www.zte.com.cn/magazine/English

ISSN 1673-5188
CODEN ZCTOAK

ZTE COMMUNICATIONS

An International ICT R&D Journal Sponsored by ZTE Corporation

December 2014, Vol.12 No. 4

SPECIAL TOPIC:

Improving Performance of Cloud Computing and Big Data Technologies and Applications

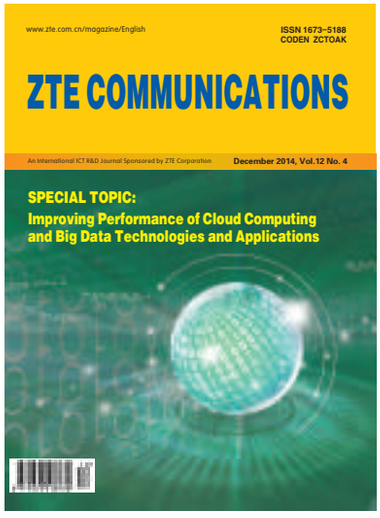
9 771673 518130 12>

ZTE中兴 ZTE COMMUNICATIONS

VOLUME 12 NUMBER 4 DECEMBER 2014

ZTE Communications Editorial Board

CONTENTS



Special Topic: Improving Performance of Cloud Computing and Big Data Technologies and Applications

CONTENTS

ZTE COMMUNICATIONS
Vol. 12 No. 4 (Issue 44)
Quarterly
First English Issue Published in 2003

Supervised by:
Anhui Science and Technology Department

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

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, 2014

Publication Licenses:

ISSN 1673-5188
CN 34-1294/ TN

Advertising License:
皖合工商广字0058号

Annual Subscription:
RMB 80

Research Papers

Roundup

Improving Performance of Cloud Computing and Big Data Technologies and Applications
Zhenjiang Dong

Improving Performance of Cloud Computing and Big Data Technologies and Applications

► Zhenjiang Dong

Zhenjiang Dong is the deputy head of the Cloud Computing and IT Research Institute of ZTE Corporation and a standing member and service team leader of the company's Committee of Corporate Strategy and Planning Experts. He is also an executive director of Chinese Association for Artificial Intelligence, a professor of Nanjing University of Science and Technology, a service computing expert of China Computer Federation. He has been responsible for more than 10 research projects supported by National High-Tech R&D Programs of China ("863" programs), Programs of Core Electronic Devices, High-End Generic Chips, and Basic Software Products of China, and National Science and Technology Major Project of China. His research interests include cloud computing, big data, and media analysis and processing.

Cloud computing technology is changing the development and usage patterns of IT infrastructure and applications. Virtualized and distributed systems as well as unified management and scheduling has greatly improved computing and storage. Management has become easier, and OAM costs have been significantly reduced. Cloud desktop technology is developing rapidly. With this technology, users can flexibly and dynamically use virtual machine resources, companies' efficiency of using and allocating resources is greatly improved, and information security is ensured. In most existing virtual cloud desktop solutions, computing and storage are bound together, and data is stored as image files. This limits the flexibility and expandability of systems and is insufficient for meeting customers' requirements in different scenarios.

In this era of big data, the annual growth rate of the data in social networks, mobile communication, e-commerce, and the Internet of Things is more than 50%. More than 80% of this data is unstructured. Therefore, it is imperative to develop an effective method for storing and managing big data and querying and analyzing big data in real time or quasi real. HBase is a distributed data storage system operating in the Hadoop environment. HBase provides a highly expandable method and platform for big data storage and management. However, it supports only primary key indexing but does not support non-primary key indexing. As a result, the data query efficiency of HBase is low and data cannot be queried in real time or quasi real time. For HBase operating in Hadoop, the capability of querying data according to non-primary keys is the most important and urgent.

The graph data structure is suitable for most big data created in social networks. Graph data is more complex and difficult to understand than traditional linked-list data or tree data, so quick and easy processing and understanding of graph data is of great significance and has become a hot topic in the industry.

Big data has a high proportion of video and image data but most of the video and image data is not utilized. Creating value with this data has been a research focus in the industry. For example, the traditional face localization and identification technology is a local optimal solution that has a large room for improvement in accuracy.

This special issue of *ZTE Communications* embodies the industry's efforts on performance improvement of cloud computing and big data technologies and applications. We invited four peer-reviewed papers based on projects supported by ZTE Industry-Academic-Research Cooperation Funds.

Hancong Duang *et al.* propose a disk mapping solution integrated with the virtual desktop technology in "A New Virtual Disk Mapping Method for the Cloud Desktop Storage Client." The virtual disk driver has a user-friendly mode for accessing desktop data and has a flexible cache space management mechanism. The file system filter driver intelligently checks I/O requests of upper applications and synchronizes

Improving Performance of Cloud Computing and Big Data Technologies and Applications

Zhenjiang Dong

file access requests to users' cloud storage services. Experimental results show that the read-write performance of our virtual disk mapping method with customizable local cache storage is almost same as that of the local hard disk.

"HMIBase: An Hierarchical Indexing System for Storing and Querying Big Data," by Shengmei Luo *et al.*, presents the design and implementation of a complete hierarchical indexing and query system called HMIBase. This system efficiently queries a value or values within a range according to non-primary key attributes. This system has good expandability. Test results based on 10 million to 1 billion data records show that regardless of whether the number of query results is large or small, HMIBase can respond to cold and hot queries one to four levels faster than standard HBase and five to twenty times faster than the open-source Hindex system.

In "MBGM: A Graph-Mining Tool Based on MapReduce and BSP," Zhenjiang Dong *et al.* propose a MapReduce and BSP-based Graph Mining (MBGM) tool. This tool uses the BSP model-based parallel graph mining algorithm and the MapReduce-based extraction-transformation-loading (ETL) algorithm, and an optimized workflow engine for cloud computing is designed for the tool. Experiments show that graph mining algorithm components, including PageRank, K-means, InDegree Count, and Closeness Centrality, in the MBGM tool has higher performance than the corresponding algorithm components of the BC-PDM and BC-BSP.

Bofei Wang *et al.* in "Facial Landmark Localization by Gibbs Sampling," present an optimized solution of the face localization technology based on key points. Instead of the traditional gradient descent algorithm, this solution uses the Gibbs sampling algorithm, which is easy to converge and can implement the global optimal solution for face localization based on key points. In this way, the local optimal solution is avoided. The posterior probability function used by the Gibbs sampling algorithm comprises the prior probability function and the likelihood function. The prior probability function is assumed to follow the Gaussian distribution and learn according to features after dimension reduction. The likelihood function is obtained through the local linear SVM algorithm. The LFW data has been used in the system for tests. The test results show that the accuracy of face localization is high.

I would like to thank all the authors for their contributions and all the reviewers who helped improve the quality of the papers.

DOI: 10.3969/j. issn. 1673-5188. 2014. 04. 001
http://www.cnki.net/kcms/detail/34.1294.TN.20141216.1723.001.html, published online December 16, 2014

A New Virtual Disk Mapping Method for the Cloud Desktop Storage Client

Hancong Duan[1], Xiaoqin Wang[1], Ping Lu[2], Shengmei Luo[2], and Zhiyong Wang[2]
(1. University of Electronic Science and Technology of China, Chengdu 610054, China;
2. ZTE Corporation, Nanjing 210012, China)

Abstract

Integration of the cloud desktop and cloud storage platform is urgent for enterprises. However, current proposals for cloud disk are not satisfactory in terms of the decoupling of virtual computing and business data storage in the cloud desktop environment. In this paper, we present a new virtual disk mapping method for cloud desktop storage. In Windows, compared with virtual hard disk method of popular cloud disks, the proposed implementation of client based on the virtual disk driver and the file system filter driver is available for widespread desktop environments, especially for the cloud desktop with limited storage resources. Furthermore, our method supports customizable local cache storage, resulting in user-friendly experience for thin-clients of the cloud desktop. The evaluation results show that our virtual disk mapping method performs well in the read-write throughput of different scale files.

Keywords

virtual disk; cloud desktop; file system filter driver; customizable cache storage

1 Introduction

In recent years, cloud storage has been one of the most popular internet applications with the maturity of cloud computing [1]. Cloud storage [2] makes various types of network storage devices work together by utilizing clustering server, network transmission and distributed file system. Compared with traditional storage devices, cloud storage is low-cost, high security, and flexible setup of capacity for easy expansion. With the popularity of mobile workforce and IT resource consumerization, desktop virtualization [3] as an important application of cloud computing has been developed.

A virtual machine (VM) which is based on thin-client data approach is emerging as a virtual desktop solution for data centers, such as Xen Desktop [4] and VMware View [5]. Most desktop storage clients of VMs run in the virtual operating system, of which the computing and storage resources are tight coupling in a virtual image file stored in the supervisor operating system. However, once the storage client in virtual desktops caches the entire user's cloud data, the size of the virtual image file increases. This results in inflexible data storage and scalability.

In this paper, we propose a virtual disk mapping method for cloud desktop storage client. This method is based on the virtual disk driver and file system filter driver and provides a flexible, user friendly cache management schema in a virtual desktop environment.

The remainder of this paper is organized as follows. Section 2 describes the related works; Section 3 presents the architecture of the cloud disk client in Windows, and proposes the mapping method based on the virtual disk drive and the file system filter driver in detail. In Section 4, we evaluate the performance of the cloud disk and compare it with the local disk with different data scales.

2 Related Works

Cloud disk services have blossomed in China and elsewhere since 2012 [6]. These services include Dropbox, Google-Drive, KingSoft KuaiPan, and Baidu CloudDisk. Most of the client sides of these cloud disks are available for Microsoft Windows, IOS, and Android. Because Windows is a widespread desktop operation system with the largest number of applications and the biggest market share, we mainly study the cloud disk client in Windows.

There are three popular technologies of the cloud disk client

This work on key technologies of the integration of cloud desktop and cloud storage Platform is supported by ZTE Industry -Academia -Research Cooperation Funds.

in Windows:

1) web browser

For the traditional network storage service, uploading/downloading files through web browsers is the main method of accessing data. Almost all the cloud disks provide web clients for users to access the cloud content through any operating system with web browsers. However, it is inconvenient to upload and download files frequently through web browsers among different user-terminals, such as smart phones, personal digital assistants (PDAs) and PCs.

2) synchronized directory

In the cloud disk service, data synchronization between terminals is a typical function. The cloud disk client in Windows represents a synchronized directory, such as Google Drive [7] and Baidu CloudDisk [8]. Users drag files between the synchronized directory and the local disk is used for data synchronization. The synchronized directory is integrated into the local system as the storage space of the cloud disk. However, this method only stores the metadata of the cloud disk user, therefore, the cloud disk is not available in an environment without network.

3) virtual disk

Virtual disk mapping supports local disk operation habits and offline operations. It is an ideal solution to cloud disk clients. The Windows clients of Dropbox [9] and KingSoft KuaiPan [10] use virtual disk mapping. Currently, this method relies on the virtual hard disk (VHD) technology. VHD [11], [12] is a virtual disk image, of which the contents exactly simulate those of a hard disk by fragmenting the file contents into fixed 512 bytes records. A virtual disk based on the VHD technique replicates all the cloud disk content. After a VHD image gets mounted, it is integrated to the local system with a drive letter. Hence, the client application invokes the ReadDirectoryChangesW API of Windows [13] to capture user's operations on the virtual disk and then synchronizes data to the back-end cloud storage.

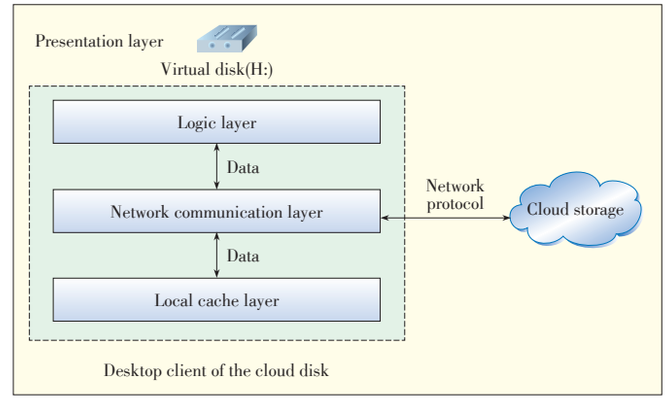
However, the client based on the VHD technology does not apply to the virtual desktop environment in which the computing and storage resources of each user are coupled and limited. When storing popular multimedia data, the virtual disk consumes a lot of storage space on the local hard disk, and this results in storage pressure on the local system with poor performance. Application-level data protection is realized by calling the specific Windows API to capture user's operations. However, this does not satisfy the needs of data security for enterprises.

To solve these problems, we propose a disk mapping scheme for the cloud disk client in Windows. With the customizable cache in the local system, this disk mapping scheme is available for widespread applied environments, especially for the virtual desktop storage. The data of cloud disk can be encrypted at the driver level. In this way, the proposed scheme can prevent the brute-force attacking the cloud content cached in the local disk.

3The Proposed Method

3.1 System Architecture of the Cloud Disk

The proposed system architecture of the cloud disk desktop client consists of the cloud storage platform and various user terminals (**Fig. 1**).



▲**Figure 1. System architecture of the cloud disk client.**

In Fig. 1, the cloud disk desktop client typically has four layers: presentation, logic, network communication, and local cache.

The presentation layer is the user interface of the client side, and most cloud disks present the cloud storage space as a virtual disk or a file directory on the user terminal systems for the intuitive file control methods.

The business rules on the logic layer are applied as data passes from the presentation layer, through the network communication and local cache layers, and back again.

In the communication layer, all the user's operations on the virtual disk can be synchronized to the cloud storage server, and the desktop client receives update notifications from the cloud server when other user terminals change the cloud content.

The local cache layer stores the user metadata and cloud content in the local disk. In order to adapt to different running environments, the cloud disk client has three local cache schemes: full, customizable, and no cache.

