper "Computation Partitioning in Mobile Cloud Computing: A Survey," Yang et al. address the issue of computation partitioning in mobile cloud. This involves partitioning the execution of applications between the mobile side and cloud side so that the execution cost is minimized. The authors survey computation partitioning in mobile cloud computing.

In the paper "MapReduce in the Cloud: Data Location Aware VM Scheduling," Nguyen el al. see the challenge of MapReduce efficiency in the cloud and develop a distributed cache system and virtual machine scheduler. They show that their prototype can improve performance significantly when running different applications.

The paper "Preventing Data Leakage in a Cloud Environment," by Cang et al., deals with the customer information security and avoidance of unauthorized data access in practical multiparty clouds. The authors survey techniques for preventing data leakages, and these techniques can be used in three trust models.

In the next paper, "CPPL: A New Chunk-Based Proportional-Power Layout with Fast Recovery," by Yin et al, the authors suggest that the size and number of data centers and cloud storage systems are dramatically increasing, and this, in turn, is dramatically increasing energy consumption and disk failures in emerging facilities. The authors propose a new chunk-based power-proportional layout called CPPL to address these problems.

In the last paper, "Virtualization of Network and Service Functions: Impact on ICT Transformation and Standardization," Khasnabish et al. review trends in the virtualization of network/service functions. They also discuss standardization of and required management and orchestration of these functions.

In this special issue, it is our intention to inform the reader of state-of-the-art research and technology on current cloud computing topics and bring to attention of the cloud computing community problems that must be investigated.

This special issue would not be possible without the help provided by many. We would like to thank all the authors for their contributions and all reviewers for their efforts and dedication. We also want to thank the editorial office of *ZTE Communications* for their support.

DOI: 10.3939/j.issn.1673-5188.2013.04.001
http://www.cnki.net/kcms/detail/34.1294.TN.20131205.1524.001.html, published online December 5, 2013

Software-Defined Data Center

Ghazanfar Ali[1], Jie Hu[1], and Bhumip Khasnabish[2]
(1. ZTE Corporation, Nanjing 210012, China;
2. ZTE USA, Morristown, NJ 07960, USA)

Abstract

Defining a software-defined data center is a vision of the future. An SDDC brings together software-defined compute, software-defined network, software-defined storage, software-defined hypervisor, software-defined availability, and software-defined security. It also unifies the control planes of each individual software-defined component. A unified control plane enables rich resource abstractions for purpose-fit orchestration systems and/or programmable infrastructures. This enables dynamic optimization according to business requirements.

Keywords

cloud computing; virtualization; security; software-defined; data center

1 Software-Defined Data Center Architecture

A software-defined data center (SDDC) architecture defines data center resources in terms of software. Specifically, it releases compute, network, storage, hypervisor, availability, and security from hardware limitations and increases service agility. This can be considered an evolution from server virtualization to complete virtualization of the data center.

1.1 Software-Defined Compute

Software-defined compute (SDC), also called server virtualization, releases CPU and memory from the limitations of underlying physical hardware. As a standard infrastructure technology, server virtualization is the basis of the SDDC, which extends the same principles to all infrastructure services.

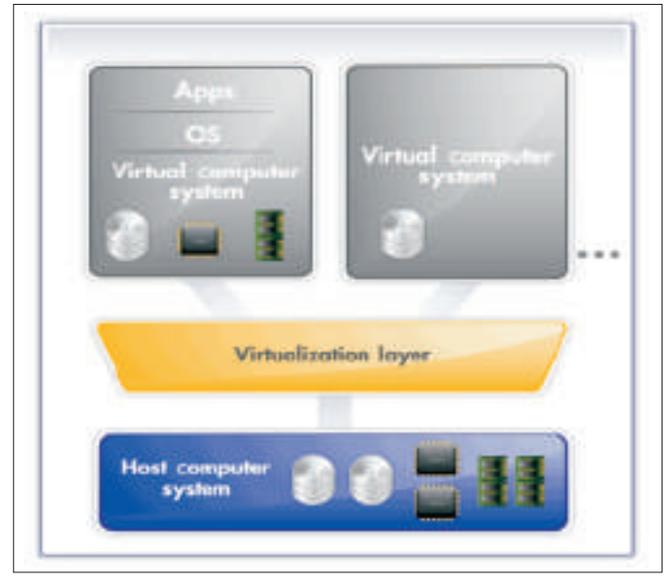
The basic elements of a virtualized environment [1] are shown in **Fig. 1**. The resources that comprise this environment are typically provided by one or more host computer systems. A virtualization layer (typically firmware or software, but sometimes hardware) manages the lifecycle of a virtual computer system, which comprises virtual resources that are allocated or assigned to it from the physical host computer system. A virtual computer system may be active and run an operating system and applications with a full complement of defined, allocated virtual devices.

The virtual computer system may also be inactive with no software running and only a subset of the virtual devices actually allocated. In this environment, a primary responsibility of the administrator is to manage the operational lifecycle of these virtual systems.

Resources of the virtual computer system may have properties or qualities that are different to those of the underlying physical resources. For example, virtual resources may have different capacities or QoS (for performance or reliability) than the underlying physical resources. Managing relationships between virtual and physical resources makes administration tasks in a virtualized environment more complex.

1.2 Software-Defined Network

In a software-defined network (SDN), the network control



▲Figure 1. Elements of virtualized system management.

plane is moved from the switch to the software running on a server. This improves programmability, efficiency, and extensibility.

There has been much technical development and implementation of SDN. This paper does not delve into the details of this vibrant software-defined component.

1.3 Software-Defined Storage

Software-defined storage (SDS) is an ecosystem of products that decouple software from underlying storage network hardware and places it in a centralized controller or hypervisor. This centralized software makes visible all physical and virtual resources and enables programmability and automated provisioning based on consumption or need. The software can live on a server or be part of an operating system or hypervisor, but it is no longer firmware on a hardware device. The software can also control hardware from multiple vendors and enable engineers to build non-proprietary environments [2].

In other words, SDS separates the control plane from the data plane and dynamically leverages heterogeneity of storage to respond to changing workload demands. The SDS enables the publishing of storage service catalogs and enables resources to be provisioned on-demand and consumed according to policy. The characteristics of software-defined storage could include any or all of the following [3], [4]:

- pooling and abstraction of the logical storage services and capabilities from the underlying physical storage systems. This is reflected in the formerly used term, storage virtualization.
- automation with policy-driven storage provisioning. This requires management interfaces that span traditional storage-array products, and it requires defining the separation of the control plane from the data plane (in the spirit of OpenFlow). This issue is not new. Prior industry standardization efforts, such as SMI-S, began in 2002.
- virtual volumes for better performance and optimized data management. This is not a new capability for virtual infrastructure administrators (it is already possible using NFS), but it does give arrays using iSCSI or Fibrechannel a path to higher administrator leverage for cross-array management apps that are written to the virtual infrastructure.
- commodity hardware with storage logic abstracted into a software layer. This is conventionally described as a clustered file system for converged storage.
- scaled-out storage architecture.

VMWare defines software-defined storage as a fundamental component of the SDDC. With software-defined storage, resources are abstracted to enable pooling, replication, and on-demand distribution. The result is a storage layer much like that of virtualized compute: aggregated, flexible, efficient, and scalable. The benefits are across-the-board reduction of the cost and complexity of storage infrastructure [5].

International Data Corporation (IDC) defines SDS as any storage software stack that can be installed on commodity resources (e.g. x86 hardware, hypervisors, or cloud) and/or off-the-shelf computing hardware. Furthermore, to conform to this definition, software-based storage stacks should offer a full suite of storage services and integration of the underlying persistent data placement resources so that tenants can move freely between these resources [6].

IDC asserts that SBS platforms offer a compelling proposition for both incumbent and upcoming storage suppliers [7]. It is in the long-term interest of incumbents to change their hardware-centric mind frame and join the ranks of emerging start-ups. This will bring about a paradigm shift. With the proliferation of SDS platforms, the delineation between hardware, software, and cloud storage suppliers will blur and eventually disappear [7].

IDC also observes that the SDS market has picked up steam. Nexenta Systems, based in Santa Clara, CA, raised $24 million in February 2013, to help advance its NexentaStor open storage platform [8]. VMware also made waves that month by snapping up virtual-storage specialist Virsto for an undisclosed amount. In December 2012, storage startup ScaleIO announced that it had raised $12 million to boost its ScaleIO ECS software operations [6].

SDS, on the other hand, was conceived with cloud environments in mind. According to Debbie Moynihan, VP of marketing at InkTank (a storage vendor), "Software-defined storage was designed to scale-out to thousands of nodes and to support multi-petabytes of data, which will be the norm as the amount of stored data continues to grow exponentially and as more and more storage moves to the cloud" [2].

The Storage Networking Industry Association (SNIA) Cloud Data Management Interface (CDMI) standard defines the functional interface that applications use to create, retrieve, update, and delete data elements from the cloud. As part of this interface, the client can discover the capabilities of the cloud storage offering and use the interface to manage containers and the data placed in them. Metadata can also be set on containers and the contained data elements [9].

1.3.1 Storage Virtualization versus SDS

SDS is similar to other software-defined elements, such as SDN, of the data center [10], [11].

In many respects, SDS is more about packaging and how IT users think about and design data centers. Storage has been largely software-defined for more than a decade: the vast majority of storage features have been designed and delivered as software components within a specific storage-optimized environment.

SDS is sometimes referred to as a storage hypervisor, although the two concepts are somewhat different. Both terms are evolving, and vendors use them to describe different aspects of their storage systems.

Now the question arises, "Does SDS enable you to do some-

thing that you cannot do with traditional storage?" For the most part, SDS is an attempt to provide the same functions as those in traditional storage systems. What is different is the abstraction, which provides two key capabilities.

First, the storage control function now can execute on any server hardware. That means a storage system can be built with commodity hardware and using commodity disks. This makes the purchase and implementation of a storage system more "kit-like," but it also means that system implementation and management requires more skill and time. This investment, however, can significantly reduce acquisition costs.

In addition, the storage controller can now be placed anywhere, it does not have to be installed on dedicated hardware. A growing trend is to implement the software storage controller within a virtual server infrastructure and use available compute power from the host or hosts within that infrastructure. This reduces costs further and creates a simpler scaling model. If a virtual storage controller is installed every time a host is added to the infrastructure, storage processing and capacity increases in lockstep with server growth.

In many ways, a storage hypervisor is part of SDS; it is the core element of an entire storage software stack. Again, vendors use the term differently, so its meaning is not standard.

1.3.2 Storage Virtualization and Server Virtualization

Storage vendors are trying to do for storage what server virtualization did for servers. Many of the things vendors are aiming at result in server hypervisors, where one big server is turned into multiple virtual machines. With a storage hypervisor, the opposite is true. Many disparate storage parts are combined it into one pool. The result is similarin terms of efficiency [12].

1.3.3 Open-Source Storage Software

Another emerging theme is open-source storage software. The commoditization of storage hardware and new economic imperative to do more with less has spurred activity in open-source storage in recent months [13].

It is unlikely that open-source storage will transform the storage industry overnight; after all, storage strategy is still dominated by conservative, risk-averse thinking. There is already plenty of momentum in areas where open-source may offer adequate performance and functionality at a much better price than traditional approaches.

Another area of interest is the cloud, where service providers offering storage as a service have turned to open-source storage in order to compete on price with the economies of scale enjoyed by cloud giants, such as Amazon.

This is a particularly active space right now, especially from an object storage perspective. The main area of interest is OpenStack-based efforts from companies such as Rackspace, HP, and Dell. Other companies, such as Basho Technologies and DreamHost spin-off, InkTank (with Ceph), are also lining up open-source object storage stacks that can underpin cost-effective, large-scale cloud storage services and potentially enhance or replace the Swift storage element of OpenStack.

Many other object storage suppliers are considering the open source route, so activity in this area is likely to increase. Open source storage may have its niche in small businesses and service providers, but it has yet to penetrate medium-sized and large enterprises in a meaningful way.

1.4 Software-Defined Security

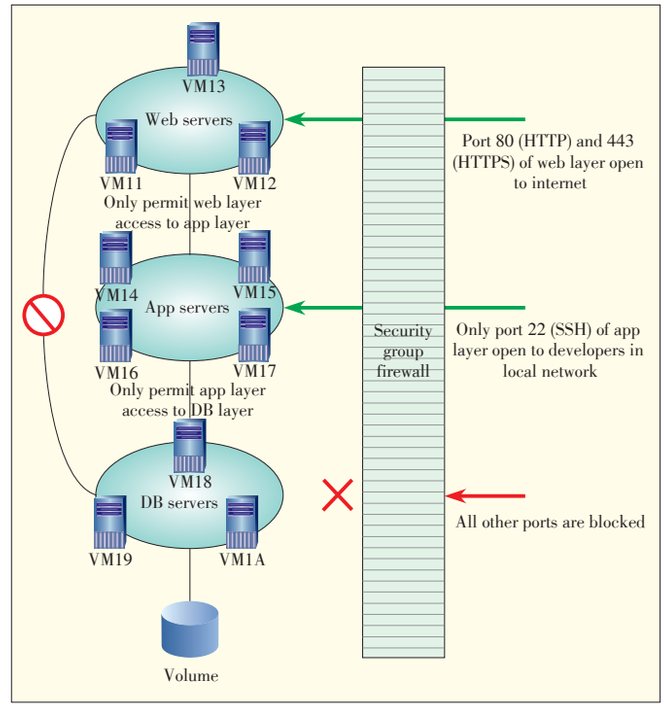
In software-defined security (SDSec), protection is based on logical policies and is not tied to any server or specialized security device. Adaptive, virtualized security is achieved by abstracting and pooling security resources across boundaries so that regardless of where a user resource is located, it can be protected. It is not assumed that the user resource will remain in the same location.

One or more instance/virtual machine, storage volume, etc. can be grouped into a logical resource security group that shares a common set of rules for controlling who can access the instances. This set of rules specifies the protocols, ports, and source IP ranges for traffic filtering.

Fig. 2 shows a basic three-tier web-hosting architecture and describes inbound and outbound traffic control using security groups. Each tier has a different security group.

The web server SG only allows access from hosts over TCP on ports 80 (HTTP) and 443 (HTTPS) and from instances in the app server SG on port 22 (SSH) for direct host management.

The app server SG allows access from the web server SG for



▲**Figure 2. Inbound and outbound traffic control using a security group.**

web requests and from local subnet over TCP on port 22 (SSH) for direct host management. Developers can directly log into the application servers from the local network. The database server SG permits only the app servers SG to access the database servers.

1.5 Software-Defined Hypervisor

In virtualization, a hypervisor is a software program that manages multiple operating systems, or multiple instances of the same operating system, on a single computer system. The hypervisor manages the system's processor, memory, and other resources in order to allocate the resources that each operating system requires. Hypervisors are designed for a particular processor architecture and may also be called virtualization managers.

Software-defined hypervisor provides enables virtualization of the hypervisor layer and decouples it from underlying virtualization management. This enables selective use of other hypervisors, such as Hyper-V, KVM and VSphere, in response to business requirements.

1.6 Software-Defined Availability

Software-defined availability (SDavailability) enables two or more virtual systems to be deployed on different platforms or at two or more locations according to availability or disaster-recovery [14]. **Table 1** gives a granular view of availability.

▼Table 1. Availability attributes

Attribute	Description
availability	The virtual systems should be placed on different virtualization platforms.
availability-geographic	The virtual systems should be placed in different geographical areas.
availability-site	The virtual systems should be placed on different operator sites.
availability-rack	The virtual systems should be placed on different physical racks.
availability-chassis	The virtual systems should be placed on different physical chassis.
availability-host	The virtual systems should be placed on different physical hosts.

In VMWare's vision of SDavailability, the SDDC provides availability for all applications, independent of the platform stack. This enables customers to establish a consistent first line of defense for the customer's entire IT infrastructure. SDavailability can automatically detect and recover from any software or operating system failure that affects virtual appliance [5].

2 DMTF Open SDDC Incubator

The SDDC [15] is an emerging area of technology that could revolutionize the IT infrastructure over the next several years. New technologies such as SDN and SDS have begun appearing on the market. Although there are many management standards for physical, virtual, and cloud-based systems, there are currently no standard architectures or definitions for SDDC. According to Dave Bartoletti of Forrester Research, "At the core of the software-defined datacenter is an abstracted and pooled set of shared resources. But the secret sauce is in the automation that slices up and allocates those shared resources on-demand, without manual tinkering" [16].

To address this demand, DMTF has proposed an Open Software Defined Data Center (OSDDC) incubator that will develop use cases, reference architectures, and requirements based on real-world customer requirements. With these inputs, the incubator will help in the development of a set of white papers and recommendations for industry standardization.

DMTF OSDDC is a pool of compute, network, storage and other resources that can be dynamically discovered, provisioned, and configured according to workload. SDDC provides abstraction that enables policy-driven orchestration of workloads as well as management and measurement of resources consumed. SDDC comprises a set of features, including [17]:

- a pool of compute, network, storage and other resources
- discovery of resource capabilities
- automated provisioning of logical resources based on workload requirements
- management and measurement of resources consumed
- policy-driven orchestration of resources to meet SLOs of the workloads.

3 OpenStack Software-Defined Infrastructure

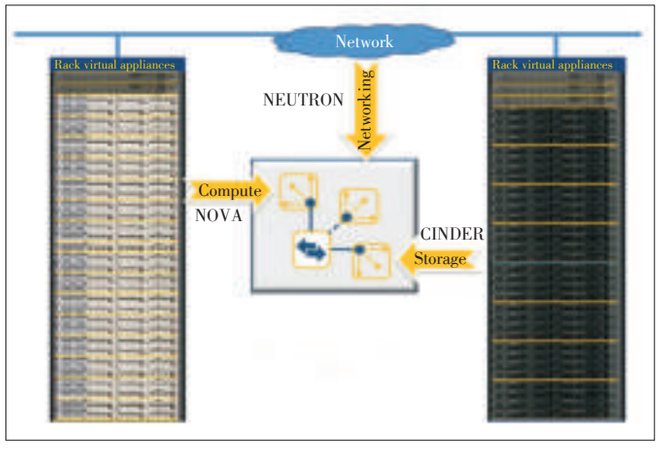
OpenStack is a cloud operating system, which means that it is the software that manages the computer resources in a cloud data center [18].

Currently, there is no data center that is entirely standardized on a single virtualization technology. Looking at compute virtualization alone, one is likely to encounter a mix of PowerVM, z/VM, KVM, VMware ESX server, Microsoft Hyper-V, and perhaps a handful of other technologies in any given data center. How is it possible to create a programming layer across such a diverse set of server virtualization technologies?

An answer to this is OpenStack, which the IT industry has settled on for software-defined infrastructure (SDI). OpenStack SDI spans all compute virtual environments and enables the integration of heterogeneous compute, storage, and network environments into a single, programmable infrastructure to support (rack of) virtual appliances.

To achieve this goal, OpenStack promotes a disaggregated resource model of three independent device controllers (**Fig. 3**) [19]: compute (NOVA), network (NEUTRON), and storage (CINDER).

Nova, also known as OpenStack Compute, is the software



▲Figure 3. OpenStack software-defined infrastructure model.

that controls the IaaS cloud computing platform. It is similar in scope to Amazon EC2 and Rackspace Cloud Servers. Nova does not include any virtualization software; rather, it defines drivers that interact with underlying virtualization mechanisms that run on the host operating system, and it provides functionality over a web API.

Neutron is an OpenStack project designed to provide network connectivity as a service between interface devices (e.g. vNICs) managed by other Openstack services (e.g. NOVA). A neutron server provides a webserver that exposes the Neutron API and passes all web service calls to the Neutron plugin for processing.

CINDER is an OpenStack project intended to provide block storage as a service.

Each device controller comprises the control API, resource allocator, and device manager.

4 OASIS Topology and Orchestration Specification for Cloud Applications

Topology and Orchestration Specification for Cloud Applications (TOSCA) is a proposed OASIS standard for portability of applications/cloud services across diverse cloud infrastructures [20].

TOSCA is intended to be the standard to describe IT services that go beyond IaaS. It is also intended to describe service templates across *aaS layers, which are built on the resource-abstraction layer comprising SDC, SDS, and SDN.

Fig. 4 gives a technical overview of TOSCA and the software-defined component model.

TOSCA defines a metamodel for defining IT services. This metamodel defines both the structure of a service as well as how to manage it. A topology template, also called the topology model, defines the structure of a service. Plans define the process models used to create and terminate a service as well as manage the service during its lifetime.

A topology template comprises a set of node templates and relationship templates. Together, all of these templates define the topology of a service as a directed graph (not necessarily a connected graph). A node in this graph is represented by a node template, which specifies the occurrence of a node type as a component of a service. A node type defines the properties

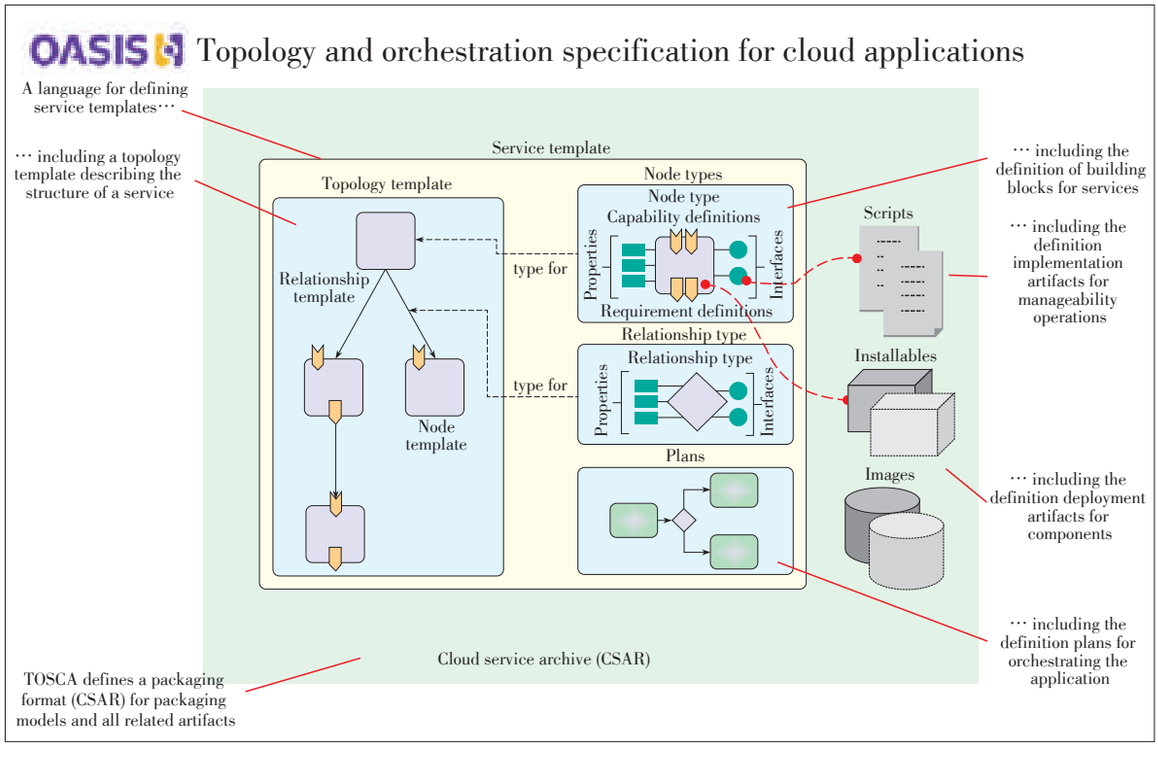


Figure 4. TOSCA technical overview.

of such a component (via node type properties) and the operations (via interfaces) available to manipulate the component.

5 Conclusion

To realize an SDDC, data center resources, such as computer, network, storage, security and availability, are expressed as software. They also need to have certain characteristics, such as multitenancy; rapid resource provisioning; elastic scaling; policy-driven resource management; shared infrastructure; instrumentation; and self service, accounting, and auditing. This ultimately entails a programmable infrastructure. that enables valuable resources to be automatically cataloged, commissioned and decommissioned, repurposed, and repositioned.

References

[1] *Virtualization MANagement (VMAN) technical note* [Online]. Available: http://dmtf.org/sites/default/files/VMAN_Overview%20Document_2010.pdf
[2] *Parsing through the software-defined storage hype* [Online]. Available: http://searchsdn.techtarget.com/tip/Parsing-through-the-software-defined-storage-hype
[3] *The fundamentals of software-defined storage* [Online]. Available: http://san.coraid.com/rs/coraid/images/SB-Coraid_SoftwareDefinedStorage.pdf
[4] Venkatraman Archana. *Software-defined datacentres demystified* [Online]. Available: http://www.computerweekly.com/feature/Software-defined-datacentres-demystified
[5] *The software-defined data center* [Online]. Available: http://www.vmware.com/software-defined-datacenter/storage.html
[6] *IDC defines software defined storage* [Online]. Available: http://www.enterprisestorageforum.com/storage-management/idc-defines-software-defined-storage.html
[7] *IDC brings clarity to software-based/software-defined storage markets* [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=prUS24068713
[8] *Storage virtualisation vs software-defined storage* [Online]. Available: http://www.computerweekly.com/blogs/StorageBuzz/2013/07/storage - virtualisation-vs-soft.html
[9] *The storage networking industry association (SNIA) cloud data management interface (CDMI) Version 1.0.2*[Online]. Available: http://www.snia.org/cdmi
[10] *What is software-defined storage and what can I do with it* [Online]. Available: http://searchstorage.techtarget.com/answer/What-is-software-defined-storage-and-what-can-I-do-with-it
[11] *Software-defined storage: answering frequently asked questions* [Online]. Available: http://searchstorage.techtarget.com/feature/Software-defined-storage-Answering-frequently-asked-questions
[12] *Defining the storage hypervisor: vendors use different methods* [Online]. Available: http://searchvirtualstorage.techtarget.com/podcast/Defining-the-storage-hypervisor-Vendors-use-different-methods
[13] *Software-defined storage: the reality beneath the hype* [Online]. Available: http://www.computerweekly.com/opinion/Software-defined-storage-The-reality-beneath-the-hype
[14] *Open virtualization Format Specification version 2.0.1*, The Distributed Management Task Force (DMTF) DSP0243, 2013.
[15] *Software Defined Data Center (SDDC) Definition White Paper*, The Distributed Management Task Force (DMTF) DSP-IS0501, 2013.
[16] *Forrester blogs: VMware doubles down on heterogeneity with NICIRA acquisition, steers cours to the software defined datacenter* [Online]. Available: http://blogs.forrester.com/dave_bartoletti/12-07-24-vmware_doubles_down_on_heterogeneity_with_nicira_acquisition_steers_course_to_the_software_defined 2013
[17] *DMTF to address need for open software defined data center standards* [Online]. Available: http://dmtf.org/news/pr/2013/5/dmtf-address-need-open-software-defined-data-center-standards
[18] *OpenStack: the open source cloud operating system* [online]. Available: http://www.openstack.org/software/
[19] *Software define storage (SDs) and its application to an openstack software defined infrastructure (SDi) implementation* [Online]. Available: http://mpstor.com/pdf/softwaredefinedinfrastructure09Oct12.pdf
[20] *Organization for the advancement of structured information standards (OASIS) topology and orchestration specification for cloud applications (TOSCA) version 1.0* [Online]. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca

Manuscript received: September 27, 2013

Biographies

Ghazanfar Ali (ghazanfarali@zte.com.cn) received his MS degree in computer science from Quad-e-Azam University, Islamabad. He works as an advanced standards research engineer in the Strategy Planing Department of ZTE Corporation. He represents ZTE Corporation as the vice-chair of the DMTF Cloud Management Subcommittee (CMSC). His has contributed about 400 technical proposals for different technical standards developed in ITU SG13, OMA, and DMTF. His research interests include software-defined compute and virtual appliance.

Jie Hu (hu.jie@zte.com.cn) received his MS degree in computer science from Southeast University, Nanjing. He is the standards director for cloud computing platforms, ZTE Corpration. He has worked as an editor for several standards organizations, including CCSA, OMA, ITU-T on CDN, Mobile Search, and Cloud Computing RA.

Bhumip Khasnabish (b.khasnabish@ieee.org), Phd, AMCPM, is a senior member of the IEEE and an emeritus distinguished lecturer of the IEEE Communications Society. He has initiated cloud and data center activities in the IETF and is vice-chair of DMTF NSM WG (previously co-chaired ATIS IPTV Interoperability Forum (IIF)). He is currently a senior director in the Strategy Planning Department of ZTE TX Inc., USA. His research interests include next-generation networking, platform and services that use virtualized computing and communication entities, tighter cross-layer communications, and automation of system configuration and services. He has worked in the Verizon/GTE next-generation laboratories, Waltham, MA, and in Bell-Northern Research (BNR) Ltd. in Ottawa, Canada. Dr. Khasnabish has published numerous articles, books, and book chapters and has been awarded several patents in his research areas.

DOI: 10.3969/j. issn. 1673-5188. 2013. 04. 002
http://www.cnki.net/kcms/detail/34.1294.TN.20131202.1347.001.html, published online December 2, 2013

Computation Partitioning in Mobile Cloud Computing: A Survey

Lei Yang and Jiannong Cao
(Department of Computing, Hong Kong Polytechnic University, Hong Kong)

Abstract

Mobile devices are increasingly interacting with clouds, and mobile cloud computing has emerged as a new paradigm. An central topic in mobile cloud computing is computation partitioning, which involves partitioning the execution of applications between the mobile side and cloud side so that execution cost is minimized. This paper discusses computation partitioning in mobile cloud computing. We first present the background and system models of mobile cloud computation partitioning systems. We then describe and compare state-of-the-art mobile computation partitioning in terms of application modeling, profiling, optimization, and implementation. We point out the main research issues and directions and summarize our own works.

Keywords

mobile cloud computing; offloading; computation partitioning

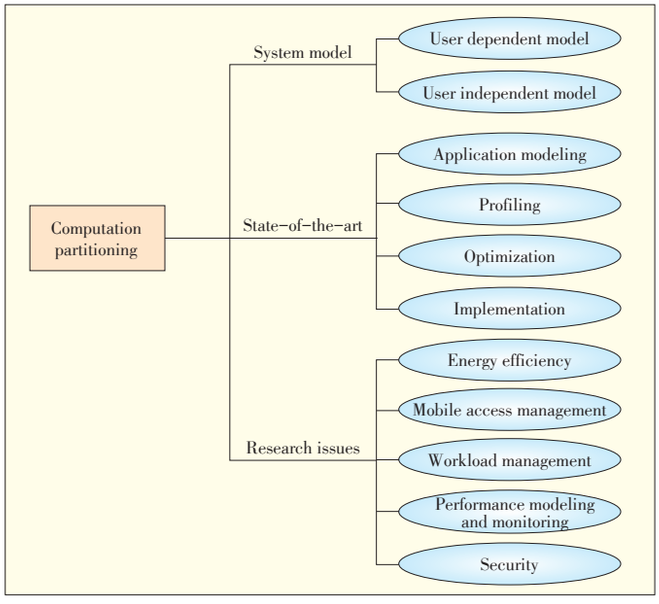
1 Introduction

Cloud computing is an important new paradigm in IT service delivery and has been driven by economies of scale. It enables a shared pool of virtualized, managed, dynamically configurable computing resources to be delivered on demand to customers over the Internet and on other available networks. With the advances in technologies for wireless communication and mobile devices, mobile computing has become integrated into the fabric of everyday life. Increased mobility means that users need to run stand-alone mobile applications and/or access remote mobile applications on their devices.

The application of cloud services to the mobile ecosystem has created a new mobile computing paradigm called mobile cloud computing (MCC). MCC offers great opportunities for the mobile service industry because it allows mobile devices to utilize elastic resources offered by the cloud. There are three MCC approaches: 1) extending cloud service access to mobile devices; 2) enabling mobile devices to work collaboratively as cloud resource providers [1], [2]; and 3) augmenting the execution of mobile applications by using cloud resources (i.e. by offloading selected computing tasks of mobile applications onto the cloud). This allows us to create applications that far exceed what is possible with a traditional mobile device.

Of the three MCC approaches, most of the recent research focuses on the third because it represents the general trend but is more challenging. This paper discusses computational partitioning, an essential problem in the third MCC approach. Computation partitioning involves partitioning the execution of application between the mobile device and cloud so that execution cost is minimized. This cost can be measured in term of completion time, throughput (if the application is to process streaming data), energy consumption, and data transmission over the network. Execution cost is usually created by the application itself, the computing capability of the mobile device, and the bandwidth/quality of connection to the cloud. If the mobile device has high computing capability or the network bandwidth is not fully utilized, we can assign more functions to the mobile side. If the device has poor computing capability but the network bandwidth is good, we can execute more functions at the cloud side.

We present the literature on computation partitioning according to the taxonomy in **Fig. 1**. We divide research on computation partitioning into two categories based on system model. One of these categories is user-independent computation partitioning, where partitioning decisions are independent, and each user can make an optimal partitioning decision. The other category is user-dependent computation partitioning, where user partitioning decisions depend on each other because particular resources that could affect user partitioning are shared. In this model, the allocation of shared resources and user partitioning decisions should be considered jointly. Partitioning decisions are made according to global information of all users, and the aim is to minimize execution cost for all users. We found that the state-of-the-art computation partitioning model



▲**Figure 1. Taxonomy of the survey.**

is the independent model [3]–[11]. In our recent works, we have studied the user–dependent partitioning model [12]. Therefore, in our survey here, we focus on the independent computation partitioning model, and user-dependent computation partitioning is dealt with in the section that summarizes our own work.

Section 2 describes the two system models. Section 3 describes state-of-the-art independent computation partitioning in terms of application modeling, device profiling, network profiling, and implementation approaches. We describe how to represent the application in a way that correctly reflects the properties of the application and makes the partitioning decision more convenient. Device profiling and network profiling involves collecting device and network information that is critical to the partitioning decision. This information includes the device computing capability, device CPU/memory state, and network bandwidth. Optimization involves minimizing the application's execution cost based on the cost model, which measures the cost of partitioning. The cost may be increased completion time, data processing throughput, energy consumption, or a combination of these. The implementation approach is the way in which the tasks of an application are remotely executed at the cloud side in a practical system. A client–server approach, VM migration approach, or agent approach may be taken to remotely execute tasks in the cloud.

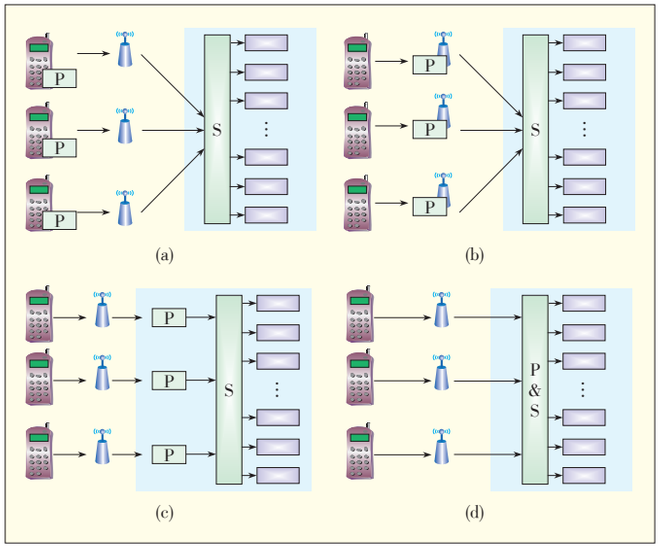
Section 4 describes research issues in computation partitioning. These issues include energy efficiency, mobile access management, workload management, performance modeling and monitoring, and security. In section 5, we present our current work on a) modeling and partitioning data-streaming applications, b) profiling network bandwidth and partitioning an application when bandwidth is fluctuating, and c) user-dependent computation partitioning. The first two areas are studied in terms of the user–independent computation partitioning model. In section 6, we conclude the paper.

2 Background and System Model of Mobile Cloud Computation Partitioning

A mobile cloud system comprises mobile devices, wireless access, and clouds. A mobile device can offload some computation to the cloud, and this can reduce computational cost (e.g. execution time or energy consumption) of the mobile device. However, such of floading creates additional overhead. If we treat the application as a black box that has computational cost when executed locally and data transmission cost when executed remotely, we can decide whether the application should be executed locally or remotely. However, this level of offloading decision-making is too coarse. For complex applications that can be divided into a set of dependable parts, we need to make offloading decisions for all the parts, and the decision made for one part depends on the other parts. Of floading decisions should be optimized for every part of the application: This is computation partitioning.

The partitioning decision depends on device information, network bandwidth, and the application itself. Device information includes the execution speed of the device and the workloads on the device when the application is launched. If the device computes very slowly and the aim is to reduce execution time, it is better to offload more computation to the cloud. Network bandwidth affects data transmission for remote execution. If the bandwidth is high, the cost in terms of data transmission will be low. In this case, it is better to of fload computation to the cloud. Each part of the application requires computation and data transmission if it is offloaded to the cloud. The ratio of the amount of computation (in term of execution instructions) to data transmitted is called the compute-to-communication ratio (CCR). If the CCR is high, the application is computation-intensive, and it is better to execute tasks remotely. If the CCR is low, the application is data-intensive, and it may be better to execute tasks locally. Unlike device and network information, CCR is a static factor that influences the partitioning decision. Different applications usually have different CCRs. Device and network information usually change over time. Collection and estimation of device and network parameters in real time is called profiling and is a challenging issue that will be discussed in the next section. When the device and network parameters are profiled, an optimization problem can be solved and the partitioning decision can be made.

We now discuss the system models for user–independent computation partitioning and user-dependent computation partitioning. In the user–independent model, each user makes their own partitioning decision, but partitioning can be done at various places in the mobile cloud system (e.g. at the device side, access network, or cloud side). **Fig. 2** shows three cases



▲Figure 2. (a), (b) and (c) are user-independent system models, and (d) is the user-dependent model.

for the user-independent model. The blocks marked P indicate the partitioning function. Partitioning can be done at the mobile side (Fig. 2a), cloud side (Fig. 2c), or within the access network (Fig. 2b). The blocks marked S indicate the scheduling of offloaded computation from mobile users to cloud servers. In this model, partitioning and scheduling are decoupled. Each user's partitioning decision is made independently.

In the user-dependent model, partitioning depends on that of other users. This model is suitable when users are competing for shared resources, such as servers at the cloud side. Because of this competition, the optimality of a partitioned execution not only depends on the partitioning itself but also on the availability of the resources (which varies according to other users' partitioning). Thus, we need to consider the profiling information of all users and make partitioning decisions that guarantee optimal average performance for all users rather than optimal performance for each individual user. Fig. 2d shows how partitioning is done at the cloud side with scheduling for all users. The block marked P&S indicates partitioning coupled with scheduling.

The user-independent model is intuitive, and most existing works pertain to this model [3]-[11]. A critical assumption of this model is that resources shared by users are always enough, and the allocation of resources does not influence the execution of application that has been partitioned in advance. It is assumed that the cloud always has enough servers to accommodate the computations offloaded from mobile devices and that execution time is the metric for optimization. Offloaded computation should be executed on the servers without delay; otherwise, the performance of the partitioned execution is sacrificed. This assumption is true when the cloud has nearly unlimited computing resources or the number of mobile users offloading computation to the cloud does not exceed to the cloud's capacity. This model is suitable for a system that serves a small or predictable number of users, and the cloud guarantees optimal partitioning for each individual user.

The user-dependent model applies when the computation loads from mobile users exceed the capacity of the cloud, and users need to compete for resources at the cloud side. In this scenario, instead of optimizing performance for each individual user (as in the user-independent model), the objective is to optimize overall performance. We suggest that the user-dependent model is suitable for use in large-scale system with unpredictable workloads. In the user-dependent model, coupling of partitioning and allocation of shared resources make partitioning more challenging. This model was first proposed in [11].

3 State-of-the-Art Computation Partitioning for Mobile Cloud Computing

3.1 Application Modeling

An application model may be the model used by programmers to develop applications. It may also be the mathematical model that represents the structure of the application. The former provides programming abstractions for application development, and the latter is the formal representation of the application to be partitioned. The latter usually depends on the programming model. Thus, we describe the application model from the perspective of programming. In our survey, three application models are: procedure-call, service-invocation, and dataflow.

In a procedure-call model, an application is represented by a set of procedures, and each procedure can call other procedures [3], [4], [8]. Thus, we can use a procedure-call tree or graph to model the structure of an application. In the tree/graph, the node represents the procedure, and the edge represents the call relationship. The programmers write the application according to a procedure-oriented paradigm. The problem in partitioning is deciding whether each procedure should be offloaded or not. This model can be applied to most applications.

In a service-invocation model, an application comprises a set of services. We usually use a service-invocation graph to model the application. In this graph, the node indicates the service, and the edge indicates the service that the programmers need in order to program the application using a service-oriented methodology. The work in [5] and [6] pertains to this model. In [5], an application with a set of weblets is decomposed. A weblet is a kind of web service that can be executed at either the mobile side or cloud side. I. Giurgiu et al. [6] build their partitioning system on a distributed service computing platform called AlfredO [13]. This platform has been used to decompose and couple Java applications into software modules.

The service-invocation model and procedure-call model decompose an application at different granularities. The service

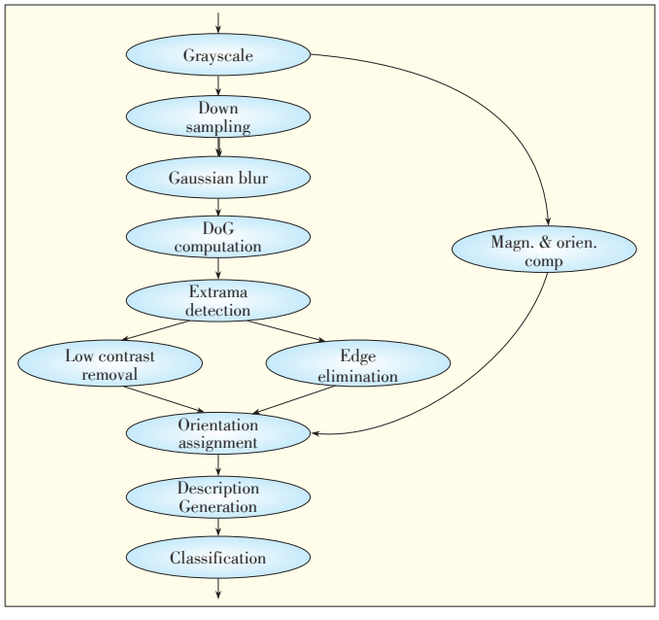
model decomposes the application at its functionalities, and the decomposed modules are loosely coupled. The procedure-call model decomposes the application according to the structure of the code. The decomposed procedures are tightly coupled with each other, which creates programming difficulties in terms of distributed execution. However, in terms of the application's representation methodology, the service-invocation model and procedure-call model are the same. They use a very similar graph to model the application.

The dataflow model is suitable for modeling most media applications that have continuous incoming data to process. In this model, the application comprises a set of dependable stages. The output data at each stage becomes the input data of the next stage. At each stage, a particular operation is performed on the incoming data. Dataflow can be represented by a directed acyclic graph in which each node is a stage and each edge indicates the data dependence between the two connecting stages. In [9] and [11], the application to be partitioned is modeled as dataflow graphs. **Fig. 3** shows the dataflow graph of a face-recognition application.

3.2 Profiling

The profiling process involves collecting information related to the application, device, and network. Application information includes the execution load (not dependent on devices) of each part of the application as well as the amount of data transmitted between two dependent parts. Because the application information is static, it can be gathered offline.

Collecting device-related and network-related information can be difficult. In terms of device profiling, we are concerned with estimating parameters such as energy consumption, computation capability, and CPU/memory workloads. In terms of network profiling, we are concerned with parameters such as bandwidth, latency, and package loss rate. Parameters such as device computing capability and energy consumption are static, so they can be acquired offline. Other parameters, such as CPU/memory workloads and network bandwidth, may vary in real environments, thus we need to estimate them online.



▲**Figure 3. Dataflow graph for an image-based face-recognition application.**

MAUI [3] profiles the energy consumption of each part of the application according to a model that shows energy consumption as a function of CPU cycles. The model is learned offline from real measurements. The authors of MAUI also evaluate the accuracy of the model and show that its error margin is less than 6%. MAUI [3] estimates network parameters, such as bandwidth and latency, online through recent offloading opportunities. It also updates its estimations when there is no offloading by transferring a 10 KB file to the server. The profiled parameters are used to make online partitioning decisions.

CloneCloud [4] profiling comprises two phases: online and offline. In the offline phase, the partitions of an application are obtained for various execution conditions (including those related to the device and network characteristics). The execution conditions are acquired by real measurements, regardless of the overhead. The partitions and corresponding execution conditions are stored in the database of the device. In the online phase, the system estimates the execution condition and searches the matched partition from the database. In [4], the authors do not describe how to estimate the execution condition accurately and efficiently in the online phase.

Odessay [9] does not profile the application, device, and network independently. Instead, it directly profiles the running time of each stage and data transmission time of each connector. The information is updated when the application starts to execute again. This information is used to determine the partition in the next execution. Most related works on computation partitioning take the approach of Odessay [5], [8]. This approach avoids the overhead created by direct profiling of networks and devices; however, it may not be as accurate as direct profiling because the partition is always made according to the last execution condition. The partition leads to bad performance when the execution conditions change quickly. This often happens in mobile environments where the wireless connection drops or bandwidth fluctuates. In our current research, we focus on estimating network bandwidth efficiently in real time and then updating the application's partition, even during the course of the execution of the application.

3.3 Optimization

Computation partitioning requires partitions to be optimized according to profiling information. The optimization metric can be execution time, energy consumption, data traffic, or a customized, weighted summation of these. MAUI [3] was proposed to optimize the energy consumption of devices. CloneCloud [4] and ThinkAir [8] support the optimization of either execution

time or energy consumption, depending on the programmers' choices. Odessay [9] aims to optimize the makespan for data-streaming applications. The framework in [11] was proposed to optimize the processing throughput for streaming applications. In [5], the authors propose hybrid optimization that can be customized by the end user.

Optimization can be done online or offline. The offline approach is taken to determine the optimal partitions for various execution conditions in the offline phase. In the online phase, one of the partitions is selected according to the current profiling conditions. The more the execution conditions traversed, the more accurate the online partition is. The online approach enables optimization on the fly according to profiling condition. Online optimization is accurate but creates overhead. In [4] and [5], the offline approach is taken, and in [3], [9] and [11], the online approach is taken.

Online optimization can be done by the mobile device, in the cloud, or even in a wireless network. Most existing works describe optimization by the mobile device [3], [4], [9], [14]. In [11], optimization is done in the cloud. If optimization is done on the mobile device, the profiled parameters do not need to be transmitted over the network. This causes additional computational overhead on the device. Optimization in the cloud can avoid this problem, but the mobile device needs to be continually connected in order to transmit the profiled parameters. This is suitable for partitioning complex applications.

3.4 Implementation

We classify implementation approaches to partitioning as client-server communication, VM migration, and mobile agent. The client-server approach requires the program codes to be pre-installed on the cloud servers. When one function of the application is offloaded onto the cloud, this function is usually performed by the Remote Procedure Call (RPC) protocol or by Remote Method Invocation (RMI). In [15] and [10], the client-server communication approach is taken to implement the partitioned execution. The drawback of this approach is that it is prone to failure due to network disconnection. The codes on the cloud/server side need to be changed from the original codes on the mobile device. Deployment of the partitioning system is not convenient.

Virtual machine (VM) migration is used to implement partitioned execution in [3], [4], [8], and [16]. At the mobile side, the application runs on a VM. When a decision is made to offload some part of the application to the cloud, the whole VM migrates to the cloud. The VM migrates back to the mobile side when the application part is finished in the cloud. This approach does not require the application to be pre-installed in the cloud; however, VM migration creates more overhead than remote procedure call because the execution state of the VM, including the memory and register state, needs to be transmitted.

Scavenger [17] uses a mobile agent to implement remote execution. It provides a platform that can help the user easily program and deploy partitioning-enabled applications. Dynamic deployment of application can be realized using this approach. However, agent management is required, and this causes overhead on the mobile device.

4 Research Challenges

4.1 Energy Efficiency

The processing capability of mobile devices is increasing, and energy consumption has become a significant issue for mobile applications. Most device vendors look for approaches to increasing the battery life. Besides inventing new battery technologies, there are many other approaches to saving energy at the system and application layers. Computation partitioning is an important approach to saving energy on devices. With this approach, energy-consuming components of the application, e.g. computationally intensive algorithms, are of floaded to the cloud. However, the difficulty with this approach is designing effective mechanisms to monitor and profile the energy consumption of applications on mobile devices. Designing models for estimating energy consumption during data transmission is also not easy. Both the profiled information and models are critical to partitioning the application in order to save energy.

We need to design energy-efficient partitioning software on the mobile device. The computationally intensive part of the computation partitioning software is the optimization. As discussed in the last section, optimization can be done offline. Partitions that are generated from offline optimization are stored on the mobile device. Whenever the execution environment changes, the application is configured with the optimal partition from all the backup partitions. Offline optimization saves energy overhead. In [4], offline optimization is proposed. Another approach to energy saving is optimization in the cloud. We have proposed a partitioning framework that implements optimization in the cloud by using a genetic algorithm [13].

Other researchers look for techniques to optimize energy consumption during data transmission. Offloading computation to the cloud requires transmission of the computation input data. Issues related to energy consumption during data transmission in computation partitioning need to be tackled. E. Uysal Biyikoglu et al. [18] design an energy-efficient data transmission mechanism that monitors network quality and transmits the data accordingly. If the network quality is good, data is transmitted; otherwise, no data is transmitted in order to save energy.

4.2 Mobile Access Management

In the mobile cloud partitioning system, mobile access networks such as 3G/4G cellular networks and wireless local area neworks (WLANs) are important components for connecting the mobile devices to the cloud. The quality or bandwidth of the user's connection to the cloud directly determines the par-

titioning of the application.

4.2.1 Network Intermittence

A practical issue is how to partition the application when the network connection is intermittent. The application is usually partitioned according to a cost model that includes computational cost (on both the mobile and cloud sides) and offloading cost (e.g. data transmission cost). In most works, it is assumed that offloading cost does not change during application runtime. This is not practical in mobile environments. In reality, network connectivity can fail because of wireless network holes, which are places where there is no signal or the signal is too weak to maintain a connection. Even when the network is connected, throughput or bandwidth can fluctuate because of the user's mobility. Fluctuating network status gives rise to offloading cost.

C. Shi et al. [19] solve this problem by assuming future network connectivity is perfectly known. They designed an offline algorithm to calculate the optimal partition given a future network bandwidth. In practical systems, we need to design an online algorithm to partition the application [20]. This algorithm can update the partition of an application from time to time during the course of execution and according to the predicted network status. The prediction of network bandwidth is also a critical issue to be addressed.

Several previous works discuss the prediction of future network status from historical mobility observations. Such an approach has been used in other applications, the first of which was wireless sensor network (WSN) data delivery. In [21], routing is improved by using a mobility-prediction algorithm. H. Lee et al. [22] study the problem of delivering data from data source nodes to the mobile sink. They predict the nodes that the mobile sink is likely to pass by and stash data on these in advance. There are also many early works on mobility prediction in cellular/Wi-Fi networks. These works discuss ways of improving network handoff by predicting the next cells/APs [23], [24].

4.2.2 Network Resource Allocation

Another issue is network resource/bandwidth allocation for user-dependent computation partitioning modeling. If mobile users offload computation to clouds through the same access networks, network resources or bandwidth will be limited. We need to allocate resources to mobile users. A user who is allocated more bandwidth has lower offloading cost, and a user allocated less bandwidth has higher offloading cost. Users' partitioning decisions depend on each other because users are competing for shared network resources. Thus, the partitioning problem needs to be solved in light of network resource allocation.

Network resources may include both cellular networks and WLANs. The research problem can be stated in different ways. One way is: we need to allocate cellular network and WLAN resources to mobile users so that overall system performance is maximized. Another way is: we need to determine how many of each type of network resource should be leased by the application provider and how to allocate resources to mobile users so that maximum performance is achieved for the lowest cost.

4.3 Workload Management

Workload management is another important issue in computation partitioning. The workload is the computations offloaded from mobile users to the cloud. The mobile cloud application needs to serve a large number of mobile users. When the scale of applications increases, properly managing the workload at cloud clusters is essential for efficient use of cloud resources and for good system performance. In the user-independent model, workload management is unrelated to the computation partition of each user. Traditional workload scheduling and balancing mechanisms can be used to tackle the problem [25]-[27]. Next, we discuss workload management in the user-dependent model.

4.3.1 Workload Scheduling in a Centralized Cloud

In the user-dependent model, the workload management is correlated with the application partition of each user. For good system performance, it is better to consider computation partitioning and cloud workload management together. First, we consider the problem using a simple system model [12]. The application is modeled as a sequence of dependent tasks, and mobile users run the same application. On the centralized cloud there is a set of server nodes that accommodate the workload (tasks) offloaded by users. The objective is to schedule the tasks of all users onto their mobile devices and the cloud servers. Each user device can only execute tasks from itself, not from other users. The problem is abstracted as a job-scheduling problem that is similar to (but more difficult than) a traditional job-scheduling problems in parallel computing.

The first classic job-scheduling problem is Task Scheduling Problem for Heterogeneous Computing (TSPHC) [28]. In this problem, an application is represented by a directed acyclic graph (DAG) in which nodes represent application tasks and edges represent inter-task data dependencies. Given a heterogeneous machine environment, where machines have different processing speeds and the data transfer rate between machines differs, the objective of the problem is to map tasks onto the machines and order their executions. In this way, task-precedence requirements are satisfied, and completion time is minimized. In general, TSPHC is NP-complete, and efficient heuristics have been proposed in the literature [28], [29].

The workload-scheduling problem [12] can be modeled as close to TSPHC as possible. In the system model [12], there are $\lambda \times n$ tasks, where λ is the number of users and n is the number of tasks in the application. The machines can be abstracted as a set of r cloud servers/VMs and one mobile device. The data transfer rate between the cloud VMs is infinite but

constrained between any pairs of mobile device and cloud VM. The problem is to map the tasks onto $(r+1)$ machines so that the precedence constraints in the application graph are satisfied, and the weighted summation of the completion time of all the tasks is minimized. The tasks that appear last in the application flow are assigned a weighting of one, and others are assigned a weighting of zero.

The key difference between the workload-scheduling problem [12] and TSPHC is the optimization objective. In TSPHC, the optimization objective is to minimize the makespan, that is, the maximum completion time for all the tasks. In [12], the objective is to minimize the total weighted completion time. Although efficient heuristics have been proposed to minimize the TSPHC makespan, few solutions have been proposed to minimize the total weighted completion time. Some early efforts were made to minimize the total weighted completion time on a single machine or parallel machine, but communication was not considered. Even simplified solutions to the workload scheduling problem in [12] are NP-hard [30].

The second classic scheduling problem is Hybrid Flow Shop (HFS) scheduling [31]. In this problem, the job is divided into stages, and there are a number of identical machines in parallel at each stage. Each job has to be processed at stage one, then stage two, and so on. At each stage, the job needs to be processed on one machine only, and any machine will do. It is assumed that all the jobs are released at the beginning, and the problem is to find a schedule to minimize the makespan. The application and its functional modules in our problem are analogous to a job and stages in HFS. The mobile devices and cloud VMs may be modeled as machines in HFS. However, the workload-scheduling problem in [12] is much different from that in HFS. In [12], there is communication overhead between stages, which makes the problem more complex than that in HFS. In [12], both the cloud VM and mobile device can execute any module of the application, so a set of machines is not partitioned into subsets according to the stages. The objective in [12] is to minimize the total completion time rather than the makespan.

4.3.2 Workload Scheduling in Distributed Clouds

Workload scheduling in the user-dependent model is more challenging when we consider that the cloud consists of geographically distributed data centers. In this model, there exists a set of mobile users from different regions. Each user has a partitioned execution of the application. The cloud contains a set of data centers that are distributed in different regions. Each data center has a certain capacity in terms of computing resources. For each user, offloading the same component to various data centers can create different offloading costs because the connection delay and/or bandwidth is different for different data centers. We need to partition the application for mobile users as well as allocate the offloaded computation to computing resources at different data centers. There are two types of workload scheduling: inter-datacenter and intra-datacenter scheduling. Both scheduling types should be considered when partitioning the computation of an application for each user so that overall system performance (e.g. the application execution time) is maximized.

Several recent works on cloud computing have described solutions to the scheduling problem for distributed clouds [30]-[33]. P. Gao et al. [32] developed an optimization framework to schedule data access requests/workloads from users to distributed data centers. The scheduling issue is studied with the aim of minimizing energy consumption in the cloud. Y. Wu et al. [33] studied the scheduling of video-on-demand access requests/workloads to geographically distributed clouds. The scheduling problem was studied together with the video-placement problem. The objective was to minimize the operational cost while satisfying the delay constraints on video access requests.

The request-scheduling problem for live video streaming applications was also studied in [34]. However, the existing works [32], [33], [34] do not apply to workload scheduling in computation partitioning systems because scheduling and partitioning are coupled in the user-dependent model and cannot be treated separately.

4.4 Performance Modeling and Monitoring

An important issue is how to design a performance model for various mobile cloud applications, including applications centered on content delivery and sensing delivery as well as user-interactive applications. All these types of applications have different performance requirements in terms of end users and systems. We need to design an accurate performance model that can illustrate both sets of performance requirements.

To develop such a performance model, we have investigated many software engineering works and ISO standards [35]-[35] that proposed appropriate performance models for various software systems. In Jain's model [37], the system is supposed to give three outcomes for a given request: correct, incorrect, or refusal to give an outcome. Three system performance metrics have also been defined: speed, reliability, and availability. The performance model for a mobile-cloud partitioning system needs to be developed by adding more practical performance metrics, such as Service-Level Agreement (SLA), time behavior, utilization, capacity, and recoverability.

Another issue is how to detect anomalies or performance degradation in a mobile-cloud partitioning system. In traditional internet applications, anomalies are detected by manually analyzing the logs [38]. This method is not feasible for a large-scale cloud system. Some researchers have taken a pattern recognition approach [39], [40] to automating the analysis of massive volumes of system logs. Because of the complex computational cost of analysis, this approach is not feasible for a system that requires real-time anomaly detection and recovery. Methods based on log analysis are not enough to detect anoma-

lies or performance degradation in mobile cloud applications. Usually, the performance of a mobile cloud application depends on more complex factors, including failure or inefficiency of the mobile device, wireless network, or clouds. Collecting and analyzing logs from mobile devices may not be possible because of high costs or privacy issues. Faced with these difficulties, we need to develop suitable approaches to detecting anomalies and performance degradation in mobile cloud applications.

To solve this detection problem, we can use a hybrid method that integrates the log analysis and real-time performance monitoring. At the cloud side, we can detect anomalies by analyzing the system logs, and at the mobile side, we can design lightweight performance monitors. We also need to design network protocols to monitor the performance of interactions between mobiles and the cloud.

4.5 Security

There are three security issues in a mobile cloud partitioning system. The first issue is authentication between computations that pertain to the same application. Authentication indicates the secure communication channel between two computations that may be executed on different mobile or cloud platforms. The second issue is access control for sensitive user data. Because the computations of an application may run on a public, untrusted cloud platform, we should design secure mechanisms for controlling access to sensitive data from mobile devices. For example, computation requires access to very sensitive data, it is better to do that computation in trusted environments, e.g. on the mobile device or on a trusted cloud platform. The third issue is to build and verify a trusted execution environment, including the mobile platform and cloud platforms. This is a fundamental problem in cloud computing. The work by X. Zhang et al. [41] is one of the few works that describe security problems specifically for mobile cloud partitioning systems. The authors propose a solution for authentication and secure session management between weblets/computations running at the device side and cloud side. The authors also propose a secure migration mechanism and approach to authorizing cloud weblets to access sensitive user data.

5 Summary of Our Own Work

We reviewed the related literature and found that several new and challenging issues need to be urgently addressed. First, little work has been done on partitioning data-streaming applications (which are quite popular today) because of a device's ability to collect streaming media data. The challenge is how to design models and architectures for partitioning these applications. Second, wireless connectivity loss and bandwidth fluctuations often occur in mobile environments. The challenge is in designing partitioning mechanisms that can still guarantee high performance in these conditions. Third, for large-scale mobile cloud applications, the number of mobile users (mobile load) can be unpredictable. The challenge is in designing partitioning mechanisms that can achieve optimal overall performance, even when the system is overloaded.

In our work, we develop and implement a real-world application, and we show that computation partitioning is a feasible way of improving performance. Then, we propose our solutions in terms of the above challenges. Our research on this topic can be categorized as computation partitioning in an RFID tracking application, performance optimization for user-independent computation partitioning, and performance optimization for user-dependent computation partitioning.

We study the application of computation partitioning in an RFID tracking application [42]. We focus on a system for attaching RFID readers on moving objects, and we deploy passive RFID tags in the environment. The moving object collects the noisy RFID readings and continuously estimates its position in real time. Traditional approaches, such as Particle Filter (PF), can by highly accurate but require much computation on the device. These approaches are difficult to implement on mobile devices that are constrained in terms of computing capability and battery. Other existing approaches, such as Weighted Centroid Localization (WCL), are cheap in terms of computational cost but are very inaccurate, especially when the object is moving quickly. Thus, we propose an adaptive approach to achieving accuracy and energy efficiency. Our approach can be used to choose costly PF or cheap WCL, depending on the estimated speed of the object. It can also be used to adaptively partition the computations between the mobile device and infrastructure servers or clouds (depending on the quality of the network connections). We evaluate our solution in real-world experiments and show that our proposed computation partitioning scheme outperforms other schemes in terms of accuracy and energy consumption.

We also study two issues in user-independent computation partitioning, where every user can make their partitioning decision according to their own information. The first issue is partitioning of the data-streaming application. Existing approaches can be taken to optimize the makespan of streaming applications. Throughput/processing speed is more important for streaming applications. We propose an algorithm to maximize throughput and develop a reference-implementation architecture for partitioning and execution of streaming applications [11]. The second issue is solving the computation partitioning problem when network connectivity is intermittent and bandwidth fluctuates. The existing one-time partitioning approach may significantly degrade performance when the network fluctuates. We develop a predictive partitioning algorithm that exploits knowledge of user's mobility to predict network status, and we update the partition based on the predicted network status [20]. We evaluate our approach according to real data traces that are collected in a campus Wi-Fi hotspot testbed. The results show that our method significantly reduces completion

time compared with previous approaches.

Last, we study the same issues in user-dependent computation partitioning. Most related works pertain to the user-independent model, but we are the first to study and propose a user-dependent computation partitioning model [12]. In this model, user partitioning decisions depend on each other because users compete for some shared resources, e.g. cloud servers and wireless access bandwidths. Thus, to achieve high system performance, allocation of shared resources needs to be considered in conjunction with user partitioning. This problem is different from and more difficult than classic job-scheduling problems. We design both offline and online algorithms to solve this problem. With benchmarks, we show that our offline algorithm outperforms listing scheduling algorithms by 10% in terms of application delay. We also validate the efficiency of our online algorithm using real-world load traces.

6 Conclusion

The rise of mobile cloud computing is rapidly changing the IT landscape. Computation partitioning has recently been studied in order to achieve high-quality service provisioning and operational efficiency for the cloud providers. Despite the significant benefits offered by the new computing paradigm, current technologies are not mature enough to realize its full potential. Challenges related to energy efficiency, mobile access management, workload management, performance modeling and monitoring, and security still exist in this domain and are beginning to attract the attention of the research community. This paper discussed state-of-the-art mobile cloud computation partitioning. It covered system models, key technologies, and research issues and directions. This work is intended to deepen the understanding of design challenges in mobile cloud computing and pave the way for further research in this area.

Acknowledgments

The research is supported in part by Hong Kong RGC under GRF Grant 510412, and the National High-Technology Research and Development Program (863 Program) of China under Grant 2013AA01A212.

References

[1] K. Kumar and Y. Lu, "Cloud computing for mobile users: can offloading computation save energy," *IEEE Computer Society*, vol. 43, issue 4, pp. 51-56, April 2010. doi: 10.1109/MC.2010.98.

[2] B.-G. Chun and P. Maniatis, "Augmented smartphone applications through clone cloud execution," in *Proc. HotOS*, Monte Verità, Switzerland, May 2009, pp. 8. doi: 10.1.1.148.8182.

[3] E. Cuervoy, A. Balasubramanianz, and D. Cho, "MAUI: making smartphones last longer with code offloading," in *Proc. MobiSys*, San Francisco, CA, USA, 2010, pp. 49-62. doi: 10.1145/1814433.1814441.

[4] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proc. EuroSys*, Salzburg, Austria, 2011, pp. 301-314. doi: 10.1145/1966445.1966473.

[5] X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, "Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 270-284, June 2011. doi: 10.1007/s11036-011-0305-7.

[6] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso, "Calling the cloud: enabling mobile phones as interfaces to cloud applications," in *Proc. Middleware*, Urbana Champaign, Illinois, US, 2009, pp. 83-102. doi: 10.1007/978-3-642-10445-9_5.

[7] Z. Li, C. Wang, and R. Xu, "Computation offloading to save energy on handheld devices: a partition scheme," in *Proc. Int. Conf. Compilers, Architecture, and Synthesis for Embedded Systems*, Taipei, Taiwan, Oct. 2001, pp. 238-246. doi> 10.1145/502217.502257.

[8] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "ThinkAir: dynamic resource allocation and parallel execution in cloud for mobile code offloading," in *Proc. IEEE Infocom*, Orlando, FL, USA, 2012, pp. 945-953. doi: 10.1109/INFCOM.2012.6195845.

[9] M. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan, "Odessa: enabling interactive perception applications on mobile devices," in *Proc. MobiSys*, Washington, DC, USA, 2011, pp. 43-56. doi:10.1145/1999995.2000000.

[10] R. Balan, M. Satyanarayanan, S. Park, and T. Okoshi, "Tactics-based remote execution for mobile computing," in *Proc. Mobisys*, San Francisco, CA, USA, 2003, pp. 273-286. doi: 10.1145/1066116.1066125.

[11] L. Yang, J. Cao et al., "A Framework for Partitioning and Execution of Data Stream Applications in Mobile Cloud Computing," in *Proc. IEEE CLOUD 2012*, Honolulu, Hawaii, USA, pp. 794-802. doi: 10.1109/CLOUD.2012.97.

[12] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," Dept. Computing, Hong Kong Polytechnical Univ., Tech. Rep., June 2013.

[13] J. Rellermeyer, O. Riva, and G. Alonso, "AlfredO: an architecture for flexible interaction with electronic devices," in *Proc. Middleware 2008*, Leuven, Belgium, pp. 22-41. doi: 10.1007/978-3-540-89856-6_2.

[14] M. Barbera, S. Kosta, A. Mei, and J. Stefa, "To offload or not to offload: the bandwidth and energy costs of mobile cloud computing," in *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 2013, pp. 1285 - 1293. doi: 10.1109/INFCOM.2013.6566921.

[15] J. Flinn, "Balancing performance, energy, and quality in pervasive computing," *In Proc. ICDCS 2002*, pp. 217-226.

[16] M. Satyanarayanan, P.Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14-23, Oct. 2009. doi: 10.1109/MPRV.2009.82.

[17] M. Kristensen, "Scavenger: transparent development of efficient cyber foraging applications," in *Proc. PerCom*, Mannheim, Germany, 2010, pp. 217-226. doi: 10.1109/PERCOM.2010.5466972.

[18] E. Uysal-Biyikoglu and A. Gamal, "On adaptive transmission for energy efficiency in wireless data networks," *IEEE Transaction on Information Theory*, vol. 50, no. 12, pp. 3081-3094, Dec. 2004. doi: 10.1109/TIT.2004.838355.

[19] C. Shi et al., "Computing in cirrus clouds: the challenge of intermittent connectivity," in *Proc. MCC' 12*, Helsinki, Finland, pp. 23-28. doi: 10.1145/2342509.2342515.

[20] L. Yang, J. Cao, S. Tang, D. Han, and N. Suri, "Foreseer: predictive mobile-cloud program partitioning under network fluctuations," Dept. Computing, Hong Kong Polytechnical Univ., Tech. Rep., Mar. 2013.

[21] B. Kusy et al., "Predictive QoS routing to mobile sinks in wireless sensor networks," in *Proc. IPSN*, San Francisco, CA, USA, Apr. 2009, pp. 109-120. doi: 10.1145/1602165.1602177.

[22] H. Lee et al., "Data stashing: energy-efficient information delivery to mobile sinks through trajectory prediction," in *Proc. IPSN*, Stockholm, Sweden, Apr. 2010, pp. 291-302. doi: 10.1145/1791212.1791247.

[23] T. Liu, P. Bahl, and I. Chlamtac, "Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 6, pp. 922-936, 1998. doi: 10.1.1.33.5575.

[24] L. Song et al., "Evaluating next-cell predictors with extensive wi-fi mobility data," *IEEE transaction on Mobile Computing*, vol. 5, no. 12, pp. 1633-1649, 2006. doi: 10.1109/TMC.2006.185.

[25] J. Hu, J. Gu, G. Sun, and T. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in *Proc. Third Int. Symp. Parallel Architectures, Algorithms and Programming*, Dalian, Liaoning, China, Dec. 2010, pp. 89-96. doi: 10.1109/PAAP.2010.65.

[26] Y. Fang, F. Wang, and J. Ge., "A task scheduling algorithm based on load balancing in cloud computing," *Web Information System and Mining Lecture Notes in Computer Science*, vol. 6318, pp. 271-277, 2010. doi: 10.1007/978-3-642-16515-3_34.

[27] S. T. Maguluri, R. Srikant, L. Ying, "Stochastic models of load balancing and scheduling in cloud computing clusters," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 702–710. doi: 10.1109/INFCOM.2012.6195815.
[28] H. Topcuoglu, S. Hariri, and M. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 1215–1223, Mar. 2002. doi: 10.1109/71.993206.
[29] Y. Lee and A. Zomaya, "A novel state transition method for metaheuristic-based scheduling in heterogeneous computing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 9, pp. 1215–1223, May 2008. doi: 10.1109/TPDS.2007.70815.
[30] L. Hall, D. Shmoys, and J. Wein, "Scheduling to minimize average completion time: off line and on-line algorithms," in *Proc. Seventh Annual ACM-SIAM Symp. Discrete Algorithms*, Atlanta, GA, USA, Jan. 1996, pp.142–151. doi: 10.1.1.33.3230.
[31] M. Pinedo, *Scheduling theory, Algorithms, and Systems*, 2nd Ed. Upper Saddle River, New Jersey, USA: Prentice Hall, 2002.
[32] P. Gao, A. Curtis, B. Wong, and S. Keshav, "It is not easy being green," in *Proc. ACM SIGCOMM*, Helsinki, Finland, 2012, pp. 211–222. doi: 10.1145/2342356.2342398.
[33] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. Lau, "Scaling social media applications into geo-distributed clouds," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 684–692. doi: 10.1109/INFCOM.2012.6195813.
[34] F. Wang, J. Liu, and M. Chen, "CALMS: cloud-assisted live media streaming for globalized demands with time/region diversities," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 199 – 207. doi:10.1109/INFCOM.2012.6195578.
[35] *Software Engineering Product Quality Part 1: Quality Model. International Organization for Standardization*, ISO/IEC9126-1:2001(E), 2001.
[36] *Systems and Software Engineering Systems and Software Product Quality Requirements and Evaluation (SQuaRE) System and Software Quality Models. International Organization for Standardization*, ISO/IEC 25010:2010(E), 2010.
[37] J. Raj, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. New York, USA: Wiley Interscience, 1991.
[38] S. Nousiainen, J. Kilpi, P. Silvonen, and M. Hiirsalmi, "Anomaly detection from server log data: a case study," VTT, Espoo, Finland, VTT Tiedotteita Research Notes 2480, 2009.
[39] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *Proc. 39th Ann. IEEE/IFIP Int. Conf. Dependable Systems and Networks*, Estoril, Lisbon, Portugal, 2009, pp. 125–134. doi: 10.1.1.163.8181.
[40] T. Lane and C. E. Brodley, "Temporal sequence learning and data reduction for anomaly detection," *ACM Transactions on Information and System Security*, vol. 2, no. 3, pp. 295–331, Aug. 1999. doi: 10.1145/322510.322526.
[41] X. Zhang, J. Schiffman, S. Gibbs, A. Kunjithapatham, and S. Jeong, "Securing elastic applications on mobile devices for cloud computing," in *Proc. CCSW*, Chicago, IL, USA, Nov. 2009, pp. 127 –134. doi: 10.1145/1655008.1655026.
[42] L. Yang, J. Cao, W.Zhu, and S. Tang, "A hybrid method for achieving high accuracy and efficiency in object tracking using passive RFID," in *Proc. IEEE Int. Conf. Pervasive Computing and Communications*, Lugano, Switzerland, Mar. 2012, pp. 109 –115. doi: 10.1109/PerCom.2012.6199856.

Manuscript received: September 10, 2013

Biographies

Lei Yang (csleiyang@comp.polyu.edu.hk) received his BSc degree in electronic engineering from Wuhan University, China, in 2007. He received his MSc degree in computer science from Institute of Computing Technology, Chinese Academy of Science, 2010. He is currently a PhD candidate of the Department of Computing, Hong Kong Polytechnic University. His research interests include mobile cloud computing, RFID systems, and social computing.