3.2 Disk Mapping Method

On the presentation layer of the cloud disk client, the cloud disk is integrated to the operation system as a local disk. In this way, users can get localized experience of reading and writing files on the cloud disk. From a user's perspective, there is no difference between accessing data from the cloud disk and accessing data from a local hard disk. All the user operations are automatically synchronized to the cloud server by the background application of the client. We also propose a

disk mapping scheme of the cloud disk. This scheme is based on the disk drive [14] and file system filter driver [15].

3.2.1 Virtual Disk Drive

The virtual disk driver maps the cloud storage space to the desktop system as a local virtual disk. The virtual disk driver virtualizes an image file into a local disk partition which allows users to read and write. After being mounted, the virtual disk is integrated into My Computer with the settable drive letter, such as the H (Fig. 1). The Windows system provides mature techniques for creating and mounting the virtual disk, involving the IOCreateDisk APIs and dispatch functions in the kernel mode. User operations on the virtual disk are sent to the file system through the I/O request packages (IRPs) controlled by Windows NT I/O Manager (**Fig. 2**).

3.2.2 File System Filter Driver

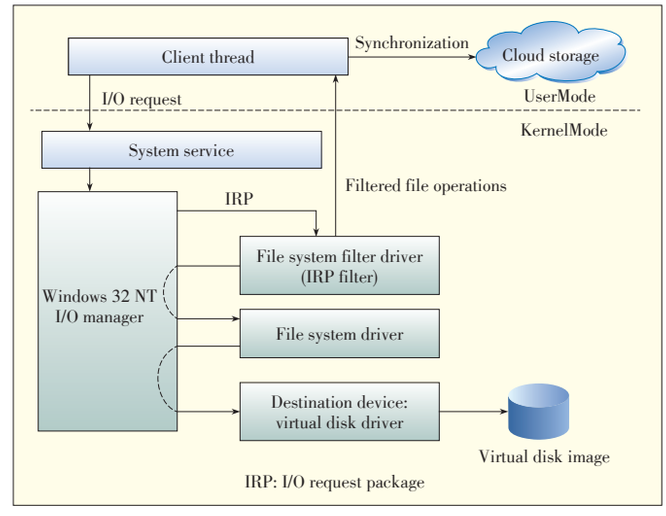
In Windows NT driver mode [16], every I/O request from the client process is controlled by Windows NT I/O Manager when a user operates on the cloud disk. Before sending an IRP, I/O Manager checks whether there is an additional device object driver besides the file system driver. If not, it sends the IRP to the file system driver normally; otherwise, the IRP is sent to the additional device object driver first. The file system filter driver is located on the file system driver or between the file system driver and storage device driver.

We design the filter driver on the file system driver (Fig. 2). The filter driver module intercepts the I/O request before it reaches the file system. The filter driver captures I/O requests from the user's operation on the virtual disk and sends them to the upper network communication layer of client application. The whole process is shown in Fig. 2. After receiving these filtered IRPs, the network communication layer optimizes them by the file operation de-duplication. The optimized operation set are then synchronized to the back-end cloud storage via the network connection.

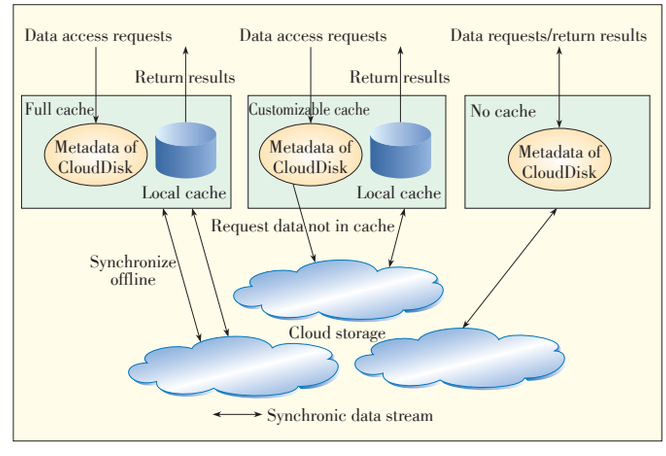
Based on the file system filter driver, the cloud disk client can implement the transparent data encryption [17] in the kernel mode when the filter driver captures the IRPs. This effectively avoids the data leakage on the client.

3.3 Client-Side Data-Caching Schemes

Currently, most cloud disk desktop clients back up all the cloud disk data onto the local hard disk, so users can access the cloud data offline. The background of client automatically synchronizes data to the cloud when the application connects to the internet. Available offline operations provide a good experience for users. However, because some devices having limited capacity of storage and computing, the full-cache storage method has poor user experience due to the large storage load on the local disk. Besides full cache, no-cache and customizable cache schemes are introduced to fit for different running environments. **Fig. 3** shows the detailed processes of these



▲**Figure 2. The disk driver and file system filter driver of the client.**



▲**Figure 3. Three local cache schemes.**

three schemes.

The full-cache scheme is the main solution of most cloud disks at present. With this scheme, a client backups all cloud data on the local disk to support offline operations.

With the no-cache scheme, a client only stores the metadata of the cloud disk, and every data accessing is mapped to the cloud server via the network. Thin-clients of many virtual desktop are based on the no-cache scheme.

The size of a client cache can be set in the customizable-cache scheme, which is applied for different environments. The customizable-cache scheme takes advantage of both full cache and no cache. It supports the offline operation, and also consumes proper amount of local storage resources. Aiming at customizing the initial cache space in different running environments, the client based on customizable-cache scheme allows users to change the capacity of local cache in need when it is running. By using cache-replacement algorithms such as Least Recently Used (LRU) and First In, First Out (FIFO), data cached in the local disk is regularly updated to become hot

spot data.

4 Evaluation

In this section, we verify the function of the customizable-cache scheme. We also present a range of read-write benchmarks to show the performance of the proposed cloud disk client and compare it with the local hard disk.

In all the fowling test-bed experiments, the proposed client is configured in the private computer, which runs on Windows 7, with dual-core 2.7 GHz Intel Pentium processor, 4 GB of memory, and a 500GB SATA 7200 RPM hard drive. CSTORE is used as the back-end server, which is a desktop-oriented distributed cloud storage platform.

4.1 Cache Data Replacement

We evaluate the cache data replacement of the cloud disk client in terms of the times of cache replacement. A directory tree stored on the local hard disk is copied to the cloud disk for the simulation of user's file operations of the cloud disk. As discussed in section 3, the previous user data cache which is least recently used is replaced until the cache capacity drops to the reserve size of the client when the current cache goes beyond the allowable size. In the following test cases, we the initial size of local cache is 100 MB. To simulate file operations of the user, we choose 200 files randomly from the file set (includes three level of directories with 500 10 kB files, 200 100 kB files, 200 1024 kB files and 100 10 MB files) stored on the local disk, copy them to the cloud disk, and estimate the cache replacement times for each test case.

The cache is successfully replaced when the size of current cache is more than 100 MB (**Fig. 4**). In the 6 test cases of Fig. 4, the average times of the cache replacement is approximately equal to 13. The proposed customizable cache management is flexible for upper applications and users.

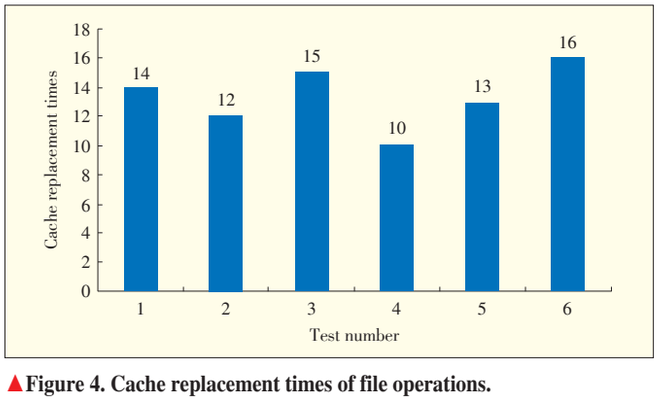
4.2 Read-Write Performance

We use IOzone [18], a disk I/O performance test tool, in Windows. Previous studies have shown that most file access involves random disk access with small request sizes [19]. Therefore, we focus on the random disk read/write and compare the throughput with the local hard disk. In our test cases, the default local cache size of a cloud disk client is 100 MB; the test file size is 50 MB; and the data block size of IO request increases from 32 kB to 8192 kB in doubling size.

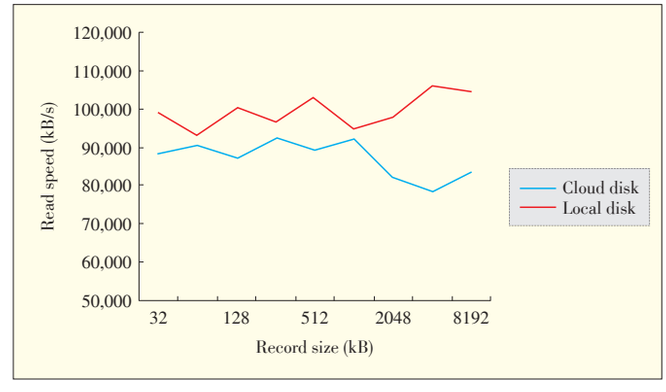
The throughput is shown in **Figs. 5** and **6**. The blue line represents the write-read speed of the cloud disk and the red line represents the write-read speed of the local disk. The speed of accessing data to the cloud disk is slightly slower than the local disk due to the extra processing of the file data. However, from the user's perspective, there is no difference between the two read-write speeds when the user manipulates his/her data from the disk device. The cloud disk performs well in the read-write throughput for users.

5 Conclusion

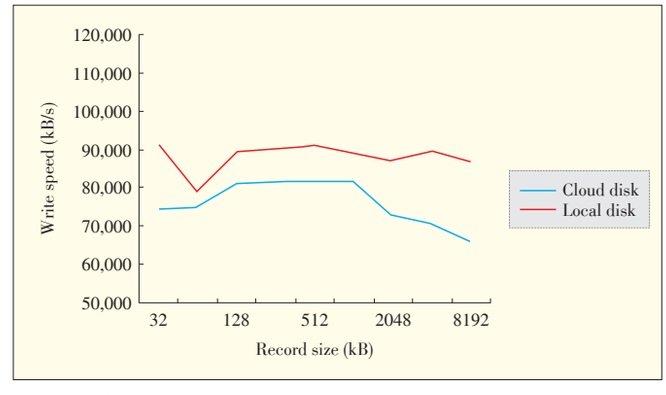
In this paper, we have presented a virtual disk mapping method for cloud desktop storage. Unlike state-of-the art proposals of the cloud disk, we use virtual disk driver and file system filter driver to simulate a virtual desktop disk associate with users' cloud storage space. The virtual disk driver provides a user-friendly mode for accessing desktop data. It also provides a flexible cache space management mechanism. The file system filter driver intelligently checks I/O requests of up-



▲Figure 4. Cache replacement times of file operations.



▲Figure 5. Read throughput.



▲Figure 6. Write throughput.

per applications and synchronizes file accessing requests to users' cloud storage services. The evaluation results show that the read-write performance of our virtual disk mapping method with customizable local cache storage is almost same as that of the local hard disk.

In the future work, the data encryption of the client based on the file system filter driver will be studied and the I/O performance of cloud disk will be continually optimized.

References

[1] J. Wu, L. Ping, X. Ge, Y. Wang, and J. Fu, "Cloud storage as the infrastructure of cloud computing," *2010 Int. Conf. Intelligent Computi. Cognitive Informatics*, Kuala Lumpur, Malaysia, pp. 380–383. doi: 10.1109/ICICCI.2010.119.
[2] W. Zeng, Y. Zhao, K. Ou, and W. Song, "Research on cloud storage architecture and key technologies," in *Proc. 2nd Int. Conf. Interaction Sci Inform. Technol., Culture and Human*, Seoul, Korea, Nov. 2009, pp. 1044–1048. doi: 10.1145/1655925.1656114.
[3] K. Beaty, A. Kochut, and H. Shaikh, "Desktop to cloud transformation planning." in *IEEE Int. Symp. Parallel & Distributed Process.*, Rome, Italy, May 2009, pp. 1–8. doi: 10.1109/IPDPS.2009.5161236.
[4] *Citrix XenDesktop* [Online]. Available: http://www.citrix.com/products/xendesktop/overview.html
[5] VMware. (2012). VMware view 4.5: modernize desktop and application management, v.2.0, brochure. [Online]. Available: http://www.vmware.com/files/pdf/VMware-View-45-DS-EN.pdf
[6] I. Drago, M. Mellia, M. M. Munafo, A. Sperotto, R. Sadre, and A. Pras, "Inside dropbox: understanding personal cloud storage services," in *Proc. 2012 ACM Conf. Internet Measurement*, Boston, USA, pp. 481–494. doi: 10.1145/2398776.2398827.
[7] *Google Drive* [Online]. Available: https://www.google.com/drive/
[8] *Baidu CloudDisk* [Online]. Available: http://yun.baidu.com/
[9] *Dropbox* [Online]. Available: https://www.dropbox.com/
[10] *Kingsoft Kuaipan* [Online]. Available: http://www.kuaipan.cn/
[11] T. Chen, K. H. Lee, Y. Sakane, and T. T. Yamashita, "Hard disk drive system having virtual contact recording," U.S. Patent 5 673 156, Sept. 30, 1997.
[12] K. Kerr. (2011, Feb. 23). The virtual disk API in Windows 7. *MSDN Mag.* [Online]. Available: http://msdn.microsoft.com/en-us/magazine/dd569754.aspx?pr=flas1
[13] A. Schulman, D. Maxey, and M. Pietrek, *Undocumented Windows: A Programmer's Guide to Reserved MIicrosoft Windows API Functions*, Boston, USA: Addison-Wesley, 1992.
[14] P. E. Soran, J. P. Guider, L. E. Aszmann, and M. J. Klemm, "Virtual disk drive system and method," U.S. Patent 7 613 945, Nov 3, 2009.
[15] S. R. McDowell, "File system filter driver apparatus and method," U.S. Patent 6 266 785, Jul. 24, 2001.
[16] R. Nagar, *Windows NT File System Internals: A Developer's Guide*. Sebastopol, USA: O'Reilly Media, 1997.
[17] L. Zheng, Z. Ma, and M. Gu, "Techniques of file system filter driver-based and security-enhanced encryption system," *J. Chinese Comput. Syst.*, vol. 28, no. 7, pp. 1181–1184, Jul. 2007. doi: 10.3969/j.issn.1000-1220.2007.07.006.
[18] *Iozone Filesystem Benchmark* [Online]. Available: http://www.iozone.org/
[19] W. Vogels, "File system usage in Windows NT 4.0," *ACM SIGOPS Operating Syst. Rev.*, vol. 34, no. 2, pp. 17–18, Apr. 2000. doi: 10.1145/346152.346177.

Manuscript received: 2014-06-16

Biographies

Hancong Duan (duanhancong@163.com) received his BS degree in computer science from Southwest Jiaotong University in 1995, ME degree in computer architecture in 2005, and PhD degree in computer system architecture from University of Electronic Science and Technology of China in 2007. He is currently a professor of computer science at the University of Electronic Science and Technology of China. His current research interests include large-scale P2P content delivery network, distributed storage, and operating system.

Xiaoqin Wang (xiaoqinwang0508@gmail.com) received his BS degree in computer science from University of Electronic Science and Technology of China. She is currently a ME candidate at the Department of Computer Software and Theory, University of Electronic Science and Technology of China. Her research interests include distributed storage and cloud computing.

Ping Lu (lu.ping@zte.com.cn) received his ME degree in automatic control theory and applications from South East University. He is the chief executive of the Service Institute of ZTE Corporation. His research interests include augmented reality and multimedia services technologies.

Shengmei Luo (luo.shengmei@zte.com.cn) received his MS degree in telecommunication and electronics from Harbin Institute of Technology in 1996. He is a chief architect at ZTE corporation. His research interests include cloud computing, cloud storage, and big data.

Zhiyong Wang (wang.zhiyong@zte.com.cn) received his master's degree in earth exploration and information technology from China University of Mining. He is currently a product planning manager at ZTE Corporation. His research interests include cloud computing, cloud storage, and big data.

DOI: 10.3969/j. issn. 1673-5188. 2014. 04. 002
http://www.cnki.net/kcms/detail/34.1294.TN.20141127.1518.002.html, published online November 27, 2014

HMIBase: An Hierarchical Indexing System for Storing and Querying Big Data

Shengmei Luo[1], Di Zhao[2], Wei Ge[2], Rong Gu[2], Chunfeng Yuan[2], and Yihua Huang[2]
(1. ZTE Corporation, Nanjing 210012, China;
2. Nanjing University, Nanjing 210046, China)

Abstract

Relational database management systems are usually deployed on single-node machines and have strict limitations in terms of data structure. This means they do not work well with big data, and NoSQL has been proposed as a solution. To make data querying more efficient, indexes and memory cache techniques are used in NoSQL databases. In this paper, we propose a hierarchical indexing mechanism and a prototype distributed data-storage system, called HMIBase, which has hierarchical indexes for non-primary keys in tables and makes data querying more efficient. HMIBase uses HBase as the lower data storage and creates a memory cache for more efficient data transmission. HMIBase supports coprocessor-to-process update requests. It also provides a client with query and update APIs and a server to support RPCs from the client and finish jobs. To improve the cache hit ratio, we propose a memory cache replacement strategy, called Hot Score algorithm, in HMIBase. The experimental results show that Hot Score algorithm is better than other cache-replacement strategies.

Keywords

NoSQL; In-Memory Index; HMIBase; Hot Score

1 Introduction

Relational databases have been around since the 1970s and have been an integral part of data storage and query applications of numerous companies and organizations. In recent years, the speed at which data is created has rapidly increased. A report of IDC in 2011 shows that the amount of global information and will surpass 1.8 ZB, or 1.8×109 GB, which growing by a factor of 9 in just five years [1]. It has become difficult (arguably impossible) to store all existing and newly created data in RDBMSs because an instance of a common RDBMS can only be installed on a single server. Another reason that RDBMSs do not work well with big data is that most data is not structured in a way that meets RDBMS requirements [1]. Not Only SQL (NoSQL) has therefore been proposed for big data storage and querying [2]. NoSQL has no limitations in terms of the type of data that can be input or queried, and it is easy to deploy in a distributed system. This makes it particularly suitable for big data. Compared with RDBMS, NoSQL is also cost efficient. Some applications have storage requirements that cannot be met by relational databases [3].

In this paper, we propose:

- a hierarchical indexing mechanism
- a prototype distributed data-storage system, called HMIBase, which has hierarchical indexes for non-primary keys in tables and enables efficient data querying
- a memory cache replacement algorithm, called Hot Score, in HMIBase.

To implement out system, we use open-source Apache HBase [4], which is designed for big data storage and querying. We use HBase because the data schema in HBase is not as strict as that in RDBMS, and users do not need to tell HBase the length of every property when creating a table. Second, the structure of a table can be modified even if the table is not empty. Third, HBase is based on Hadoop Distributed File System (HDFS), which means it is easy to scale an HBase cluster from a single server to hundreds of similar nodes. This is difficult to do with most RDBMSs.

To reduce query time, a cache is also needed. In fact, data set queries often conform to the 80-20 Rule [5]; that is, they follow a Zip-F Distribution [6] with some specific arguments. Therefore, storing frequently accessed data in a cache can improve performance significantly. Because most SQL and NoSQL systems do not support caching, a cache needs to be

This work is supported by China National Science Foundation (Grant 61223003) and ZTE Industry-Academia-Research Cooperation Funds.

built on top of a database system. We use Redis [7] for the memory cache because it is based on key-value pairs and performs very well with single-key queries. Also, most data in Redis is stored in the memory, and the cost in terms of query time is significantly less than that of HBase.

This paper proceeds as follows. In section 2, we introduce hindex, an index system in HBase. Our proposed system has some similarities to hindex. In section 3, we describe HMIBase, its modules and inner implements. In section 4, we introduce the procedure of HMIBase. In section 5, we describe the architecture of HMIBase. In section 6, provide the results of tests done on the Hot Score algorithm. Section 7 concludes the paper.

2 Related Work

2.1 Hindex

J. Mandava et al. introduced hindex to improve the performance of HBase [8]. Hindex is fully implemented on the server side implementation along with the HBase coprocessor, which preserves index data in a separate table. Indexing is region-wise, and a custom load balancer co-locates the index table regions with actual table regions. HBase with hindex supports multiple indexes on a table and multi-column indexes. An index may be based on part of a column value. And then, equal and range condition scans using index if index has been built. Also, indexing can be done with a bulk load [8].

Hindex has a put operation and scan operation. When a row is put into the HBase's user table, coprocessors generate the index information and then put it into the corresponding index table. The index table's rowkey can be spliced as follows [8]:

$$\textit{rowkey} = \textit{region startkey} + \textit{index name} + \textit{indexed column value} + \textit{user table rowkey} \quad (1)$$

When scanning on a user-table, a scanner will be created by coprocessor on the index table. Then coprocessor use this scanner to scan the index data, and seeks to extract rows in the user table. These seeks on HFiles are based on the rowkey obtained from index data. By doing this, blocks where data is not present can be skipped [8]. However, hindex locates its index on each node of a cluster, so data queries have to be sent to every node even if the result only exists in one node. It may cost too much time to execute the query for data.

2.2 Memory Cache Strategies

An appropriate memory cache strategy is necessary to discard some records, improve the hit ratio, and reduce query time. Common cache strategies include first-in first-out (FIFO) [9], which involves selecting the first record for data replacement; random, which involves randomly selecting a record to replace; and the Least Recently Used (LRU) [10], which involves selecting the least-accessed record in a recent period and replacing it. Other strategies, such as TBF, have also been proposed [11]. We propose a new strategy that involves the use of Hot Score algorithm. Hot Score uses a property, called "hot score," to decide which record to replace.

3 Preliminary

3.1 HBase and Its Coprocessor

HBase is a distributed, fault-tolerant, highly scalable, column-oriented, NoSQL database. HBase is used for real-time read/write random-access to very large databases [5]. It leverages the distributed data storage of HDFS, a distributed file system running on thousands of computers.

As with HDFS, the two main parts of HBase are HMaster and HRegionServers, both of which are managed by Zookeeper [6]. In particular, Zookeeper can manage configuration information, which is often difficult to manage in a distributed system. HMaster [1] is the master server of HBase and monitors all HRegionServer [1] instances in the cluster. HMaster also monitors the interface of all metadata modifications. Another important part of HBase is HRegionServer, which serves and manages regions. In HBase, region is an important concept that describes the basic element of availability and distribution for HBase tables. A HRegionServer often runs on an HDFS DataNode.

HBase is a NoSQL database. However, it is a quasi-relational database because it lacks some of the important features of a relational database, e.g., typed columns and triggers. Also, unlike SQL in a relational database, there is no advanced query language based on HBase. An SQL system may maintain an index for a primary key in a table to improve querying. However, data in HBase is stored in HDFS and may be located in different machines. Therefore, it is difficult for HBase to maintain an index similar to that of an SQL system.

The HBase coprocessor [5] is based on the BigTable coprocessor of Jeff Dean [6]. With a BigTable coprocessor, each tablet in the table server can run arbitrary code and support high-level call interface for clients. A BigTable coprocessor is a very flexible model for building distributed services and enables automatic scaling, load balancing, and request routing. The main modules in a coprocessor are Observer and Endpoint [5].

An instance of Observer contains three tree observers:

1) RegionObserver, which provides hooks for data manipulation events such as Get, Put, Delete, and Scan.
2) WALObserver, which provides hooks for operations related to write-ahead log (WAL).
3) MasterObserver, which provides hooks for DDL-type operations such as Create, Delete, and Modify Table.

Endpoint is an interface for dynamic remote procedure call (RPC) extension. Endpoint is installed on the server side and

can be invoked with HBase RPC. The client library provides convenient methods for invoking dynamic interfaces.

The coprocessor procedure is as follows. The coprocessor initiates RPC invocations of the registered dynamic protocol on every target table region. The results of those invocations are returned as they become available. The client library manages this parallel communication on behalf of the application. The client library manages messy details, such as retries and errors, until all results are returned or there is an unrecoverable error. Then the client library rolls up the responses into a Map and hands it over to the application. If an unrecoverable error occurs, an exception will be thrown to the application code so that it can take action. **Fig. 1** shows a typical process of a coprocessor.

3.2 Redis

Besides HBase, another important NoSQL database is Redis [7]. Redis is a key-value store but is often referred to as a data structure server because keys can contain strings, hashes, lists, sets and sorted sets. By supporting different types of data structures shown above, Redis can process a wide variety of problems that can be naturally mapped into what Redis offers. Redis allows its users to solve their problems without having to perform the conceptual gymnastics required by other databases [13].

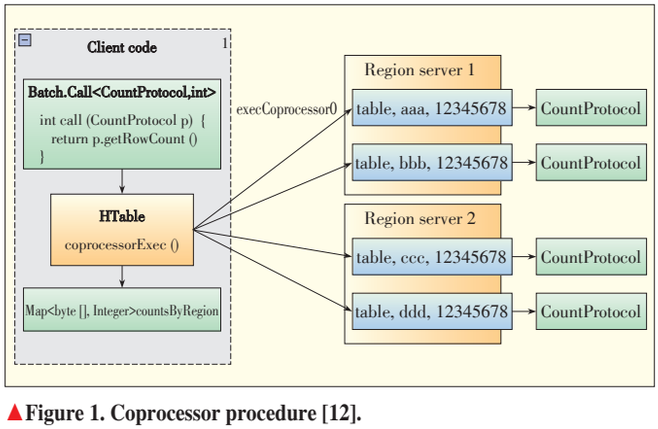
In our system, Redis can be used as a memory cache to reduce the time of queries and improve system performance.

4 System Architecture

We propose a new index mechanism for a system called HMIBase, which is based on HBase with coprocessors. HMIBase has two parts: client and server. The server contains a memory cache, and the client provides query and update APIs similar to HBase.

4.1 HMIBase Index Mechanism

Assume we query a specific table to obtain a record with its value in a specific column. We create an index table that contains only a rowkey but no other columns. The rowkey of indextable contains the table's column information, the value in column and rowkey of origin table. For example, a table called *telephoneBook* has three columns: *id*, which contains people's ids; *name*, which contains names; and *telephone*, which contains telephone numbers. To obtain the name of a person with a specific telephone number, we build the index when inputting the data into the table. The index is constructed as follows: *t, number, id*. Here, *t* describes to the column information and can be replaced with a column name, abbreviation, or any other content. The *number* and *id* is the value in the column for telephone number and rowkey in *telehphoneBook*, respectively.



▲**Figure 1. Coprocessor procedure [12].**

In order to query all values in a specific range, we also propose a value table for each column to be queried. There may be more than two value tables for a single table. Referring to the example previously mentioned in subsection 4.1, the prefix digits of a telephone number indicate the zone of the number; e.g., a telephone number starting with 025 indicates it comes from Nanjing. We can therefore send a range query starting with 025-00000000 and ending with 025-99999999 to search for people who live in Nanjing. The telephone numbers in corresponding value table are sorted in lexicographic order as a rowkey.

With this index mechanism, non-primary keys can be queried by querying two rowkeys. First, the value table is queried; values in a specific range are obtained; and values in the index table are queried in order to obtain the rowkeys. Finally, an ordinary query of the rowkeys will provide the answers. This mechanism is especially suitable to range queries because HBase has to scan all tables to obtain records of a non-primary key. Querying rowkeys results in much better performance much than querying non-primary keys.