Jiannong Cao (csjcao@comp.polyu.edu.hk) is currently a chair professor and head of the Department of Computing at Hong Kong Polytechnic University. He received his BSc degree in computer science from Nanjing University, China, in 1982. He received his MSc and PhD degrees in computer science from Washington State University in 1986 and 1990. His research interests include parallel and distributed computing, computer networks, mobile and pervasive computing, fault tolerance, and middleware. He has co-authored four books, co-edited nine books, and published more than 300 technical papers in major international journals and conference proceedings. He has directed and participated in numerous research and development projects and, as a principal investigator, obtained more than HK$25 million in grants. He is the chair of Technical Committee on Distributed Computing, IEEE Computer Society; a senior member of IEEE; a member of ACM; and a senior member of China Computer Federation. He has been an associate editor and member of editorial boards of many international journals. He has been the chair and a member of organizing/program committees for many international conferences.

ZTE Contributed to Launch of 4G Network of BASE Company in Belgium

4 November 2013, Shenzhen, China–ZTE Corporation, a publicly-listed global provider of telecommunications equipment, network solutions and mobile devices, and BASE Company, the Belgian subsidiary of Dutch mobile operator KPN, have launched BASE Company's 4G services in Belgium.

BASE Company has launched 4G services directly in 15 cities and was the second operator to start 4G in Belgium. During the press conference at the announcement in early October, a 4G smart phone with only two signal bars reached download speeds of up to 42 Mbps, and multimedia files played clearly and smoothly.

As the radio equipment supplier for BASE Company, ZTE helped to construct high-quality UMTS and HSPA dual carrier networks in Belgium. With a Uni-RAN solution that supports smooth evolution and good equipment performance. ZTE has been selected as a 4G radio equipment supplier for BASE Company. ZTE's engineering team helped BASE Company to construct the high-quality 4G network.

ZTE is a major global provider of end-to-end integrated 4G solutions and services and a reliable strategic partner. By June 2013, ZTE had obtained 60 4G commercial contracts and 40 EPC commercial contracts. With leading 4G end-to-end solutions as well as partnerships with major operators around the world, ZTE has carried out 4G trial networks with more than 140 operators. ZTE has successfully launched commercial 4G services for China Mobile, Telenor, TeliaSonera, Bharti, Hutchison, and Telstra. (ZTE Corporation)

DOI: 10.3939/j.issn.1673-5188.2013.04.003
http://www.cnki.net/kcms/detail/34.1294.TN.20131203.0858.001.html, published December 3, 2013

MapReduce in the Cloud: Data-Location-Aware VM Scheduling

Tung Nguyen and Weisong Shi
(Department of Computer Science, Wayne State University, Detroit, MI 48202, USA)

Abstract

We have witnessed the fast-growing deployment of Hadoop, an open-source implementation of the MapReduce programming model, for purpose of data-intensive computing in the cloud. However, Hadoop was not originally designed to run transient jobs in which users need to move data back and forth between storage and computing facilities. As a result, Hadoop is inefficient and wastes resources when operating in the cloud. This paper discusses the inefficiency of MapReduce in the cloud. We study the causes of this inefficiency and propose a solution. Inefficiency mainly occurs during data movement. Transferring large data to computing nodes is very time-consuming and also violates the rationale of Hadoop, which is to move computation to the data. To address this issue, we developed a distributed cache system and virtual machine scheduler. We show that our prototype can improve performance significantly when running different applications.

Keywords

cloud; MapReduce; VM scheduling; data location; Hadoop

1 Introduction

Recently, the volume of data being generated as a result of simulation and data mining in the physical and life sciences has increased significantly. This trend has necessitated an efficient model of computation. Google MapReduce [1] is a popular computation model because it is suitable for data-intensive applications, such as web access logging, inverted index construction, document clustering, machine learning, and statistical machine translation. There are several implementations of MapReduce, including Phoenix [2], Sector/Sphere [3], open-source Hadoop [4], and Mars [5]. In fact, Hadoop is so popular that Amazon offers a separate service, called Elastic MapReduce (EMR), based on it. In the past few years, we have witnessed the fast-growing deployment of Hadoop for data-intensive computing in the cloud.

In our analysis in section 2, we found that Hadoop is not as efficient as expected when running in the cloud. The first drawback is virtual machine (VM) overhead, which includes JVM overhead because Hadoop was developed with Java. A Hadoop MapReduce job is typically executed on top of a JVM operated inside another VM if run in the cloud. Our experiment shows that the execution time of an application run on VMs is about four times longer than the execution time of the same application run on physical machines. The second drawback is the extra overhead created by data movement between storage and computing facilities in the cloud. In Elastic MapReduce, data has to be transiently moved online from Amazon's simple storage services (i.e. S3) to the Hadoop VM cluster. Hadoop is often used to process extremely large volumes of data, and transient movement of this data puts a great burden on infrastructure. Resources such as network bandwidth, energy, and disk I/Os can be greatly wasted. For example, when sorting 1 GB of data on our testbed, the time taken to move the data was 4.8 times longer than the time taken to sort it. In this work, we focus on the data-movement problem and do not deal with VM overhead reduction.

We designed and implemented a distributed cache system and VM scheduler to reduce this costly data movement. In a warm-cache scenario, which is usually occurs after the cloud has been running for a while, our system improved performance by up to 56.4% in two MapReduce-based applications in the life sciences. It also improved performance by 75.1% in the traditional Sort application and by 83.7% in the Grep application.

The rest of the paper is organized as follows: In section 2, we first show the inefficiency of EMR in terms of performance and data access. In sections 3 and 4, we describe the design and implementation of our distributed cache system for improving EMR data movement. The performance of our prototype is evaluated in section 5, and related work is discussed in section

6. Conclusions are drawn in section 7.

2 Problem Statement

In this section, we explore how the current cloud and EMR system works and identify potential issues. In our case, the first question is: How do Cloud providers generally store user data? The next question we ask is: How does EMR work with respect to data processing?

2.1 Background

In this paper, we focus on an Infrastructure as a Service (IaaS) system, such as Amazon Cloud or Eucalyptus [6]. Because insight into Amazon Cloud is very limited, we have to use Eucalyptus, an open-source cloud computing platform that has a similar interface to Amazon EC2, a computing service, and Amazon S3, a storage service. Eucalyptus supports both ATA over Ethernet (AoE) and SCSI over Infiniband (iSCSI) storage networking standards, but the Amazon S3 architecture has not been published. Whenever possible, we run our experiments on Amazon.

We believe that S3 is in a storage-area network that is separate from EC2 for two reasons. First, the bandwidth between S3 and EC2 instances is smaller than that between EC2 instances [7]. Second, separation between computation and storage in data centers is a common design feature. S3 is designed specifically for storing persistent data and has strict requirements in terms of security, availability, reliability, and scalability. EC2 instances are usually used to store transient data which, if not moved to Amazon Elastic Block Store (EBS) or S3, is destroyed when the instance is terminated. In the Eucalyptus Community Cloud, the volumes and bucket directories (similar to EBS and S3) are also located in the front-end node, which is separate from the nodes that host VMs [8]. Further information about the differences between S3 and EBS can be found in [9].

To use EMR, the user first prepares the execution jar files (in the MapReduce framework) as well as the input data. They then create and launch a job flow and obtain the results. The job flow contains all the information, such as the number of instances, instance types, application jar and parameters, needed to execute a job.

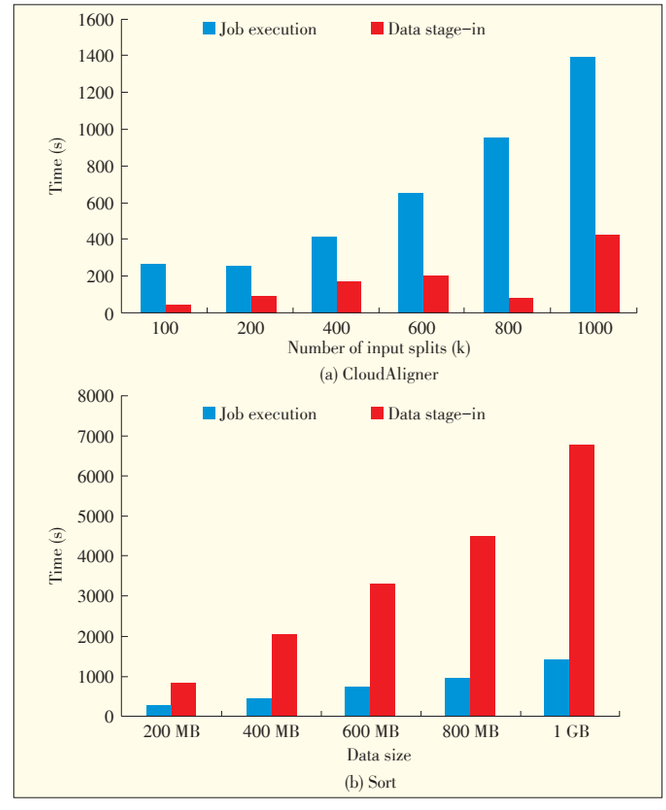
2.2 Problem Identification

There are two issues with EMR: 1) performance degradation from using VMs and 2) overhead created by data movement. In this work, we focus on the data-movement problem and do not deal with VM overhead reduction.

Overhead created by the use of VMs instead of physical machines is a conventional problem that has been well studied [10]-[12]. Most solutions to this problem are based on general platforms, such as Xen, KVM and VMware, providing better virtualization. In [13]-[15], the limited performance of Hadoop on VMs was investigated [15]. However, this investigation was problematic because the physical nodes had two quad-core 2.33 GHz Xeon processors and 8 GB of memory, and the VMs only had 1 VCPU and 1 GB of memory.

The second issue we deal with in this work is the overhead created by data movement. The above EMR workflow implies that the user data has to be transferred between S3 and EC2 every time a user runs a job flow. Although this transfer is free of charge from the user's perspective, it consumes resources, such as energy and network bandwidth. In fact, the cost may be considerable because the MapReduce framework is often used for data-intensive computing, and the data to be processed is massive in scale. The cost of moving all this data is not negligible.

We carried out two experiments to test our hypothesis that data movement is not a negligible part of job flow. Both experiments were done on our private cloud with Eucalyptus. The correlation between our private cloud and Amazon will be discussed in the next section. CloudAligner and Sort applications were executed with varied workloads in order to measure the execution and data-movement times. Sort is a benchmark built into Hadoop. **Fig. 1** shows that data movement is the dominant part of the Sort application (it is 4.8 times the execution time) and also an increasing part of the CloudAligner application. The largest amount of input data in the CloudAligner experiment, although extracted from the real data, is less than one-tenth the size of the real data. Also, the selected reference



▲**Figure 1. Data movement vs. job execution.**

chromosome, chr22, is the smallest of other chromosomes. The data movement part in the CloudAligner experiment indicates that data movement is would also be significant if real-sized data was used.

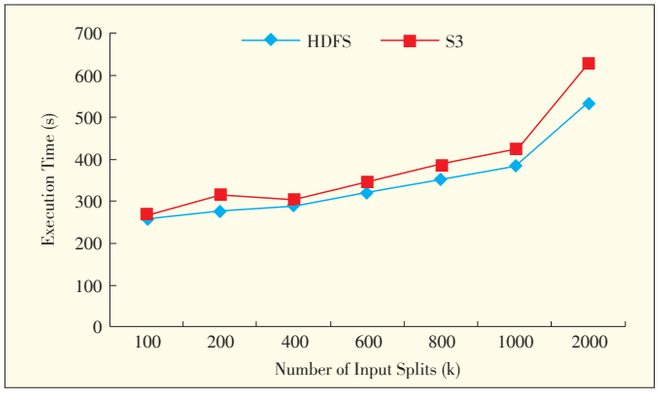
In general, the current EMR approach violates the rationale behind the success of Hadoop, i.e. moving computation to the data. Although optimized to perform on data stored on S3, EMR still performs better on HDFS (**Fig. 2**). Fig. 2 shows the results of running CloudBurst on an EMR cluster of 11 small EC2 instances with different data sizes and data stored on HDFS and S3. CloudBurst is a MapReduce application used to map DNA sequences to the reference genome [16]. Like CloudAligner, this type of application is fundamental in bioinformatics. It is used because it is also an exemplar application of EMR.

Data movement is time-consuming, and in EMR, data has to be moved every time a user executes a job flow. With the elasticity and transiency of the cloud, data and VMs are deleted after the job flow has finished. The situation is even worse when, after the first run, the user wants to tune the parameters and re-run the application on the same data set. The whole process of moving the huge data set would be triggered again. One may argue that the original job could be kept alive, and the EMR command line tool could be used to add steps with modified parameters. However, for simplicity many users only use the web-based tool, which is currently rather simple and does not support such features. Keeping the system alive also means the user keeps paying.

This wastes a lot of bandwidth. To overcome this, user persistent data and computation instances have to be close to each other. We propose a system that caches user data persistently in the physical hosts' storage (where the VMs of that user are hosted). This way, when the same user comes back to the system, the data is ready to be processed.

3 System Design

As stated in the previous section, our goal is to improve the movement of data by EMR by reducing the amount of data to be transferred to the computing system from the persistent storage. To achieve this goal, the loaded data should be kept at the computing instances persistently so that it can be used later. When a user returns to the system, their VMs should be scheduled close to the data.



▲**Figure 2. Execution time of CloudBurst on HDFS and S3.**

3.1 Terminology, Assumptions and Requirements

Before getting into the details of our solution, we first clearly identify terminologies, the target system, the intended applications, and requirements.

We use the terms VM, instance, node, and computing node interchangeably. The terms physical machine and physical hosts are likewise used interchangeably. The back-end storage is referred to as persistent storage services, such as S3 or Walrus. The front-end, or ephemeral storage, is the storage at the computing nodes. However, the front-end server is the head node of the Eucalyptus cloud.

The specific system to which we want to add our cache should be similar to EMR because EMR is a proprietary production system and therefore cannot be accessed.

The intended application of our system is the same as that of the MapReduce framework: a data-intensive application. Using our modified EMR system, a user would not see any change in the system except for improved performance. The interaction between the user and system is also unchanged. The user still needs to create a job flow and specify instances, parameters, and executable files.

The input assumption of our system, which is the same as that of EMR, is that the user data and its executions are already stored in the persistent storage (SAN), such as S3. Therefore, it is not necessary to deal with data availability, reliability, and durability here. If our cache does not contain enough data (replicas) of a user, it can always be retrieved from the back-end storage.

Like with Hadoop, the data we are targeting has the property of "write once, read many times" because it is often extremely large and difficult to modifying or editing. Usually, the files in such a system are read-only or append-only. Therefore, strong consistency between the cached data and back-end (S3) is not required.

Because the user data is stored in the physical machines that host the VMs of different users, the system needs to secure this data. In other words, the cache needs to be isolated from the local hard drives allocated to VMs.

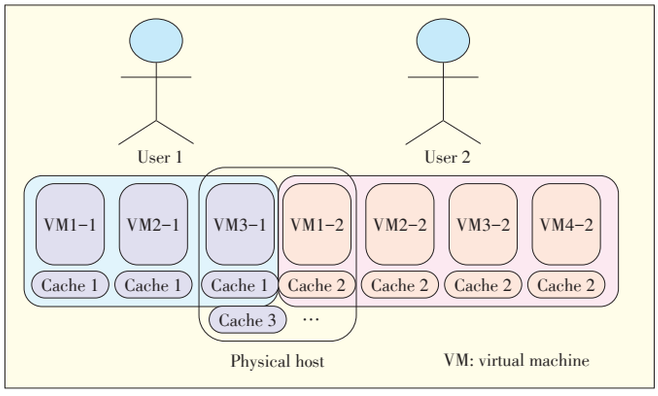
3.2 Solution

VMs and all retrieved data vanish after the job flow has finished. Therefore, to reduce wasted data movement, this fetched data is cached persistently in the computing cluster. The data should not be deleted alongside the VM termination. VMs are hosted on physical machines, which reserve a part of their local hard drives for this cache. This part is separated from the partitions for VMs. Another option is to use EBS because it is

also persistent.

Fig. 3 shows how a user views the system. Each user has a set of VMs, and each VM has its own cache. Many VMs and caches of different users can share the same physical host.

However, the cluster comprises many physical servers, and



▲**Figure 3. The system from user's point of view.**

most cloud users only have their VMs running on a very small part of the cluster. This means that a user's data is only stored on a small number of physical servers in the cluster. There is no guarantee that a user's VMs are hosted on the same set of physical servers between two different job flow executions. This may result in the cached data being unavailable to the user.

There are two possible solutions to this problem. We can move the missed retrieved data to the machines that host the user's new VMs or we can modify the VM scheduler to host the user's VMs on the same set of physical machines among different job flows. Each solution has its pros and cons. In the first solution, although the VM scheduler does not need to be modified, we need to keep track of the location of the cached data and the new VM hosts of all users. With the new VM-host mapping, we can then identify which parts of the cached data are missing and copy or move them to the new hosts. In the second solution, the data does not need to be moved, but the load in the system may become unbalanced because of the affinity of VMs for a set of physical servers. Even if the preferred physical machines of the return client do not have enough remaining resources, the system still has to put VMs into other available physical machines. In our implementation, we take the second approach. However, the delayed-scheduling approach [17] can also be taken to achieve both locality and fairness.

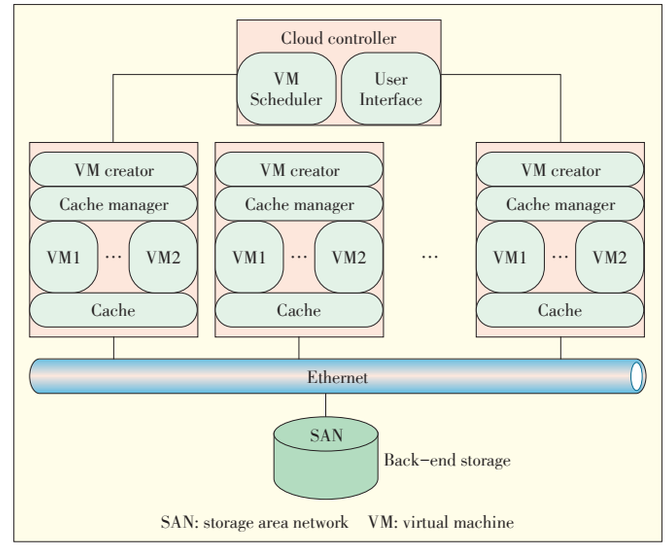
Fig. 4 shows the logical view of the system. We employ the traditional master-slave model, which also matches with Eucalyptus (on which we implement our prototype). The master node, also called front-end in Eucalyptus, is the cloud controller in Fig. 4 and contains the web service interface and VM scheduler. The scheduler communicates with the nodes in order to start VMs on them. Each node also uses web services to handle requests from the scheduler. On the nodes are running VMs, the cache partition (system cache), the VM creator (Xen, KVM, libvirt), and the cache manager. The cache manager attaches the cache to the VM and also implements replacement algorithms, such as FIFO and LRU.

When a user submits a job, the VM scheduler system tries to allocate their VMs to the physical hosts that already contained their data. Otherwise, if the user is new and has not uploaded any data to the system (i.e. HDFS, not S3), the system can schedule their VMs to any available hosts that do not trigger the cache-replacement process (or only trigger the smallest part of it).

To ensure isolation, the cache is allocated to separate partitions on the physical hosts. During the cache-replacement process, the cache is automatically attached to the VMs according to the appropriate user, and it persists after the VM has been terminated. Another design-related decision is whether each user should have a separate partition in the hosts or if all users should share the same partition as a cache. For security and isolation reasons, we choose the former.

The size of the cache partition is important and can be varied or fixed. For simplicity, we fix the cache size for each user (user cache). However, the partition for the cache at a physical machine (system cache) can be dynamic. The same physical machine can host a different number of VMs depending on the VM types (m1.small, c1.large, etc.). The user cache size is also proportional to the VM types, although it is fixed. Different VM types have different fixed user cache sizes. To ensure fairness, all users have the same sized cache for the same type of VM.

The size of the system cache is also important. If it is too small, it does not efficiently reduce data movement because the system has to replace old data with new. The old data does not have many chances to be reused. If it is too large, part used for the ephemeral storage of the VMs (regular storage of the



▲**Figure 4. Logical view of the system.**

VMs) is reduced. Hence, the number of VMs available for hosting on a physical machine is also reduced. In fact, the cache size depends on the system usage parameters, such as average number of users and user cache size. If the system has too many simultaneous users or the user cache is too large, data replacement can be triggered too many times. We can extend the system cache or reduce the user cache to relieve this.

There are two cache-replacement levels in our system. The first is at the physical node and the second is across the whole system. When the local cache is full, the node should migrate the data in its cache system to other available machines. When the global cache is full, the old cache should be removed by following traditional policies, such as FIFO or LRU. There are two options available here: remove all the data of users or remove just enough data to accommodate the newest data.

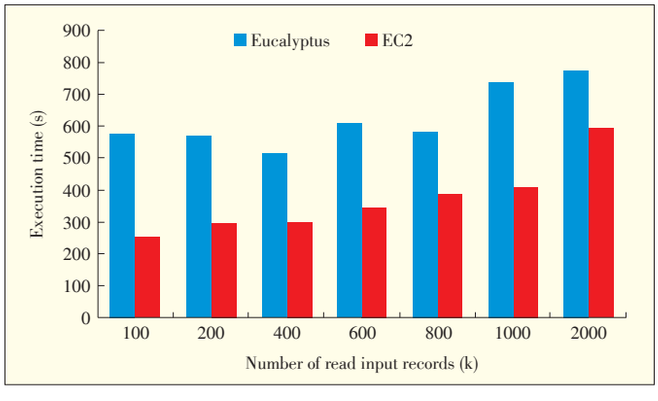
The cache-replacement mechanism depends on cache availability and VM availability. These two factors are independent. A physical host may not be able to host new VMs, but it may still have available cache (and verse versa).

4 Implementation

4.1 EMR versus Private Cloud

Because we do not have insight into Elastic MapReduQ, we use an open-source private cloud to measure the steps in the job flow. **Fig. 5** shows the execution time for our cloud and EC2 when running CloudBurst with the same data set and with the same VM configurations. We also used the same number of VMs with almost the same configurations. The correlation between our system and EMR is 0.85.

Fig. 5 shows that our system performs worse than EMR because of limitations in our network. According to [7], the bandwidth between EC2 instances (computing nodes) is 1 Gbit/s, and the bandwidth between EC2 and S3 is 400 Mbit/s. In our network, the average bandwidth between the physical nodes (computing nodes) is only 2.77 Mbit/s, between computing nodes and persistent storage node Walrus is about 2.67 Mbit/s, and between the client and front-end is 1.2 Mbit/s. CloudBurst itself is a network-intensive application [18].



▲Figure 5. Execution time for EC2 and Eucalyptus.

4.2 Our Implementation

Generally, to implement the proposed system, the cache partition has to be created at the physical nodes; the VM creation script has to be changed to attach the cache to the VM; and the VM scheduler of the cloud has to be modified. Creating a partition in any operating system should be a minor task. Most cloud implementations have libvirt toolkit for virtualization because it is free, supports many different operating systems, and supports the main hypervisors, such as Xen, KVM/QEMU, VMWare, Virtual Box, and Hyper-V. Therefore, the VM cache attachment task can be applied to almost any cloud. Modification of the VM scheduler depends on the implementation of the cloud. However, many well-designed implementations enable the administrator to easily add new schedulers. Such implementations also allow configurability so that the desired scheduler can be selected. In our implementation, we use Hadoop 0.20.2 and Eucalyptus 2.0.3, and the EMI image is CentOS.

To add our cache system to Eucalyptus, we need to create a cache partition on each physical node and specify the fixed size for each type of instance. We then mount the cache system to a suitable point of the VM so that Hadoop can access the VM. In our prototype, the size of the user cache is only 1.2 GB. For simplicity, we also only ran experiments with one type of instance.

Although the user cache is fixed, the system cache is not. To enable dynamic sizing of the system cache partition, we use a logical volume manager (LVM). To attach the user cache as a block device in a VM, we modify the VM XML creator script of Eucalyptus. The block device (cache) is not actually mounted to the file system; therefore, it cannot be used yet.

To mimic EMR, we need to create a script receiving information like job flow. This script then invokes other scripts to create instances, start Hadoop, run the job, and terminate the instances. The standard Eucalyptus VM image (EMI) does not have Hadoop; therefore, we modify the EMI so that Hadoop can be installed and configured, and we also enable the EMI to run the user script. This enables us to automate the starting/stopping script easily and to mount the user cache to the directory used by HDFS.

So far, we have only mimicked the EMR and prepared the cache storage at the node. The main task in our system is to schedule user VMs close to their data. When a user comes back to the system, the data should already be in the Hadoop cluster. To realize this, we modified the Eucalyptus scheduler.

We added code to the Eucalyptus cluster controller to record the map between the user and their VM locations. Then, we modified the VM scheduler so that if it detects a returning user with the same number of instances requested (by looking at the recorded map), it schedules the user's VMs to the previous locations if possible. If there are no resources remaining for the new VMs, the system uses the default scheduling policy

to move the corresponding cache to other available nodes and starts the VMs there. If there is no available cache slot in the whole system, we should swap the cache in the original node with the (oldest or random) one on the "available to run VM" node.

5 Performance Evaluation

5.1 Experiment Setup

We implemented a prototype cache system on Eucalyptus. Our testbed consisted of 12 machines, the configurations of which are shown in **Table 1**.

The workloads used in this section derive from CloudAligner, CloudBurst, Hadoop Sort benchmarks and Grep, which is a built-in MapReduce application used to search for an expression on the input files. The input for Sort is generated by RandomWriter. The input for Grep comprises the Hadoop log files and a Shakespeare play. The search expression is "the*".

The data for both CloudBurst and CloudAligner comprises two pieces. The first piece is the read sequences produced by a sequencer such as Illumina GAII, HiSeq 2000 or Pacific Biosciences. We obtained real data from the 1000 Genomes project; in particular, we used the accession SRR035459 file, which is 956 MB. We extracted subsets from that file to use in our experiments. **Table 2** shows the size and number of input splits as well as the size of the data in MapReduce Writable format and size of the original text file. Another piece of data is the reference genome. We choose chromosome 22 of the human genome (with original text size of 50 MB, and 9.2 MB of this is in MapReduce Writable format). This data is small and is only used here as a proof of concept. The real data is much larger. For example, there are 22 chromosomes in the human genome and normally, the alignment software needs to align the read to all of these.

▼**Table 1. Testbed configuration**

Type	Number of Machines	CPU	Memory (GB)	HDD (GB)	OS
Client	1	AMD 2 GHz	6	250	Ubuntu
Front-end	1	AMD Phenom II	8	500	CentOS
Computing	10	Xeon 2.80 GHz	2	40	CentOS

▼**Table 2. Size of the input read files**

Input Split (kB)	Writable (MB)	Text (MB)
100	4.3	5.2
200	8.6	11.0
400	18	21
600	26	32
800	35	42
1000	43	53
2000	86	106

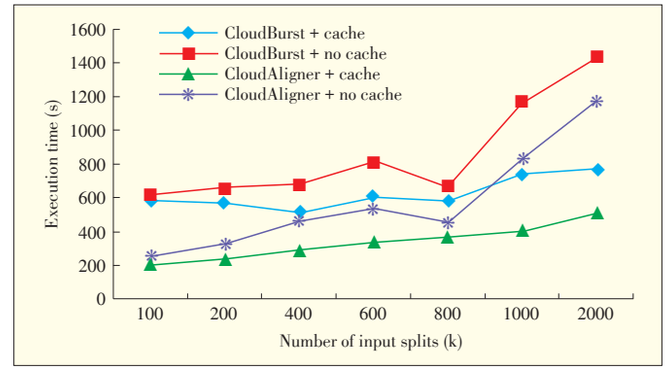
5.2 Improvements

To show the improvement in performance brought about by our cache system and the VM scheduler, we should ideally run experiments on data stored in Walrus and compare the results with those obtained using HDFS because with a warm cache, the data already exists in HDFS. However, the jets3t 0.6 library in the Hadoop version we used does not support communication with Walrus. As a result, we took the following two approaches.

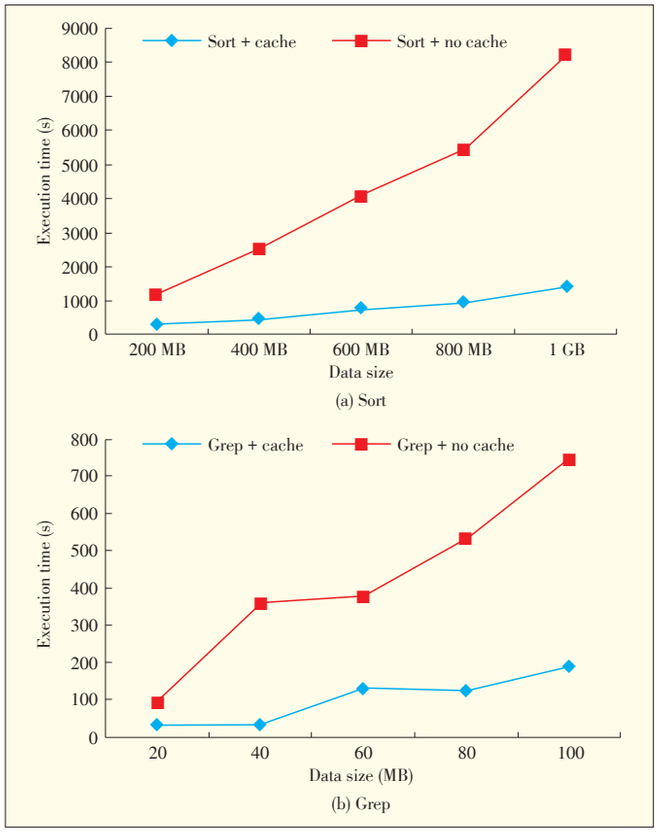
First, the data was moved from the Walrus server to HDFS, and data- movement durations were recorded. Without the cache, the data always has to be moved in; therefore, the execution time of a job flow should include this data- movement time. This approach is called addition.

Fig. 6 shows the performance of CloudBurst and CloudAligner when using and not using cache. When the data size increases, especially after 800 k, the difference between using and not using cache is wider because the data-movement time increases faster than the processing time. **Fig. 7** supports this observation. If the data is large and time to process it is small (due to MapReduce), the benefit of using our cache is greater. For example, in the Sort experiment, moving 1 GB of data to the system takes 6795 s in our network, but sorting it takes only 1405 s.

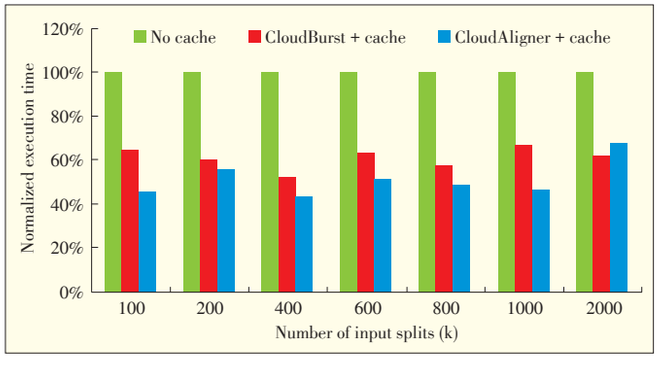
Second, we compare the results of running the same application with data in HDFS and in Amazon S3. In this case, the computing nodes are in our testbed, not at Amazon. We use our local cluster to process the data from HDFS and S3. This method is called HDFS_S3. The execution times for CloudBurst and CloudAligner with different configurations are shown in **Fig. 8**. Figs. 6 to 8 show the actual time to move and process the data; however, **Fig. 9** shows the relative performance for the applications without cache and with warm cache. This helps us see to what degree our system improves performance. For CloudAligner and CloudBurst, our system improves performance by up to 56.4% and 47.4%, respectively. Performance improvement in Sort and Grep is more significant



▲**Figure 6. Performance of the CloudAligner and CloudBurst applications with and without warm cache when addition method is used.**



▲Figure 7. Performance of the Sort and Grep with and without warm cache when addition method is used.

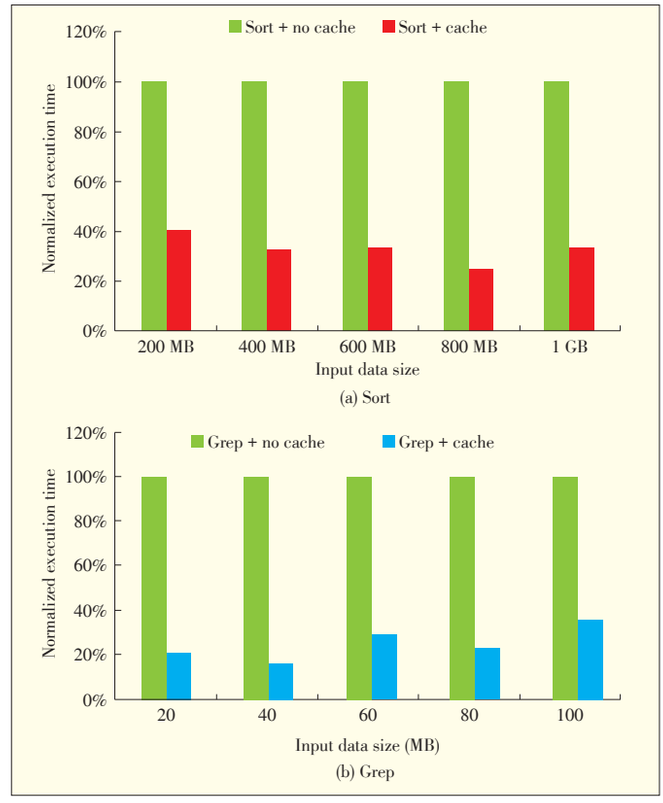


▲Figure 8. Performance of CloudAligner and CloudBurst with warm cache when HDFS_S3 method is used.

because these need less time to process larger data. In particular, the Sort application can benefit up to 75.1%, and the number of the Grep application is 83.7%. This result makes sense because all Grep does is just to cut the large data into small parts and to search each part in parallel for the pattern linearly.

5.3 Overhead and Scalability

To the old system we introduced new images, added a cache partition to the VM, mounted the cache partition, started Hadoop, and modified the scheduler. We made modifications in



▲Figure 9. Performance of Sort and Grep with warm cache when HDFS_S3 method is used.

three main areas: the cluster controller, the node controller, and the VM. In terms of VM startup time, overhead is negligible but may arise from the differences between the modified and original EMIs. These differences are in the installation of Hadoop, the VM startup xml (libvirt.xml), and the mounting script. This overhead is negligible because the installation only needs to be done once, and the size of the image does not change after this installation. The mounting script also contains only one Linux mount command, and the VM startup xml only has one additional device. In addition, the VM startup time is very small compared with the time needed to prepare for the instance (i.e. copy root, kernel, ramdisk files from the cache or from Walrus, create ephemeral file, etc).

In terms of execution time, overhead is also negligible because our additional scheduler is active only when the user has already visited the system. In addition, our scheduler improves the scheduling process because it does not need to spend time iterating each node in order to find an available node for the user request.

The complexity of our scheduler is given by $O(U+M\times N)$, where U is the number of EMR users, M is the maximum number of physical nodes that host a user's VMs, and N is the total number of physical hosts of the cloud.

Scalability can be expressed in terms of the number of users and size of the system (i.e. number of physical nodes). In terms

of system size, our solution does not affect the scalability of the existing cloud platform. If U is not taken into account, our scheduler has the same complexity as the original schedulers, which is given by $O(M \times N)$.

However, because we store user VM schedule plans, our approach does not scale well in terms of the number of users. Our work focuses on MapReduce applications only, so our cache system only serves a proportion of cloud users (not all cloud users use EMR).

Our cache solution is a best-effort solution, not an optimal one. This means that data can still move between different physical nodes if the previously scheduled physical nodes are not available. Without our solution, such movement would occur always.

6 Related Works

There is a distributed cache built into Hadoop [19], but our cache is different. With the Hadoop distributed cache, files are already in the HDFS. In our cache, the files are in the back-end storage system (S3). The data in [19] is cached between the map tasks and reduce tasks. In addition, the target platform in [19] is the small cluster, but ours is the cloud. Another recent work that describes cache in the cloud is [20]. Although this work mainly focuses on improving the caching of the VM image template, its concept can be directly applied to our case in order to further improve overall system performance.

EBS is suitable to use as our cache because it is independent from the instances. However, the current EMR does not support it. To use EBS, a user has to manually create suitable AMI with Hadoop, start it, attach the EBS, configure, start Hadoop, and run a job. If the user wants to run their job again, the whole process has to be repeated manually, even though the data remains on the EBS volumes.

The performance of MapReduce has received much attention recently. For example, in [21] and [22]m it is argued that Hadoop was designed for a single user and described ways of improving the performance of MapReduce in multiuser environment. A more recent work on Hadoop MapReduce for data-intensive applications is [23], but the authors evaluate the performance of data-intensive operations, such as filtering, merging, and reordering. Also, their context is high-performance computing, not cloud computing.