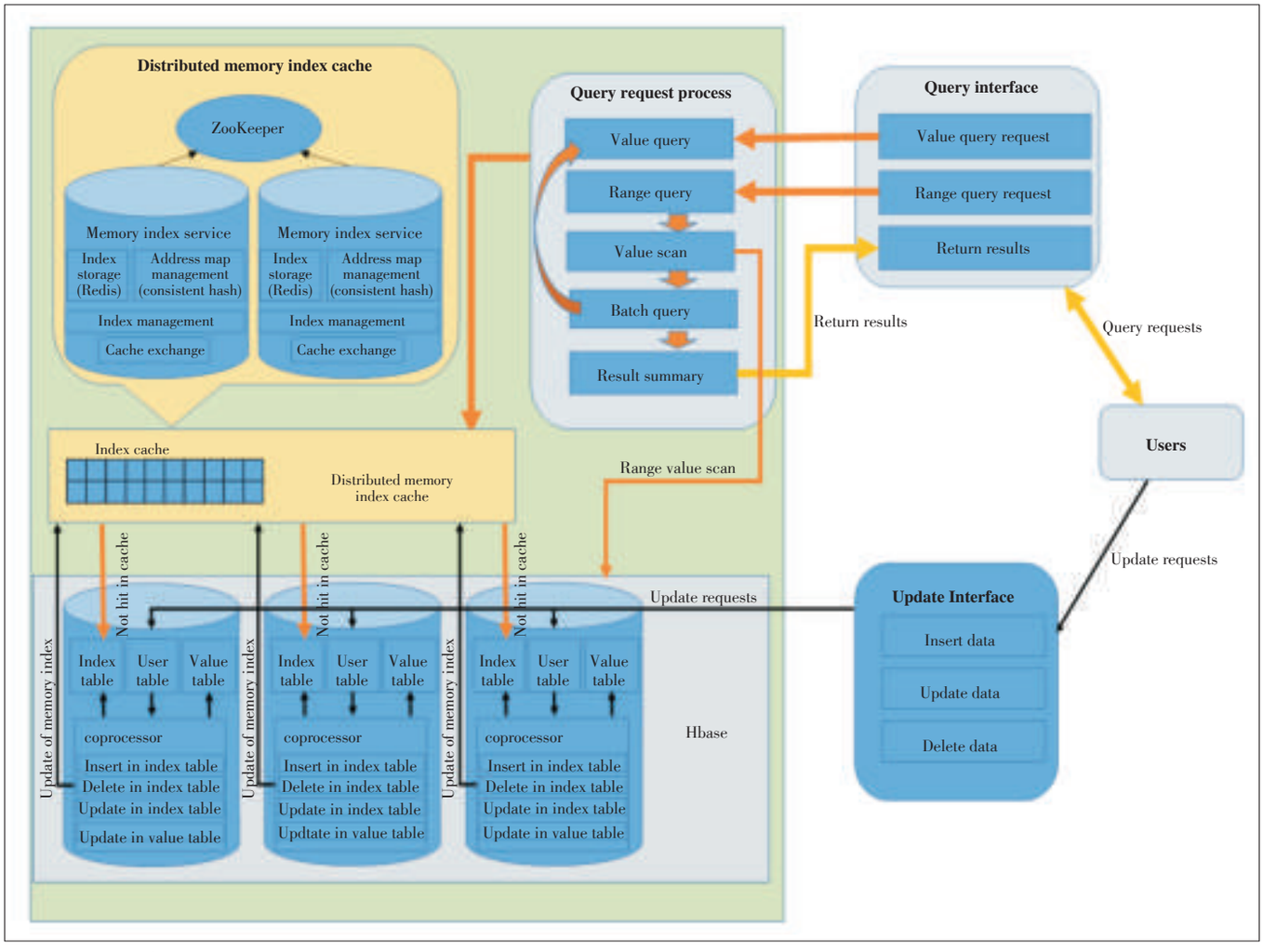
4.2 HMIBase Framework

There are two types of data in HMIBase: hot and cold. The former is data stored in both the memory cache and HBase, and the latter is data stored only in HBase.

Fig. 2 shows the HMIBase framework. On the client side, the main modules in HMIBase are Query Interface, Query Request Process and Update Interface. On the server side, the main modules in HMIBase are Index-Cache in Memory, Data-Cache in Memory, and HBase with coprocessor.

The Query Interface module can receive queries from the user and return results to the user. In HMIBase, there are two types of query request: point and range. A point query request can be used to query information about a specific key whereas a range query can be used to query all information between the lower and upper bounds of the user's input.

The procedure of HMIBase is as follows. APIs are used to first send a query or update request. After receiving the request from a user, the Query Interface module passes it to the Query Request Process module. A point query request is sent directly to the Index-Cache in Memory module. However, a



▲Figure 2. HMIBase architecture.

range query request is sent to HBase to find the values within the range specified by bounds. Then, the range query request is transformed into a number of point query request and processed one by one. Finally, the results of all point query requests are merged and returned to the Query Interface module and then the user.

Index-Cache in Memory is a cache of the hot data index, and Data-Cache in Memory stores the data whose index are Index-Cached in Memory. Like in other cache systems, the selection of data from HBase depends on the cache strategy. We propose an algorithm to improve system performance. Memory access is much faster than HBase access (HBase data is stored in HDFS and HDFS files are stored on the local file system). Therefore, the time spent on a query is reduced. We use Redis for Data-Cache in Memory.

If not in memory, a query request is sent to HBase. An HBase module executes a native query and sends the result back. Whether the data of the queried key is stored in memory depends on the strategy.

The Update Interface module has update APIs, including insert, delete and update. The module sends update requests to HBase. An update request is a trigger to HBase coprocessors. If the data corresponding to an update request is duplicated in the memory cache, the memory cache is temporarily locked and deleted or replaced with new data.

4.3 HMIBase Implementation

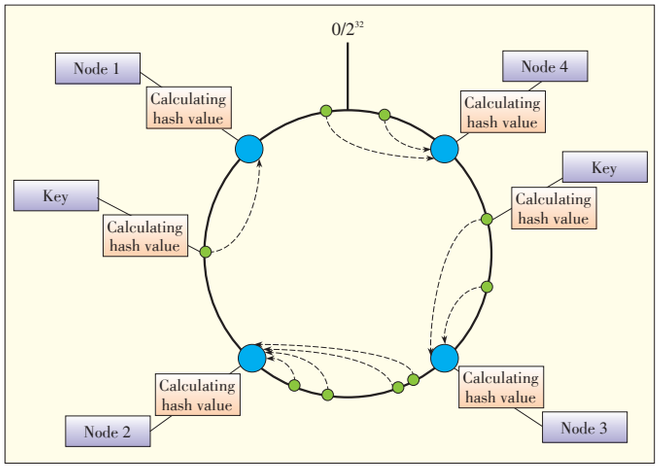
In the client, HMIBase implements a query client object for each session. A query client tries to establish a connection with the memory cache server when initiating and disconnects exiting. Then, a user can call query APIs supported by Query Client Object in order to finish a job.

The query client contains a memory client object that processes the connection with the cache server. The memory client sends a remote process call to the server and obtains a value in return. Both point queries and range queries are processed by same the RPC; the only difference is the argument passed to the query client by the user.

The query client does not support Update APIs because update operations trigger the HBase coprocessor. Current HBase APIs work efficiently in HMIBase, and it is not necessary to implement new APIs.

When a range query is received, HMIBase turns to HBase to search for values within the range. HMIBase uses a first-key filter [1] to search the index table and obtain suitable values within the range in HBase. Then, HMIBase executes a number of value queries.

HMIBase implements a memory cache server at the server side. The reason the memory cache is called a "server" is that it can receive remote process call requests from the client and return results, just like a server. When initiated, the memory cache server uses Apache ZooKeeper to move data to different machine nodes. If a table is created by a user, a global index table corresponding to the table is created simultaneously. Data transfer is much slower than memory caching, and a global index table can reduce the time spent accessing each node because it maps the query key to its location. In HMIBase, ZooKeeper uses a consistent hashing algorithm to map data to different nodes [14] (**Fig. 3**).



▲**Figure 3. Consistent hashing.**

ZooKeeper arranges nodes in a logical ring, and the locations of nodes within this ring are uniformly distributed. After receiving a remote process call from the memory client or HBase coprocessor, the server tries to hash the query request (including table name, column information, and value) to a value of the logical ring. The metadata of this value is stored in its next node, as indicated by the arrows in Fig. 3. If a node within the cluster is disconnected, the value is mapped to the next node in the logical ring.

HMIBase also focuses on robustness. Maintaining data is usually difficult in a distributed database because different nodes in cluster may get different system status when there are multiple operations on the same record. HMIBase adds a synchronized lock on the server's update interface. One request of update can access the data record only after it gets the server's update lock. An update session can only be initiated because an update lock has been obtained from the server. After a session ends, it unlocks itself and sends an event to ZooKeeper, which then updates the system status. Another session in the waiting queue obtains the lock in order to do its job.

5 Hot Score Algorithm

To improve the hit ratio and reduce query time, HMIBase uses Redis as a memory cache. However, we adopt a new strategy—the use of Hot Score algorithm in the cache. Hot Score assigns a hot score to each record in HBase. This score is given by:

$$Hot\ Score = \frac{visitCount}{countPeriod} \times \alpha + (1-\alpha) \times lastHotScore \quad (2)$$

where *countPeriod* is the cycle of the Hot Score algorithm, i.e., the number of query requests between current execution and the next, and *visitCount* is the number of times a record has been queried during the last *countPeriod* Query Requests. The parameter α is the weight of the current statistical result when calculating a new hot score. The *lastHotScore* is the original value of the hot score and has a weight of 1 - α.

The Hot Score algorithm is part of a memory cache server query procedure, so we describe this algorithm in the function query. The function query with Hot Score algorithm is described in pseudo code in **Algorithm 1**.

Algorithm 1: Query with Hot Score Algorithm

```
Input: table t, column col, Value value
Output: query result res
res.key = generate_key(t, col, value);
if Cache.contains(res) = true then
    get res.data from redis;
end
else
    get res.data from HBase;
end
if res.data = null then
    return res;
end
Cur_Access_Set.add(res);
res.increase_visit_count();
increase_query_counter();
if get_query_counter() = count_period then
    reset_query_counter();
    foreach record in Cur_Access_Set do
        record.hot_score = record.hot_score * α +
        record.get_visit_count()/count_period * (1−α);
    sort Cur_Access_Set by hot_score order;
    foreach i in [0, MAX_CACHE_SIZE −1] do
```

```
        record = Cur_Access_Set.get(i);
        if Cache.contains(record) = false then
            Cache.add(res);
            get record.data from HBase;
            put record.data into Redis;
        end
    end
    foreach i in [MAX_CACHE_SIZE, Cur_Access_Set.size()-1] do
        record = Cur_Access_Set.get(i);
        if Cache.contains(record) = true then
            Cache.remove(res);
            delete record.data from Redis;
        end
    end
    Cur_Access_Set.clean();
end
return res;
```

In Algorithm 1, *Cache* is the current memory cache, which only needs to be updated when Hot Score algorithm is executed. *Cur_Access_Set* is a set of metadata of all the queries over a period and needs to be cleaned after the Hot Score algorithm has ended. For each query, HMIBase generates a key using the table name, columns, and values input by the user. HMIBase then tries to access the data from *Cache* using the generated key. If the data cannot be accessed, it is obtained from HBase. There is no data exchange between the cache memory and HBase unless the Hot Score algorithm is executed. When the number of queries reaches *countPeriod*, Hot Score Algorithm is executed. First, hot scores are calculated for all records accessed in *Cur_Access_Set*, regardless of whether they are in memory or not. This calculation is made using (1). Then, records are sorted according to their hot scores and the Top-K records are chosen. The chosen records are loaded into memory and others, even if they remained in memory during the last period, are discarded. After Hot Score is executed, *Cur_Access_Set* is cleaned in preparation for the next period.

6 Experimental Evaluation

We conducted experiments (**Table 1**) to determine how our methods affect system performance. Brown University's MapReduce database benchmark was used [15] because it supports a Python script. This enables data records to be generated as much one wants. Ten million records were generated and stored in HBase (data was put into a test table). After this, the query time for native HBase, Hindex and our HMIBase was compared. We compared the query time of different cache strategies to see the improvement brought about by Hot Score algorithm.

▼**Table 1. Experiment environment**

Environment	Information (or Version)
Server	x86-64, 2.8GHz × 8 cores/32G memory
Number of nodes	2
OS kernel	Linux kernel 2.6.32
JVM	Java-1.6.38 for Linux
Hbase	Hbase-0.94.14
Redis	Redis-2.6.8

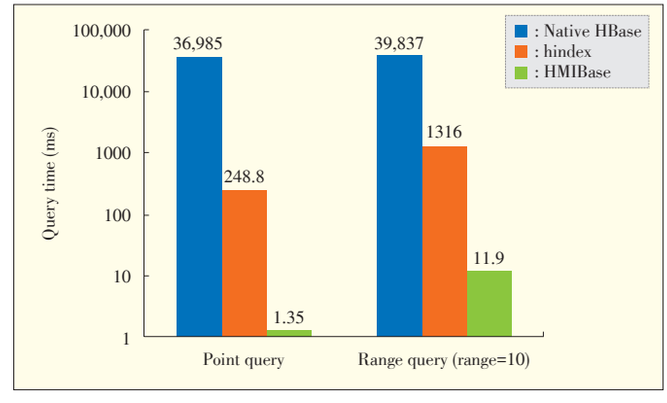
We also generated some query requests, including point query and range query request on this benchmark. The values of point and range query are generated in ZipF distribution [11] and the ranges here are set to be constant values (here, 10 and 100).

We ran some random queries, including point queries and range queries (range = 10) on HBase without any modification (native HBase), hindex, and HMIBase. **Fig. 4** shows results of these queries on the different systems.

In Fig. 4, the time cost for point and range queries on HMIBase is much shorter than that for point and range queries on HBase and hindex. In HMIBase, time cost for a range query is nearly 10 times that for a point query. HBase uses a filter to query and scans all the records in the table; therefore, the time cost for point and range queries are similar. Hindex creates an index on each region of HBase; therefore, the time spent on each node becomes shorter, but there is no obvious improvement in accessing nodes from the master. With HMIBase, the speed of access to nodes from the master has an improvement with a factor of five to eight. HMIBase locates the index directly and stores hot data in the memory, so it performs much better than native HBase and hindex. Because HMIBase executes a range query by executing multiple point queries, the time cost of a range query depends on the size of result set.

In another experiment, we applied different cache strategies to HMIBase and add strategy as a property in the configuration so that it was easy to switch between strategies. We executed some point queries and range queries using different strategies. **Fig. 5** shows the time cost of these queries using different cache strategies.

Hot Score algorithm results in faster point and range querying than other caching strategies. As the size of data increases,



▲**Figure 4. Experiment results.**

◀Figure 5.
Time cost of point and range queries using different cache strategies.

the difference in time cost between Hot Score and other strategies increases. The main reason for this difference is that Hot Score algorithm has a higher hit ratio than other caching strategies. The hit ratios for each query executed in the experiment are shown in Figs 5(b), (d) and (f). We use a coefficient of 0.2 to allow the size of the cache to increase in line with the size of queried data. In this way, there is no visible difference in hit ratio when the size of queried data increases.

The hit ratio for LRU is higher than that for random data selection because LRU can calculate when the data was last used and store hot data for faster retrieval. However, LRU may discard some hot data because it is only concerned with the last use of the data. Hot Score, on the other hand, calculates the use of data over a whole period and thus has a higher hit ratio than LRU.

Another reason that Hot Score algorithm results in fast que-

rying is that it reduces data transmission. In Redis, read and write operations need to be synchronized in order to maintain data consistency, and this increases time cost. With random data selection or LRU, if not in memory, the memory cache needs to decide which key to exchange and do a read/write operation between Redis and HBase files. This means the synchronization operations may increase the time of the query. Hot Score algorithm only exchanges data when the number of queries is large enough, e.g., 10,000. Even though there may be several updates between HBase and cache, Redis can process these updates in a pipeline because there are no read operations between the updates, and only one synchronization operation is required in the end. Therefore, Hot Score also reduces transmission time.

7 Conclusion

NoSQL has become more and more popular for data storage in recent years. To make data querying more efficient, indexes and memory cache techniques have been used in NoSQL databases. In this paper, we have described HMIBase, a NoSQL system built on Hbase. HMIBase has indexing and has transparent features and modules. HMIBase uses HBase for data storage and creates a memory cache for more efficient data transmission. HMIBase offers a client query and update APIs as well as a server to support RPCs from the client and finish jobs.

HMIBase supports a variety of cache strategies for data exchange. These strategies include random, LRU, and Hot Score, a new algorithm introduced in this paper. After the introducing Hot Score, we described how HMIBase is based on an HBase cluster. We then performed experiments on HMIBased using the data cache strategies just mentioned. These experiments revealed that Hot Score algorithm is a better cache strategy than LRU or random.

Future research may be done on NoSQL systems (other than HBase and Redis) that have not been well researched and the addition of an exception process procedure for HMIBase in order to improve the robustness of HMIBase. This will make HMIBase more suitable for different scenarios.

References

[1] L. George, *HBase: The Definitive Guide*, 1st ed. Sebastopol: O'Reilly Media, 2011, pp. 319–320.
[2] K. Grolinger, W. A. Higashino, A. Tiwari, and M. AM Capretz, "Data management in cloud environments: NoSQL and NewSQL data stores," *Journal of Cloud Computing*, vol. 2, no. 22, 2013. doi: 10.1186/2192-113X-2-22.
[3] R. Hecht and S. Jablonski, "NoSQL evaluation: a use case oriented survey," in *2011 Int. Conf. CSC*, Hong Kong, China, pp. 336–341. doi: 10.1109/CSC.2011.6138544.
[4] Apache. (2014). *Apache Hbase* [Online]. Available: http://hbase.apache.org
[5] W. A. Britten, "A use statistic for collection management: the 80/20 rule revisited," *Library Acquisitions: Practice & Theory*, vol. 14, no. 2, pp. 183–189, 1990. doi: 10.1016/0364-6408(90)90061-X.
[6] B. C. Brookes, "The derivation and application of the Bradford-Zipf distribution," *Journal of Documentation*, vol. 24, no. 4, pp. 247–265, 1968. doi: 10.1108/eb026457.
[7] S. Sanfilippo and P. Noordhuis. (2010). *Redis* [Online]. Available: http://redis.io/
[8] J. Mandava, R. Chintaguntla and P. Rastogi. (2013). *Hindex* [Online]. Available: https://github.com
[9] R. L. Mattson, J. Gecsei, D. R. Slutz, *et al.*, "Evaluation techniques for storage hierarchies," *IBM Systems journal*, vol. 9, no. 2, pp. 78–117, 1970. doi: 10.1147/sj.92.0078.
[10] E. G. Coffman and L. Varian, "Further experimental data on the behavior of programs in a paging environment," *Communications of the ACM*, vol. 11, no. 7, pp. 471–474, 1968. doi: 10.1145/363397.363398.
[11] C. Ungureanu, B. Debnath, S. Rago, *et al.*, "TBF: A memory-efficient replacement policy for flash-based caches," in *2013 IEEE 29th Int. Conf. Data Engineering*, Brisbane, Australia, pp. 1117–1128. doi: 10.1109/ICDE.2013.6544902.
[12] M. Lai, E. Koontz, and A. Purtell. (2012). *Coprocessor Introduction* [Online]. Available: https://blogs.apache.org/hbase/entry/coprocessor_introduction
[13] J. L. Carlson, *Redis in Action*. Shelter Island: Manning Publications Co., 2013.
[14] D. Karger, E. Lehman, T. Leighton, *et al.*, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web," in *Proc. 29th Annual ACM Symp. Theory of Computing*, 1997, pp. 654–663. doi: 10.1145/258533.258660.
[15] A. Pavlo, E. Paulson, A. Rasin, *et al.*, "A comparison of approaches to large-scale data analysis," in *Proc. 2009 ACM SIGMOD Int. Conf. Management of Data*, Providence, USA, pp. 165–178. doi: 10.1145/1559845.1559865.

Manuscript received: 2014-05-23

Biographies

Shengmei Luo (luo.shengmei@zte.com.cn) received his MS degree in telecommunication and electronics from Harbin Institute of Technology in 1996. He is a chief architect at ZTE Corporation. His research interests include cloud computing, cloud storage, and big data.

Di Zhao (zd08135@126.com) received his BS degree in computer science and technology from Nanjing University in 2012. He is currently a master's degree candidate at the Department of Computer Science and Technology, Nanjing University. His research interests include parallel computing and analyzing and processing of big data.

Wei Ge (gloria.w.ge@qq.com) received her MS degree from Northeastern University in 2003. She is currently a PhD candidate in computer science at Nanjing University. Her research interests include data management, database query optimization, and big data query optimization. She has published 10 papers in journals and conference proceedings, including in *Science in China* (series F), APWeb 2003, and *Journal of Electronics*.

Rong Gu (gurongwalker@gmail.com) received his BS degree in computer science from Nanjing University of Aeronautics and Astronautics in 2011. He is currently a PhD candidate in computer science at Nanjing University. His research interests include parallel and distributed computing, cloud computing, and big-data parallel processing.

Chunfeng Yuan (cfyuan@nju.edu.cn) is a professor at the Department of Computer Science, Nanjing University. She received her BS and MS degrees in computer science from Nanjing University. Her main research interests include compute system architecture, big data parallel processing, and Web information mining.

Yihua Huang (yhuang@nju.edu.cn) is a professor at the Department of Computer Science, Nanjing University. He received his BS, MS, and PhD degrees in computer science from Nanjing University. His research interests include parallel and distributed computing, big-data parallel processing, and Web information mining.

DOI: 10.3969/j. issn. 1673-5188. 2014. 04. 003
http://www.cnki.net/kcms/detail/34.1294.TN.20141127.1518.002.html, published online November 27, 2014

MBGM: A Graph-Mining Tool Based on MapReduce and BSP

Zhenjiang Dong[1], Lixia Liu[1], Bin Wu[2], and Yang Liu[2]
(1. ZTE Corporation, Nanjing 210012, China;
2. Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract

This paper proposes an analytical mining tool for big graph data based on MapReduce and bulk synchronous parallel (BSP) computing model. The tool is named Mapreduce and BSP based Graph-mining tool (MBGM). The core of this mining system are four sets of parallel graph-mining algorithms programmed in the BSP parallel model and one set of data extraction-transformation-loading (ETL) algorithms implemented in MapReduce. To invoke these algorithm sets, we designed a workflow engine which optimized for cloud computing. Finally, a well-designed data management function enables users to view, delete and input data in the Hadoop distributed file system (HDFS). Experiments on artificial data show that the components of graph-mining algorithm in MBGM are efficient.

Keywords

cloud computing; parallel algorithms; graph data analysis; data mining; social network analysis

1 Introduction

Rapid development in fields such as the Internet of Things, cloud computing, social networking, mobile communication and social media, has ushered in the era of big data. Now, most big data exists as a graph or can be expressed using graph structure, which is one of the most widely used abstract data structures in computer science. Graph structure enables more complex, comprehensive presentation of data than link tables or tree structures. Graph-mining theories and techniques are improving all the time; however, the amount of information is growing exponentially, and the scale of graph-based data is increasing significantly. The Internet Data Center in the United States has pointed out the amount of data on the Internet is expanding by 50% each year, and more than 90% of data worldwide was generated in the past two years [1]. Millions of smart mobile devices, both enterprise and personal, are produce data in everywhere and at any time. YouTube users upload about 70,000 hours of video every day. According to statistics from the China Internet Network Information Center (CNNIC), at the end of December 2013, the number of webpages in China was 150 billion, a 22.2% increase over last year. At the same time, there were approximately 308 million microblog users, approximately 54.7% of all Internet users [2]. The amount of big graph data being generated is huge, and it will be challenging to efficiently analyze all this data. We therefore propose a system combining MapReduce and a BSP-based graph-mining tool (MBGM). This system has a series of parallel graph-mining algorithms based on the bulk synchronous parallel (BSP) method. It also has a set of data extract-transformation-loading algorithms based on MapReduce. Distributed file systems are also used to store and manage the graph data, and a workflow engine is used to invoke the algorithms.

The remainder of this paper is structured as follows. In section 2, we review related works. In section 3, we describe the MBGM system architecture. In section 4, we discuss the implementation of several typical graph-analysis algorithms and discuss an application. In section 5, we discuss the testing and performance of the proposed system. In section 6, we discuss future directions.

2 Related Work

MBGM is closely related to parallel-computing platforms and graph-mining tools.

Message-passing interface (MPI) is a model for passing on messages. Many companies and universities have implemented jobs that can be run on almost any type of parallel computer.

This work is supported by ZTE Industry-Academia-Research Cooperaton Funds.

Such computers support all existing graph algorithms [3]. However, because the MPI model uses a communication method to integrate computing resources, this model has several drawbacks. For example, the inefficiency of parallel computing and high consumption of memory makes it difficult to properly manage the resources and communication.

MapReduce was created by Google [4]. The best known and most successful open-source implementation of MapReduce is Hadoop. An application based on the MapReduce framework can run on large-scale clusters in a parallel, fault-tolerant way. A MapReduce program has two main steps: map and reduce. Each phase comprises an input, computation, and output step: the output of each phase is the input for the next phase. When a phase is finished, every machine writes its output data to shared memory, and the data is synchronized. Therefore, each machine is allowed to read the data written in the previous phase. The MapReduce framework relies on the <key, value> pair to transfer data between the steps. A user can implement their own map and reduce function to finish the computing task. The input and output files are stored in a distributed file system. MapReduce is suitable for processing large-scale data and executing algorithms that do not need much iteration. However, it performs badly with algorithms that are highly iterative. This means that it is not suitable for most graph algorithms.

Bulk synchronous parallel (BSP) [5] is also a widely used parallel computing framework. It overcomes some of the weaknesses of MapReduce and performs well when a program has much iteration or requires a lot of communication. A BSP program can be divided into several super-steps, each of which consists of three stages: local computation, communication, and barrier synchronization. A BSP system has a number of computers with local memory and disks. Each computer can run several computing processes called peers. In the local computation stage, each peer is computed using locally stored data. After finishing local computation, each peer communicates only necessary data to other peers. When a peer finishes the communication stage, it waits until all the peers reach the barrier synchronization, and a super-step is completed.

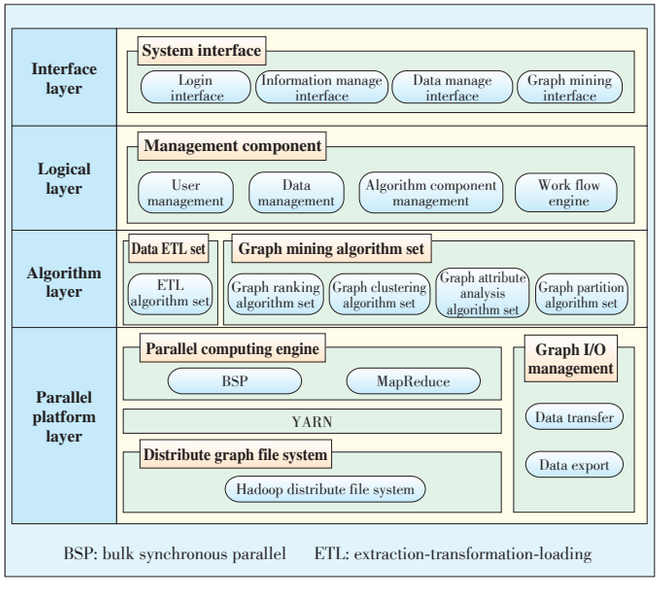
Spark [6] is newly proposed parallel computing framework. The basis of Spark is the resilient distributed dataset (RDD), which is a read-only collection of objects partitioned across a set of machines that can be rebuilt if a partition is lost. The elements of an RDD do not need to exist in physical storage. A handle to an RDD contains enough information to compute the RDD starting from data in reliable storage. Another feature of Spark is the abstract of parallel operation. Instead of map-reduce in the MapReduce model and super-step in the BSP model, the computing elements in Spark are reduce, collect and foreach. Spark performs well with iterative jobs and interactive analytics.