The closest work to ours on MapReduce and the cloud is [24]. In this work, Hamoud et al. also exploit data locality to improve the performance of the system. The main difference between their work and ours is that they schedule reduce tasks whereas we schedule VMs. In their evaluations, the authors of [25] use MapReduce, the cloud, and bioinformatics applications and in this context, their work is similar to ours. The authors propose Azure MapReduce as a MapReduce framework on Microsoft Azure Cloud infrastructure. Interestingly, their work supports our observation that running MapReduce on a bare metal cluster results in better performance than running MapReduce on a cloud-based cluster. It did not try to bridge this gap like ours.

7 Conclusion and Future Work

We have highlighted several existing issues with MapReduce in the cloud. We have also proposed a distributed cache system and VM scheduler to address the data-movement issue. By doing this, we can improve system performance. We have implemented a prototype of the system with Eucalyptus and Hadoop, and the experimental results show significant improvement in performance for certain applications.

If the user application associates with an extremely large data set that cannot fit into our distributed cache, our solution can still help reducing scheduling overhead. In this case, moving data from back-end storage is inevitable.

In the future, we plan to make Hadoop work with Walrus, obtain a real trace to measure cache hit or miss ratios, implement cache movement across the whole system, and monitor the removed data.

References

[1] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107-113, 2008. doi: 10.1145/1327452.1327492.

[2] R. M. Yoo, A. Romano and C. Kozyrakis, "Phoenix rebirth: Scalable MapReduce on a large-scale shared-memory system," *IEEE Workload Characterization Symposium*, pp. 198-207, 2009. doi: 10.1109/IISWC.2009.5306783.

[3] Y. Gu and R. L. Grossman, "Sector and Sphere: the design and implementation of a high-performance data cloud," *Philosophical Transactions of the Royal Society A: Mathematical,Physical and Engineering Sciences*, vol. 367, no. 1897, pp. 2429-2445, 2009. doi: 10.1098/rsta.2009.0053.

[4] *Apache Hadoop* [Online]. Available: http://wiki.apache.org/hadoop/

[5] B. He, W. Fang, Q. Luo, N. K. Govindaraju and T. Wang, "Mars: a MapReduce framework on graphics processors," in *Proc. 17th international conference on Parallel architectures and compilation techniques*, New York, NY, USA, 2008, pp. 260-269. doi: 10.1145/1454115.1454152.

[6] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff and D. Zagorodnov, "The eucalyptus open-source cloud-computing system," in Proc. *2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, Shanghai, May 18-21, 2009, PP. 124-131. doi: 10.1109/CCGRID.2009.93.

[7] T. Von Eicken, *Amazon EC2 Network and S3 performance* [Online]. Available: http://www.cloudiquity.com/2009/01/amazon-ec2-network-and-s3-performance/

[8] *File system layout* [Online]. Available: http://open.eucalyptus.com/learn/InstallingECC

[9] *Differences between S3 and EBS* [Online]. http://www.cloudiquity.com/2009/03/differences-between-s3-and-ebs/

[10] W. Huang, M. Koop and D. Panda, "Efficient one-copy MPI shared memory communication in virtual machines," in *Cluster Computing, 2008 IEEE International Conference on*, Tsukuba, Sept. 29-Oct.1 2008, pp. 107-115. doi: 10.1109/CLUSTR.2008.4663761.

[11] J. Liu, W. Huang, B. Abali and D. Panda, "High performance VMM-bypass I/O in virtual machines," in *Proc. USENIX Annual Technical Conference*, 2006.

[12] W. Huang, J. Liu, B. Abali and D. K. Panda, "A case for high performance computing with virtual machines," in *Proc. 20th annual international conference on Supercomputing*, New York, NY, USA, 2006, pp. 125-134. doi: 10.1145/1183401.1183421.

[13] S. Ibrahim, H. Jin, B. Cheng, H. Cao, S. Wu and L. Qi, "CLOUDLET: towards mapreduce implementation on virtual machines," in *Proc. 18th ACM interna-*

tional symposium on High performance distributed computing, New York, NY, USA, 2009, pp. 65-66. doi: 10.1145/1551609.1551624.
[14] S. Ibrahim, H. Jin, L. Lu, L. Qi, S. Wu and X. Shi, "Evaluating MapReduce on Virtual Machines: The Hadoop Case," in *Proc. 1st International Conference on Cloud Computing*, Berlin, Heidelberg, 2009, pp. 519-528. doi: 10.1007/978-3-642-10665-1_47.
[15] J. Shafer, S. Rixner and A. Cox, "The Hadoop distributed filesystem: Balancing portability and performance," in *Performance Analysis of Systems & Software (ISPASS), 2010 IEEE International Symposium on*, 2010, pp. 122-133. doi: 10.1109/ISPASS.2010.5452045.
[16] M. Schatz, "CloudBurst: highly sensitive read mapping with MapReduce," *Bioinformatics*, vol. 25, no. 11, pp. 1363-1369, 2009. doi: 10.1093/bioinformatics/btp236.
[17] M. Zaharia, D. Borthakur, J. Sarma, K. Elmeleegy, S. Shenker and I. Stoica, "Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling," in *EuroSys 2010*, New York, 2010, pp. 265-278. doi: 10.1145/1755913.1755940.
[18] T. Nguyen, W. Shi and D. Ruden, "CloudAligner: A fast and full-featured MapReduce based tool for sequence mapping," *BMC Research Notes*, vol. 4, no. 1, pp. 171, 2011.
[19] S. Zhang, J. Han, Z. Liu, K. Wang and S. Feng, "Accelerating MapReduce with Distributed Memory Cache," in *Parallel and Distributed Systems (ICPADS), 2009 15th International Conference on*, Shenzhen, China, 2009, pp. 472-478. doi: 10.1109/ICPADS.2009.88.
[20] D. Jeswani, M. Gupta, P. De, A. Malani and U. Bellur, "Minimizing Latency in Serving Requests through Differential Template Caching in a Cloud," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, Honolulu, HI, 2012, pp. 269-276. doi: 10.1109/CLOUD.2012.17.
[21] S. Seo, I. Jang, K. Woo, I. Kim, J. Kim and S. Maeng, "Hpmr: Prefetching and pre-shuffling in shared mapreduce computation environment," in *Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on*, New Orleans, LA, 2009, pp. 1-8. doi: 10.1109/CLUSTR.2009.5289171.
[22] M. Zaharia, D. Borthakur, J. Sarma, K. Elmeleegy, S. Shenker and I. Stoica, "Job scheduling for multi-user mapreduce clusters," *EECS Department*, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-55, 2009.
[23] Z. Fadika, M. Govindaraju, R. Canon and L. Ramakrishnan, "Evaluating Hadoop for Data-Intensive Scientific Operations," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, Honolulu, HI, 2012, pp. 67-74. doi: 10.1109/CLOUD.2012.118.
[24] M. Hammoud, M. S. Rehman and M. F. Sakr, "Center-of-Gravity Reduce Task Scheduling to Lower MapReduce Network Traffic," in *IEEE CLOUD*, Honolulu, HI, 2012, pp. 29-58. doi: 10.1109/CLOUD.2012.92.
[25] T. Gunarathne, T.-L. Wu, J. Qiu and G. Fox, "MapReduce in the Clouds for Science," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, Indianapolis, IN, 2010, pp. 565-572. doi: 10.1109/CloudCom.2010.107.

Manuscript received: April 22, 2013

Biographies

Tung Nguyen (tnguyen@i-a-i.com) is a research scientist at Intelligent Automation Inc. He plays a key role in many projects on data-intensive distributed processing, cloud computing, and mass-data analysis using topological features. Dr. Nguyen received his PhD degree from Wayne State University in 2012. He received his BS and MS degrees in computer science and engineering from Ho Chi Minh City University of Technology, Vietnam, in 2001 and 2006. His research interests include green computing, cloud computing, data-intensive computing, and application of cloud computing to life sciences. He has published several papers on computer science and bioinformatics and has been published in the proceedings of *OSDI* and in *NPC*, *SUSCOM*, and *BMC Frontiers Genetics* journals. He has also been a peer reviewer at many conferences, including *Euro-Par* and *CollaborateCom*. His homepage is http://www.cs.wayne.edu/tung/

Weisong Shi (weisong@wayne.edu) is an associate professor of computer science at Wayne State University. He received his BS degree in computer engineering from Xidian University in 1995. He received his PhD degree in computer engineering from the Chinese Academy of Sciences in 2000. His research interests include computer systems, mobile computing, and cloud computing. Dr. Shi has published 120 peer-reviewed journal and conference papers and has an H-index of 24. He has been the program chair and technical program committee member of numerous international conferences, including WWW and ICDCS. In 2002, he received the NSF CAREER award for outstanding PhD dissertation (China). In 2009, he received the Career Development Chair Award of Wayne State University. He has also won the Best Paper Award at ICWE'04, IPDPS'05, HPCChina'12, and IISWC'12.

ZTE Achieves TM Forum's Frameworx 12.0 Conformance Certification

19 November 2013, Shenzhen-ZTE Corporation has announced that ZTE's ZSmart BSS/OSS V8 has successfully achieved TM Forum's Frameworx 12.0 Conformance Certification.

TM Forum's Frameworx Conformance Certification is an independent verification of how a solution's business processes and information models align with the industry standards found in Frameworx, including the Business Process Framework (eTOM) and Information Framework (SID).

ZTE ZSmart BSS/OSS V8 is the latest IT software and service suite developed by ZTE, which can provide end-to-end operation support for operators. Based on the unified SOA architecture, ZSmart BSS/OSS provides comprehensive functional components and product suites that can be flexibly combined and configured to help operators cope with rapidly changing markets and customer requirements. In addition, ZSmart BSS/OSS V8 provides many innovative solutions, including intelligent charging and policy control which can be adopted to help operators realise intelligent traffic operation and precise marketing, complete OSS upgrades and IT reconstruction, and explore new commercial models to face the new challenges of the age of data.

ZTE has always promoted the standardisation of telecommunication products and has applied for more than 40,000 domestic and international patents. More than 90 percent of all patents held by ZTE have covered basic patents for international telecommunication technology standards. Over 60 percent of those patents are in new technology fields such as LTE, cloud computing, the Internet of Things and smart terminals. (ZTE Corporation)

DOI: 10.3969/j. issn. 1673-5188. 2013. 04. 004
http://www.cnki.net/kcms/detail/34.1294.TN.20131127.1347.001.html, published online November 27, 2013

Preventing Data Leakage in a Cloud Environment

Fuzhi Cang, Mingxing Zhang, Yongwei Wu, and Weimin Zheng
(Department of Computer Science and Technology, Tsinghua National Laboratory for Information Science and Technology Tsinghua University, Beijing 100084, China)

Abstract

Despite the multifaceted advantages of cloud computing, concerns about data leakage or abuse impedes its adoption for security-sensitive tasks. Recent investigations have revealed that the risk of unauthorized data access is one of the biggest concerns of users of cloud-based services. Transparency and accountability for data managed in the cloud is necessary. Specifically, when using a cloud-host service, a user typically has to trust both the cloud service provider and cloud infrastructure provider to properly handling private data. This is a multi-party system. Three particular trust models can be used according to the credibility of these providers. This paper describes techniques for preventing data leakage that can be used with these different models.

Keywords

cloud computing; data leakage; data tracking; data provenance; homomorphic encryption

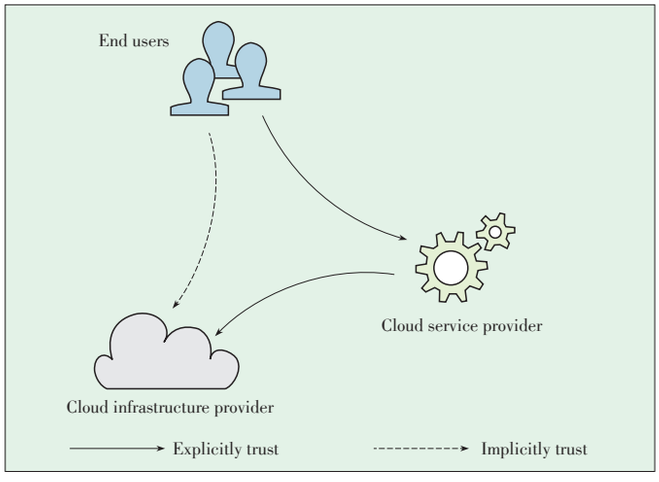
1 Introduction

Data protection is a top priority in the business world, especially in today's electronic era. In prior times, critical information, such as accounts and treaties, were recorded on paper and kept in safes. Such documents could not be stolen without great effort. However, such documents are now saved as digital bits and stored, transferred, and even traded using computers. As hacking methods become more advanced and electronic technologies become more complicated, the risk of important records being compromised is growing.

Compounding the problem is the fact that cloud computing has various benefits that have led to it being adopted at rapid speed by both companies and individuals. This quick uptake further weakens the user's control over their data. In the cloud-computing model, customers plug into the cloud via a network and access a shared pool of computing resources (e.g. networks, servers, storage, and applications) to process their data or hosting their own services. Although on-demand pricing and scaling are attractive, the fact that private customer data is handled and stored on systems outside the customer's control should not be forgotten. This is especially true for companies, such as Dropbox, whose entire business is built upon infrastructure as a service (IAAS). Such companies depend on the honesty and competency of the IaaS provider for data security. This honesty and competency has been called into question with a number of past security incidents and data breaches [1]-[3]. However, even if a service provider is considered trusted and follows best security practices, unauthorized access due to software bugs or misconfiguration is still a threat. A great deal of work has been done on the prevention, detection, and mitigation of such risks. Data owners want to have transparency and accountability so that they can see how their data is being handled by the cloud-hosted service and ensure their data is not being abused or leaked.

An investigation by M. Armbrust et al. [4] revealed that data confidentiality was the third-biggest obstacle to faster adoption of cloud computing. In 2010, Fujitsu Research Institute conducted a survey on potential cloud customers [5] and found that 88% of these potential customers were worried about who could access their data and demanded more awareness of what goes on in the back end.

In recent years, many techniques have been proposed to address these challenges, which, in our survey, divided into three main categories. There are three distinct participants in a typical cloud computing scenario: end users, cloud service provider, and cloud infrastructure provider (**Fig. 1**). In this work, we refer to both IaaS and PaaS providers as cloud infrastructure providers because they all provide base infrastructures for online services that are hosted on them. In this scenario, users need to implicitly trust the service provider to secure data. Users also need to implicitly trust the cloud infrastructure provider that hosts these services. Thus, three trust models can be established according to the credibility of these providers. In the first model, both the cloud service provider and cloud infrastructure provider are trusted (e.g. when using a private cloud). In the second model, the cloud infrastructure provider is usually a large and reputable company, like Google or Amazon), and is trusted. However, the cloud service providers are not trust-



▲Figure 1. Participants in cloud Computing.

ed. In the third model, Neither the cloud infrastructure nor the cloud service provider is trusted. In the remainder of this paper, we discuss current mechanisms that can be used when these different trust models are applied.

2 Tracking Data When Services are Trusted

If both the cloud service provider and cloud infrastructure provider are trusted, or when a private cloud is used, the trust scenario is not much worse than if a cloud were not being used. However, unauthorized access caused due to software vulnerabilities or bugs in the cloud service are could still cause data leakage. To close all loopholes, a data-tracking technique is proposed. With a standalone data-tracking system, all data access is traced and recorded. The administrator can prevent unauthorized data access by checking the operator's privilege before an operation is actually executed. An administrator can also audit a user's behavior by checking the logs afterward. This is an added security measure that alleviates user concerns about security in industries such as banking.

Although this tracking can be done within various layers of abstraction, the best choice is the file abstraction layer. Specifically, in a file-level tracking system, access to every file is intercepted before any action is taken, and relevant information is recorded for further analysis after the execution. This technique creates low additional overhead, is highly effective, and has already been adopted by several popular cloud providers. For example, Rackspace offers a feature called Access Log Delivery, which allows a user to enable logging for their private cloud files containers. Rackspace's blog states, "Customers sharing an account with multiple users can track which users are accessing their data, which are uploading the most content, etc. This gives IT departments more information for better serving their customers and possibly identifying problem users." [6]

Fig. 2 shows a typical architecture for those who want to implement a file-level tracking system from scratch. In this system, clients are only permitted to access cloud utilities indirectly via a portal. When a remote procedure call (RPC) is invoked by the client, the portal queries a database to determine whether a user has the authority to perform an operation. If the user does not have the authority, the request is denied. If the user does have the authority, subsequent operations are executed in the cloud via the application programming interfaces (APIs) provided by the service, and results are returned to the client. The portal also records information such as the user's IP address and operation type in a log database for further reference. If a smaller granularity, such as byte-level granularity, is desired, the range of read/write operations should also be recorded.

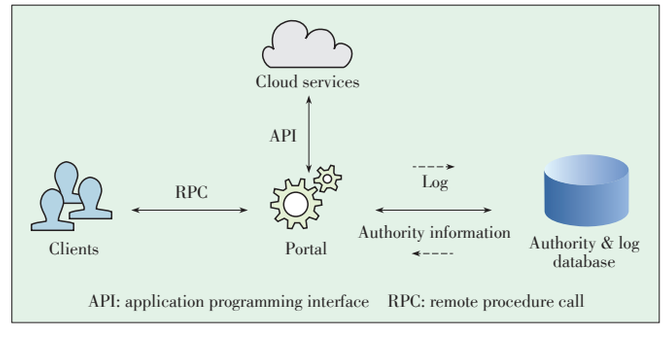
However, tracking data access may not be enough. Even applications are authorized to read a user's data and access output channels, leakage is still possible because an authorized application may be malwares or have been hijacked by another unauthorized program. In an attempt to prevent such leakage, much research has been done on information flow tracking (IFT). Privacy Scope is a novel system for tracking the movement of sensitive user data as it flows through applications [7]. It empowers users to run applications in their own environment and pinpoints information leaks, even when the data is encrypted. Privacy Scope systems are usually based on dynamic taint analysis, which is beyond the scope of this paper.

3 Auditing a Cloud Service Provider when the Cloud Infrastructure Provider is Trusted

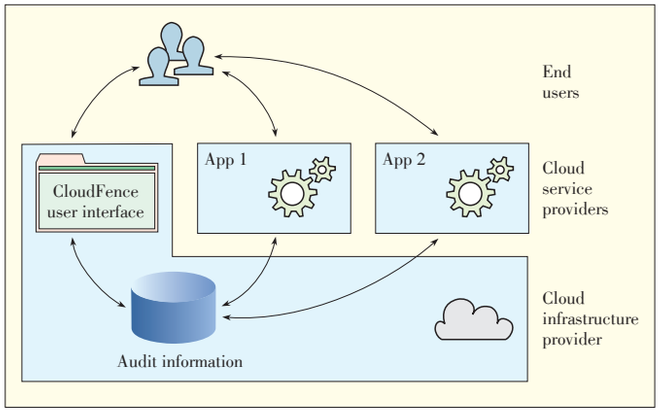
Compared with lesser-known companies that provide online services, big cloud infrastructure providers, such as Amazon, Google, and Microsoft, are well-known and highly reputable. Customers usually feel more comfortable handing control of their data to such providers. By enabling both direct and indirect users to inspect data uploaded to the cloud infrastructures, these companies create a direct relationship between the end user and themselves. This increases user confidence.

3.1 CloudFence

Much research has been done in this area. **Fig. 3** shows a high-level overview of CloudFence [8] as well as the main in-



▲Figure 2. The architecture of a file-level tracking system.



▲Figure 3. Overview of CloudFence.

teractions between parties involved in a CloudFence service. First, a user acquires a universally unique ID from the cloud infrastructure provider. Then, the user's sensitive data is tagged by the service provider with the supplied user ID and is tracked throughout the cloud infrastructure. At the same time, audit information is gathered and stored in the cloud infrastructure provider's database. CloudFence relies on data flow tracking (DFT) techniques to track the flow of data within all participating services hosted by the same cloud provider and to generate audit logs. Each cloud-hosted service has a separate audit log that can be correlated with other logs to create an audit report for the data owner. At any time, a user can monitor their data by using a user-friendly web interface or API. The main component of CloudFence is a DFT subsystem, which provides explicit, fine-grained, byte-level data flow tracking, and no modification of applications or underlying OS is needed. The subsystem can handle 232 different tags at the same time. It uses Intel Pin, a dynamic binary instrumentation toolkit, to instrument all instructions that operate tagged data, and it injects the tag propagation logic before the corresponding instructions. Both the original and additional instrumentation codes (i.e. the data tracking logic) are re-translated using Pin's just-in-time compiler.

The CloudFence system offers service providers an extra feature that reinforces the trust relationship between them and their users. It can also be used as the basis of a tag-based security system that prevents inadvertent leaks or unauthorized data access. The cloud infrastructure provider can improve transparency for their users by integrating CloudFence into their infrastructure, and this potentially increases the customer base.

3.2 Provenance

Besides the tags used in CloudFence, there are several other systems that provide similar functionality (called provenance). In [9], provenance is also referred to as lineage and pedigree and is defined as "the information that helps determine the derivation history of a data product, starting from its original resources." The provenance of a data block mainly includes information about where the data is derived and how the root ancestors transformed into the current data. Provenance-aware Storage System (PASS) [10] is one of the most-cited early provenance systems. In automatic provenance collection and maintenance, provenance is treated as a first-class object. The second PASS prototype [11] allowed the original system to operate one level of abstraction (which better supports a user's need to determine their data's movement) and answer questions that require an integrated view of the provenance. To facilitate cloud computing, Macko et al. extend the second PASS prototype and modify the Xen hypervisor to collect provenance from running guest kernels [12].

Adding provenance mechanisms to cloud computing and storage is essential to improve transparency and ensure accountability for data managed in a cloud. Provenance mechanisms are crucial for further adoption of cloud services [9].

4 Protecting Data Without Trusting a Cloud Service Provider

In light of past security incidents and the disclosure of the NSA's PRISM program, even big, reputable online services cannot be fully trusted. Thus, the most sensitive data, such as passwords, should not be devolved to cloud providers. The question becomes: Can we still leverage the benefits of cloud computing without needing to trust the cloud provider?

4.1 Fully Homomorphic Encryption

The most straightforward answer to this question is encryption. If all data is encrypted before being uploaded to the cloud, and the key is kept with the user, there is no need to worry about the risk presented by the cloud provider. However, even though encryption helps secure data stored in the cloud, it does not necessarily protect the data when it is being processed on remote infrastructure. That is, although the data can be stored in an encrypted form in the cloud, it generally has to be decrypted before being processed. The data is vulnerable when processed in a cloud. To address this challenge, fully homomorphic encryption, has emerged. In the future, this technique might allow encrypted data to be directly processed in the cloud.

The basic idea of this technique is that a specific encryption function allows a defined set of operations (e.g. addition, multiplication) to be performed directly on the encrypted data to produce an encrypted result. When this result is decrypted, it is the same as that obtained by operating on plain-text data.

In other words, for the numbers a and β, the suitable encryption function is $Encrypt(x)$, and the corresponding decryption function is $Decrypt(x)$. If $a \oplus \beta = \gamma$ and $Encrypt(a) \oplus Encrypt(\beta) = \gamma^*$, then $Decrypt(\gamma^*) = \gamma$, where $\oplus$ is a binary function supported by the encryption function. Therefore, if addition is supported, two encrypted values are

added to get a sum that, when decrypted, is the sum of the original two numbers. To be fully homomorphic, the code must allow a third party to add and multiply numbers contained within it without the need for decryption.

This may sound like a magic. In fact, until 2009, no one was sure whether homomorphic encryption was even possible. Then, a Stanford student named Craig Gentry proved its practicability in his PhD thesis [13]. This was a step in the right direction, but Gentry only provided an existence proof that showed that homomorphic encryption was no longer impossible. He did not implement the encryption in practice. Gentry estimated that performing a Google search with encrypted keywords would increase the amount of computing time a trillion times, which is unacceptable.

However, since Gentry's work, there have been a number of improvements that have made the scheme practical. At the end of 2009, M. Dijk et al. presented a simplified, fully homomorphic system called Brakerski-Gentry-Vaikuntanathan (BGV) homomorphic encryption scheme. This scheme uses integers only [14].

Moving forward, Victor Shoup and Shai Halevi, working at the IBM T J Watson Research Center, have just released an open source (GPL) C++ library called Helib. The code incorporates many optimizations to make the encryption faster [15]. As described in its package description, "HElib is a software library that implements homomorphic encryption (HE). Currently available is an implementation of the Brakerski- Gentry-Vaikuntanathan (BGV) scheme, along with many optimizations to make homomorphic evaluation run faster, focusing mostly on effective use of the Smart- Vercauteren ciphertext packing techniques and the Gentry-Halevi-Smart optimizations."

4.2 Trusted Cloud Computing

Besides advanced encryption, many other methods have been proposed for achieving trusted cloud computing. These methods involve a trusted cloud provider attesting service to end users. Although cloud service providers are making substantial efforts to secure their systems, insiders that manage these software systems at the provider's backend ultimately still have the technical means of accessing customer VMs [16]. Thus, there is a pressing need for a technical solution that guarantees confidentiality and integrity during processing and that can be verified by the customers.

To this end, the Trusted Computing Group (TCG) has proposed a set of hardware and software technologies that enable the construction of trusted platforms [17]. Specifically, the TCG proposed a standard for the design of the trusted platform module (TPM) chip, which is often bundled with commodity hardware. The TPM contains a private key that uniquely identifies the TPM, and it also contains some cryptographic functions that cannot be modified. Just like secure sockets layer (SSL), some manufacturers sign the corresponding public key to guarantee the correctness and validity of the chip. Then, a trusted platform can be implemented by leveraging the features of TPM chips to build a TPM attestation chain. This enables trustworthy remote attestations. Typically, this mechanism works as follows [18]: At boot time, the host (a physical machine) computes a measurement list (ML) comprising a sequence of hashes of the software involved in the boot sequence (i.e. BIOS, bootloader, and the software implementing the platform). This ML is securely stored inside the host's TPM, which cannot be modified even by the administrator of the machine. Then, a remote client can verify the platform in the following three steps:

1) The client sends the platform running at the host with a nonce N
2) The platform asks its local TPM to create a message containing both the stored ML and the received nonce N. Then, the platform encrypts the message with the TPM's private key.
3) The host sends the message back to the remote client, which can decrypt the message using the TPM's corresponding public key. This authenticates the host. This is achieved by checking that the nonce matches and the ML corresponds to a configuration that deems to be trusted.

Now the remote client can reliably identify the platform on an untrusted host, and this platform secures the users' data. For example, the trusted platform Terra implements a thin virtual machine manager (VMM) that enforces a closed-box execution environment [18]. This means that a guest VM running on it cannot be inspected or modified by a user with full privileges over the host. To put it simply, TPM allows a user to reliably detect whether or not the host is running a platform that can be trusted by the remote party. This trusted platform effectively secures the VMs running on it.

This attesting chain can be broken if the administrator can migrate the VM from one tested host to another untrusted host in the cloud environment. Therefore, Santos et al. [16] proposed the Trusted Cloud Computing Platform (TCCP), which provides a closed-box execution environment by extending the trusted platform to an entire IaaS back end. The TCCP guarantees confidentiality and integrity for a user's VM and allows a user to determine up front whether the IaaS enforces these.

5 Conclusion

This paper describes current mechanisms for preventing data leakage, one of the biggest concerns with using cloud-hosted services. Transparency and accountability for data managed in the cloud needs to be improved so that users can be more confident in the safety of their data stored in the cloud.

Usually, a user has to trust both the third-party cloud service provider and the cloud infrastructure provider to properly handle the user's data. This is a multiparty system. Three trust models can be used according to the credibility of these providers. There are techniques that can be used for each of these models.

When both the cloud services provider and cloud infrastructure provider are trusted, the main cause of data leakage is unauthorized access. However, this can be prevented by using a file-level tracking system. With a standalone tracking system, all data access attempts must be authorized. The administrator is empowered to audit user behaviors by checking recorded information.

If the cloud infrastructure provider is trusted but the cloud service provider is not, the infrastructure provider can be more transparent by offering auditing systems that can be used by indirect users to inspect the movement of their data. This kind of method often involves additional meta information, such as tags or provenance, that is propagated with the data.

It is still possible to use the cloud services without handing over control of data. Fully homomorphic encryption is an encryption technique that allows encrypted data to be directly processed in the cloud without decryption. Much effort has been put into achieving trusted cloud computing, which leverages the TPM to build a chain of trust that allows users to remotely check on trustworthiness.

Preventing data leakage must not only be limited to a single cloud service provider solution. Instead, data movement within multiple clouds or between the internet and cloud should also be investigated.

Acknowledgements

This Work is supported by National Basic Research (973) Program of China (2011CB302505), Natural Science Foundation of China (61373145, 61170210), National High-Tech R&D (863) Program of China (2012AA012600, 2011AA01A203), Chinese Special Project of Science and Technology (2012ZX01039001).

References

[1] Computerworld. (2010 Dec). *Microsoft BPOS cloud service hit with data breach* [Online]. Available: http://www.computerworld.com/s/article/9202078/Microsoft_BPOS_cloud_service_hit_ with_data_breach
[2] Sophos. (2011 Jun). *Groupon subsidiary leaks 300k logins, fixes fail, fails again* [Online]. http://nakedsecurity.sophos.com/2011/06/30/groupon- subsidiary-leaks-300k-logins-fixes- fail-fails-again/
[3] The Wall Street Journal. (2009 Mar). *Google Discloses Privacy Glitch* [Online]. Available: http://blogs.wsj.com/digits/2009/03/08/1214/
[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, pp. 50-58, 2010. doi: 10.1145/1721654.1721672.
[5] Fujitsu Research Institute. (2010). *Personal data in the cloud: a global survey of consumer attitudes* [Online]. Available: http://www.fujitsu.com/downloads/SOL/fai/reports/fujitsu personal- data-in-the-cloud.pdf
[6] Rackspace. (2012). *Logging for private cloud files containers* [Online]. Available: http://www.rackspace.com/blog/logging-for-private-cloud-files-containers/
[7] D. Zhu, J. Jung, D. Song, T. Kohno, and D. Wetherall, "Privacy scope: a precise information flow tracking system for finding application leaks," Department of Computer Science, UC Berkeley, Tech. Rep. EECS-2009-145, 2009.
[8] V. Pappas, V. P. Kemerlis, A. Zavou, M. Polychronakis, and A. D. Keromytis, "CloudFence: data flow tracking as a cloud service," *Proc. 16th Int. Symp. Research in Attacks, Intrusions and Defenses (RAID)*, Saint Lucia, October 2013. doi: 10.1007/978-3-642-41284-4_21.
[9] OQ Zhang, M Kirchberg, et al., "How to track your data: The case for cloud computing provenance," *3rd IEEE Int. Conf. Cloud Computing Technology and Science*, Athens, Greece, 2011, pp. 446-453. doi: 10.1109/CloudCom.2011.66.
[10] K.-K. Muniswamy-Reddy, et al., "Provenance-aware storage systems," *Proc. USENIX Ann. Technical Conf.(General Track)*, Boston, MA, USA, 2006, pp. 43-56. doi: 10.1.1.110.1442.
[11] K.-K. Muniswamy-Reddy, U. Braun, D. A. Holland, P. Macko, D. Maclean, D. Margo, M. Seltzer, and R. Smogor, "Layering in provenance systems," *Proc. USENIX Ann. Technical Conf.*, San Diego, CA, USA, 2009, pp. 129-142. doi: 10.1109/SSDBM.2004.23.
[12] P. Macko, M. Chiarini, and M. Seltzer, "Collecting provenance via the Xen hypervisor," *3rd USENIX Workshop Theory and Practice of Provenance*, Heraklio, Greece, June 2011.
[13] C. Gentry, "A fully homomorphic encryption scheme," Dissertation, Stanford Univ., CA, USA, 2009.
[14] M. V. Dijk, et al., "Fully homomorphic encryption over the integers," in *Advances in Cryptology-EUROCRYPT 2010*, Germany: Springer Berlin Heidelberg, 2010, pp. 24-43. doi: 10.1007/978-3-642-13190-5_2.
[15] HElib. [Online]. Available: https://github.com/shaih/HElib
[16] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing," *Proc. 2009 conf. Hot topics in cloud computing*, San Diego, CA, USA, June 2009, Article no. 3. doi: 10.1.1.149.2162.
[17] *Trusted Computing Group* [Online]. Available: https://www.trustedcomputinggroup.org
[18] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpa ch, I. Pratt, and A. Warfield, "Live migration of virtual machines," *Proc. 2nd ACM/USENIX Symp. Networked Systems Design and Implementation (NSDI)*, Berkeley, CA, USA, 2005, pp. 273-286. doi: 10.1.1.59.6685.

Manuscript received: October 10, 2013

Biographies

Fuzhi Cang (cfz05@mails.tsinghua.edu.cn) received his BE degree in computer science and technology from Tsinghua University in 2009. He is currently a MS candidate in Department of Computer Science and Technology, Tsinghua University, China. His research interests include distributed systems and cloud security.

Mingxing Zhang (zhangmx12@mails.tsinghua.edu.cn) received his BS degree in computer science and technology from Beijing University of Posts and Telecommunications in 2012. He is currently a PhD student in computer science at Tsinghua University, China. His research interests include distributed and parallel systems.

Yongwei Wu (wuyw@tsinghua.edu.cn) received his PhD degree in applied mathematics from the Chinese Academy of Sciences in 2002. He is currently a professor in computer science and technology at Tsinghua University, China. His research interests include parallel and distributed processing, and cloud storage. Dr. Wu has published more than 80 research papers and has received two Best Paper Awards. He is currently on the editorial board of the *International Journal of Networked and Distributed Computing and Communication* of China Computer Federation. He is an IEEE member.

Weimin Zheng (zwm-dcs@tsinghua.edu.cn) is a professor of computer science and technology at Tsinghua University, China. He received his BS degree and MS degree from Tsinghua University in 1970 and 1982. He is currently the director of the Chinese Computer Society. His research interests include computer architecture, operating systems, storage networks, and distributed computing. He is a senior member of the IEEE.

DOI: 10.3939/j.issn.1673-5188.2013.04.005
http://www.cnki.net/kcms/doi/10.3939/j.issn.1673-5188.2013.04.000.html, published online December 12, 2013

CPPL: A New Chunk-Based Proportional-Power Layout with Fast Recovery

Jiangling Yin, Junyao Zhang, and Jun Wang
(Department of Electrical Engineering & Computer Science, University of Central Florida, Orlando, Florida 32816, USA)

Abstract

In recent years, the number and size of data centers and cloud storage systems has increased. These two corresponding trends are dramatically increasing energy consumption and disk failure in emerging facilities. This paper describes a new chunk-based proportional-power layout called CPPL to address the issues. Our basic idea is to leverage current proportional-power layouts by using declustering techniques. In this way, we can manage power at a much finer-grained level. CPPL includes a primary disk group and a large number of secondary disks. A primary disk group contains one copy of available datasets and is always active in order to respond to incoming requests. Other copies of data are placed on secondary disks in declusterd way for power-efficiency and parallel recovery at a finer-grained level. Through comprehensive theoretical proofs and experiments, we conclude that CPPL can save more power and achieve a higher recovery speed than current solutions.

Keywords

power proportionality; parallelism recovery; declustering; layout

1 Introduction

To satisfy the requirements for performance, reliability, and availability in large-scale data analytics, distributed processing frameworks, such as Yahoo! Hadoop and Google MapReduce, have been adopted by many companies. The increasing number and size of data centers raises the problem of power consumption. To reduce power consumption, efficiency must be increased, and low-power modes for underutilized resources are required. Power proportionality has been used to evaluate the efficiency of computer power usage [1]-[3]. Power proportionality means that power used should be proportional to the amount of work performed, even though the system is provisioned to handle a peak load.

Achieving power proportionality in a CPU requires dynamic voltage scaling [1], [4]-[6]. However, achieving power proportionality in a storage system is very difficult because most hard drives do not work in multipower states. There are very few dual-rotation-rate hard drives on the market, and it is impossible to finely scale the power consumed by disks. A feasible alternative for a large data center is to use dynamic server provisioning to turn off unnecessary disks and save power [7]-[10]. A CPU's dynamic provisioning schemes depend on the specific data layout, but powering disks on and off does not. At any point, all active disks in a storage system need to contain an entire data set in order to guarantee uninterrupted service for incoming requests.

In recent years, several research efforts have resulted in group-based proportional-power layout schemes for storage systems. Lu et al. [11] introduced a family of power-efficient disk layouts for simple RAID1 data mirroring. Thereskaet et al. [2] developed a power-aware data layout in which all servers are divided into groups of equal size, and each group contains a copy of the entire data set. Amur et al. [10] developed a layout that is different from the power-aware grouped layout and in which the size of each group is different. To achieve ideal power-proportionality, the groups comprise an increasing number of disks.

All of these layouts use disk/server groups for simplicity.

There are two main limitations for group-based proportional-power layouts. First, the whole disk group is either powered on or off, so unneeded disks often consume power and result in performance loss. Second, a group-based layout usually requires a whole group of disks to be powered on, even if a single disk fails. Recovering a failed disk is also slow because of limited recovery parallelism across groups and no data overlap within a group.

In this paper, we develop a new chunk-based power-proportional layout called CPPL to solve these problems. Our basic

This work is supported in part by the US National Science Foundation Grant CCF-0811413, CNS-1115665, CCF-1337244 and National Science Foundation Early Career Award 0953946.

idea is to leverage current power-proportional layouts by using declustering techniques to manage power much more finely. We summarize the contributions of our paper as follows:

1) We establish a set of theoretical rules to study the feasibility of implementing ideal power proportionality with practical data layouts. We also study the effects of ideal power proportionality on disk recovery.
2) We study specific CPPL layouts according to the theory and address power proportionality by configuring the overlapped data between the primary disks, i.e. disks in a primary replica group (p disks) and non-primary disks, i.e. disks not in the primary replica group (non-p disks).
3) We conduct comprehensive experiments using a disksim-based framework. Our experimental results show that CPPL can save more energy than current solutions and is capable of higher recovery speed than current solutions.

2 Power-Proportional Storage System

Fault tolerance and load balancing [12], [13] are of primary concern in a traditional storage system and are usually achieved by randomly placing replicas of each block on a number of disks comprising the storage system. Shifted declustering [14] is a concrete placement scheme that has these properties but also maintains mapping efficiency. However, the data is distributed, which means subsets of disks cannot be powered down to save energy without affecting data availability.

To discuss these properties by metric, we suppose that the total number of data chunks (data units) is Q, and each chunk has k replicas. The chunks are stored on a system that runs in a data center comprising n disks. We can number the data chunks with an ID $1,2,3,...,Q$ and name all the disks $1, 2,...,n$. The replicas of chunk i $(1 \leqslant i \leqslant Q)$ are stored on different disks, and (i, j) $(1 \leqslant i \leqslant Q$ and $1 \leqslant j \leqslant k)$ represents the jth replica of chunk i.

2.1 Fault Tolerance

We define v_θ as the number of overlapping chunks of θ disks. It is an important factor in disk recovery because if disks fail, only those containing overlapping chunks can provide recovery data for the failed disks. For example, if one disk fails, another active disk needs to provide data for recovery. For the best recovery perfomance, the failed disk should have at least one chunk that overlaps with all other active disks.

2.1.1 Distributed Reconstruction

Lemma: In a layout, if $\beta\ (2 \leqslant \beta \leqslant n)$ disks contain the same amount of distinct data δ_β, then β disks also have the same number of overlapping data chunks v_β.

Proof: We map β disks to β sets containing the ID of the data chunks and record them as $S=(S_1, S_2, \cdots, S_\beta)$. This statement can be proved by induction, and $|S_i|$ is the number of chunks in set S_i.

The basic case is $\beta=2$; $\delta_2 = |S_1 \cup S_2| = |S_1| + |S_2| - |S_1 \cap S_2|$; and $\delta_2 = |S_1 \cup S_2| = |S_1| + |S_2| - |S_1 \cap S_2|$. Because $\delta_2 = |S_1 \cup S_2| = |S_1| + |S_2| - |S_1 \cap S_2|$ (assuming that all disks have the same capacity) and all δ_2 are equal on this precondition, v_β is a constant for any two sets from S.

Assuming that this statement is true for $x-1$ $(2 \leqslant x-1 < \beta)$, then v_{x-1} is a constant for any number of sets smaller than $x-1$ in S. From the inclusion-exclusion principle, we have

$$\delta_x = \sum_{i=1}^{x} |S_i| - \sum_{i,j:1 \leqslant i < j \leqslant x} |S_i \cap S_j| + \ldots + \left((-1)^{x-1}\right) |S_1 \cap \ldots S_n| = \sum_{i=1}^{x} |S_i| - v_2 \times \sum_{i,j:1 \leqslant i < j \leqslant x} 1 + \ldots + \left((-1)^{x-1}\right) v_x \quad (1)$$

By precondition, δ_x is a constant and so v_x is also a constant. The proof is complete.

Theorem: If a layout can support parallel recovery for θ failed disks, then the disks will be laid with data chunks in such a way that the difference of ρ_θ should be at most 1 for $1 < \theta < n$.

Proof: Suppose the storage system conforms to a degradation model where x $(n > x \geqslant 1)$ disks fail. We rename these disks $(d_1, d_2, ..., d_x)$. The system needs to recover these disks by requesting available chunks from the remaining active disks. Through parallel recovery, the data chunks on the failed disks should be as distinct as possible so that there is more available data on the active disks. According to lemma 2, x failed disks will have the same v_x. The total v_x for n disks is $\binom{n}{x}$, and the total number of overlapping data chunks based on x is $\binom{n}{x} \times vx$.

To fully support fault tolerance, chunks with identical IDs should be stored on different disks. Thus, each different x chunk with identical chunk ID counts once, and the total count for Q different data chunks is $Q \times \binom{k}{x}$. The count from a disk and chunk perspective should be equal:

$$\binom{n}{x} \times v_x = Q \times \binom{k}{x} \quad (2)$$

where $v_x = Q \times \frac{k!}{x!(k-x)!} \times \frac{x!(n-x)!}{n!}$. We have $\rho_\theta = \lfloor v_x \rfloor or \lceil v_x \rceil$, $\lfloor v_x \rfloor - \lceil v_x \rceil = 0$ or 1, so the proof is complete.

2.1.2 Power-Proportional Reconstruction

In this section, we describe how a layout can satisfy the distributed reconstruction while supporting power proportionality. Specifically, we describe how the overlap of chunks changes between any two disks when the powering-up disks change. Suppose that m disks are active at time t and any pair of

disks $(i \text{ and } j)$ from m have overlap chunks of $x_{i,j}$. The number of paired disks among m active disks is $\sum_{i=0}^{m-2}\sum_{j=i+1}^{m-1}=\binom{m}{2}$. Thus, the sum of pair chunks among any two disks is $\binom{m}{2}\times x_{i,j}$. On the other hand, consider m active disks storing Q chunks in portions. The total number of paired chunks is $Q\times\left(\binom{\frac{m}{n}\times k}{2}\right)$.

A storage system should be highly available; that is, it should be k-way $(k\geqslant 2)$ replication storage that can at least provide $(k-1)$ failure correction. This means that there no two replicated chunks are located on any one disk. Thus, the sum of paired chunks from the two perspectives are equal:

$$\binom{m}{2}\times x_{i,j}=Q\times\left(\binom{\frac{m}{n}\times k}{2}\right) \tag{3}$$

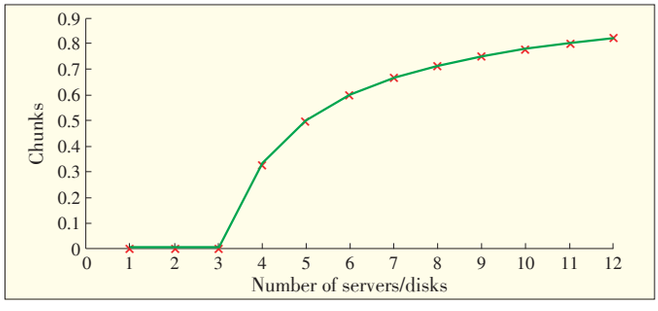
where $x_{i,j}=\frac{Q\times k\times(m\times k-n)}{n^2\times(m-1)}=\frac{Q\times k^2}{n^2}-\frac{Q\times k}{n^2}\times\frac{(n-k)}{m-1}$

In (3), k, Q and n in a stable storage system are usually constant, and $x_{i,j}$ is a function of m and increases with m. **Fig. 1** shows how $x_{i,j}$ changes with m. In Fig. 1, $k=4$, $n=12$, and $Q=9$. Values of $x_{i,j}$ below 0 are taken to be 0 because a negative $x_{i,j}$ means no overlapping chunks for any pair of disks in the m active disks. From Fig. 1, overlapping chunks on any two disks change with the number of powering-up disks. Specifically, if the disks are numbered 0 to 11 and m = 12 or 4, the overlap chunks that could be found between disk 0 and disk 1 will be 0.8 or 0.35 at different times. Therefore, the chunks need to be redistributed on disk 0 and disk 1.

As mentioned in the introduction, there is the problem of migrating petabytes of data and frequently switching servers on and off. It is very difficult to keep a fully distributed reconstruction while maintaining power proportionality.

2.2 Group-Based Power-Proportional Data-Layout Policy

Group declustering was first introduced to improve the performance of standard mirroring. It was then extended to multi-way replication for high-throughput mediaserver systems [15].



▲**Figure 1. Overlap chunks shared by any two active disks for $k=4$, $n=12$, $Q=9$.**

Group declustering involves partitioning all disks into several groups, and the number of groups is equal to the number of data copies. Each group stores a complete copy of all data. In contrast to standard mirroring, data in the first group is scattered across all of the servers in the second group. In [2], the authors propose a power-aware group-data policy that can be implemented using a group-declustering scheme.

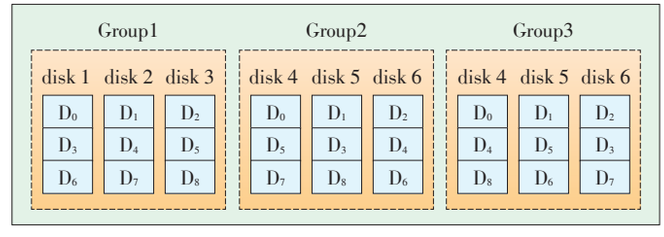
Fig. 2 shows the group-based layout, which can achieve power proportionality at the group level because any group can be used to provide one copy of the data service. The first problem with this method is that disks in each group share nothing, and better reconstruction parallelism cannot be achieved. For example, when all disks are powered on during busy periods but disk 4 fails, then disks 5 and 6 cannot provide available data to disk 4 for recovery. The second problem is that power is not proportional during recovery because all of the disks in other groups need to be powered on, even though only one disk has failed. For example, if disk 1 in the first group fails, then all of the disks in the second or third groups should be powered on for recovery. The third problem is that when the incoming request comes with bias, which causes overload on a certain disk, the system may need to power on another entire group of disks to share the workload of the busy disk. This occurs because the data on the busy disk is evenly scattered across other group of disks.

2.3 Ideal Power Proportional Service

This section discusses the possibility that power used is proportional to the services provided by the storage system. Powering down or putting disks on standby means that the chunk data in the corresponding disks is not available for service, and user requests could be viewed as the total number of chunks retrieved. The service provided by the disks is depends on two things: the required data must be available on that disk and the service request on the disk must not exceed the maximum workload (requests per second).

Observation 1: It is impossible for a storage system to achieve power proportionality by powering- down or idling disks unless each disk stores exactly one data chunk.

Proof: We evaluate power proportionality in the following discussion. In the storage system, the amount of chunk data that is available for service is $k\times Q$ when all of the disks are powered on. Over a period of time, suppose partial data (service) is accessed and the number of chunks for retrieval is x.



▲**Figure 2. Three-way replication data layouts by group declustering.**

Then, the proportion of chunks that are available is $\frac{x}{kQ}(0 \leqslant x \leqslant kQ)$.

The power used by the disks could be represented by the number of active disks. We assume that all disks consume the same amount of energy if they are powered on over a period of time. Thus, the maximum amount of power used is n; that is, all disks are powered on. Also, suppose that not all disks are active so that they can provide service, and the number of active disks is y. Then, the proportion is $\frac{y}{n}(0 \leqslant y \leqslant n)$.

The power used is proportional to the service performed, which satisfies $\frac{x}{kQ} = \frac{y}{n}$; that is,

$$y = \frac{n}{kQ} \times x \ (0 \leqslant x \leqslant kQ, 0 \leqslant y \leqslant n) \tag{4}$$

Given a fixed storage system, k, Q and n should be constants. Thus, the number of active disks y is a linear function of the number of requested chunks x. Both x and y should be integers $(0 \leqslant x \leqslant kQ, 0 \leqslant y \leqslant n)$. If $\frac{n}{kQ} < 1$, y will be a non-integer, and the power will not be proportional. If $\frac{n}{kQ} > 1$, y cannot get all the integers from 0 to n. This does not make sense because $k \times Q < n$, which means that the number of disks is more than all the chunks. If $\frac{kQ}{n} = 1$ (i.e. $kQ = n$), then $x = y$ and the power used is proportional to the service. Thus, power proportionality can be achieved if and only if $kQ = n$. This means that each disk only stores exactly one data chunk.

3 Design of a Chunk-Based Power-Proportional Layout

In our proposed scheme, we follow theorems previously discussed. First, the proposed scheme keeps has the fast recovery property of the storage system, and combination theory is used to select disks on which to place chunks. Then, we modify the layout scheme to support power efficiency. According to our observations in section 2, it is impossible to achieve absolute power proportionality. Thus, the design goal of the proposed scheme is to approximate power proportionality, that is, to maximize the efficiency of additional power usage when the system has to power-on another disk.

3.1 Chunk Based Data-Layout Scheme

To achieve better recovery parallelism, any β disks $(2 < \beta < n)$ should have the same number of overlap chunks v_β. This number will only be greater than zero when the number of disks under consideration is no larger than the replicas k. If $k+1$ disks have overlapping chunks, one chunk will be required to have more than k copies. Thus, to place k copies of a chunk, k disks are selected as a set on which to place them. For n different disks, the total number of subsets is $\binom{n}{k}$, and we count them from 1 to $\binom{n}{k}$. The k copies of chunk i are placed into the disks of the ith subset. If $\binom{n}{k} < Q$, the combination sets are repeated to place chunks. With $k=2$ and $n=6$, the subsets of disks in **Table 1** are mapping, with count from 1 to $\binom{6}{2} = 15$. The $(i=26)\%15 = 11$ and the subset comprising disks $x_1 = 3$ and $x_2 = 5$ (the two copies of chunk 26) are stored on disks 3 and 5.

▼**Table 1. Subcounting example**

Subset count	Subsets
1	1, 2
2	1, 3
3	1, 4
4	1, 5
5	1, 6
6	2, 3
7	2, 4
8	2, 5
9	2, 6
10	3, 4
11	3, 5
12	3, 6
13	4, 5
14	4, 6
15	5, 6

In general, the following equation tells us that the ith subset comprises k disks with an index value of $(x_1, x_2, ..., x_k)$. In Table 1, $(i=26)\%15 = 11 = \sum_{i=1}^{3-1} \sum_{i+1}^{6} 1 + \sum_{3}^{5-1} 1$, the index is 3 and 5:

$$\sum_{i_1=1}^{x_1-1} \sum_{i_2=i_1+1}^{n-k+1} \cdots \sum_{i_k=i_{k-1}+1}^{n} 1 + \cdots + \sum_{i_i=x_1+1}^{x_1-1} \cdots \sum_{i_k=j+1}^{n} 1 + \cdots + \sum_{i_k=x_{k-1}+1}^{x_k} 1 = i\,\%\binom{n}{k}$$

According to the combination series, any two disks share the same number of overlap chunks when $Q\%\binom{n}{k} = 0$. Because we place k copies of each chunk into k disks and we take these k disks as a set from the combination series, any two disks have an equal chance of being together. This ensures that any two disks in the series of sets will have the same number of overlapping chunks. The data in one disk will be evenly declustered to other disks. For the example, in Table 1, disk 1 has the same number of chunks as any other disk. We have proved that shifted declustering can provide full parallelism recovery [14].

3.2 Approximating Power Proportionality

It is very difficult to achieve perfect or ideal power propor-

tionality. Researchers exploit different data layouts to approximate power proportionality on different levels. In practice, the storage system always keeps at least one copy of all data chunks active:

$$x = Q + \Delta x \tag{6}$$

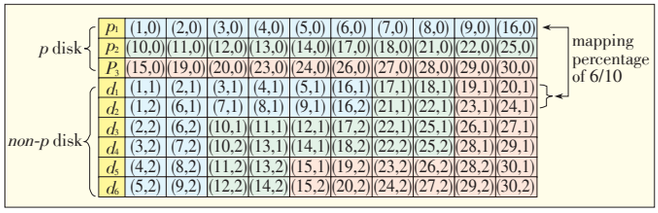
Suppose a layout has p disks as an isolated group, and $\varphi(\Delta x)$ is the extra number of disks that contain Δx for service. According to the discussion in section 2.1, the goal is to achieve

$$\frac{Q+\Delta x}{kQ} \approx \frac{p+\varphi(\Delta x)}{n} \tag{7}$$

The three relative variables are $\Delta x, \varphi(\Delta x), and\ p$, and the key is to find the correlation between Δx and $\varphi(\Delta x)$. Generally, $\varphi(\Delta x)$ has an increasing relationship with Δx; that is, more service requests lead to more active disks. If $\Delta x = 0$, then only one copy of the chunk data is needed, and $\varphi(\Delta x)$ is also zero. Thus, $p \approx \frac{n}{k}$. If $\frac{n}{k}$ is not an integer, we let $p = \left\lfloor \frac{n}{k} \right\rfloor$ to make p smaller and save more power.

Dynamic provisioning and reload dispatching can create a suitable relationship between Δx and $\varphi(\Delta x)$ if extra disks need to be powered on to respond to incoming requests. However, this usually entails the migration of petabytes of data, which makes the storage system more volatile. In the group-based layout, if the primary group is busy, another group of disks can be powered on to ensure service quality. Sometimes it may be not necessary to power on the entire group of disks when only one or two disks are overloaded (i.e. the I/O requests per second exceeds the setting). Thus, the policy of turning on a whole group of disks does not maximize efficiency for additional power usage.

Our proposed layout maximizes efficiency for additional power used. Specifically, we map each non $-$ p disk to a p disk with different percentages of overlap chunks and also preserving the declustering. The advantage of this policy is that the mapping relationship with different percentages provides a much more flexible power-up selection. In **Fig. 3**, n = 9, k = 3, Q = 30, p = 3, and non- p = 6. The subsets (1,0), (1,1), and (1,2) show there are three copies of chunk 1. If over a period, an extra disk is needed to share the workload of p_1, then d_1 or d_2 could be powered on because they share more overlap chunks (6/10) with p_1. To preserve the declustering, d_1 or d_2 also share some chunks with p_2 and p_3.



▲**Figure 3. An example of 30 chunks with 3 copies distributed into *p* disks and non- *p* disks.**

We record the p disk as $p_1, p_2, ..., p_p$ and the non- p disks as $d_1, d_2, ..., d_{n-p}$, and the percentage of overlap between pi and d_j is $vp_{i,j}$. The value of $vp_{i,j}$ is fixed to the value of p, which is $\sum_{j=1}^{n-p} vp_{i,j} = \frac{n-p}{p} (i = 1, 2, ..., p)$. For $i = 1$ in Fig. 3, we have

$$\sum_{j=1}^{9-3} vp_{1,j} = \left(\frac{6}{10}\right) + \left(\frac{6}{10}\right) + \left(\frac{2}{10}\right) + \left(\frac{2}{10}\right) + \left(\frac{2}{10}\right) + \left(\frac{2}{10}\right) = \frac{n-p}{p} = \frac{9-3}{3}.$$

Specifically, if the data on p disks is evenly declustered on non $-$ p disks, then $vp_{i,j} = \frac{1}{p}$.

For specific storage systems with different incoming request distributions, the chosen percentages between p disks and non- p disks could be different. With more bias services to provide (many I/O requests to a certain disk), a larger overlap percentage should be chosen for data declustering. The power on/off strategy is very simple in our layout. Most of the time, the system only needs one copy of the data for general service, and the system leaves p disks powered on. If the workload on a p disk exceeds the maximum setting, the system activates the disk that has more overlap chunks in order to share the workload of the busy disk.

3.3 Fast Recovery and Power Efficiency

Distributed reconstruction cannot be maintained when the system provides proportional service by switching disks on or off. The overlapping chunks between any two disks change with respect to the number of active disks. In this section, we describe the policy of the proposed layout that allows the storage system to remain power efficient when disks fail. Disk failure, especially multidisk failure, is a difficult issue to address because many combinational failures need to be considered. In this paper, we consider disk crash failures, not arbitrary Byzantine failures, and we assume that the failed disks are providing services. If the disks are not in service, we can take time to perform the recovery. The recovery policy takes into account availablility and load balance when powering on extra disks for recovery. For availability, one copy of every chunk needs to be active in the system because all of the data may be accessed immediately. Load balancing involves how to schedule the ongoing access load of the failed disks and the recovery load for the failed disk.

The ongoing access load of the failed disks must be handled immediately by the storage system in order to retain QoS. To power up fewer disks, the disks that have greater data overlap with the failed disk are invoked first. A complete data copy of the failed disks should be found in the active disks, and the group of disks that contain a complete copy of the failed disks are called the recovery group. From Fig. 3, d_1 and d_2 could be

recovery group of p_1. For power efficiency, we select a smaller number of disks (the default is 2) as a recovery group for each p disk. The disks in the recovery group have a higher percentage of overlap. If a primary server fails, the file system activates the corresponding recovery group. This is the failure-recovery strategy. The percentage of overlap is determined in the following manner: each p disk will map to two disks with a higher percentage of overlap and which contain a c omplete copy of the data on the p disk. The p disk also maps to disks with an equal percentage of overlap.

Thus, if the workload in the whole system is low, the extra number of power-up disks could be smaller (2 is default). In the example drawn from Fig. 3, the failed disk only causes two disks to be powered on. Also, the non-p disks share the same number of overlapping chunks, and this fully supports parallel recovery. For example, when all disks are powered on and d_1 has failed, all the other eight disks can provide data to d_1. The number of overlapping chunks determines the recovery speed. In fact, the maximum recovery parallelism of CPPL is $n-p$ and that of group-based recovery is $\frac{n}{k}$ [2].

4 Experiments and Evaluation

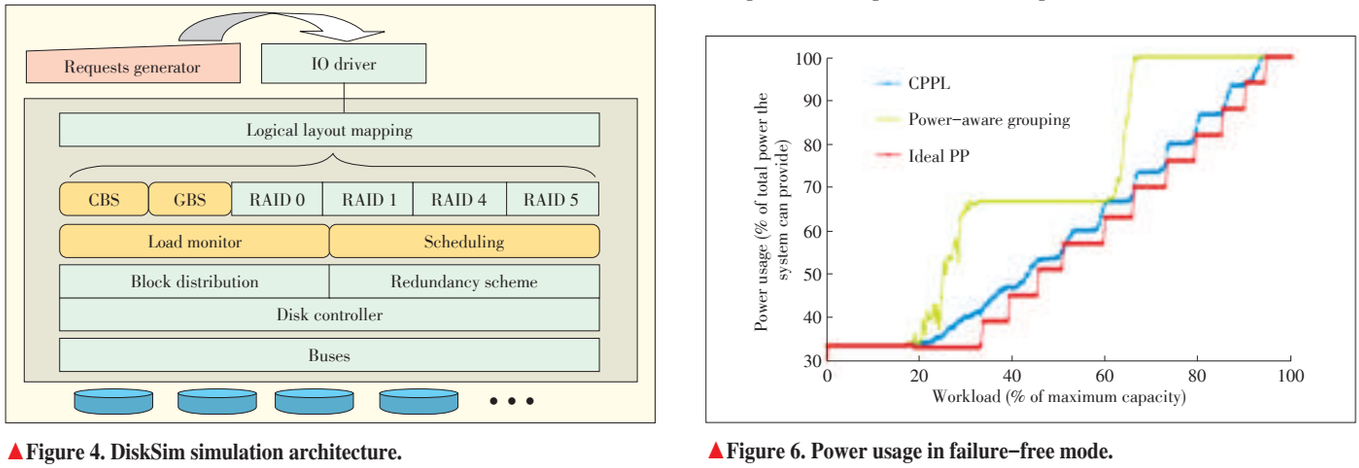
We run simulations on DiskSim [16] in order to determine the performance of the chunk-based layout in a multiway replication architecture. We implemented the address-mapping algorithms for CPPL, and we also implemented a group-based layout called Power-aware grouping [2]. We then determined performance through service-request-driven simulations.

The architecture simulated on DiskSim is shown in **Fig. 4**. At the top layer, a trace generator or input trace is read. The green boxes represent existing DiskSim modules, and the white boxes represent added modules for implementing our mapping algorithms, load monitor, and scheduling policy [17].

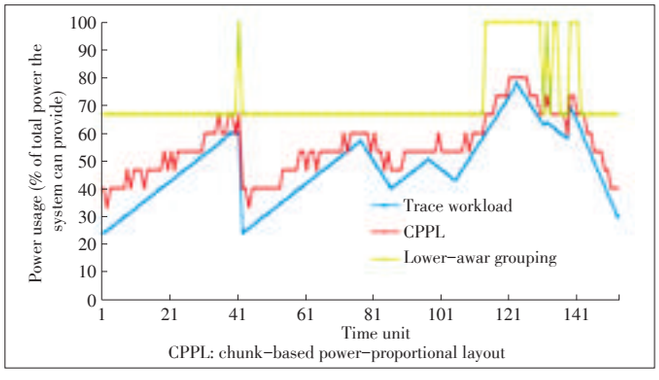
4.1 Proportional Service Performance

In normal service, the disks containing a copy of data are powered on in order to respond to incoming requests. We use Shortest Queue First (SQF) to assign the requests to active disks. The algorithm always tries to assign the new request to the active disk with the lowest workload. If the requested data is not on the disk, the process will continue to search until the request can be responded to by an active disk. If the process fails to assign the request to active disks, more disks need to be activated. With a power-aware grouped layout, another group of disks is powered up if the active disks cannot provide the requested service. With a chunk-based layout, the power-up order of the disks is first considered by the disks that can leverage better load balancing with the highest ongoing service of the p disks (as discussed in section 3.2). Also, we add the metric ideal power proportionality (ideal PP) with respect to workload.

With numerous requests under different workloads, we found that the chunk-based layout saves more power than the group-based layout without affecting user experience. **Fig. 5** shows power usage for the OLTP I/O trace called Financial I trace, which is a typical storage I/O trace from Umass Trace Repository [18]. Because any real trace has a specific bias, we use statistics to evaluate the performance under a variety of workloads (**Fig. 6**). For each workload, 100 different request cases are run. The graphs show the overall average percentage of power used when all disks are powered on. The power usage



▲Figure 4. DiskSim simulation architecture.



▲Figure 5. Power performance comaprison for Financial I trace.

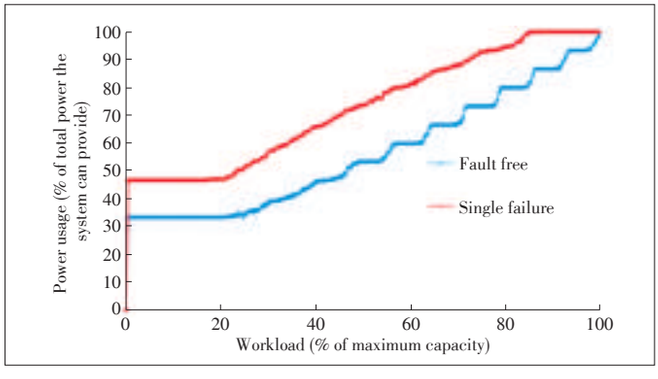
▲Figure 6. Power usage in failure-free mode.

curve of a chunk-based layout is lower than that of a power-aware grouping. These results agree with our design principle. When only a few disks are overloaded, a smaller number of disks are powered on in order to share the workload of the busy disks.

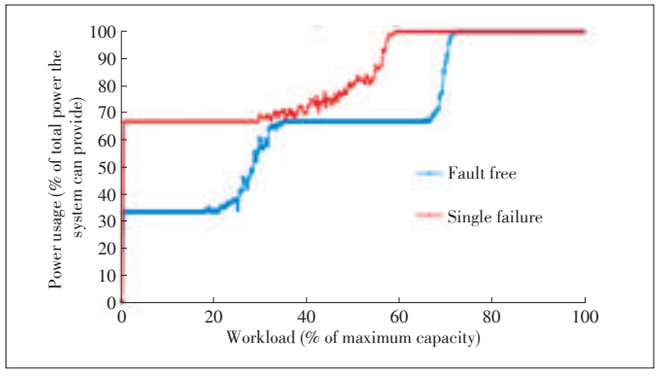
4.2 Degraded Mode Performance

In this section, we discuss power performance when the system is in degraded mode. The system should be able to handle the ongoing access requests of the failed disks as well as the recovery work for the failed disks. For both layouts, we apply the SQF algorithm so that the workload can be handled without affecting user experience. The algorithm tries to assign the workload of the failed disks to the active disk with the lowest workload. If this workload cannot be assigned to these active disks, more disks need to be activated. The system first tries to respond to the requested service without failure. During the runtime, one or two active disks are randomly chosen to fail.

Fig. 7 shows the power usage for CPPL in failure-free mode and with a single server failure. The power curves are parallel and increase proportionally with workload. The moderate gap between the two curves is the extra power used by the recovery group. This result agrees with our design in section 3.3. The extra number of power-up disks is smaller when one disk fails. However, with the group-based layout in **Fig. 8**, power usage



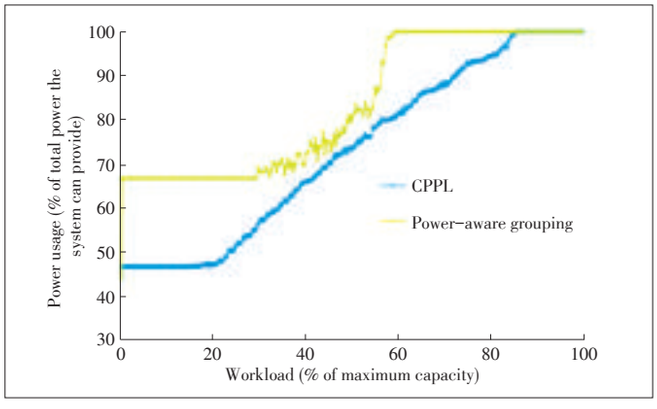
▲Figure 7. Power usage for CPPL: Failure-free mode vs. degraded mode.



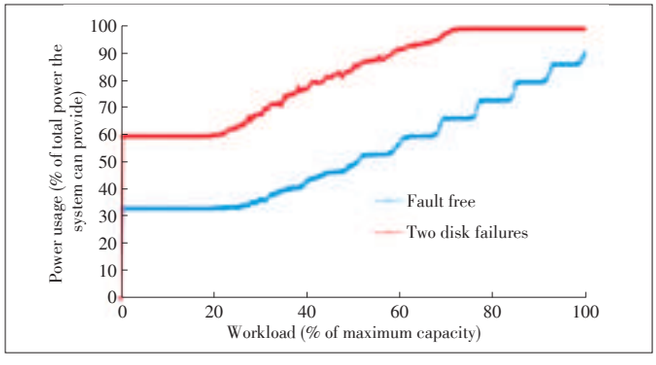
▲Figure 8. Power usage for power-aware grouping layout: Failure-free mode vs. degraded mode.

in failure-free mode moves in three steps that are not proportional to the workload. This disproportionality occurs because all of the disks are divided into three groups, i.e. $k=3$. Moreover, a whole group of disks needs to be powered on, even if the workload is light. **Fig. 9** shows power usage in degraded mode. We found that more than 30% of power can be saved by using CPPL instead of a group-based layout.

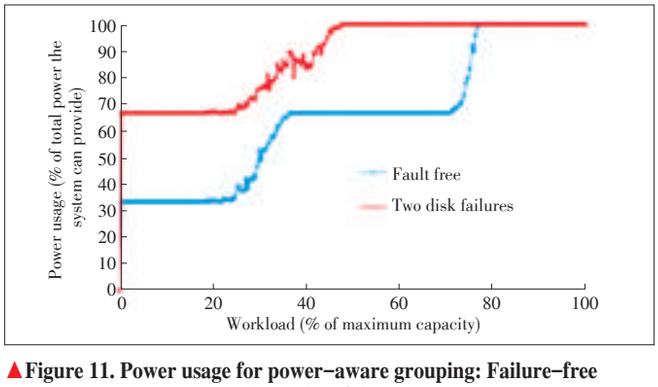
Figs. 10 and **11** show power usage of CPPL and power-aware grouping, respectively, in failure-free mode and two-disk-failure modes. Two disks fail in the simulation, so the whole workload is the work capability of $n-2$ disks in failure-



▲Figure 9. Power usage in degraded mode.

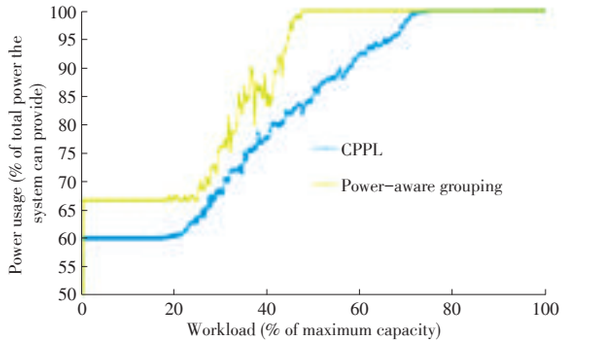


▲Figure 10. Power usage for CPPL: Failure-free mode vs. degraded mode (two disk failures).



▲Figure 11. Power usage for power-aware grouping: Failure-free mode vs. degraded mode (two disks failures).

free mode. In Fig. 10, the increasing relationship between the power curves is proportional, which means that the extra power used for recovery is the same for different workloads. However, gap between the curves in Fig. 10 is bigger than that for a single failure. This is expected because more disks need to be powered on in two-disks-failure mode. Fig. 11 shows that less power is saved with two-disk failure and all disks are powered on from the workload of 50%. **Fig. 12** is the power usage comparison for CPPL and power- aware grouping in degraded



▲**Figure 12. Power usage in degraded mode.**

mode.

5 Conclusions

In this paper, we have exploited the data-placement layout in storage architecture based on multiway replication. We have theoretically analyzed the characteristics of an ideal layout, which can support power proportionality and parallel recovery. We proposed a data-placement layout based on chunks. The proposed scheme manages power much more finely and still enables parallel recovery. Compared with group-based layout schemes, our proposed scheme saves much more power than group-based schemes and provides better recovery without affecting user experience.

References

[1] L. Andr, et al., "The Case for Energy-Proportional Computing," *Computer*, vol. 40, no. 12, pp. 33-37, 2007. doi: 10.1109/MC.2007.443.
[2] E. Thereska, , A. Donnelly, and D. Narayanan, "Sierra: practical power-proportionality for data center storage," in *Proc. of the sixth conference on Computer systems*, ACM: New York, NY, USA, 2011, pp. 169- 182. doi: 10.1145/1966445.1966461.
[3] H. Amur, et al., "Robust and flexible power-proportional storage," in *Proc. of the 1st ACM symposium on Cloud computing*, ACM: Indianapolis, Indiana, USA, 2010, pp. 217-228. doi: 10.1145/1807128.1807164.
[4] L. Keqin, "Performance Analysis of Power-Aware Task Scheduling Algorithms on Multiprocessor Computers with Dynamic Voltage and Speed," *Parallel and Distributed Systems*, IEEE Transactions on, vol. 19, no. 11, pp. 1484- 1497, 2008. doi: 10.1109/TPDS.2008.122.
[5] K. Li, "Design and Analysis of Heuristic Algorithms for Power-Aware Scheduling of Precedence Constrained Tasks," in *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW)*, 2011 IEEE International Symposium on, shanghai, China, May, 2011, pp. 804-813. doi: 10.1109/IPDPS.2011.224.
[6] X. Fan, W. D. Weber, and L.A. Barroso, "Power provisioning for a warehouse-sized computer," *SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 13-23, 2007. doi: 10.1145/1273440.1250665.
[7] J.S.Chase, et al., "Managing energy and server resources in hosting centers," *SIGOPS Oper. Syst.* Rev., vol. 35, no. 5, pp. 103- 116, 2001. doi: 10.1145/502034.502045.
[8] R.P. Doyle, et al., "Model-based resource provisioning in a web service utility," in *Proc. of the 4th conference on USENIX Symposium on Internet Technologies and Systems*, Seattle, WA, USA, 2003.
[9] G. Chen, et al., "Energy-aware server provisioning and load dispatching for connection-intensive internet services", in *Proc. 5th USENIX Symposium on Networked Systems Design and Implementation*, 2008, San Francisco, California , USA, 2008.
[10] J.Leverich and C. Kozyrakis, "On the energy (in)efficiency of Hadoop clusters," *SIGOPS Oper. Syst. Rev.*, vol. 44, no. 1, 2010, pp. 61-65. doi: 10.1145/1740390.1740405.
[11] L.Lu, P.Varman, J.Wang. "DiskGroup: Energy Efficient Disk Layout for RAID1 Systems," *The 2007 IEEE conference on networking, architecture, and storage (NAS' 07)*, Guilin, China, 2007, pp. 233- 242. doi: 10.1109/NAS.2007.21.
[12] G.A.Alvarez, et al., "Declustered disk array architectures with optimal and near-optimal parallelism," in *Proc. of the 25th annual international symposium on Computer architecture*, Barcelona, Spain, 1998, pp. 109-120. doi: 10.1145/279358.279374.
[13] M. Holland, and G.A. Gibson, "Parity declustering for continuous operation in redundant disk arrays," *SIGPLAN Not.*, vol. 27, no. 9, pp. 23-35, 2007. doi: 10.1145/279361.279374.
[14] H. Zhu, P. Gu, and J. Wang, "Shifted declustering: a placement-ideal layout scheme for multi-way replication storage architecture," in *Proc. of the 22nd annual international conference on Supercomputing, Island of Kos*, Greece, 2008, pp. 134-144. doi: 10.1145/1375527.1375549.
[15] M. S.Chen, et al., "Using rotational mirrored declustering for replica placement in a disk-array-based video server," *Multimedia Syst.*, vol. 5, no. 16, pp. 371-379, 2007. doi: 10.1007/s005300050068.
[16] *The disksim simulation environment (v4.0)*[Online]. Available: http://www.pdl.cmu.edu/DiskSim/
[17] D.Narayanan, et al., "Everest: scaling down peak loads through I/O off-loading," in *Proc. of the 8th USENIX conference on Operating systems design and implementation*, San Diego, California, 2008.
[18] *Umasstracerepository* [Online].Available: http://traces.cs.umass.edu/index.php/Storage/Storage

Manuscript received: May 23, 2013

Biographies

Jiangling Yin (jyin@eecs.ucf.edu) received his MS degree in software engineering from the University of Macau in 2011. He is working towards his PhD degree in computer engineering from the Electrical Engineering and Computer Science Department, University of Central Florida. His research focuses on energy-efficiency computing and file/storage systems.