Apache Mahout [7] supports supply classification, clustering, pattern mining, regression, and dimension reduction, and machine-learning algorithms. However, it lacks a graph-mining function. GraphLab [8] improves on MapReduce abstraction by compactly expressing asynchronous iterative algorithms with sparse computational dependencies. However, the asynchronous mechanism may cause non-convergence problem while implement synchronous iterative graph algorithm. PEGASUS [9] is a large open-source graph-mining system implemented on Hadoop. The key idea of PEGASUS is to convert graph-mining operations into iterative matrix-vector multiplications. PEGASUS supports large-scale graph data; however, in practice, not all the graph-mining algorithms can be modeled by matrix-vector multiplications. Dryad [10] is a general parallel computing platform proposed by Microsoft Research. This platform abstracts the computing and communication in data-mining operations into vertexes and edges in order to form a dataflow graph. The platform executes the vertexes on work nodes and refines the dataflow graph to optimize the running process. Big Cloud parallel data mining (BC-PDM) [11], developed by China Mobile Research Institute (CMRI), provides visualization for data mining and analysis of graph data. However, it is based on Hadoop, so the graph-mining algorithms do not perform well. Pregel [12] was premised on BSP and implemented by Google. It is a complete solution for large-scale graph computing but has not been published in the public domain. BC-BSP [13] is another implementation of the BSP parallel platform. Although most BSP platforms use memory to exchange the temporary data, BC-BSP a data-spill mechanism (including static data and dynamic data) on the local disk. This improves data processing for a small cluster, but the management and updating of this data-spill mechanism requires extra communication and system resources and creates new defects in the platform. GraphX [14], a resilient distribute graph system based on Spark, was proposed by UC Berkeley. The system has a Spark computing engine and uses the vertex-cut method instead of the edge-cut method for data partitioning.

3 System Architecture

This system focuses on big-graph management and graph-mining. We noticed that, while the original data is huge, but most graph-mining application using part of the original data. For example, the user phone call data provided by mobile communication companies is GB sized, and contains much information about the detail of caller and answerer, but most graph analysis does not calculate all the information. We use MapReduce to extract the graph data from original data and construct the graph. To manage graph data and original data, we designed a data I/O management component in the parallel platform layer and a data-management component in the logical layer. The algorithm layer is divided into two parts: extraction-transformation-loading (ETL) algorithm set and graph-mining algorithm set. We built the graph-mining algorithm set to implement graph ranking algorithms, graph clustering algorithms, and graph attribute analysis algorithms and graph partition al-

gorithms as the foundation of the graph analysis. Finally, we made this system extendable to enable the addition of other algorithm components. An overview of the architecture of MBGM is shown in **Fig. 1**. The system consists of four layers, the function of which are described here.



▲**Figure 1. Architecture of MBGM.**

3.1 Parallel Platform Layer

The parallel-platform Layer comprises distributed graph file system, YARN, parallel computing engine, and graph I/O Management component. We used Hadoop Distributed File System (HDFS) to construct the Distributed Graph File System, enabling the storage of big graph data. YARN, a framework for cluster resource management and job scheduling, comprises an application master (AM), which should integrate multiple computing frameworks, e.g., MR, BSP. Because the BSP model performs well with graph-mining algorithms, we chose Hama BSP [15] as the parallel computing engine and used it to handle message communication, data distribution, and fault tolerance. We used MapReduce as the graph data pre-processing engine to extract graph information from original data. The graph I/O management component transfers data from the database into the graph data form that the MBGM can handle and then exports the MBGM data.

3.2 Algorithm Layer

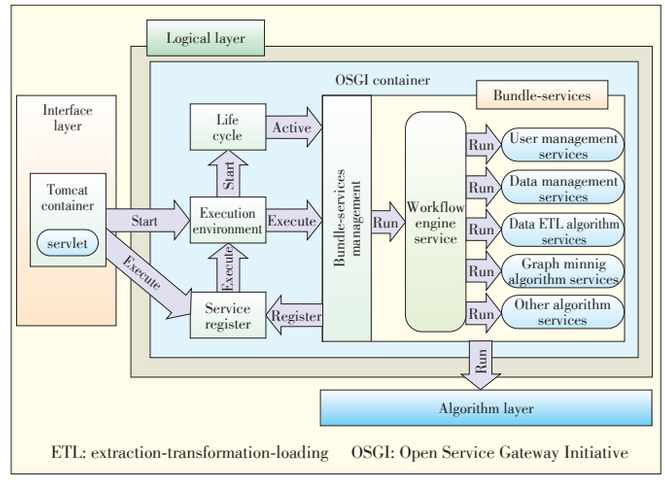
The algorithm layer is the main layer of MBGM. This layer can be roughly divided into two parts: ETL algorithm and graph-mining algorithm. In the graph-mining algorithm part, we implemented four sets of 20 graph-mining algorithm components in the BSP parallel model and four group of data ETL algorithm for transform original into graph data. The ETL algorithm set comprises data-cleaning set for detecting and removing error value, data transform set for transform value into the format user needed, data extract set and data update set. Those graph-mining algorithm components can be divided into four sets: graph ranking set comprises PageRank [16], Hyperlink-Induced Topic Search (HITS) [17], and Random Walk with Restart (RWR) [18] algorithms components, the graph clustering set comprises Gauss–Newton

(GN) [19], Clauset, Newman, and Moore (CNM) [20], Clique Percolation method (CPM) [21], and Label Propagation Algorithm (LPA) [22] algorithm components, and the K-means algorithm is used for the processing of general data. The graph attribute analysis set contains the graph diameter, closeness centrality, clustering coefficient, network density, betweenness centrality, and five other algorithm components. The graph partition set contains components that are based on the Multilevel Subgraph Patrolling (MSP) [23] algorithm component and the Metis [24] algorithm component. These algorithm components can be run either in the console or can be invoked in a user-defined workflow from the user interface.

3.3 Logical Layer

The logical layer is based on Open Service Gateway Initiative (OSGI) and is used to implement a stable, efficient, scalable system[25]. This is service platform and manager all kinds of services that are supported by this layer, such as User Management Service, Data (HDFS) Management Service , Algorithm Service (Each algorithm is a kind of service), and the workflow engine Service.

Fig. 2 shows the operation of the logical layer. Getting the start command from tomcat servlet which is a container in the interface layer, OSGi container will execute a train of operations, starting the lifecycle, activating and registering the each service that hosted in bundle-services management. With the tomcat servlet invoking one of services, Service Register queries and gets service that is requested by the tomcat from the services pool, and then execution environment (EE) calls the



▲**Figure 2. Operation flow of the logical layer.**

workflow engine service to perform the requested service which is ultimately implemented in the algorithm layer.

3.4 The Interface Layer

The interface layer is built in HTML, and flex provides an interactive interface with which the user can login to the MBGM, management information, and most importantly, use the graph algorithms' mining graph data.

4 Implementation of Parallel Graph Analysis Algorithm

Parallel graph-mining algorithms based on the BSP model are the most important aspects of the BPGM. In the BSP parallel model, graph-mining algorithms are "think as vertex." Programmers use the perspective of the vertex to manage the gathering of information from other vertexes and to send information to other vertexes.

4.1 Implementation of PageRank in BSP

PageRank is one of the most important and famous rank algorithm in graph analysis algorithm. The basic idea of PageRank is the importance of a vertex is depending on the quantity and quality of its neighbor vertex. The formal of PageRank computation is:

$$PR(p_i)=\frac{1-d}{N}+d\sum_{p_j\in M(p_i)}\frac{PR(p_j)}{L(p_j)} \quad (1)$$

The N is vertex quantity of the graph, $M(p_j)$ is the indegree set of vertex p_j, and $L(p_j)$ is the outdegree of vertex p_j, d is an empirical value, in this implementation we set $d=0.85$. The pseudo-code of PageRank in BSP method described in **Algorithm 1**.

Algorithm 1. Implementation of PageRank in BSP

```
Input: G(V, E) G is the graph data, V is the
       vertex set, E is the edge set;
Output: PageRank value of each vertex
PageRank:
    1. for each p∈V {
    2. p.pr = 1/N;
    3. for superstep from 1 to k {
    4.   for each p∈V
    5.   sum = 0.0;
    6.     for each q∈M(p) {
    7.     sum += q.pr/L(q);
    8.     p.pr = (1-d)/N + d*sum;
    9.     }
    10.  }
    11. return p.pr;
    12. }
```

In the Algorithm 1, *p.pr* is the PageRank value of vertex *p*, and *k* is the iteration limited. While compute PageRank value in a small graph data, it is necessary to set a threshold of PageRank value change between super-steps, if a vertex reached the threshold it vote to halt and when all vertex vote to halt, the procession is finished. But in the big graph data, it is hard to determine the threshold, so we simply limited the maximal superstep number, usually set k = 30. The application scenario of PageRank is showed in section 5, and performance over various data sets is listed in **Table 1**.

4.2 Implementation of LPA in BSP

Community-detection and analysis is an important methodology for understanding the organization of various real-world networks and has applications in problems as diverse as consensus formation in social communities or identification of functional modules in biochemical networks. Most community-detection algorithms in large-scale real-world networks require a priori information, such as the number and sizes of communities, or are computationally expensive.

▼**Table 1. Runtime of some graph-mining algorithm components (s)**

Graph data set	Eigenvector centrality measure	InDegree count	MSP	PageRank	Closeness centrality	Personal centrality	Clustering coefficient	RWR
data_set_1	16.2	13.2	166.1	25.2	31.2	13.1	13.1	22.2
data_set_2	25.1	16.2	310.1	40.2	70.5	16.1	19.1	28.1
data_set_3	28.2	22.1	343.0	61.5	31.4	19.1	19.2	37.2
data_set_4	88.2	64.2	696.1	202.4	25.4	22.0	31.1	169.2
data_set_5	173.6	73.2	995.2	439.6	32.1	31.2	55.2	313.4
data_set_6	643.7	199.3	1278.3	1241.6	55.7	52.2	151.2	688.7
GoogleWeb	199.2	79.2	721.3	304.6	31.7	34.2	52.2	331.5

MSP: Multilevel Subgraph Patrolling algorithm RWR: Random Walk with Restart algorithm

The label propagation algorithm (LPA) is a community-detection algorithm based on label propagation. Suppose a vertex v has neighbors $v_1,v_2,v_3,...,v_k$ and that each neighbor vertex carries a label denoting the community to which they belong. Then v determines its community based on the label of its neighbors, assuming that each vertex in the network chooses to join the community to which the maximum number of its neighbors belong. The pseudo-code of LPA in BSP method is described in **Algorithm 2**.

Algorithm 2. Implementation of LPA in BSP

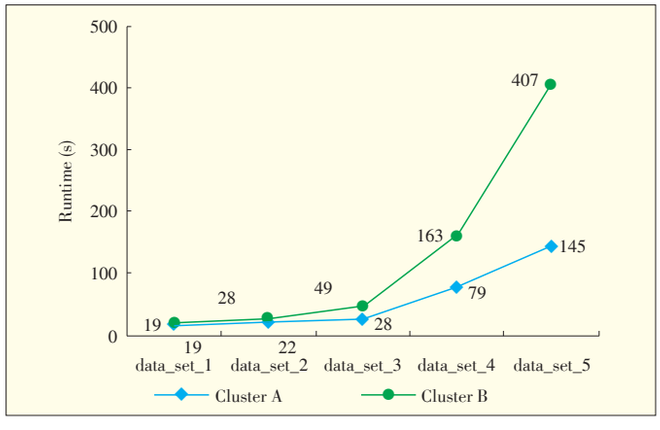
```
Input: G(V, E) G is the graph data, V is the vertex set,
       E is the edge set;
Output: Vertex set with community label
LPA:
    1.for each p∈V {
```

```
2.v.setLable(RandomLabel);
3.}
4.while(Iteration < Max_Iteration)
5.{
6.    For ( v⊆V ){
7.        v.sendLabel(adjVertex );
8.    }
9.    Peer.sync();
10.    For ( v⊆V ){
11.        v.getLabel();
12.        v.setLable(MaxNumLabel);
13.    }
14.    Peer.Sync();
15.    Iteration++;
16.}
17.Peers. sync();
```

In Algorithm 2, line 2 initial ever vertex in graph with random label which indicate the community it belong to. Line 7, sendLabel operator, is based on the operation of "sendMessageToNeighbors" provided by Hama and sends a message of vertex label to its neighbors, and in line 11, each vertex received messages from neighbors, and update its label as the most community label it received in line 12. If a vertex received an equal number of label types, it should randomly pick one as its new label. The Max_Iteration is the threshold of the superstep number, and we set it as 60. The performance of the LPA on various scale of graph data and different scale of cluster is shown in **Fig. 3**.

4.3 Implementation of CPM in BSP

A real-life graph tends to have a very complicated structure, and a vertex always not only belongs to one community, most traditional community detection algorithm cannot catch this feature. In response to this challenge, researchers have proposed the concept of overlapping community (CPM). The clique percolation method is one of the most important algorithms in this field.



▲**Figure 3. LPA performance of MBGM on Cluster A and Cluster B.**

The CPM defines a *k-clique-community* as a union of all *k-cliques* (complete subgraphs of size k) that can be reached from each other through a series of adjacent k-cliques (where adjacency means sharing $k-1$ vertex). This definition is aimed at representing the fact that it is an essential feature of a community that its members can be reached through well-connected subsets of nodes. There are other parts of the whole network that are not reachable from a particular *k-clique*, but they potentially contain further *k-clique-communities*. In turn, a single node can belong to several communities. All these can be explored systematically and can result in a large number of overlapping communities.

The CPM mainly has two steps, locate all *cliques* and find community structure. The pseudo-code of CPM in BSP method described in **Algorithm 3**.

Algorithm 3. Implementation of CPM in BSP

```
Input: G(V,E) G is the graph data, V is the vertex set,
       E is the edge set; k is the target clique size
Output: Vertex set with community label
LPA:
locate k-clique
    1. for each p∈V {
    2. A=Ø;
    3. B=getAdjacentVertex(p);
    4. for each v∈B {
    5.    A=A∪v;
    6.    B.removeVertex ( v );
    7.    B.removeVertexConnect(A);
    8.    if(|A|<k and B=Ø){
    9.        goto step 1;
    10.       }
    11.   else if (|A|=k){
    12.       return A∪p;
    13.       A=Ø;
    14.       goto step 4;
    15.       }
    16.   }
    17. }
find community structure
    18. List L= Ø ;color = 1;
    19. for each c∈C {
    20. c.setColor(0);
    21. }
    22. for each c∈C {
    23. if (c.getColor()=0){
    24.     c.setColor(color++);
    25.     L.add(c);
    26.   }
    27.   while(L.isNotEmpty()){
    28.     for( l∈L ){
    29.       for each c∈C and c.getColor=0;
    30.       if isOverLap(l,c)
    31.       c.SetColor(l.getColor ());
    32.       update C;
    33.     }
    34.   }
    35.  L.remove(l);
    36. }
    37. return C;
```

In algorithm 2, line 2 the function *getAdjacentVertex*() return the k-1 hops adjacent vertex of p because to complete a k-clique, only k-1 different adjacent vertex information is needed. This strategy helps graph data spread over work node in cluster and reduce communication. The line 7 function removeVertexConnect() is to remove all the vertex connect to the vertex in set A, to make sure no vertex is compute duplicate. To keep the pseudo-code brief, we skipped the steps between locate k-clique and find community structure that is build the clique set C and put the result cliques of "locate k-clique" into C. In the "find community structure" process, we use a chromatic method to prevent recomputed cliques and use setColor and getColor to implement this method. The function isOverLap () in line 31 used for compute whether two k-cliques have k-1 identical vertex, which means overlapped.

5 Performance

We tested MBGM for functionality, reliability, usability, efficiency, maintainability, and portability. The evaluation was performed on clusters with 9 nodes, each of which comprises two Intel(R) Xeon(R) CPU E5530 processors, 48 GB main memory, and 1024 GB hard drive. The evaluation data is a randomly generated graph data set scale ranging from 10,000 edges to 2000,000 edges. We also deployed a BC-PDM on the same cluster and ran some social network analysis algorithms using Google web data. Some of the results are shown in **Fig. 4**. We compared MBGM and BC-BSP with the PageRank algorithm on a 4-node cluster, but where the nodes have the same hardware. The results are recorded in **Fig. 5**. Finally, we use two clusters to test the performance of LPA algorithm, cluster A is the 9 nodes cluster described previous, and cluster B uses 4 nodes of cluster A. The characteristics of these graphs' data are shown in **Table 2**.

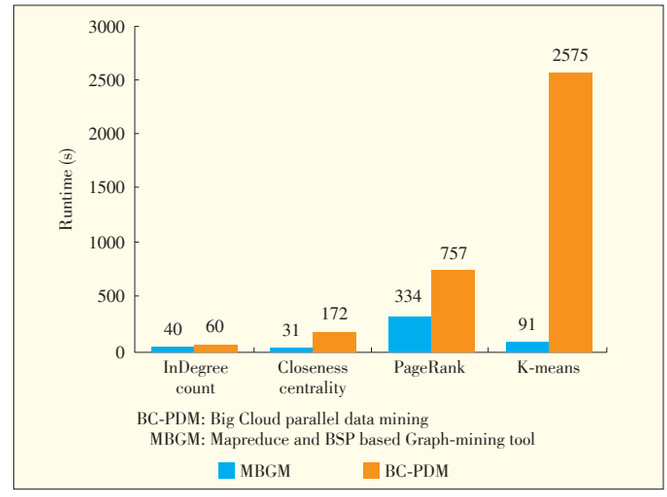
The results show that most graph-mining jobs can be accomplished in a short time and benefit from well-designed architecture. Also, the MBGM has a higher performance than BC-PDM and BC-BSP. Fig. 3 shows the LPA has good performance and scalability to handle various graph data.

6 Discussion

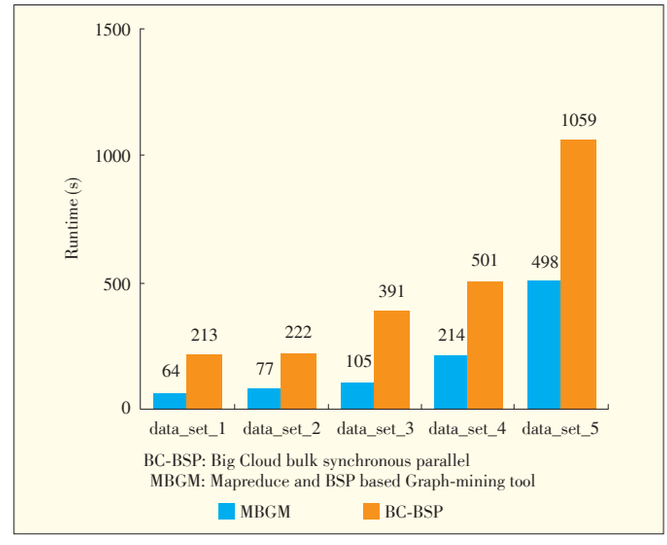
The main reason that MBGM performed better than BC-PDM and BC-BSP is the different parallel computing engine. MBGM uses BSP as the graph computing engine and MapReduce as pre-processing computing engine. However, BC-PDM uses MapReduce as the computing engine and BC-BSP uses a modified BSP model as the computing engine.

Most graph computing algorithms require quantity communication among vertex and much iteration. The MapReduce model computing engine can compute big data, but needs to write temporary results to disk every Map-Reduce phase. This limits the iterative computing ability. BC-BSP is based on BSP computing model, but modified the mechanism of read and write data to memory in order to extend the data capability. We believe this mechanism also increase the complexity of BC-BSP and reduced the performance.

The MBGM takes advantage of both MapReduce and BSP



▲**Figure 4. Comparison of MBGM and BC-PDM on data_set_5.**



▲**Figure 5. PageRank performance of MBGM and BC-BSP.**

▼**Table 2. Networks basic structural properties**

Name	$\|N\|$	$\|E\|$	Type
data_set_1	17,500	100,000	Random
data_set_2	72,000	500,000	Random
data_set_3	175,000	1,000,000	Random
data_set_4	720,000	5,000,000	Random
data_set_5	1,750,000	10,000,000	Random
data_set_6	3,500,000	20,000,000	Random
GoogleWeb	875,713	5,105,039	Web

computing models. The BSP model parallel computing engine ensures the graph algorithm performance and the MapReduce pre-processing engine to reduce the original data scale which also improves the data computing capability of MBGM. Also, with the help of operation flow, the MBGM can run several graph-mining tasks in user defined order without people watching. This saves manpower and, compared with other command-line parallel computing tool, saves the time of operator configure tasks one after previous one finished.

7 Conclusion and Future Work

In this study, we introduced MBGM based on Cloud computing. It has the ability to analyze big graph data and achieved a better performance than the Hadoop-based data mining tools BC-PDM and BSP-based parallel platform BC-BSP. We expected to mix more parallel computing model to achieve a higher performance of graph-mining both in data scale and computing speed.

Acknowledgment

We thank the following individuals who contributed ideas, feedback, and guidance: Wu Bin, Yang Juan and Wang Bai. We are very grateful of assistance and research that Technology Software Engineering Group has provided.

References

[1] Z. H. Liu and Q. L. Zhang, "Research overview of big data technology," *Journal of Zhejiang University (Engineering Science)*, vol. 48, no. 6, pp. 957–972, 2014.
[2] CNNIC. (2014 Jan.). Statistical report on internet development in China. [Online]. Available: http://cnnic.cn/hlwfzyj/hlwxzbg/hlwtjbg/201403/P020140305346585959798.pdf
[3] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI: The Complete Reference*. Cambridge, USA: MIT press, 1995.
[4] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.
[5] L. G. Valiant, "A bridging model for parallel computation," *Communications of the ACM*, vol. 33, no. 8, pp. 103–111, Aug. 1990. doi: 10.1145/79173.79181.
[6] M. Zaharia, M. Chowdhury, M. Franklin, S. shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proc. 2nd USENIX Conf. on Hot Topics in Cloud Computing*, Boston, USA, Jun. 2010.
[7] S. Owen, R. Anil, T. Dunning, and E. Friedman, *Mahout in Action*. Greenwich, USA: Manning Publications, 2011.
[8] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein, "Graphlab: A new framework for parallel machine learning," CMU Tech. Rep., GraphLab Project arXiv:1006.4990, 2010.
[9] U. Kang, C. E. Tsourakakis, and C. Faloutsos, "PEGASUS: A peta-scale graph-mining system implementation and observations," in *Ninth IEEE Int. Conf. on Data Mining*, Miami, USA, Dec. 2009, pp. 229– 238. doi: 10.1109/ICDM.2009.14.
[10] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, "Dryad: Distributed data-parallel programs from sequential building blocks," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 3, pp. 59– 72, Jun. 2007. doi: 10.1145/1272998.1273005.
[11] L. Yu, J. Zheng, W. Shen, B. Wu, B. Wang, L. Qian, and B. Zhang, "BC-PDM: Data mining, social network analysis and text mining system based on cloud computing," presented at *18th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Beijing, China, 2012.
[12] G. Malewicz, M. H. Austern, A. J. C. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: A system forlarge-scale graph processing," in *Proc. SIGMOD'10*, Indianapolis, USA, pp. 135–145.
[13] Y. Bao, Z. Wang, Y. Gu, G. Yu, F. Leng, H. Zhang, B. Chen, C. Deng, and L. Guo, "BC-BSP: A BSP-based parallel iterative processing system for big data on cloud architecture," in *Proc. DASFAA Workshops 2013*, Wuhan, China, pp. 31–45. doi: 10.1007/978-3-642-40270-8_3.
[14] R. S. Xin, J. E. Gonzalez, M. J. Franklin, and Ion Stoica, "Graphx: A resilient distributed graph system on spark," in *First International Workshop on Graph Data Management Experiences and Systems*, New York, USA, 2013, No. 2. doi: 10.1145/2484425.2484427.
[15] S. Seo, E. J. Yoon, J. Kim, S. Jin, J. Kim, and S. Maeng, "Hama: An efficient matrix computation with the MapReduce framework," in *2010 IEEE Second Int. Conf. on Cloud Computing Technology and Science*, Indianapolis, USA, 2010, pp. 721–726. doi: 10.1109/CloudCom.2010.17.
[16] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford InfoLab, Tech. Rep. SIDL-WP-1999-0120, 1999.
[17] H. Tong, C. Faloutsos, and J. Pan. "Fast random walk with restart and its applications," in *Proc. 6th IEEE Int. Conf. on DataMining, Hong Kong*, China, 2006, pp. 613–622.
[18] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, Sept. 1999. doi: 10.1145/324133.324140.
[19] M. Girvan and M. E. J. Newman, "Community structurein social and biological networks," *Proc. Natl Acad Sci USA*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.
[20] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, 2004. doi: 10.1103/PhysRevE.70.066111.
[21] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, 2004. doi: 10.1103/PhysRevE.69.026113.
[22] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in largescale networks," *Physical Review E*, vol. 76, no. 3, 2007. doi: 10.1103/PhysRevE.76.036106.
[23] Z. Zeng, B. Wu, and H. Wang, "A parallel graph partitioning algorithm to speed up the largescale distributed graph-mining," in *Proc. 1st Int. Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, Beijing, China, 2012, pp. 61–68. doi: 10.1145/2351316.2351325.
[24] G. Karypis and V. Kumar, "Metis-unstructured graph partitioning and sparse matrix ordering system, version 2.0," Dept. Computer Science, Univ. of Minnesota, Tech. Rep., 1995.
[25] J. Yang and D. Zhang, "Lightweight workflow engine based on Hadoop and OSGI," presented at *5th IEEE Int. Conf. on Broadband Network & Multimedia Technology*, Beijing, China, 2013.

Manuscript received: 2014-04-04

Biographies

Zhenjiang Dong (dong.zhengjiang@zte.com.cn) reveived his MS degree in telecommunication from Harbin Instituteof Technology in 1996. He is the deputy head of the Service Institute of ZTE Corporation,. His research interests include cloud computing and mobile internet.

Lixia Liu (liu.lixia@zte.com.cn) is a senior engineer in the pre-research department of ZTE. She received her MS degree from Ocean University of China in 2008. Her research interests include natural language processing, text mining, data mining, machine learning, mathematical statistics and cloud computing.

Bin Wu (wubin@bupt.edu.cn) received his PhD degree from the Institute of Computing Technology, Chinese Academy of Science, Beijing, in 2002. He is a senior member of CCF. He is current a professor at the School of Computer Science, Beijing University of Posts and Telecommunications, China. His research interests include data mining, complex network, and cloud computing. He has published more than 100 papers in refereed journals and conferences.

Yang Liu (liuyang1984@bupt.edu.cn) received his BS degree in computer science from Henan University of Technology in 2007. He is currently a PhD candidate at the School of Computer Science, Beijing University of Posts and Telecommunication, China. His research interests include social network analysis, data mining, and cloud computing.