Junyao Zhang (junyao@eecs.ucf.edu) received his MS degree in software engineering from Jilin University in 2009. He is currently a Ph.D student in the Computer Science Department at University of Central Florida. His research interests include scalability, reliability and energy-efficiency issues in file/storage systems.

Jun Wang (jwang@eecs.ucf.edu) received his PhD degree in computer science and engineering from University of Cincinnati in 2002.He is currently an associate professor with tenure in Department of Electrical Engineering and Computer Science, University of Central Florida. He is the recipient of National Science Foundation Early Career Award 2009 (news report) and Department of Energy Early Career Principal Investigator Award 2005. He is currently an associate editor of *IEEE Transactions on Parallel and Distributed Systems*. He has authored more than 60 publications in premier journals and leading HPC and systems conferences proceedings.

DOI: 10.3969/j. issn. 1673-5188. 2013. 04. 006
http://www.cnki.net/kcms/detail/34.1294.TN.20131211.1452.002.html, published online December 11, 2013

Virtualizing Network and Service Functions: Impact on ICT Transformation and Standardization

Bhumip Khasnabish[1], Jie Hu[2], and Ghazanfar Ali[2]
(1. Strategy Planning and Standards Development, ZTE TX Inc., Morristown, NJ 07960, USA;
2. Strategy Planning Department, ZTE Nanjing R&D Center, Nanjing 210012, China)

Abstract

Virtualization of network/service functions means time-sharing network/service (and affiliated) resources in a hyper-speed manner. The concept of time sharing was popularized in the 1970s with mainframe computing. The same concept has recently resurfaced under the guise of cloud computing and virtualized computing. Although cloud computing was originally used in IT for server virtualization, the ICT industry is taking a new look at virtualization. This paradigm shift is shaking up the computing, storage, networking, and service industries. The hope is that virtualizing and automating configuration and service management/orchestration will save both capes and opex for network transformation. A complimentary trend is the separation (over an open interface) of control and transmission. This is commonly referred to as software-defined networking (SDN). This paper reviews trends in network/service functions, efforts to standardize these functions, and required management and orchestration.

Keywords

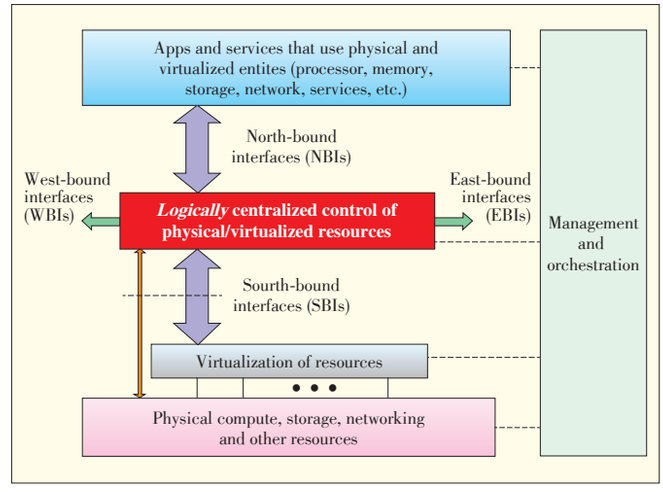
network function virtualization (NFV) and chaining; service function virtualization (SFV) and chaining; network virtualization overlay (NVO); software-defined networking (SDN); networking economics

1 Introduction

Network function virtualization (NFV) [1]–[6] has its origin in virtualized computing and storage services. However, there are many significant differences between NFV and virtualized computing and between NFV and virtualized storage. The primary objective of NFV is to share physical networking resources in a hyperspeed manner and minimize common overheads.

However, without any common application programming interfaces (APIs) or interoperable resource format, network infrastructure resources cannot be shared or efficiently used. Service providers also do not like proprietary APIs and solutions because network migration, transformation, and upgrade can be a major issue. **Fig. 1** shows a high-level architecture for separating network and service infrastructure resources, separating the control of these resources, and separating their usage by applications and services [7]–[9].

Both computing (e.g. DMTF [10]) and networking (e.g. ETSI [1], [2], [5], [6], IETF/IRTF [11]–[15]) standardization organizations have initiated activities based on use cases and aimed at developing NFV and integrating it into legacy networks. A number of proof-of-concept (POC) and testbed/integration activities have also begun in the laboratories of various service providers. These activities are being augmented by cost/benefit analyses and return on investment (ROI) calculations[1] for the field deployment of NFV. Initial results, reported by the migra-



▲**Figure 1. High-level architecture for network/service function virtualization and software-defined networking.**

[1] For example, Amazon provides a tool for calculating the TCO or total cost of ownership (http://aws.amazon.com/tco-calculator/) for the Amazon Web Service (AWS), and Microsoft Windows Azure also supports a similar tool (http://cloud-assessment.com/home/calc). The objective is to demonstrate reduction in growth and administrative costs for the same services when virtualized infrastructures are utilized instead of clusters of in-house physical (server, storage, CPE and networking gears) resources.

tion working groups of various service providers, customers and SDOs, are promising.

2 Use Cases

2.1 Distributed Management Task Force

DMTF's Network Services Management (NSM) working group is focused on network-service profiles for routed and routing protocols, which ensure IPv4, IPv6 and layer-2 (L2) connectivity that relates to the services provided by the network infrastructure to applications running in the cloud. Recently, DMTF NSM WG has developed a set of use cases, which we describe here.

2.1.1 Predefined Template-Based Network Configuration

In this use case, the end user is not concerned with the network topology (**Fig. 2**). The network service required by virtual machines (VMs) can be predefined in network templates. For example, the cloud service provider can define a standard network topology and service for a three-tiered website.

To build a website in the cloud, a user can select a predefined three-tiered website and assign roles, such as front-end web server, application server or database server, to VMs. Once the VM roles are assigned, high-level network services can be automatically provisioned to these VMs. For example, firewalls may be set up between web servers and application servers or between application servers and database servers to control access of these servers. Furthermore, a load balancer acting as front-end web servers can be automatically configured to distribute external requests to VMs.

From a network provider's perspective, the network template and role assignment information provided by users should be mapped to configurations on physical network devices and VMs (when network services are provided by software). A cloud service provider should be capable of managing network topology, flows, and services so that the most frequently used network architectures can be deployed inside the virtual network environment.

2.2.2 Network Configuration Based on the Existing Physical Network Topology of a User's Data Center

A cloud consumer may have already deployed their own private network and server clusters. When users move their existing IT infrastructures to the cloud, services in the existing physical networks should also be moved to the virtual network so that VMs migrated from existing physical servers can work properly. In this case, users should first extract network service configurations, such as ACLs in firewall and policy settings in load balancer, from the deployed physical network (**Fig. 3**).

To facilitate network migration, users may map their network configurations to a standardized format or template, e.g. network service model in CIMI interface or OVF-2 package. After the virtual network has been set up by the cloud service provider, a user can seamlessly plug in the VMs to the virtual network interfaces mapped to their existing physical network.

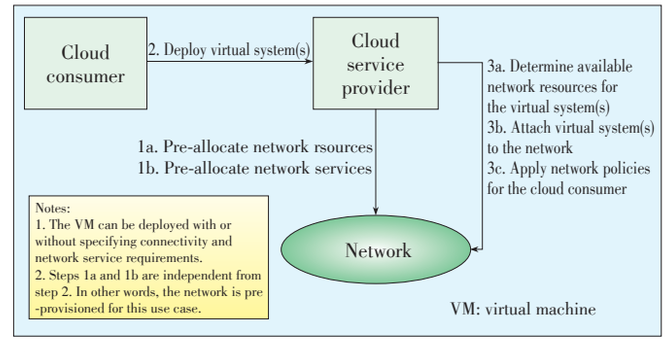
For each DMTF use case, both administrative and operational costs can be significantly reduced. Initial setup and learning costs can be paid off in a few years, and this justifies the investment in a predefined template and virtualizing the existing physical networks.

As well as developing service profiles for DNS, DHCP and NetConf, the DMTF NSM working group is developing network policy management profiles for firewall, load balancer, QoS, routing, access control list, and network resource security group. The DMTF NSM working group is also developing network services management profiles for BGP, layer 3 interface, and routing service.

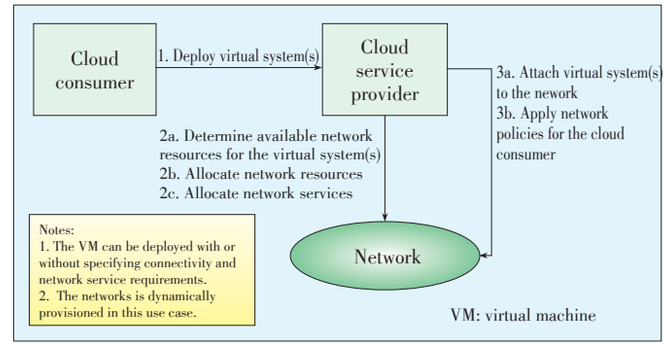
2.2 ETSI/ISG NFV Use Cases

It is widely believed that NFV was created to virtualize functions of OSI Layer 4 (transport layer) through to OSI Layer-7 (application layer) as an alternative to using high-cost, purpose-built network appliances and devices.

Consequently, the NFV use cases mostly arise as a result of the need for large service providers to save capex and opex. For example, capex can be saved by using commercial off-the-shelf (COTS) equipment, and opex can be saved by seamless, automated network and service management. Automated net-



▲**Figure 2. Precondition for template-based network configuration [10].**



▲**Figure 3. Precondition for network configuration based on existing physical network topology of a user's data center [10].**

work configuration means automatically configuring virtualized network entities for a desired service. Automated service configuration means automatically steering the flows (packet streams) to the most desirable service nodes through the most desirable network path/route.

As described in [5], NFV is current exploring the following use cases:

- network function virtualization infrastructure as a service (IaaS)
- virtual network function as a service (VNFaaS)
- virtual network platform as a service (VNPaaS)
- VNF forwarding graphs (VFGs)
- virtualization of mobile core network and IMS
- virtualization of mobile base stations
- virtualization of the home environment
- virtualization of CDNs (vCDN)
- virtualization of fixed-access network functions.

The main objective is to implement the above use cases by chaining a set of abstracted network entities, called virtual network functions (VNFs), that derive from the physical network resources via a virtualization layer (**Fig. 4**).

Using VNFs for just-in-time implementation of services provides the desired level of flexibility and deployment efficiency; however, automation/management resiliency and complexity requires further investigation.

Fig. 5 shows how physical and virtual network functions can be concatenated to dynamically create and update new service.

The logical view of the set of VNFs touched by a service can be illustrated by the flow paths (packet streams) (**Fig. 6**).

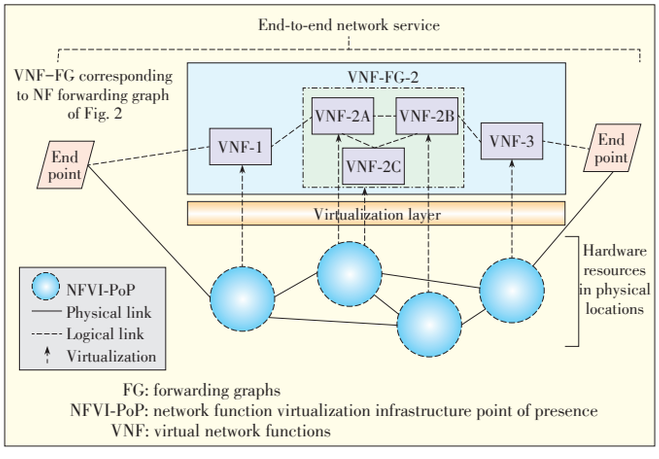
NFV recognizes that maintaining a consistent view of the statuses of virtualized devices and VNFs across multiple administrative domains in order to trace and diagnose faults and meet end-to-end SLAs.

Other challenges include distributed implementation of a logical and centralized control system using COTS hardware; developing common APIs, authentication, and virtualization; and uniform, transparent benchmarking of performance, capacity, scaling, handover, system-integration (including software upgrade), and resiliency across physical network boundaries.

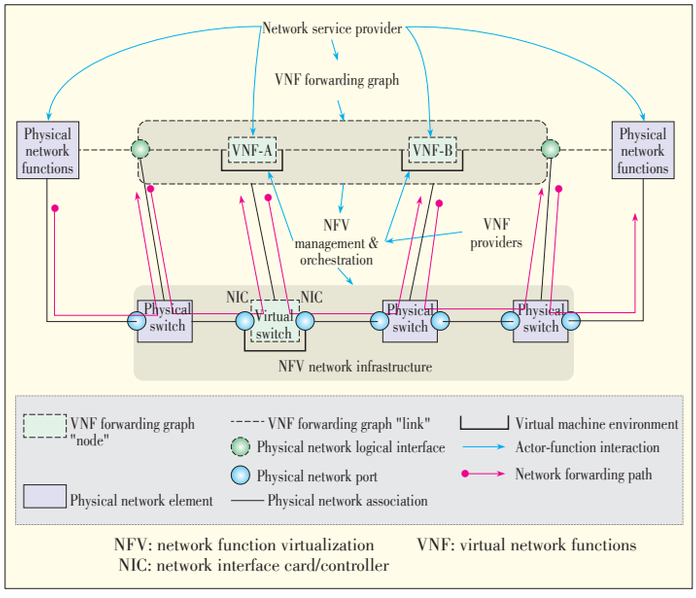
However, once these challenges have been dealt with, deployment and operation of virtualized CPE, Broadband Network Gateway, firewall/load-balancer, and address translator (IPv4 to IPv4/IPv6/name-or-content-based address) will be flexible and cost-effective. Service providers are expected to reduce both infrastructure costs and operation costs once provisioning and capacity adjustment are automated.

2.3 IETF and IRTF Use Cases

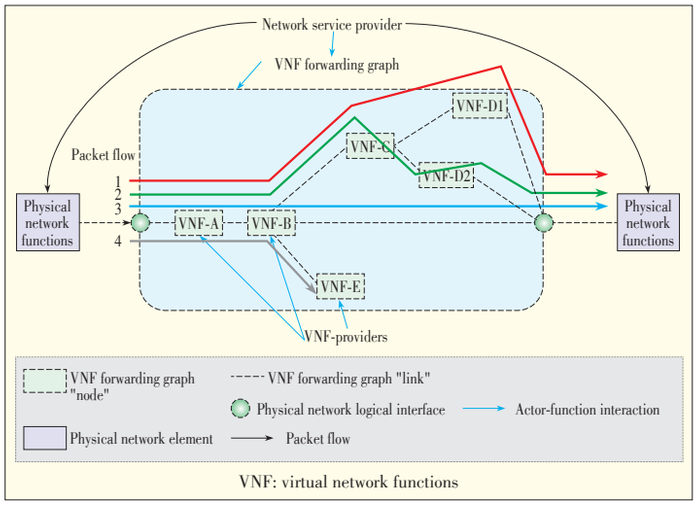
The first bar BoF on Cloud Computing, Networking and Ser-



▲**Figure 4. The NFV concept: end-to-end network service through chained VNFs [6].**



▲**Figure 5. NFV physical view: Virtual network function forwarding graph [5].**



▲**Figure 6. NFV logical view of virtual network function forwarding graph [5].**

vices was held during IETF-87 in March 2010 [11]. Since then, a number of groups have been created within IETF and IRTF to discuss network virtualization. The most important of these are the Network Virtualization Overlays (NVO3) working group [12], System for Cross-Domain Identity Management (SCIM) working group [13], and the Software-Defined Networking Research Group (SDN-RG).

The main focus of IETF NVO3 working group is to support multi-tenancy, which has become a core requirement of data centers (DCs), especially data centers that support virtualized hosts and VMs. The NVO3 working group will investigate the interconnection of DC virtual private networks (VPNs) and their tenants with non-NVO3 Internet-protocol-based networks to determine whether any specific work is needed [12]–[15].

IETF NVO3 use cases are focused on DC network virtualization and its applications. DC virtualization use cases include DC virtual network (VN) access via Internet and DC VN and Enterprise site interconnection via service provider's wide area network (WAN). Use cases related to DC network applications include support for multiple technologies and applications in a DC, tenant network with multiple subnets or across multiple DCs, and virtual data centers (VDCs) [12]. These use cases focus on virtualizing access, networking, and tenant systems within a DC in order to create virtualized DCs. The objective is to save both fixed costs and operational costs for DC-based services as well as providing the desired level of flexibility.

The main focus of the IETF SCIM working group is to standardize methods for creating, reading, searching, modifying, and deleting user identities and identity-related objects across administrative domains. The goal is to simplify common tasks related to user-identity management in services and applications [13], [14].

The use cases of IETF SCIM working group [13] include cloud service provider to cloud service provider flows as well as enterprise cloud subscriber to cloud service provider flows with an emphasis on the following scenarios:

- change of the ownership of an entity (e.g. a file)
- migration of identities
- single sign-on (SSO) service
- provisioning of user accounts for a community of interest (CoI)
- transfer of attributes to a relying party web site
- notification of changes.

Here again, the objective is not only to provide flexibility but also to offer a desired level of security across different physical/virtual administrative domains without incurring excessive infrastructure and operations costs.

The SDN-RG of IRTF provides a forum for researchers to investigate key problems in software-defined networking (SDN). SDN-RG investigates SDN from various perspectives with the goal of identifying approaches that can be defined and used in the near term and to identify future research challenges. Key areas of interest include solution scalability, abstractions, and programming languages and paradigms that are particularly useful in the context of SDN [15].

The main focus of SDN-RG [15] is adapting the network configuration at the speed that service development requires within a mixture of legacy and advanced networking where operation and debugging are becoming increasingly complex in enterprise, data center, and service provider networks. The use cases include the following areas: network description languages, abstractions, interfaces and compilers including the methods and mechanisms for (on-line) verification of both configuration and operation of network/node functions.

3 Reference Framework Architecture

3.1 DMTF

In modern data centers, multiple network and service elements, such as firewalls, routers, AAA servers, DNS, QoS managers, and load balancers exist in LANs and SANs, both of which can be used to provide advanced network services. These elements may be implemented as virtual appliances or traditional dedicated devices and applications. For unified management access to such network and service elements, we introduce virtualized networking. We look at the externally manageable functionality of such entities abstracted from their actual realization.

DMTF NSM working group focuses on developing specifications that help present a unified management view of the virtualized networking, services and their components to both cloud consumers and providers.

In a virtualized network, there are several challenging network-related problems, including configuring for network topology and service deployment, and configuring for physical network hosting in a virtualized environment.

3.1.1 Virtualized Networking Components

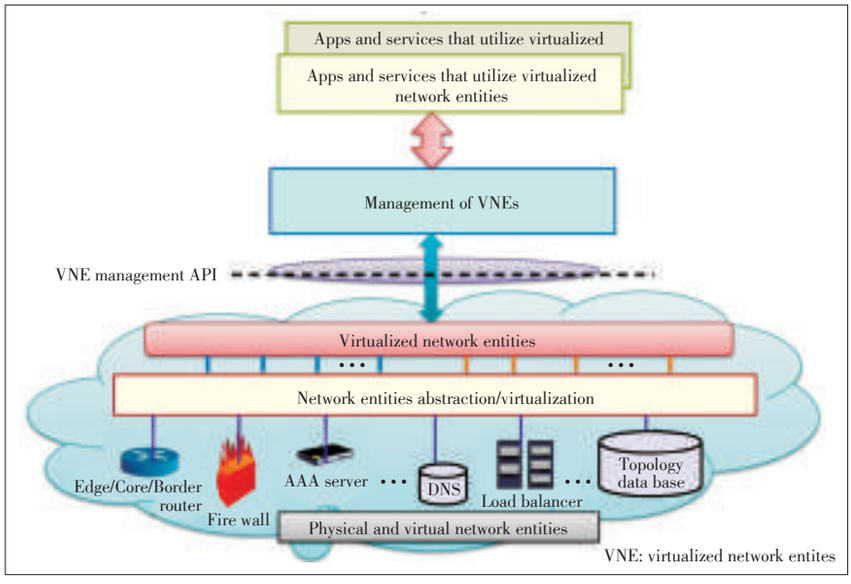
Fig. 7 shows a high-level schematic for abstracting network elements in order to expose them as the virtualized network entities (VNEs) for management.

The main components of virtualized networking are: physical and virtual network elements/entities, VNEs, and API for VNE management (Fig. 7).

3.1.2 Network Entities

Network entities include routers, firewalls, AAA servers, DNS, and load balancers, all of which can be interconnected to support network services. Such entities can be realized both as physical devices or virtual appliances.

A common mechanism is needed to virtualize these generic network entities and achieve seamless interoperability. Once these generic network entities are virtualized, the VNEs can be exposed through an open API so that various applications and



▲**Figure 7. Network entities (resources and services) abstraction, virtualization and management [10].**

services can manage and use them.

3.1.3 Virtualized Network Entities

VNEs are the abstraction of physical network entities and the network entities realized as virtual appliances. VNEs can be flexibly combined to support virtualized networking services.

These VNEs can be exposed via a management API to the upper management layers. The management API can be used to create, assign, monitor, update, and release the VNEs.

3.2 ETSI/ISG NFV Architecture

The ETSI/ISG NFV [1], [2], [5], [6] architecture identifies functional blocks and the main reference points between these blocks. The main functional blocks are:

- VNF
- element management system (EMS)
- NFV Infrastructure, including hardware and virtualised resources, and virtualization layer
- virtualized infrastructure manager(s)
- orchestrator
- VNF manager(s)
- service, VNF and infrastructure description
- operation support systems (OSS) and business support systems (BSS)

Fig. 8 shows the NFV architecture with functional blocks and reference points. The reference points that are shown by solid lines are potential targets for standardisation. Reference points Vn-Nf and Ve-Vnfm are of interest in terms of SDN controller NBI because these are involved in lifecycle and portability management of VNFs.

For reference point Vn-Nf (VNF to NFVI), Puppet, Chef OpenSource driver, and configuration file management (IETF NetConf) may be useful [7] along with VM Manager MIB (currently being discussed in IETF OPSA WG).

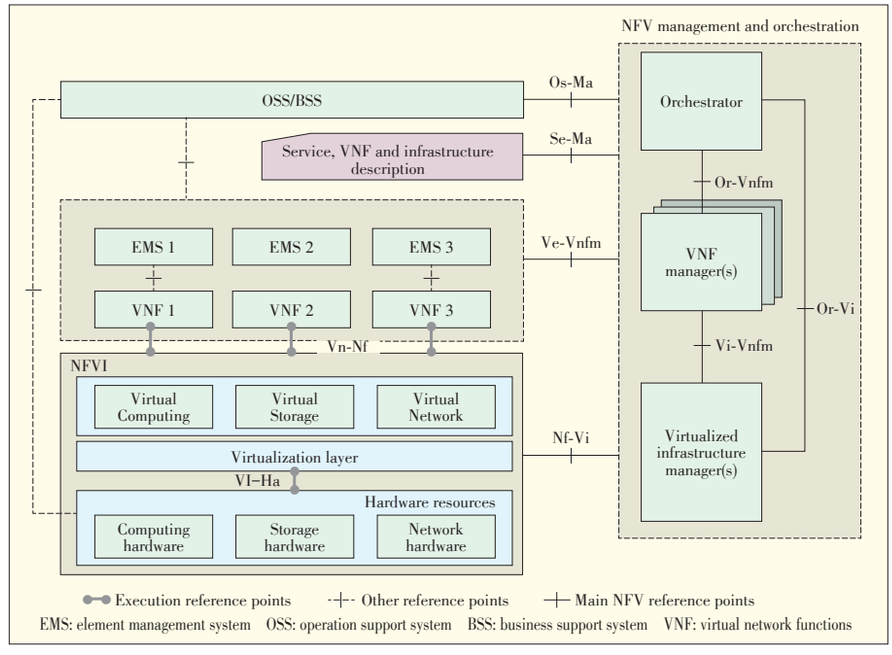
Reference point Ve-Vnfm (VNF.Environment to VNF.Manager-Orchestrator) needs to support a mechanism for a distributed and centralized VNF environment to request scale out/in, up/down may use extensions to OpenSource cloud computing APIs (e.g. CloudStack, extensions to OpenStack).

For NFVI to VIM (reference points Nf-Vi and/or Or-Vi and/or Vi-Vnfm), we may need extensions to 1) OpenStack for VMM Configuration Management, 2) IETF (OPSAWG) VM Manager MIB, and 3) DMTF Open Virtualization Format (OVF). In addition, special purpose plugins and/or adaptors to OpenStack Neutron can be used in to expose the northbound APIs of ONF.

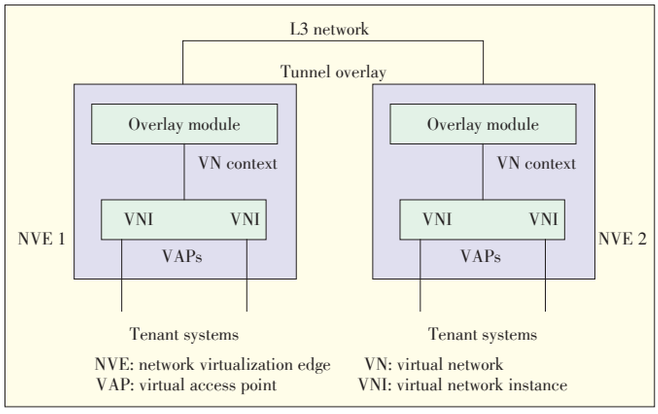
3.3 IETF and IRTF

IETF NVO3 WG is developing network virtualization overlays framework, and SCIM WG is focusing of schema and protocol development for a set of cross-domain identity management use cases.

Fig. 9 shows the network virtualization edge (NVE) refer-



▲**Figure 8. ETSI/ISG NFV reference architecture [1], [2], [5], [6].**



▲**Figure 9. Generic network virtualization edge reference model (IETF NVO3 Ref. Draft [12])**

ence model from IETF NVO3 WG [12].

One or more virtual network instances (VNIs) can be instantiated on an NVE. A tenant system interfaces with a corresponding VNI via a virtual access point (VAP). An overlay module provides tunneling overlay functions that include encapsulation and de-capsulation of tenant traffic, tenant identification, and mapping. Some NVE functions, such as data plane and control plane functions may reside in one device or may be implemented separately in different devices. NVE functions can also be hierarchically implemented, e.g. an end device can act as an NVE spoke, and an access switch can act as an NVE hub.

The SCIM WG is currently developing use cases, management protocol, and management core schema. The objective is to develop the core schema and interfaces based on HTTP and REST.

At IETF87 in July 2013 and IETF88 in November 2013, the IETF hosted the Network Service Chaining (NSC) and Service Function Chaining (SFC [14]) BoFs with the objective of supporting the implementation of use cases that ETSI/ISG NFV is (or will be) putting forward to industry. The proposed milestones for SFC include developing a framework so that the requirements for information model, encapsulation protocol, control plane, management information base, and IETF protocol adaptation/update can be determined.

The SFC architecture includes flow/packet classifier and appropriate NBI based control of packet/flow path by encapsulating the path information in the header of the flows/packets. These will reduce the number of hardware devices in the path of service flows and result in cheaper services.

4 Service-Level Agreement in Virtual Environments

When virtualized network and service functions are used, it may be very difficult to maintain a consistent end-to-end SLA. The TM Forum has recently published a document [16] called *Enabling End-to-End Cloud SLA Management* that emphasizes a set of business considerations and architecture design principles for supporting end-to-end cloud SLA management. The main barriers identified in that document are:

- lack of a single point of authority/accountability
- lack of integration of the service models and dependencies of various infrastructure and platform providers, and lack of interoperability, including brokering
- lack of consistent monitoring APIs and reporting tools
- lack of uniformity between business and service-level objectives and their correlation with the SLA

5 Regulatory and Security Requirements in Virtual Environments

There are many new security and regulatory requirements for both enterprise and carrier services that use virtualized network and service functions. These requirements are mainly related to data, and information and knowledge-base security, as discussed in a recent IETF draft [17]. These arise from lack of mandatory application security in protocols. Other gaps include

- systems security and new requirements
- network security and new requirements
- mobile security and new requirements
- physical security and new requirements
- OAM security and new requirements

DMTF is also working [18] with the Cloud Security Alliance (CSA) to address many of these issues and develop an audit process and trust models that are suitable for virtual network/service functions.

6 Interoperability in Virtual Environments

One of the main impediments to using virtualized network and service functions across different administrative domains is that the network and service functions (resources) must be exposed and orchestrated through the same interface. Thus, it is necessary to have either a common API or abstract the differences between the APIs in order to support seamless orchestration and interoperability.

Deltacloud [19] offers an API based on Representational State Transfer (REST), which supports an alternative to SOAP-based web services protocols. This API abstracts the differences between clouds. It offers the following three front-ends: Classic Deltacloud, DMTF Cloud Infrastructure Management Interface (CIMI) [20], and Amazon EC2 [21], and supports all of the other major Cloud service providers.

DMTF's CIMI allows interoperability between a cloud consumer and any cloud providers which supports standard CIMI interface for managing a cloud infrastructure. The interface uses REST-full HTTP to send and receive messages. These messages are formatted by using either Java Script Object Notation

(JSON) or XML.

7 Conclusions, Recommended Action and Future Research

It is widely believed that the following technical trends will be dominant in the ICT industry in the foreseeable future: 1) virtualized network and service functions, 2) automated configuration and service management/orchestration, and 3) separation (over an open interface) of control and media forwarding/transmission, which is also known as SDN.

Service providers, including enterprises, campuses, Internet, and common wireline and wireless/mobile carrier, are jumping on the bandwagon. The objective is to save capex and opex and make the network infrastructure more agile so that any new service can be introduced without needing to spend significant money on network resources. However, the following need to be addressed before there is any measurable benefit from deploying these new technologies:

- interoperable virtualization of network and service functions (i.e. interoperability across multiple hypervisors)
- visualization of virtualized network and service resources
- multidomain management (discover, add, move, change/update, release) and orchestration of network and service functions to dynamically create network and service chains and graphs
- lifecycle management of virtualized network and service functions
- end-to-end management of service-level objective (SLO) and service-level agreement (SLA) by automatically monitoring overloads and failures and recovering from these. Standard distributed resiliency and efficiency management methods may be very useful for this purpose.
- authentication and security management for virtualized network and service functions. Standard management/orchestration capabilities may be necessary for reliable, secure management of connectivity over multiple administrative domains.
- Maintenance of geographical boundaries for using virtualized resources in support of privacy, secrecy, and Regulations

Over the next decade, service providers, ICT equipment and solution providers, academics, and SDOs will work together to support seamless virtualization, automation, and separation of control and media transmission across network infrastructures.

References

[1] ETSI/ISG NFV. (Oct. 2012). *Network functions virtualization-introductory white paper* [Online]. Available: http://portal.etsi.org/NFV/NFV_White_Paper.pdf
[2] ETSI/ISG NFV. (Oct. 2013). *Network functions virtualisation-update white paper* [Online]. Available: http://portal.etsi.org/NFV/NFV_White_Paper2.pdf
[3] IETF87. (Jul. 2013). *Network service chaining (NSC) BoF* [Online]. Available: http://www.ietf.org/proceedings/87/nsc.html
[4] ZTE Core Expert Group, "ZTE SDN and NFV architecture," ZTE Corporation, Shenzhen, China, Rep., August, 2013.
[5] *Network Function Virtualisation - Use Cases*, ETSI/ISG NFV, GS NFV 009 V015, Sept. 2013.
[6] *Network Functions Virtualisation - Architectural Framework*, ETSI/ISG NFV, GS NFV-0010 V0.1.7, Sept. 2013.
[7] ZTE Core Expert Group, "ZTE SDN controller northbound interface (NBI)," ZTE Corporation, Shenzhen, China, Rep., Sept. 2013.
[8] ZTE Core Expert Group, "ZTE SDN controller southbound interface (SBI)," ZTE Corporation, Shenzhen, China, Rep., Sept. 2013.
[9] ZTE Core Expert Group, "ZTE SDN controller," ZTE Corporation, Shenzhen, China, Rep., Oct. 2013.
[10] DMTF. (Oct. 2013). *Network services management use cases* [Online]. Available: http://www.dmtf.org/sites/default/files/standards/documents/DSP2034_1.0.0a.pdf.
[11] IETF. (Mar. 2010). *Clouds initiative* [Online]. Available: http://trac.tools.ietf.org/area/app/trac/wiki/Clouds#
[12] IETF. (2012). *Network virtualization overlays (NVO3)* [Online]. Available: http://datatracker.ietf.org/wg/nvo3/
[13] IETF. (2012). *System for cross-domain Identity Management (SCIM) WG* [Online]. Available: http://datatracker.ietf.org/wg/scim/
[14] IETF88. (Nov. 2013). *Service function chaining (SFC) BoF* [Online]. Available: https://datatracker.ietf.org/wg/sfc/charter/
[15] IRTF. (2012). *Software-defined networking research group (SDN-RG)* [Online]. Available: http://irtf.org/sdnrg
[16] TM Forum. (Sept. 2012). *TR 178, enabling end-to-end cloud SLA management* [Online]. Available: http://www.tmforum.org/TechnicalReports/TR178EnablingEndtoEnd/50148/article.html
[17] S. Karavettil et al. (Dec. 2012). *Security framework for virtualized data center services* [Online]. Available: http://tools.ietf.org/html/draft-karavettil-vdcs-security-framework-05
[18] DMTF. (Jun. 2011). *Cloud security alliance (CSA) DMTF work register* [Online]. Available: http://dmtf.org/sites/default/files/CSA-DMTF_WorkRegister2_1.pdf
[19] Apache. (2012). *DeltaCloud* [Online]. Available: http://deltacloud.apache.org
[20] DMTF. (Oct. 2012). *Cloud infrastructure management interface (CIMI)* [Online]. Available: http://dmtf.org/sites/default/files/TechNoteCIMIv6_comments_10.31.12_0.pdf
[21] Amazon. (Nov. 2013). *Amazon EC2* [Online]. Available: https://aws.amazon.com/ec2/

Manuscript received: September 30, 2013

Biographies

Bhumip Khasnabish (bhumip.khasnabish@ztetx.com, b.khasnabish@ieee.org), Phd, AMCPM, is a senior member of the IEEE and an emeritus distinguished lecturer of the IEEE Communications Society. He has initiated cloud and data center activities in IETF and is vice-chair of DMTF NSM WG (previously co-chaired ATIS IPTV Interoperability Forum (IIF), and founded and chaired ATIS NG-CI TF and MSF Services WG). He is currently a senior director in the Strategy Planning Department of ZTE TX Inc., USA. His research interests include next-generation networking, platform and services that use virtualized computing and communication entities, tighter cross-layer communications, and system configuration/services automation. He has worked in the Verizon/GTE next-generation Laboratories, Waltham, MA, and in Bell-Northern Research (BNR) Ltd. in Ottawa, Canada. Dr. Khasnabish has published numerous articles, books, and book chapters and has been awarded several patents in his research areas.

Jie Hu (hu.jie@zte.com.cn) received his MS degree in computer science from Southeast University, China. He is the standards director of cloud computing platform of ZTE Corporation. He has been an editor in several standards organizations, including CCSA, OMA, ITU-T on CDN, Mobile Search, and Cloud Computing RA.

Ghazanfar Ali (ghazanfarali@zte.com.cn) received his MS degree in computer science from Quad-e-Azam University, Islamabad. He is an advanced standards research engineer in the Strategy Planning Department of ZTE Corporation and represents ZTE Corporation as a vice chair at DMTF Cloud Management Sub-Committee (CMSC). His major distinctions include contribution of about 400 technical proposals in different technical standards developed in ITU SG13, OMA, and DMTF. His research interests include software-defined computing and virtual appliances.

DOI: 10.3969/j. issn. 1673-5188. 2013. 04. 007
http://www.cnki.net/kcms/detail/34.1294.TN.20131205.1837.002.html, published online December 5, 2013

Cooperative Communication Protocols for Performance Improvement in Mobile Satellite Systems

Ashagrie Getnet Flattie
(Engineering Department, Ethio Telecom, Addis Ababa, Ethiopia)

Abstract

A mobile satellite indoor signal is proposed to model performance of cooperative communication protocols and maximal ratio combining. Cooperative diversity can improve the reliability of satellite system and increase data speed or expand cell radius by lessening the effects of fading. Performance is determined by measured bit error rates (BERs) in different types of cooperative protocols and indoor systems (e.g. GSM and WCDMA networks). The effect of performance on cooperative terminals located at different distances from an indoor cellular system is also discussed. The proposed schemes provide higher signal-to-noise ratio (SNR)—around 1.6 dB and 2.6 dB gap at BER 10^{-2} for amplify-and-forward (AF) and decode-and-forward (DF) cooperative protocols, respectively, when the cooperative terminal is located 10 m from the WCDMA indoor system. Cooperative protocols improve effective power utilization and, hence, improve performance and cell coverage of the mobile satellite network.

Keywords

cooperative communication; amplify-and-forward; mobile satellite system; signal-to-noise ratio; maximal ratio combining

1 Introduction

Mobile communication via satellite is an integral part of IMT2000 and UMTS [1]. Unlike in a terrestrial cellular network, transmission in a mobile satellite network is constrained by available power [2]. Satellite communication using land-based mobile terminals suffers from shadowing, multipath fading, and strong variations in the received signal power because the signal is reflected by buildings and/or terrain. Shadowing of the satellite signal is caused by obstacles , such as buildings, bridges and trees, in the propagation path and results in attenuation over the entire signal bandwidth [3]-[6]. More than 80% of users are usually inside buildings, and it is a challenge to provide high-performance indoor coverage, especially in terms of higher data rates. It is much more than a technical challenge; the business case must also be evaluated, and any solutions implemented must be future-proof. Nevertheless, buildings (in particular, high-rise office buildings) contain many potential users of telecommunications systems and could be significant sources of revenue if high-quality radio communication could be delivered to them.

Coverage and capacity problem can be expanded by installing more cellular base stations, but this requires more complex and costly hardware, not to mention expensive real estate on which to physically locate the base stations [7]. In addition to this, applications for future ubiquitous communications will require wireless networks to have diverse network architectures and technologies [8].

Many sophisticated transmission technologies have been developed to improve the robustness and throughput of mobile systems. For example, multiple antennas can increase the capacity and reliability of mobile communications. Cooperative communication systems have also been developed as a low-cost alternative to multiple-input multiple-output (MIMO) systems [9]. Over the past few years, cooperative communications has been one of the most widely explored topics. Cooperation involves generalizing the relay channel to multiple sources that have information to transmit, and these sources also act as relays for each other. The main idea behind cooperation is that each cooperating entity gains by means of the unified activity [10], [11]. The benefits of cooperative communication in terms of link reliability and coverage extension have become better known within academia and the telecommunication industry over the past few years [12]. With cooperative communication, most of the benefits of MIMO are also leveraged. Benefits such as array gain, diversity gain, spatial multiplexing gain, and interference reduction can be obtained without using conventional MIMO technology and without increasing expenditure in terms of transmission time or bandwidth [13], [14].

This paper describes a cooperative communication scenario in a satellite and cellular system. MATLAB is used to analyze the bit error rate (BER) of different cooperative communication protocols used in this scenario.

When analyzing satellite-to-indoor channels, we can use either empirical measurements or deterministic channel modeling methods, such as ray tracing. In order to attain good reception, it is crucial to know the propagation conditions inside a room [15]. Link budgets are calculated in order to analyze critical factors in the transmission chain and optimize performance in areas such as transmission power and bit rate. This ensures that a target quality of service (QoS) can be reached [2].

The cooperative setup is shown in **Fig. 1**. In general, there

Cooperative Communication Protocols for Performance Improvement in Mobile Satellite Systems

Ashagrie Getnet Flattie

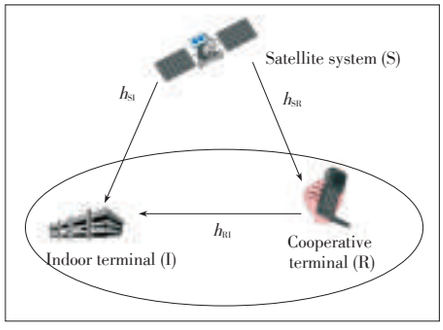


Figure 1. Basic cooperative communication system with a single cooperative terminal.

are two main relaying modes decode–and–forward (DF) and amplify–and–forward (AF). In DF mode, the message received by the source is fully decoded, and the detected symbols are re-modulated into the same or different alphabet. The resulting data is forwarded to the destination. In this mode, propagation of decoding errors may lead to a wrong decision at the destination. In AF mode, the relaying node simply amplifies the received signal subject to power constraints [10].

2 System Model

The basic cooperative communication system comprises a satellite system (S), cooperative terminal (R), and indoor terminal (I) (Fig.1). It is assumed that all the links use quadrature phase–shift keying (QPSK) modulation. Channel gains are given by h = (h_{SI},h_{SR} *and* h_{RI}). Several relaying protocols and MRC combining methods are examined to assess their effect on performance [16]. Different numbers of cooperating terminals are used to examine the AF and DF transmission protocols.

Cooperative communication typically refers to a system where users share and coordinate resources it order to improve transmission quality. To achieve cooperative communication cooperating terminals can be specifically assigned portable relay nodes or other user terminals that temporarily form indoor system links. An indoor user must also act as a receiver for other cooperating users or terminals.

In the first time slot, the satellite transmits the signal directly to the indoor destination and also to the cooperative terminal. In the second time slot, the cooperative terminal uses the AF or DF cooperative diversity strategy to retransmit the signal received from the satellite to the indoor terminal.

A fixed AF relaying protocol, often simply called an AF protocol, amplifies the received signal and transmits it to the destination. In phase one, the satellite sends out the signal x with transmit power Ps. The corresponding received signal y_{SR} at the cooperative terminal and y_{SI} at the indoor terminal can be written as [17], [18]:

$$y_{SR} = \sqrt{Ps}\, h_{SR}\, x + n_{SR} \tag{1a}$$

$$y_{SI} = \sqrt{Ps}\, h_{SI}\, x + n_{SI} \tag{1b}$$

where Ps is the transmission power at the satellite; x is the transmitted signal; n_{SR} and n_{SI} are the additive noise; and h_{SR} and h_{SI} capture the effects of path loss, shadowing, and fading between the source and relay and between the source and destination, respectively. For simplicity, all channels are modeled as Rayleigh flat fading channels. The additive noise is white Gaussian noise with zero–mean and variance N_0 [19].

In phase two, the satellite remains silent while the cooperative terminal amplifies the received signal and forwards it to the indoor terminal with transmit power P_{coop} . The received signal at the destination can be modeled as [16], [17]:

$$y_{RI} = h_{RI} q(y_{SR}) + n_{SI} \tag{2}$$

where $q(.)$ depends on which processing is implemented at the cooperative terminals.

2.1 Maximum Ratio Combining Output SNR

The maximum ratio combining (MRC) is an optimal combining technique with only linear complexity [7]. MRC provides the best possible performance by multiplying each input signal with its corresponding conjugated channel gain MRC is based on the assumption that channel attenuations and phase shifts are perfectly known [20].

Using MRC at the destination node and based on (2), the total signal–to–noise ratio (SNR) given by $\gamma = C/N$, from the i th cooperative terminal is [21]:

$$\gamma = |h_{SI}|^2 \frac{E_s}{N_{SI}} + \sum_{i=1}^{M} \frac{\dfrac{|h_{SR}|^2 E_s}{N_{SR}} \cdot \dfrac{|h_{RI}|^2 E_i}{N_{RI}}}{\dfrac{|h_{SR}|^2 E_s}{N_{SR}} + \dfrac{|h_{RI}|^2 E_i}{N_{RI}} + 1} \tag{3}$$

where h_{SI}, h_{SR} and h_{RI} are independent complex Gaussian distributed channel coefficients of the source destination, source–relay i, and relay i destination channels, respectively. E_s and E_i are the average energy transmitted at the source and ith cooperative node, respectively. These can also be considered as the transmission power, assuming that each transmission has unit duration. To simplify:

$$\gamma = \gamma_{SI} + \sum_{i=1}^{M} \frac{\gamma_{SR} \times \gamma_{RI}}{\gamma_{SR} + \gamma_{RI} + 1} \tag{4}$$

where $\gamma_{SR} = \dfrac{|h_{SR}|^2 E_s}{N_{SR}}$, $\gamma_{RI} = \dfrac{|h_{RI}|^2 E_i}{N_{RI}}$ and $\gamma_{SI} = |h_{SI}|^2 \dfrac{E_s}{N_{SI}}$ are the instantaneous SNRs between satellite and cooperative terminal, cooperative terminal and indoor terminal, and satellite system and indoor terminal, respectively [12], [22].

Assuming all cooperative terminals have the same characteristics; $\gamma_{SI} = \gamma_{SR}$ in (4):

$$\gamma = \gamma_{SI}\left[1 + M \frac{\gamma_{RI}}{\gamma_{SI} + \gamma_{RI} + 1}\right] \tag{5}$$

$$\gamma = \frac{P_s |h_{SI}|^2}{N}\left[1 + M\frac{\beta_i^2 |h_{RI}|^2}{\beta_i^2 |h_{RI}|^2 + 1}\right] \quad (6)$$

where β_i^2 ith relay amplifier gain $= \frac{P_{coopmax}}{P_s |h_{SR}|^2 + N}$; the noise variance $N = KT_{sys}B_s$; P_s is the satellite downlink power, $P_{coopmax}$ is the cooperative maximum power, and M is the number of cooperative terminals [23].

2.2 Cell Coverage

If we assume that the cooperative terminal is d km from the indoor system, then

$$P_{indoor} = P_{coop} - P_L(d) \quad (7)$$

where $P_L(d)$ is the mean path loss at the distance d km, P_{coop} *is* the average power transmitted by the cooperative terminal, and P_{indoor} is the average power received at the indoor system (in decibels) [24], [25]. Coverage and required average received power are inversely related to each other.

3 Basic Principles of Satellite Communication

Every satellite application becomes effective by building on the strengths of the satellite link [26]. Terrestrial cellular systems also experience dynamic fading as the subscriber drives past buildings and trees, under overpasses, or into isolated locations where base stations cannot reach. This fading is very rapid because it is produced primarily by multipath propagation. The strongest signal that is received may at times be a reflected signal off of one or more buildings [26]. QoS can be expressed in terms of BER performance, which depends on the carrier-to-noise (C/N_0) density ratio, and service reliability can be expressed in terms of service availability [27], [28].

Link budgets are calculated in order to analyze critical factors in the transmission chain and optimize performance in areas such as transmission power and bit rate. Link budgets also ensure that a QoS target can be reached [2], and often help determine the ratio of carrier power to noise power at the receiver input. This ratio is denoted C/N or CNR [29].

In the case of fading in the downlink, the downlink C/N is (8) [2], [5], [30]:

$$\frac{C}{N} = P_t G_t \frac{G_t}{T}\left(\frac{\lambda}{4\pi R}\right)^2 \frac{1}{kB}\frac{1}{Ap} \quad (8)$$

or expressing (8) in decibels:

$$\frac{C}{N_0} = 10\log(P_t G_t) - 20\log\left(\frac{4\pi R}{\lambda}\right) + 10\log\left(\frac{G}{T}\right) - 10\log Ap - 10\log kB \text{ dBWHz} \quad (9)$$

where N_0 is the one-sided noise power spectral density (dB/W/Hz^{-1}) and is equal to $N - 10\log(B)$, and $-10\log Ap$ is the atmospheric attenuation in decibels.

Equation (9) is valid for both the uplink and downlink. In the downlink, ERIP refers to the satellite transmission and G/T refers to the Earth station or mobile terminal. G/T is the ratio of receiver antenna gain to system noise temperature, in dB/K, measured at the input to the receiving system [31]. A significant condition in digital satellite link design is to ensure that E_b/N_0 is sufficiently large to guarantee that the BER performance criteria are met. The relationship between C/N and E_b/N_0 is [32]:

$$\frac{E_b}{N_0} = \left(\frac{C}{N}\right)\left[\frac{1+\alpha}{\log_2 M}\right] \quad (10)$$

where α is the channel filter roll-off factor, and M is the possible values or signals the phase of the carrier takes in a modulation scheme.

The relationship between C/N and SNR is:

$$SNR = C/N - \frac{C}{N} = \frac{S}{N} - K_{rolloff} \quad (11)$$

where $K_{rolloff}$ (assumed in DVB-S) is −0.3977 dB.

The minimum mean received signal (carrier power) to noise spectral density $\left(\frac{C}{N_0}\right)_{min}$ is related to the bit rate R_b via [33], [34]

$$\left(\frac{C}{N_0}\right)_{min} \geq R_b \left(\frac{E_b}{N_0}\right)_{min} \quad (12)$$

where E_b = energy/bit (W), N_0 = noise energy density (W/Hz), and R_b = bit rate (bit/s).

3.1 Estimating SNR Using DAF

To calculate the SNR using DAF, the BER of the link must be calculated first and translated into an equivalent SNR. BER of one cooperative link can be calculated as [20], [35]:

$$BER_{s,r,I} = BER_{s,r}\left(1 - BER_{r,I}\right) + \left(1 - BER_{s,r}\right)BER_{r,I} \quad (13)$$

BER performance for a QPSK modulated signal is given by

$$SNR = \left[Q^{-1}(BER)\right]^2 \quad (14)$$

where Q (.) is the Gaussian-Q function.

4 Indoor Channel

The satellite to indoor propagation channel depends heavily on the layout and material properties, i.e. the construction materials used for walls, windows and ceilings, of the building where the receiver is located [36]. Either empirical measurements or deterministic channel modeling methods can be used

to analyze satellite-to-indoor channel. In order to achieve a good reception it is crucial to know the propagation conditions inside a room [15]. In this study, the log-distance path-loss model [37], [38] is considered. The log-distance path-loss model is a site-general model, and the received power decreases linearly with distance on a logarithmic scale:

$$L(d) = L(d_0) + 10\gamma \log\left(\frac{d}{d_0}\right) + Xs \text{ dB} \quad (15)$$

$$L(d_0) = -120\log(\lambda) + 20\log(4\pi) + 20\log(1) \quad (16)$$

where $L(d_0)$ is the path loss at the reference distance (usually taken to be the (theoretical) free-space loss at 1 m); γ is the path-loss distance exponent; *and* Xs is a Gaussian random variable with zero mean and standard deviation of σ dB.

When Rayleigh fading is applied on a per-bit level, given by R, the energy per bit to noise power spectral density ratio is

$$\frac{E_b}{N_0} = P_{tx} + L(d) + R - N_{\text{floor}} \quad (17)$$

This empirical model takes into account the effects of shadowing by introducing Xs, which describes the statistical character of slow fading within the indoor link and, as a random variable, satisfies the long normal distribution with a standard deviation of σ in decibels [37], [39].

5 Simulation Assumption

Table 1 shows the key parameters for calculating and simulating SNR, C/N, E_b/N_b, and cell coverage.

6 Analysis of Simulation Results

In this section, we investigate the effect of cooperative partners on the SNR for each of the channels involved in cooperative cooperation. That is, we investigate satellite to indoor channel, satellite to cooperative terminal channel, cooperative terminal to indoor channel, and satellite to cooperative terminal to indoor system channel when the AF and DF cooperative diversity schemes are used. The results are expressed as BER versus total SNR at the indoor terminal and cell radius and are given as a percentage.

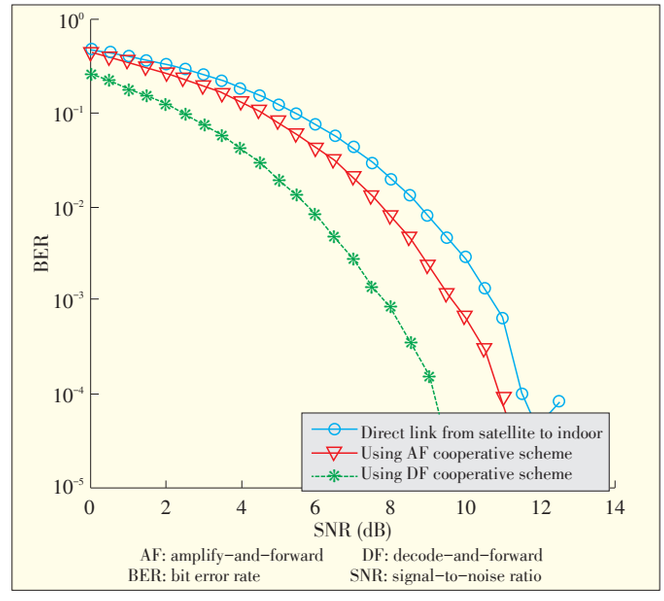
BER and equivalent percentage in cell coverage are the metrics used to test overall system performance in two different cellular networks. One network is GSM at 900 MHz and the other is a 3G/WCDMA network at 2100 MHz. Direct transmission from satellite to indoor terminal is simulated ,and cooperative transmission using Rayleigh fading channel models and log-distance path loss model are also simulated. The results of all these simulations are compared. Computation was done with MATLAB using QPSK modulation schemes.

Fig. 2 shows that with AF and DF, cooperative relaying provides gains of approximately 1.5 dB and 3.1 dB, respectively, over direct transmission at BER 10^{-3}. Cooperative communication performs significantly better than direct transmission, and AF performs better than direct transmission. However, the best performance is achieved with the DF scheme because with AF protocol, there is amplified noise in the transmitted signal.

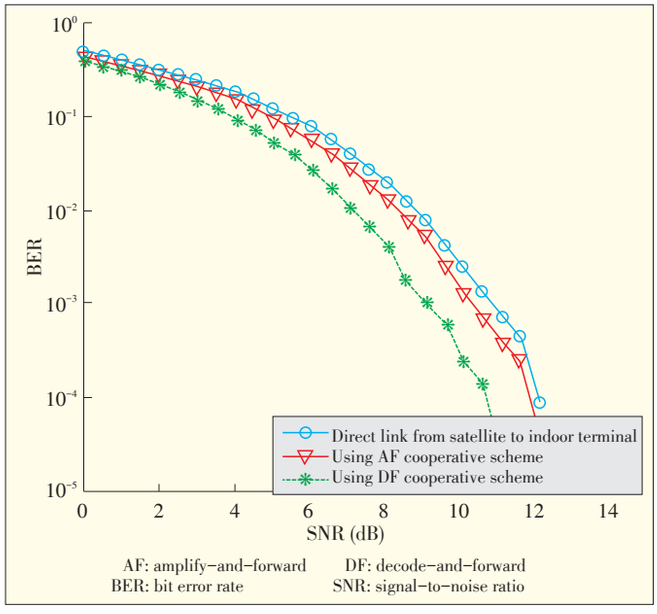
When transmitting directly from the satellite to indoor terminal, using the cooperative scheme with AF and DF results in performance that is 0.6 dB and 1.8 dB away , respectively, from that when the MRC scheme at a BER of 10^{-3} (**Fig. 3**). This is expected because the cooperation protocol effectively realizes a distributed receiver diversity. The direct transmission schemes require higher SNRs in order to provide the same performance as AF. This approach increases cell coverage inside the building because higher required average receiving power lowers coverage (and vice versa), as in (7), (15) and (17).

▼**Table 1. Simulation parameters**

Parameter	Value
Satellite transponder bandwidth	36 MHz
Satellite downlink frequency	11 GHz
Distance of Earth station to satellite	39,000 km
System temperature	290 K
Cooperative terminal maximum power	250 mW
FEC code rate	2/3
Indoor required coverage	96%
Antenna pointing loss	1 dB
Type of modulation	QPSK
Satellite maximum EIRP	100 dB
Cooperative terminal frequency	900 MHz and 2100 MHz
Standard deviation for office, soft partition	9.6 dB for GSM and 14.1 dB for WCDMA



▲**Figure 2. Direct transmission vs. different cooperative relaying protocols: Performance in terms of BER vs. SNR when a cooperative terminal located 8 m from the GSM indoor system.**



▲**Figure 3. BER as a function of SNR for d = 1000 m and GSM 900 MHz indoor terminal.**

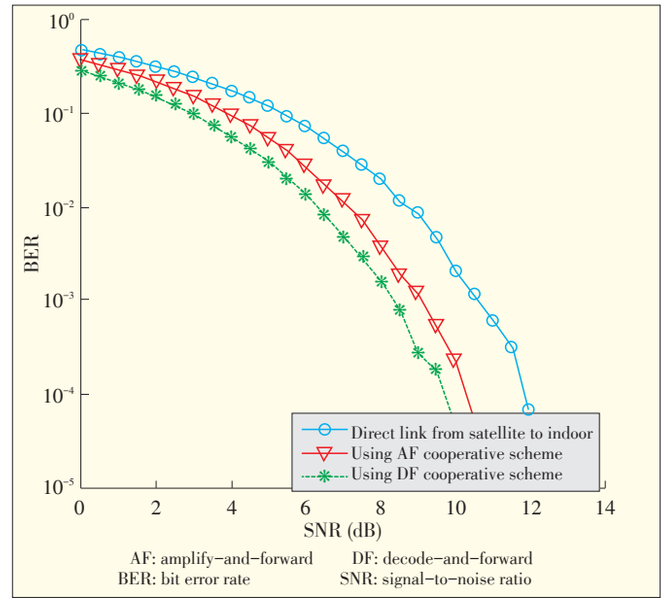
Fig. 4 shows the BER as a function of SNR for different cooperative communication schemes. So far, the performance evaluation has only been done for basic satellite and cooperative scenarios. These scenarios will be extended to allow comparison for E_b/N_0 in WCDMA indoor terminal. When BER = 10^{-3}, SNRs are 10.8 dB, 9.2 dB, and 8.6 dB for direct link from satellite to indoor, AF cooperative scheme, and DF cooperative scheme, respectively. The SNR values are calculated using (9), (10), (16), (18) and the values in Table 1.

In the case of DVB-S, we use (9), (10), (16) and Table 1 values. The corresponding C/N for direct transmission, cooperative transmission using AF protocol, and cooperative transmission using DF protocol are 11.2 dB, 9.6 dB and 9 dB, respectively. The E_b/N_0 for direct transmission, cooperative transmission using AF protocol, and cooperative transmission using DF protocol are 9.6 dB, 8 dB and 7.4 dB, respectively (assuming modulated QPSK signal and FEC to read Solomon = 2/3). The target BER 10^{-3} E_b/N_0 improvement is more than 1.6 dB.

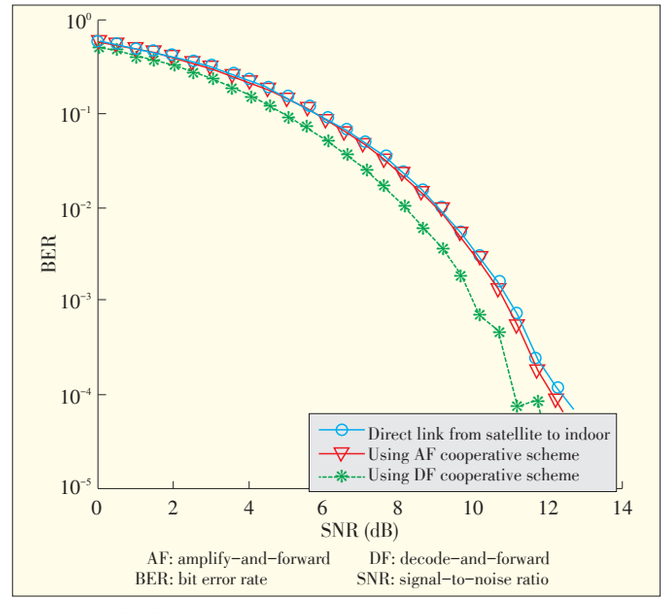
One of our goals is to minimize the overall BER of the systems with the cooperative terminal located at different distances from the indoor terminal. Comparing **Figs. 4** and **5**, at BER 10^{-3} with AF protocol, the cooperative terminal at 1000 m requires 1 dB higher SNR in order to provide the same performance as at 10 m. Also from Figs. 4 and 5 lower SNR is required when the separation distance between the cooperative and indoor terminal decreases.

7 Conclusion

In this paper, the simulation results of a cooperative communication system are described for different cooperative protocols. These results are compared with the performance results of a traditional wireless systems operating over a single link (direct transmission from satellite to indoor terminal). The BER performance of all cooperative protocols (using MRC combining type) is better than that of the direct link. When the cooperative terminal is located 10 m from the WCDMA indoor terminal, the proposed schemes outperform in all cases, with approximately 1.2 dB and 2.1 dB gap at BER 10^{-3} for AF and DF cooperative cases, respectively. Moreover, the position of the cooperative terminal affects overall system performance. The



▲**Figure 4. BER vs SNR with a cooperative terminal 10 m from the WCDMA indoor terminal.**



▲**Figure 5. BER vs SNR with a cooperative terminal at 1000 m (total path loss 140 dB) from WCDMA indoor terminal.**

best performance is achieved when the cooperative terminal is located near the indoor terminal. At BER 10^{-2} and with AF protocol, the cooperative terminal at 1000 m requires 1.4 dB higher SNR to provide the same performance at 10 m in a WCDMA network. Finally, cell coverage is increased by reducing the required average received power.

Acknowledgments

I would like to thank my wife Dr. Alemnesh Woldeyes for her valuable comments and continuous encouragement.

References

[1] B. G. Evans, *Satellite Communication Systems*, 3rd ed. Stevenage, UK: The Institution of Engineering and Technology, 2008, pp. 172, 500–508.
[2] R. E. Sheriff and Y. F. Hu, *Mobile Satellite Communication Networks*. England: John Wiley & Sons, 2001, pp. 147–158.
[3] A. Jamalipour, *Low Earth Orbital Satellites for Personal Communication Networks*. Norwood, MA, USA: Artech House, 1998, pp.12–13.
[4] M. Richharia and L. D. Westbrook, *Satellite System for Personal Application: Concepts and Technology*. United Kingdom: John Wiley & Sons Ltd, 2010, pp. 57, 310.
[5] G. Corazza, *Digital Satellite Communications*. Italy: Springer Science + Business Media, LLC, 2007, pp.66–98.
[6] A. Lehner and A. Steingass, "A novel channel model for land mobile satellite navigation", in *Proc. ION GNSS 2005*, Long Beach, CA, USA, September 13–16, 2005, pp. 2132–2138. doi: 10.1.1.175.4035.
[7] M. Dohler and Y. Li, *Cooperative Communications Hardware, Channel & Phy*, UK: John Wiley & Sons, Ltd, 2010, pp. 9–13, 210–215.
[8] T. Abe, H. Shi, T. Asai, and H. Yoshino, "Relay techniques for MIMO wireless networks with multiple source and destination pairs," *EURASIP J. Wireless Commun. Networking*, vol. 2006, Issue 2, pp. 113–118, April 2006. doi: 10.1155/WCN/2006/64159.
[9] Z. Ding, I. Krikidis, J. Thompson, and K. K. Leung, "Physical layer network coding and precoding for the two–way relay channel in cellular systems," *IEEE Trans. Signal Processing*, vol. 59, No. 2, pp. 696–712, February 2011. doi: 10.1109/TSP.2010.2081985.
[10] F. H. P. Fitzek and M. D. Katz, *Cooperation in Wireless Networks: Principles and Applications*. Netherlands: Springer, 2006, pp. 168–170.
[11] A. Scaglione, D. Goeckel, and J. N. Laneman, "Cooperative communications in mobile ad–hoc networks: Rethinking the link abstraction," *IEEE Signal Processing Mag.*, vol. 23, pp. 18–29, September 2006. doi: 10.1.1.134.3364.
[12] M. Uysal, *Cooperative Communications For Improved Wireless Network Transmission: Framework For Virtual Antenna Array Applications*. Hershey, PA USA: IGI Publishing, 2010, pp. 260–262,.
[13] S. Said, S. El–Arabie, M. I. Dessouky, and S. El–Arabie, "Performance of cooperative diversity in cognitive relay networks," *Journal of Theoretical and Applied Information Technology*, vol. 39, no.1, pp. 98–103, May 2012.
[14] S. Shamai and A. D. Wyner, "Information–theoretic considerations for symmetric, cellular, multiple–access fading channels—part I," *IEEE Trans. Information Theory*, vol. 43, no. 6, pp. 1877– 1894. Nov. 1997. doi: 10.1109/18.641553.
[15] N. Song, G. Del. Galdo, M. Milojević, M. Haardt, and A. Heuberger, "Spatial availability in satellite–to–indoor broadcasting communications," in *Proc. 7th Workshop Digital Broadcasting*, Erlangen, Germany, Sep. 2006, pp. 113–118.
[16] H. Shin, and J. B. Song, "MRC analysis of cooperative diversity with fixed–gain relays in nakagami–m fading channels," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2069–2074, June. 2008. doi: 10.1109/TWC.2008.070812.
[17] K. J. Ray Liu, A. K. Sadek, W. Su and A. Kwasinski, *Cooperative Communications and Networking*. UK: Cambridge University Press, Jan. 2009, pp. 121–127.
[18] T. Himsoon, W. Su, and K. J. Ray Liu, "Differential transmission for amplify–and–forward cooperative communications," *IEEE Signal Processing Lett.*, vol. 12, no. 9, pp. 597–600, Sep. 2005. doi: 10.1109/LSP.2005.853067.
[19] S. G. Glisic, *Advanced Wireless Communications: 4G Cognitive and Cooperative Broadband Technology*, 2nd Ed. England: John Wiley & Sons Ltd, 2007, pp.592–596.
[20] J. Proakis and M. Salehi, *Digital Communications*, 5th Ed. Berkshire, UK: McGraw–Hill, 2008, pp. 174–192, 852.
[21] Y. Zhao, R. Adve, and T. J. Lim, "Symbol error rate of selection amplify–and–forward relay systems," *IEEE Commun. Lett.*, vol. 10, no. 11, pp. 757–759, Nov. 2006. doi: 10.1109/LCOMM.2006.060774.
[22] T. Nechiporenko, K. T. Phan, C. Tellambura, and H. H. Nguyen, "On the capacity of Rayleigh fading cooperative systems under Adaptive Transmission," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1626–1631, April. 2009. doi: 10.1109/T–WC.2008.071098.
[23] J. N. Laneman, D. N .C. Tse, and G.W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Trans. Information Theory*, vol. 50, no. 12, pp. 3062 – 3080, Dec. 2004. doi: 10.1109/TIT.2004.838089.
[24] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity—part I: system description," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1927–1938, Nov. 2003. doi: 10.1109/TCOMM.2003.818096.
[25] G. Durgin, T. S. Rappaport, and H. Xu, "Measurements and models for radio path loss and penetration loss in and around homes and trees at 5.85 GHz," *IEEE Trans. Commun.*, vol. 46, no. 11, pp. 1484– 1496, Nov. 1998. doi: 10.1109/26.729393.
[26] B. R. Elbert, *The Satellite Communication Applications Handbook*, 2nd ed. Norwood, MA, USA: Artech House, 2001, pp. 2, 395–400.
[27] S. Dimov Ilcev, *Global Mobile Satellite Communications: For Maritime, Land and Aeronautical Applications*, New York, USA: Springer–Verlag New York Inc., 2005, pp. 235–242.
[28] G. Maral, M. Bousquet, Z. Sun, *Satellite Communications Systems: Systems, Techniques and Technology*, 5th Ed. UK: John Wiley & Sons Ltd., 2009, pp. 163–170, 364–367.
[29] D. Roddy, *Satellite Communications*, 3rd Ed. USA: McGraw–Hill Professional, 2001, pp. 316–330.
[30] T. T. Ha, *Theory and Design of Digital Communication Systems*. New York, USA: Cambridge University Press, 2011, pp. 215–218.
[31] B. R. Elbert, *The Satellite Communication Ground Segment and Earth Station Handbook*. USA: Artech House, Inc., 2001, pp. 343–345.
[32] M. O. Kolawole, *Satellite Communication Engineering*. New York, USA: CRC Press 2002, pp. 146–151.
[33] J. E. Kadish and T. W. R. East, *Satellite Communication Fundamental*. USA: Artech House, 2000, pp. 286–288.
[34] M. Richharia and L. D. Westbrook, *Satellite Systems for Personal Applications: Concepts and Technology*. UK: John Wiley & Sons, 2010, pp. 161–162.
[35] M. Ahmed, Md. A. Hossain, A. F. M. Zainul Abadin, and P. Mohon Ghosh, "BER performance evaluation of a cooperative wireless communication system with CDMA implementation of fixed relaying protocols," *Global Journal of Computer Science and Tech. Network, Web & Security*, vol. 13, issue 1, version 1.0, pp. 27–33, Sep. 2013.
[36] R. Hoppe, T. Hager, T. Heyn, A. Heuberger, H.Widmer, et al., "Simulation and measurement of the satellite to indoor propagation channel at L– and S–band," in *Proc. EuCAP 2006*, Nice, France, Nov. 2006, pp. 1–6. doi: 10.1109/EUCAP.2006.4584781.
[37] J. S. Seybold, *Introduction to RF Propagation*. New Jersey, USA: John Wiley & Sons, 2005, pp. 209–214.
[38] N. Blaunstein and C. Christodoulou, Radio Propagation and Adaptive Antennas *for Wireless Communication Links: Terrestrial, Atmospheric and Ionosphere*. New Jersey, USA: John Wiley & Sons, 2006, pp. 305–315.
[39] M. Barkat, *Signal Detection and Estimation*, 2nd Ed., USA: Artech House, 2005, pp. 131.

Manuscript received: October 11, 2013

Biography

Ashagrie Getnet Flattie (edenashagrie@gmail.com) received his BSc degree in electrical engineering (specializing in communication technology) from Defense University, Debre Zeit, Ethiopia, in 2003. He received his MSc degree in electrical and computer engineering (majoring in communication engineering) from Addis Ababa University, Ethiopia, in 2008. In 2003, he was a lecturer in the Ethiopian Air Force and later worked at the Information Network Security Agency (INSA) of Ethiopia. Over the past six years, he has worked in telecommunications as a mobile planning and optimization manager and also a quality performance manager. Currently, he is an engineering RAN manager at Ethio–Telecom, Addis Ababa. His fields of interest include wireless communications. He has published six research papers in international conference proceedings and has been awarded best oral presenter at the International Conference on Wireless Networks (ICWN'12) in Thailand.

DOI: 10.3939/j.issn.1673-5188.2013.04.008
http://www.cnki.net/kcms/detail/34.1294.TN.20130923.1023.001.html, published online September 23, 2013

Capacity Scaling Limits and New Advancements in Optical Transmission Systems

Zhensheng Jia
(Optics Lab, ZTE USA, NJ 07960, USA)

Abstract

Optical transmission technologies have gone through several generations of development. Spectral efficiency has significantly improved, and industry has begun to search for an answer to a basic question: What are the fundamental linear and nonlinear signal channel limitations of the Shannon theory when there is no compensation in an optical fiber transmission system? Next-generation technologies should exceed the 100G transmission capability of coherent systems in order to approach the Shannon limit. Spectral efficiency first needs to be improved before overall transmission capability can be improved. The means to improve spectral efficiency include more complex modulation formats and channel encoding/decoding algorithms, prefiltering with multisymbol detection, optical OFDM and Nyquist WDM multicarrier technologies, and nonlinearity compensation. With further optimization, these technologies will most likely be incorporated into beyond-100G optical transport systems to meet bandwidth demand.

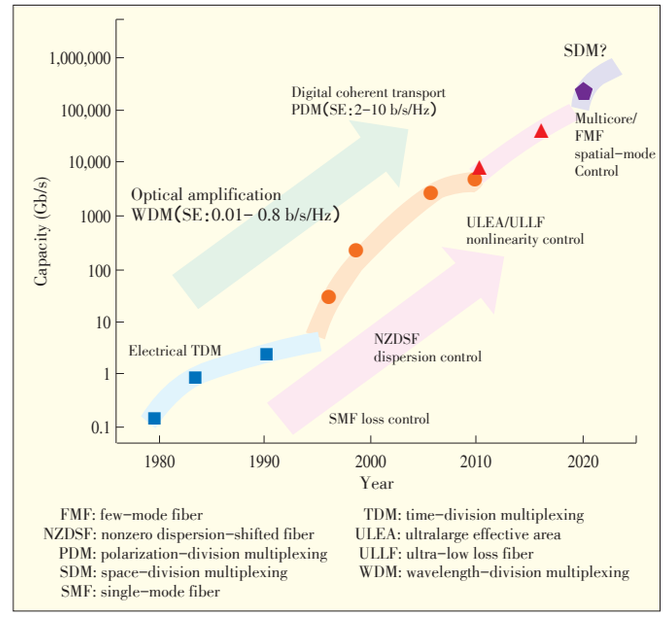
Keywords

spectral efficiency; Shannon limit; Gaussian noise; optical signal noise ratio; modulation; nonlinearity compensation

1 Service and Optical Transmission Capacity Requirements

With the rapid development of online video, large-scale cloud computing, and mobile internet, the amount of traffic flowing through telecom networks will continue to grow. The Minnesota Internet Traffic Studies and the Discovery Institute in North America has predicted that the bandwidth demand of internet services has increased 50%-60% since 1996 [1], [2]. This prediction is fully consistent with the current service development. Underlying optical transmission technologies have undergone many changes to meet upper-layer service requirements. After the breakthroughs in semiconductor lasers and low-loss single-mode fiber (SMF) in the 1970s, optical transmission technologies have been developing rapidly for dozens of years. **Fig. 1** shows the important stages of in this development. From the 1980s to early 1990s (the first stage of development), electrical time-division multiplexing (ETDM) was the core technology. The main technical issue in optical transmission was performance stability of optical components such as lasers and filters. The invention of the Erbium-doped fiber amplifier (EDFA) in the 1990s and the first commercial use of 8 × 2.5 Gbit/s wavelength-division multiplexing (WDM) in 1996 were important milestones along the path to improving optical fiber capacity. Optical fiber has evolved from early loss-reduction optical fiber to first-order and second-order dispersion-managed fiber (DMF). Dispersion-shifted fiber (DSF) and nonzero dispersion- shifted fiber (NZDSF) have also emerged. These developments have greatly helped overcome linear impairments in optical fibers; they have made long- distance transmission possible; and they have dramatically improved the spectral efficiency of optical signals. At this stage, optical signal modulation, coding, detection, and L- band utilization are the main hotspots in optical transmission research.

The third technological leap occurred during the mid to late 2000s. The rapid development of silicon- based electronic chips and maturing of signal processing technologies reinvigorated the coherent detection, which are now core components for digital signal processing (DSP) assisted optical coherent



▲Figure 1. The evolution of optical communication technologies for commercial use.

This work is supported by National High-Tech Research and Development Program of China under Grant No. 2013AA010501.

transmission. Solutions to dispersion compensation; PDM; dispersion recovery; and carrier frequency, phase and clock synchronization have all been found from the coherent receiver chip based on a DSP algorithm. This has increased the spectral efficiency of optical signals to 2 bit/s/Hz, and optical transmission has entered the stage of digital coherent transmission of four-dimensional orthogonal signals (i.e. X-polarized and Y-polarized I and Q signals). To further improve spectral efficiency, QPSK modulation has been evolved to multilayer signaling modulation, such as 16-QAM. Multicarrier multiplexing technologies, such as OFDM, Nyquist WDM, and electric and optical variants, have become hot research topics, and attempts have been made to use these commercially. Channel-coding technologies have also incorporated soft-decision forward error correction (FEC), which improves the signal decoding quality so that the technologies are compatible with multiple nodes and do not affect transmission distance. Optical fiber capacity, efficiency, and transmission distance needs to be balanced with different levels of complexity and cost.

Nonlinear impairment is another technical problem to be tackled. There are a variety of nonlinear, digital-domain compensation methods, such as digital back propagation (DBP). However, compensation algorithms are difficult to implement in chips because such algorithms are complex. Therefore, compensation algorithms are still being studied in the labs. New optical fibers have evolved to SMFs with increased effective area (ULEA) and reduced propagation loss (ULL). A standard SMF has relatively more difficult nonlinear phase-matching conditions because of the existence of large dispersion; thus, it more tolerant than NZDSF to nonlinearity effect. In the future, SDM is likely to become a technological turning point for further increasing capacity. Multicore and multimode fibers (MMFs) still need to be technically improved, and many factors need to be researched. Generally speaking, MMFs have been experimentally shown to have superior performance and are thus predicted to be used widely in future applications.

Before SDM is used, requirements related to ubiquitous service growth and during technological development need to be taken into account. Three basic questions need answering: What is the fundamental optical fiber capacity? What is the highest possible spectral efficiency within 4-5 THz bandwidth at band C? What modulation and coding technologies can approach the ideal upper limit? In the subsequent sections, we try to answer these questions.

2 Shannon Limit

2.1 Linearity

Claude E. Shannon described channel system capacity in 1948 [5]. Shannon's description focused on the additive white Gaussian noise (AWGN) channel, which can reliably transmit information at the upper signal-rate limit. In other words, when the signal rate is lower than the theoretical Shannon limit, complex (but effective) modulation and coding technologies can be used for reliable transmission. The applicable prerequisite is that the input power is limited and the noise variance is not zero. The basic relationship is defined in the following equation:

$$\begin{aligned} SE &= C/B \\ &= \log_2(1+SNR_S) \quad \text{or} \\ SE &= \log_2(1+SNR_b \times SE) \end{aligned} \tag{1}$$

where C is the system capacity, B is the channel bandwidth, SE is the system capacity per bandwidth (also called spectral efficiency). The signal-to-noise ratio (SNRs) is the ratio of the energy per symbol to noise and is given by

$$\begin{aligned} SNR_S &= P/N_O R_S \\ &= E_S/N_O \end{aligned} \tag{2}$$

where E_S represents the energy per symbol, R_S represents the symbol rate of a signal, $P = E_S R_S$, and N_O represents the noise power spectral density. For every bit,

$$\begin{aligned} SNR_b &= E_b / N_O \\ &= SNR_S / \log_2 M \\ &= SNR_S / SE \end{aligned} \tag{3}$$

where $\log_2 M$ is the number of bits per symbol, M is the size of the alphabet, and E_b is the energy per bit. **Fig. 2(a)** shows several typical modulation formats for a memoryless single polarization single channel in terms of the linear Shannon limit of the SNR function per symbol, (based on Gaussian noise distribution). This figure shows that all modulation formats are converged to their own spectral efficiencies as the SNR per symbol is increased. In **Fig. 2(b)**, as the dotted lines are increased, the high-order modulation format approaches the Shannon limit, and the SNR per symbol poses higher requirements for reaching saturation. In addition, QAM formats, such as PSK and ASK, converge much faster than phase modulation formats because of different Euclidean distances when the 16-PSK with 16-QAM curves are compared.

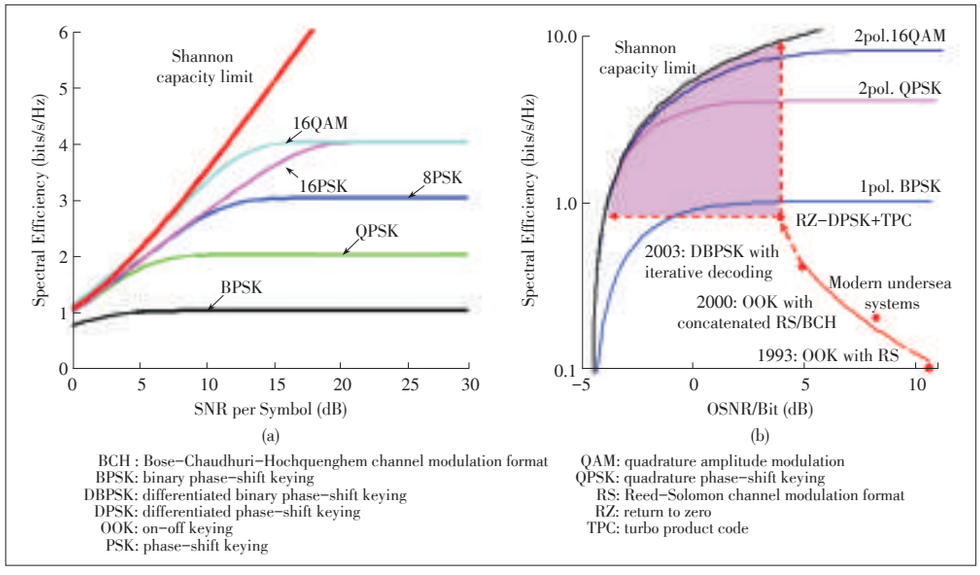
This theory can be applied to optical communication [6], [7]. Using the one dimension polarization space, optical signal SEs can be doubled. At the same time, the SNR per symbol or the SNR per bit can be replaced by the $OSNR$:

$$\begin{aligned} OSNR_{0.1\text{nm}} &= P / 2N_{ase}B_{ref} \\ &= SNR_b R_b / 2 \times 12.5\ \text{GHz} \end{aligned}$$

or

$$\begin{aligned} OSNR_b &= P / 2N_{ase}R_b \\ &= SNR_S / (2 \times SE) \end{aligned} \tag{4}$$

where $OSNR_{0.1\ \text{nm}}$ is the $OSNR$ in 0.1 nm. Fig. 2 (b) shows the dual-polarization Shannon limits of the $OSNR$ per bit function for several modulation formats. This figure also shows that commercial systems have evolved through multiple generations of technology to approach the Shannon limit. These systems have



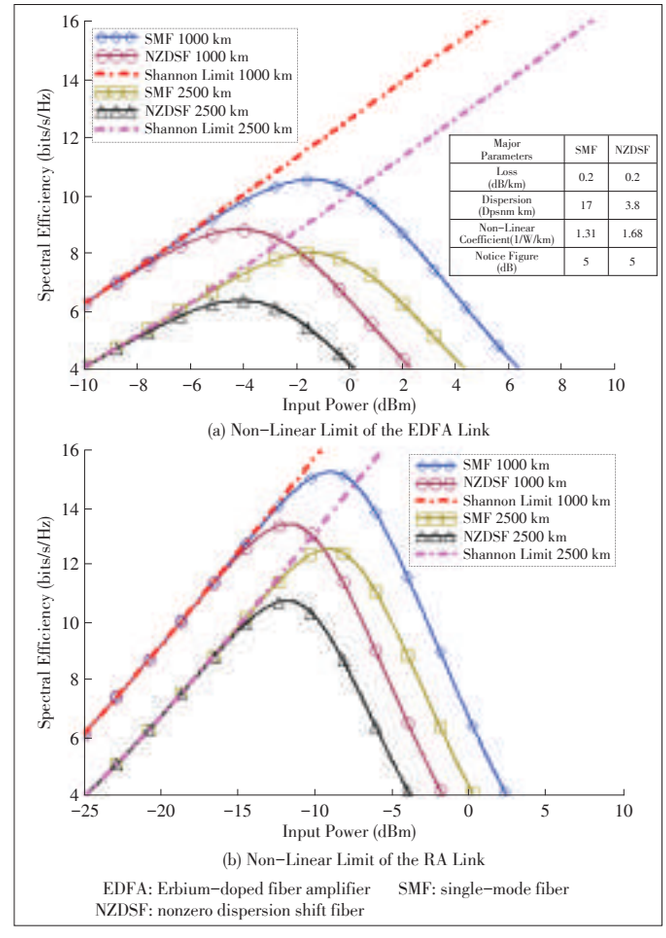
▲Figure 2. a) Single-polarization and b) dual-polarization linear Shannon limits of typical modulation formats.

evolved through the early strength modulation and Reed-Solomon FEC coding to the later differential binary phase-shift keying (DBPSK) and FEC coding/decoding technologies. The required *OSNR* per bit has been reduced while the SE has increased. The RZ-DPSK + TPC point is closest to the experimental Shannon limit in the case of non-coherent receiving. The shaded area in Fig. 2(b) is the space the SE can be enhanced (i.e. improved *OSNR*) using QPSK or 16-QAM, more complex FEC technology, and the DSP algorithm.

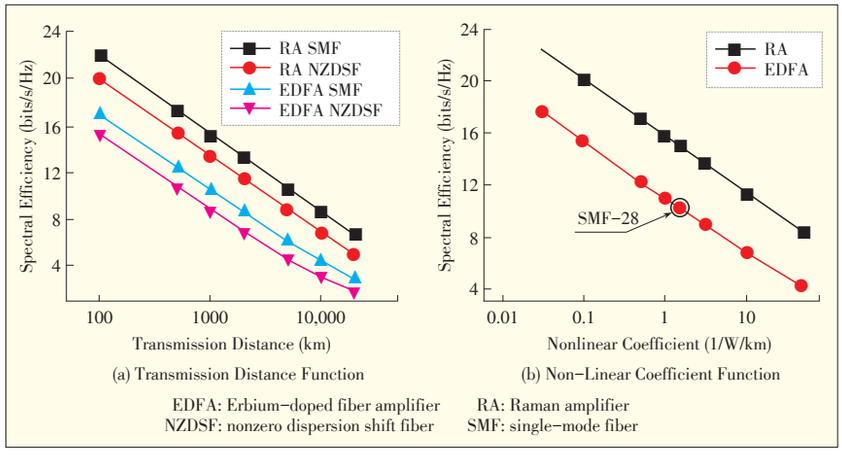
2.2 Non-Linear Requirements

Unlike in a wireless channel, an optical fiber demonstrates the non-linear Kerr effect when there is high input power. This significantly changes the refractive index, which introduces the nonlinear effects, such as the self-phase modulation (SPM), cross-phase modulation (XPM), and four-wave modulation (FWM) of an optical fiber. Therefore, there are two boundaries in a non-linear optical channel. In the case of low power, a nonlinear optical channel is limited by amplified spontaneous emission (ASE) noises from optical amplifiers. In the case of high power, the nonlinear effect of an optical channel controls the achievable channel capacity. In nonlinear conditions, the noise within the whole signal bandwidth needs to be considered, and interchannel interaction has a severe effect. **Fig. 3(a)** shows the highest spectral efficiency of the EDFA link in the optimized Gaussian constellation diagram of signal distribution and without nonlinear compensation. **Fig. 3(b)** shows the effects of Raman amplification. These two figures also show comparisons of commonly used SMFs with NZDSFs over 1000 km and 2500 km. Fig. 3(a) shows some parameters of the main optical fibers and components. There are two distinct features: The maximum value of the same optical fiber is reached at the same EDFA (SMF = −1.3 dBm, NZDSF = −4 dBm) or RA (SMF = −9 dBm, NZDSF = −11.7 dBm), irrespective of the transmission distance. Over the same transmission distance, SMF is superior to NZDSF because NZDSF has a higher nonlinear coefficient and smaller dispersion in favorable nonlinear phase- matching conditions. A Raman amplifier (RA) link is better than an EDFA link over the same transmission distance. However, it is easy to ignore in Raman link because the whole link stably provides a high-power level and lower nonlinearity tolerance, and its input power is lower than that of the EDFA link. When comparing the benefits of an RA link with those of an EDFA link, these factors need to be taken into account.

The basic physical parameters remain unchanged. **Fig. 4(a)**



▲Figure 3. Nonlinear limits of the EDFA link and RA links.



▲**Figure 4. Relationship between the transmission distance and nonlinear coefficient for achieving the highest spectral efficiency.**

shows the relationship between the transmission distance (including access, metro, long-distance, and transoceanic submarine communication networks) and the SE of two optical fibers using different amplification mechanisms. As the distance increases, SE diminishes linearly. When the transmission distance decreases by three orders of magnitude from the submarine communication to the access, there is only a threefold increase in SE. Therefore, it is very difficult to increase the SE of an optical communication network. In addition, a Raman or standard SMF performs better than an NZDSF. **Fig. 4 (b)** shows the highest possible SE of the EDFA and RA links when the nonlinear coefficient of the optical fiber is changed over 1000 km. When the nonlinear coefficient decreases from ten orders of magnitude to three orders of magnitude, there is only a threefold increase in SE. Fig. 4(b) also shows the location of a standby SMF. When the EDFA is amplified, the SE reaches 10 bit/s/Hz, and the RA reaches 14 bit/s/Hz.

3 Forward-Error Correction Channel Coding and Decoding

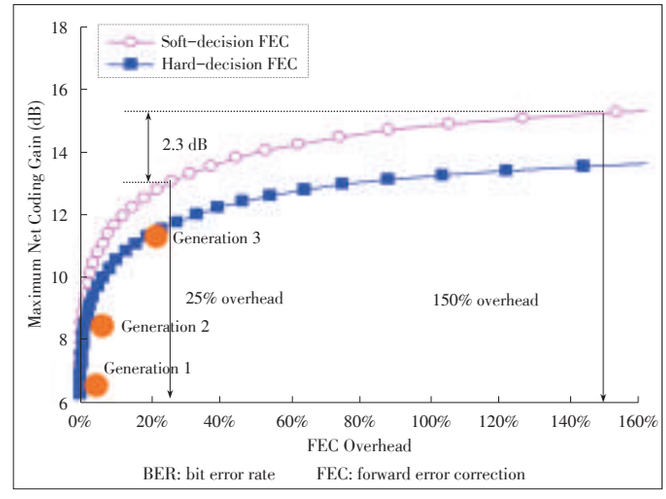
Despite the impact from the modulation format and nonlinear effect, FEC is another very powerful tool to enhance transmission performance. FEC is a channel coding/decoding technology that has evolved through three generations of technology: from the early classic Reed-Solomon (255, 239) hard-decision with 6 dB coding gain to a cascade coder and crossing/iterating/convolutional decoder with an additional 2-3 dB coding gain. Current FEC technology has soft-decision turbo product code (TPC) or low-density parity check (LDPC) with larger than 11 dB net coding gain (NCG). Another fundamental question comes up: what is the theoretical limit of the FEC coding and decoding process? **Fig. 5** shows the maximum theoretical limit of an optimal soft/hard decision FEC with different proportions of overhead. It is easily seen that when the overhead increases from 25% to 150%, the theoretical NCG increases by 2.3 dB. With different proportions of overhead or code rate, the difference between a soft-decision FEC and a hard-decision FEC is approximately 1-2 dB. The mathematical algorithm of the soft decision is mature; however, it was not actually used in optical communications until the processing speed, power consumption, and integration level of semiconductors matured. Decreasing the error floor (EF) and using more complicated decoding technologies can further improve soft-decision FED for approaching the theoretical coding gain limit.

4 Key Technologies for Approaching the Shannon Limit

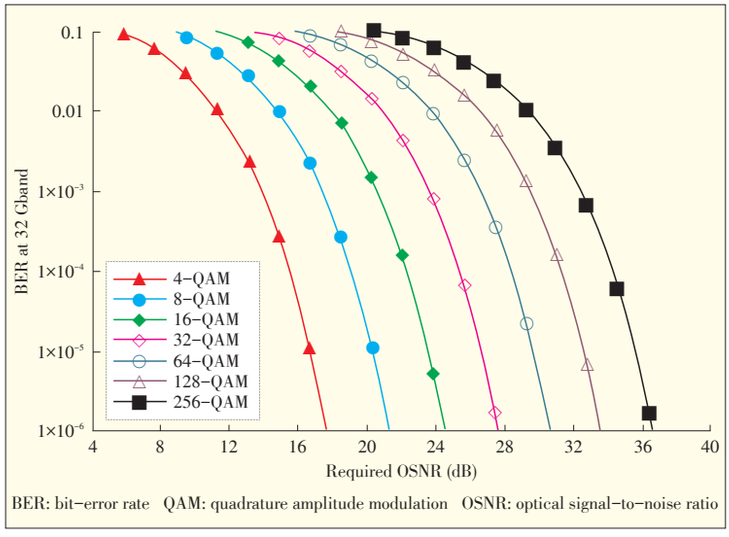
Although besides the adoption of new optical fibers with low non-linearity and ultra low loss and complicated soft-decision FEC to improve transmission performance, The other key technologies for approaching the Shannon limit include more complex modulation formats and effective nonlinear compensation. In addition, an enhanced algorithm that creates signals with memory can also break the existing Shannon limit of memoryless signals.

4.1 More Complex Modulation Format

From the Shannon limit curve, the greater the amplitude and phase modulation (e.g. from QPSK to 16-QAM), the closer the constellation diagram approaches the optimized Gaussian distribution and the closer the theoretical limit becomes to the Shannon limit. Signals from 8-QAM, 16-QAM, 32-QAM to 256-QAM formats have been demonstrated in the laboratory. However, the transmission symbol rate and distance are very limited because of high OSNR requirements and high implementation costs. **Fig. 6** shows the OSNR BER curves of multi-



▲**Figure 5. Maximum theoretical coding gains of the soft-decision FEC and hard-decision FEC for BER = 10^{-15}.**



▲Figure 6. OSNR-BER curves of multiple modulation formats.

ple modulation formats. Comparing QPSK with 16-QAM and 256-QAM, the required OSNR of 6.7 dB is different from that of 18.6 dB when the BER is 1×10^{-3}. This means a shorter transmission distance (Fig. 4).

4.2 Multisymbol Simultaneous Detection from Memoryless Signals to Memory Signals

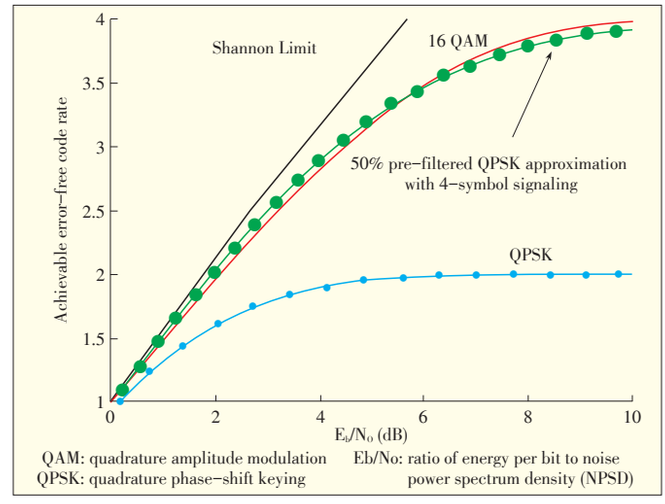
The memory signals refers to the intersymbol correlation within the time domain (e.g. intersymbol interference (ISI) resulting from dispersion or strong filtering). This correlation leads to intersymbol energy penetration and exchange. In this case, the best decision criterion is not the single symbol or bit decision but the multisymbol sequence detection decision, which can be implemented through a DSP algorithm such as maximum likelihood sequence estimation (MLSE) or maximum a posteriori (MAP). The algorithm for simultaneously detecting strong filter signals (e.g. diminishing the signal power and bandwidth to 0.8 W or even 0.5 W by using the original filter with the W bandwidth) with the sequence detection at the receiver can exceed the theoretical Shannon limit for the signals without any memory in the same modulation format. The pre-filtered QPSK and 16-QAM are shown in **Fig. 7**. The transmission capability of 50% filtered QPSK signals is already approaching 16-QAM. In terms of hardware and algorithm, however, the complexities of the transmitter and the receiver are significantly increased.

4.3 Sinc-Function-Shaped Signals

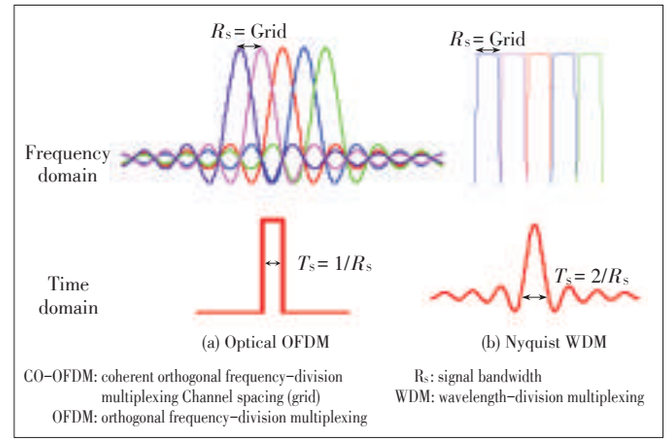
By proactively introducing the intersymbol interference, a strong filtering is to use simultaneous multi-signal detection at the receiving end to improve spectral efficiency. By ideally introducing the zero-cost interchannel interference (ICI) or ISI in the frequency or time domain, a similar technology can come closer to the Shannon limit through spectrum shaping at the sending end. **Fig. 8** shows the OFDM and Nyquist WDM [13] in the frequency domain (spectrum) and in the time domain (pulse). Coherent orthogonal frequency-division multiplexing (CO-OFDM) indicates that the ideal ISI of the rectangular-shaped transmission pulse is zero in the time domain; however, in the frequency domain, each signal can be demodulated without any impairment because of the orthogonality of CO- ODFM (even though multiple Sinc- function- shaped subcarriers overlap). In this domain, the Nyquist WDM is rectangular, and its ideal ICI is zero. In the time domain, each carrier channel carries Sinc-function-shaped signals. These two technologies have become the first choices for establishing a super channel.

4.4 Nonlinear Compensation

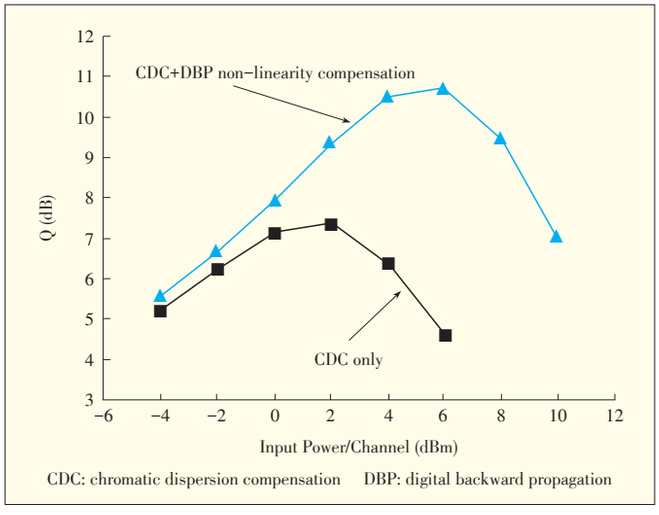
Because of nonlinearity, performance is degraded although OSNR is increased, and the work region enters the nonlinear area accordingly when the input power is increased high enough (**Fig. 9**). Nonlinear compensation can im-



▲Figure 7. Prefiltered QPSK and 16-QAM comparison.



▲Figure 8. CO-OFDM and Nyquist WDM signals in both the frequency and time domains.



▲**Figure 9. Performance improvement ssing the non-linear DBP compensation.**

prove the optimum input power, and can be used to approach the Shannon limit and improve system transmission capacity. Nonlinear compensation algorithms include MLSE, Volterra series equalizer, digital backward propagation (DBP), and radio frequency (RF) pilot tone [14], [15]. Without further algorithm simplification, MLSE and Volterra methods are difficult to be applied in 100G systems (or higher) for nonlinear compensation because of hardware implementation. The DBP method with Fourier Transform (FT) can compensate for the SPM. Interchannel XPM compensation requires the information of the entire optical fiber channel. If its steps and algorithm were improved, DBP would be the first choice for use in a dispersion-compensation channel. Some studies in optical OFDM systems have proven that the RF pilot tone can compensate for SPM and XPM to some extent. These algorithms are not completely separated but can be used together. Before they can be used in a real commercial system, the complexity associated with implementing them needs to be lowered. Meanwhile, system performance also needs to be maintained.

5 Conclusion

The Shannon limit is a fundamental theory in the communication systems. With the rapid increase of signal bandwidth for various services, underlying optical transmission technologies have gone through several technical evolutions. For the reasonable transmission distance, higher requirements have been put on spectral efficiency (i.e. total optical fiber transmission capacity). Under this context, besides the evolution of optical fiber like multi-core or multi-mode fiber, technologies for approaching the Shannon limit have become the research hotspots. These technologies include more complicated modulation format, channel coding and channel decoding; pre-filtering and associated simultaneous multisymbol detection algorithms; CO-OFDM and Nyquist WDM multicarrier technologies; and compensation solutions for fiber nonlinearity. With the optimization of these technologies individually or collectively and advancement of semiconductor chips, Next beyond-100G systems will approach the Shannon limit more closely to meet future bandwidth demands.

References

[1] R. W. Tkach, "Scaling optical communications for the next decade and beyond," *Bell Labs Technical Journal*, vol. 14, no. 4, pp. 3-9, 2010.
[2] University of Minnesota, Minnesota Internet Traffic Studies (MINTS) [Online]. Available: http://www.dtc.umn.edu/mints/home.php
[3] P. J. Winzer, "Energy-efficient optical transport capacity scaling through spatial multiplexing," *IEEE Photonic Technology Letters*, vol. 23, no. 13, pp. 851-853, 2011.
[4] R. Ryf, C. A. Bolle, and J .Von. Hoyningen-Huene, "Optical coupling components for spatial multiplexing in multi-mode," in *Proc. of the 37th European Conference on Optical Communication(ECOC' 11)*, Geneva, Switzerland, Sep 18-22,2011.
[5] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, no. 27, pp. 379-423, 1948.
[6] Y. Cai and A. N. Pilipetskii, "Channel capacity of fiberoptic communication systems with amplified spontaneous emission noise," in *Proc. of the Optical Fiber Communication/ National Fiber Optic Engineers Conference (OFC/NFOEC' 05)*, Anaheim, CA, USA, Mar 6-11, 2005.
[7] R. J. Essiambre, G. J. Foschini, G. Kramer, et al. "Capacity limits of information transmission in optically-routed fiber networks," *Bell Labs Technical Journal*, vol.14, no. 4, pp. 149-162, 2010 .
[8] Y. Cai, "Coherent detection in long-haul transmission systems," in *Proc. of the Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference (OFC/NFOEC' 08)*, San Diego, CA, USA ,Feb 24-28, 2008.
[9] P. Poggiolini, A. Carena, V .Curri et al, "Analytical modeling of nonlinear propagation in uncompensated optical transmission links," *IEEE Photonic Technology Letters*, vol. 23, no. 11, pp. 742-744, 2011.
[10] K. Ouchi, K. Kubo, T. Mizuochi et al, "A fully integrated block turbo code FEC for 10 Gb/s optical communication systems," in *Proc. Optical Fiber Communication/National Fiber Optic Engineers Conference (OFC/NFOEC' 06)*, Anaheim, CA, USA, Mar 5-10, 2006.
[11] Z. S. Jia, J. J. Yu, H. C. Chien, et al, "Field transmission of 100 G and beyond: Multiple baud rates and mixed line rates using Nyquist- WDM technology," *Journal of Lightwave Technology*, vol. 30, no. 24, pp. 3793-3804, 2012.
[12] Y. Cai, J. X. Cai, A. Pilipetskii, et al, "Spectral efficiency limits of pre-filtered modulation formats," *Optics Express*, vol. 18, no. 19, pp. 20273-20281, 2010.
[13] J. J. Yu, Z. Dong, H. Chien et al, "Transmission of 200 G PDM-CSRZ-QPSK and PDM-16 QAM with a SE of 4 b/s/Hz," *Journal of Lightwave Technology*, vol. 31, no. 4, pp. 515-522, 2013.
[14] X. X. Li, X. Chen, G. Goldfarb, et al, "Electronic post-compensation of WDM transmission impairments using coherent detection and digital signal processing," *Optics Express*, vol. 15, no. 2, pp. 880-888, 2008.
[15] A. Diaz, A. Napoli, S. Adhikari, et al.,"Analysis of back-propagation and RF pilot- tone based nonlinearity compensation for a 9 × 224Gb/s POLMUX-16QAM system," in *Proc. of the Optical Fiber Communication/National Fiber Optic Engineers Conference (OFC/NFOEC' 12)*, Los Angeles, CA, USA, Mar 4-8, 2012.

Manuscript received: September 25, 2013

Biography

Zhensheng Jia (Zhensheng.jia@zte.com.cn) received his BE and MSE degrees from Tsinghua University, China. He received his PhD degree from Georgia Institute of Technology, USA. He is currently the assistant director of the Optics Lab, ZTE USA. Prior to joining ZTE, he worked for Beijing Research Institute of China Telecom Corporation Limited and Telcordia Technologies, US. Dr. Jia has authored more than 100 articles and has given a number of invited talks.

ZTE Communications Call for Papers

Special Issue on
Wireless Body Area Networks for Pervasive Healthcare and Smart Environments

Wireless body area networks (WBANs) employ radio-frequency communications to interconnect tiny sensor nodes in, on, or around human bodies to provide continuous monitoring of physiological signals, physical activities or body positions. It is challenging to design WBANs due to the limited energy capacity and processing capabilities of sensor nodes, and the highly variable radio communication environment that is prone to interference. Recent technological advances in wearable and implantable biosensors, short-range wireless communications, and low-power embedded processors are contributing to increasing pace of research and development on WBANs to address these challenges. WBANs usually function as signal sources in larger intelligent systems that enable useful applications with the potentials for great societal and economic impacts. Such intelligent systems are formed by connecting WBANs with off-body communications and computing infrastructure; e.g., accessing cloud-computing services through a smartphone that connects to the Internet via a wide-area wireless network. There is a strong interest among researchers and practitioners in the development of WBAN-based intelligent systems are enabling pervasive healthcare applications, such as ambulatory monitoring of out-patients, and smart environments that support monitoring of emergency personnel and mission-critical workers, sport training, video gaming, context-aware applications, etc. The purpose of this special issue is to survey the state of the arts and disseminate the latest developments in WBAN technologies, systems and applications. Original papers are solicited, and submissions will be peer-reviewed before they are accepted for publication. Topics of interest include, but are not limited to the following:

- Measurements, modeling and regulatory aspects of WBAN radio channels
- Short-range communication and networking standards for WBANs
- Nano technologies for WBAN sensors and communications
- Signaling design and access techniques for WBANs
- Interference mitigation techniques for WBANs
- Energy harvesting techniques
- Cognitive and cooperative networking techniques for WBANs
- Interworking WBANs with wide-area networks and the Internet
- Architectures and protocols of WBAN-based intelligent systems
- Cloud computing services for WBAN-based intelligent systems
- Security and privacy issues in WBAN-based intelligent systems
- Pervasive healthcare applications enabled by WBANs
- Smart environment applications enabled by WBANs

Important Dates

Submission: 31 March, 2014
First decision: 28 April, 2014
Final manuscript due: 19 May, 2014

Guest Editors

Prof. Victor C. M. Leung (The University of British Columbia, vleung@ece.ubc.ca)
Prof. Hongke Zhang (Beijing Jiaotong University, hkzhang@bjtu.edu.cn)

2014 International Conference on Information and Communications Technologies

ICT 2014, Nanjing, China

16th –29th May, 2014

www.ietict.org

The 2014 International Conference on Information and Communications Technologies (ICT2014) will be held 16th – 18th May 2014 in Nanjing, China. ICT 2014 aims to provide an international platform for both academic scholars and industry leaders in fields of information and communication technologies to exchange novel ideas and latest research results. The conference includes not only technical sessions, but also invited sessions and keynote addresses.

You are invited to submit original papers to the conference. Submitted papers should not have been previously published or currently under review for any other publication. All accepted papers will be published in the conference proceedings and be submitted to IET inspection for IET Inspec and IEEE Xplore. After the conference, the accepted paper will be submitted for EI Compendex.

TOPICS OF INTEREST INCLUDE, BUT NOT LIMITED TO:

Communication Technology:
Optics and Optoelectronics
Secure Communications
Video and Broadcasting
Wireless and Satellite Communications
Communication Software and Services
Wireless Communications and Networking
Optical Communications and Networking
Signal Processing for Communications
Multimedia Communications
Green Communications & Computing

Networking Technology and Application:
Smart Networking
Optical Networks and Systems
Next-Generation Networking
Ad-hoc, Sensor and Mesh Networking
Network Architectures
Internet of Things
Cognitive Radio and Networks

Network and Information Security:
Network Security
Cryptology and Information Security
Mobile and Wireless Security
Information Hiding and Watermarking
Disaster Recovery

Information Theory and Application:
Digital Systems
Electronics, Computing and Control
Quality, Reliability, Security & Safety
Semiconductors and device
Signal Processing
Information Engineering

Computational Intelligence:
Artificial Intelligence and Its Application
Intelligent Data Management
Genetic Algorithms
Interactive computational models
Information Retrieval
Machine learning
Hybrid methods
Grid Computing
Natural Language Processing

Important Dates:
Paper Submission Deadline: Feb.25th, 2013
Acceptance Notification: Mar.15th, 2013
Final Manuscript Deadline: Mar.29th, 2013
Conference Date: Apr.27th-29th, 2013

Contact Us:
Website: www.ietict.org, www.ieccr.net
E-mail: ietict2013@gmail.com
Tel: 86-15712895816

ZTE Communications
Table of Contents, Volume 11, Numbers 1–4, 2013

Volume–Number–Page

SPECIAL TOPICS

QoE Modeling and Applications for Multimedia Systems

Big Data: Where Dreams Take Flight

Physical Layer Security for Wireless and Quantum Communications

ZTE Communications
Table of Contents, Volume 11, Numbers 1–4, 2013

Volume–Number–Page

Cloud Computing

RESEARCH PAPERS