DOI: 10.3969/j. issn. 1673-5188. 2014. 04. 004
http://www.cnki.net/kcms/detail/34.1294.TN.20141208.1906.001.html, published online December 8, 2014

Facial Landmark Localization by Gibbs Sampling

Bofei Wang[1], Diankai Zhang[1], Chi Zhang[2], Jiani Hu[2], and Weihong Deng[2]
(1. ZTE Corporation, Shenzhen 518057, China;
2. Beijing University of Posts and Telecommunication, Beijing 100876, China)

Abstract

In this paper, we introduce a novel method for facial landmark detection. We localize facial landmarks according to the MAP criterion. Conventional gradient ascent algorithms get stuck at the local optimal solution. Gibbs sampling is a kind of Markov Chain Monte Carlo (MCMC) algorithm. We choose it for optimization because it is easy to implement and it guarantees global convergence. The posterior distribution is obtained by learning prior distribution and likelihood function. Prior distribution is assumed Gaussian. We use Principle Component Analysis (PCA) to reduce the dimensionality and learn the prior distribution. Local Linear Support Vector Machine (LL-SVM) is used to get the likelihood function of every key point. In our experiment, we compare our detector with some other well-known methods. The results show that the proposed method is very simple and efficient. It can avoid trapping in local optimal solution.

Keywords

facial landmarks; MAP; Gibbs sampling; MCMC; LL-SVM

1 Introduction

Facial landmark detection is a crucial step for face-related tasks such as face recognition [1]–[3], face tracking, face animation and 3D face modeling. The accuracy of detection significantly affects the performance of such face-related systems.

Existing methods [4]–[15] for facial landmark detection are:

1) A component detector, which is usually a classifier trained for the local feature of each component, is used to search the whole face image and decide which subwindow is the relevant component. To make it robust for the degradation and corruption of local feature, shape constraint is combined to choose the optimal figuration of the key points.

2) A regressor is trained according to the whole image region or local feature. The regressor directly predicts the position of the key points. It is more efficient because it predicts the key points without scanning. Facial landmark detection is quite challenging when a face image is affected by the face angle, facial expression, and accessories such as glasses. In **Fig. 1**, we show some such challenging images.

In an Active Shape Model (ASM) [6], the deformable face shapes are represented by a set of key points that are localized with feature detection methods. The shape variations are modeled by Principal Component Analysis (PCA) so that the face shape can only vary in controlled direction which is learned during training. An Active Appearance Model (AAM) [4] solves the problem by jointly modeling holistic appearance and shape. In AAM, the shape and texture are combined in the PCA subspace so that PCA coefficients are jointly tuned to reduce the geometry and texture differences from the mean face.

▲**Figure 1. Examples of face image with wider range of pose, expression and affiliations.**

Everingham et al. [9] model the face configuration by using pictorial structures and handle a wider range of pose, lighting, and expression by modeling the joint probability of the location of nine fiducials relative to the bounding box with a mixture of Gaussian trees. Belhumeur et al. [12] propose a Bayesian model that combines the outputs of the local detectors with a consensus of non-parametric global models for part locations. Uricar et al. [10] jointly optimize appearance similarity and deformation cost with a parameterized scoring function where the parameters are learned from training instances rather than validation instances using the structured output Support Vector Machine (SVM) classifier. In recent years, many regression methods have been proposed. These methods make it possible

This work is supported by ZTE Industry-Academia-Research Cooperation Funds.

to precisely localize facial landmarks. Dantone et al. [11] use the head pose as a global feature and uses conditional regression forests to learn the distributions conditional to global face properties. Cao et al. [7] directly learns a two-level boosted regression function based on shape indexed features to infer the whole facial shape from the image and explicitly minimize the alignment errors over the training data. Xiong et al. [16] predict shape increment by applying linear regression on SIFT features.

In this paper, we introduce a novel method for detecting facial landmarks. We localize the landmarks according to the Maximum a posteriori probability (MAP) criterion. The posterior distribution is obtained by learning prior distribution and likelihood function. Prior distribution is assumed to be Gaussian. Local Linear SVM (LL-SVM) [17], [18] is used to obtain the likelihood function of every key point. To maximize the posterior distribution and guarantee global convergence, we use Gibbs sampling [19]. Compare to the existing methods, our method can efficiently optimize the posterior probability in a huge probability space.

The remaining of this paper is organized as follows. Section 2 describes the detailed methodology for localizing the facial landmarks. Section 3 explains the experiment configuration. We summarize this paper in Section 4.

2 Localization of Facial Landmarks

We localize facial landmarks on the face image. We first obtain the face box by an off-the-shelf face detector, and then we normalize the face image to 100 × 100 size. We implement the localization work on the normalized face image and then convert back to the original image. We denote the facial landmark position as a vector $X=[x_1,y_1,x_2,y_2,\ldots,x_m,y_m]^T$ where x_i, y_i are the horizontal and vertical coordinates of the landmark, the gray face image I_G. Localizing the position is used to find an optimal X^* in the face image by maximizing the posterior probability. The posterior probability is given by:

$$X^* = \arg\max_X p(X|I_G) = \arg\max_X p(X)p(I_G|X) \quad (1)$$

We first learn the prior probability density distribution $p(X)$ and the likelihood $p(I_G|X)$. Then, we use a novel method to find the optimal X^*. To locate the key points on the face box detected by our off-the-shelf face detector, the key points are restricted to a small region relative to the face box. Therefore, we define a search window for every key point and locate the key points in the corresponding window. We calculate the distribution of each key point. Then the search window is defined to include almost all points.

2.1 Likelihood Learning

The likelihood is used to measure the feature similarity of the probe points and the truth points. To obtain the likelihood score, we use methods such as statistic model, regression method, and discriminative method. We choose the discriminative method to obtain the likelihood score. We call this likelihood score in the searching window as salient map in the following part. We calculate the likelihood probability of normalizing the likelihood score to 0-1.0. The intensity value of each pixel in the neighborhood is used as the local feature. The LL-SVM can produce the likelihood score for every point. Next, we simply depict a fast version of LL-SVM. As described in [17], the coding vector of orthogonal coordinate coding (OCC) should be normalized in L1-norm. In our experiment, we find that, if we have normalized x, it is unnecessary to normalize the coding value C_x because such L1-normalization is trivial in improving localization accuracy. So, to simplify our deduction, we omit the last step of OCC. Further, the localization task is concerned only with the relative value of the LL-SVM output on the detected area. Therefore, the bias b can be ignored, and the decision function is a quadratic form, given by:

$$\sum_j y_j\alpha_j K(x,x_j) = \sum_j y_j\alpha_j x^T GG^T x_j x^T x_j =$$
$$x^T\left(\sum_j y_j\alpha_j GG^T x_j x_j^T\right)x = x^T Ax \quad (2)$$

where x_j, y_j, α_j are the support vector, the label of the support vector, and the coefficient of the support vector. $A=\sum_j y_j\alpha_j GG^T x_j x_j^T$. G is the generator matrix composed of the normalized orthogonal bases using SVD. Then we can rewrite A to a real symmetric matrix. $A'=(A+A^T)/2$, which can be given as:

$$A' = Z\Lambda Z^T \quad (3)$$

where $Z=[Z_1,Z_2,Z_3,Z_4,\cdots]$ is the orthogonal matrix consisting of eigenvectors of A' and $\Lambda=\begin{pmatrix}\lambda_1\cdots 0\\ \cdots\ \vdots\ \cdots\\ 0\cdots\lambda_n\end{pmatrix}$ is a diagonal matrix whose diagonal element is the responding Eigenvalues.

$$x^T Ax = x^T A'x = x^T Z\Lambda Z^T x = x^T\left(\sum_{i=1}^{d}\lambda_i z_i z_i^T\right)x = \sum_{i=1}^{d}\lambda_i\left(z_i^T x\right)^2 \quad (4)$$

where d is the number of nonzero eigenvalues. If λ_i is sorted by its absolute value in descend order. We can approximate $x^T Ax$ as:

$$x^T Ax = \sum_{i=1}^{n_0}\lambda_i\left(z_i^T x\right)^2 \quad (5)$$

From experience, we find that n_0 can be highly compressed

relative to n. The result is guaranteed even when $n_0/n = 1/10$.

The training procedure for Fast LL-SVM is shown in **Algorithm 1**. In experiment, the fast LL-SVM can be 3 times faster than LL-SVM.

Algorithm 1. Training for Fast LL-SVM

Denotes: D is the matrix of training samples. Each columns of D is a feature vector of a train sample.

1: Learning Orthogonal Basis Vectors (u, s, v) = $SVD(D)$, select the top N columns of u as the basis vectors. These vectors assemble the generating matrix G.

2: Compute training instance matrix K

$$K_{ij} = K(i,j) = < C_{D_i D_i^T}, C_{D_j D_j^T} >$$

3: Use the traditional SVM package to solve Feed matrix K, training labels and other parameters to the SVM package, get the support vectors SVs and responding weight α.

4: Compute A′, then the approximate coefficients λ_i and vectors z_i.

2.2 Prior Learning

The prior probability of X can be learned from the training set. To make it convenient to deduce, we describe X as a matrix with dimension n^2. Each row represents one key point. First, we decompose X into several independent parameters: scale, angle, translation and inherent shape factor. Then we get

$$X = s*V*T_r + C_{xy} \tag{6}$$

where s is the scale factor, T_r is the orthogonal matrix $\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\vartheta & \cos\theta \end{bmatrix}$ related to the rotation of key points; C_{xy} is the translation component with each row as [C_x C_y]; and V is the normalized shape. Because these parameters are independent, the prior probability is:

$$p(X) = p(s)p(\theta)p(C_{xy})p(V) \tag{7}$$

We model different parameters in different ways. Similar to ASM, the distribution of the inherent shape w is also assumed Gaussian. Then we use PCA to reduce the dimension and learn the distribution.

$$u = B^T(V - u_v) \tag{8}$$

In this formula, we unfold V and u_v in vector B is the matrix composed by the eigenvectors of the covariance matrix of V. is The mean normalized shape is denoted u_v. The eigenvalue is the variance in each dimension. We could obtain the prior distribution of V by distribution of u.

$$u = Bu + u_v$$
$$p(u) = \frac{1}{z}\exp\{-u^T \Lambda^{-1} u\} \tag{9}$$

2.3 The Optimization by Gibbs Sampling

Once the prior and likelihood probability have been determined, we search for the optimal X to maximize the posterior probability. We find the globally optimal solution by traversal, but the computation complexity is very high. For example, if we intend to locate seven key points, every key point has 900 candidate locations. Therefore we have to perform 900^7 times. Markov-chain Monte Carlo (MCMC) method is a technique for sampling from probability distributions by constructing a Markov chain whose equilibrium distribution is equal to the target distribution. MCMC is used in our system for the property of guaranteed global convergence. It has been successfully used in face prior learning and image segmentation. There are different kinds of MCMC methods, including Metropolis-Hastings algorithm, Gibbs sampling, and Slice sampling. Gibbs sampling does not require any tuning if all the conditional distributions of the target distribution can be sampled exactly. Thus, we choose Gibbs sampling.

In our Gibbs sampling, the key points location X is decomposed into different parameters. The probability space is simply controlled by $P = \{u, s, \theta, C_{xy}\}$, where u is the m-dimensional vector decided by PCA and C_{xy} is two-dimensional vector. That is to say, we can sample X by sampling the parameters. We write P in $P = \{P_1, P_2, P_3, \ldots, P_K\}$, where K is the total number of parameters of the model. We denote the ith sample P^i. The sampling process is as follows:

1) Begin with initial value P^0;

2) For each sample P^i, sample each variable P_j^i from its conditional distribution up on the others $p(P_j^i | P_1^i, P_2^i, \ldots, P_K^{i-1})$.

That is, sample each variable from its condition distribution upon all other variables, using of the most recent value for each variable.

The conditional distribution of each variable can be computed from the joint density. The conditional distribution is the marginal distribution of each random variable because we have defined them as independent. Repeating step 2, we obtain the samples $P^1, P^2, P^3, \ldots$ subjected to the target distribution. Then, we get samples of locations $X^1, X^2, X^3, \ldots$, Finally, X^* is computed using (1).

For initialization, we have tried two methods: Average of Synthetic Exact Filters (ASEF) [20] and SVM. We choose SVM because it is more precise than ASEF. The accuracy of initialization affects the convergence rate of Gibbs Sampling. In this paper, we use LL-SVM to obtain the likelihood score for every key point. Details are given in the last section. We choose the

maximum probability point as the initial points.

3 Experiment and Comparison

In this section, we present the experiment to evaluate the proposed facial landmark detector. We also compare the proposed method with such methods as the independently SVM detector, active shape models, and the detector proposed by Everingham *et al.*

3.1 Experiment Settings

To determine the effectiveness of the proposed method, we test our detector in the Labeled Faces in the Wild (LFW) [21] database. There are 13,233 images, each 250 × 250 pixels. It contains a great variance and the image quality is very low, which is realistic. Dantone et al. describe the manual annotation of the eight interested landmarks of LFW.

To keep comparability with [10], we randomly split the LFW database into training, testing and validation sets. The proportion of the three sets is 6:2:2. We compared with other competing detectors on the same testing set. The training and validation parts are selected using the same method as in [10].

We used the proposed detector and base line independent SVM detector. The other competing detectors had their own training databases. Different evaluation criteria are used to measure the detectors: mean normalized deviation, defined by (10); and maximal normalized deviation, defined by (11).

$$E(X,X^*)=\frac{K(X)}{M}\sum_{j=1}^{M}\left\|x_j-x_j^*\right\| \tag{10}$$

$$E(X,X^*)=K(X)\max_{j=1\cdots M}\left\|x_j-x_j^*\right\| \tag{11}$$

where x_j is the jth points ground truth location and x_j^* is the jth points predict location. M is the number of key points. $K(X)$ is the normalization factor [10], defined as the length of the line connecting the mid-point between the eye centers with the mouth center.

In our experiment, we test the detector with 7-landmarks: corners of the eyes (4 landmarks), corners of the mouth (2 landmarks), and the nose.

3.2 Compared Methods

In this sub-section, we introduce all the methods compared in our experiments. Some examples of localization are shown in **Fig. 2**.

3.2.1 Proposed Method

First, we use the Fast LL-SVM to determine the likelihood score for every key point. Details are given in the last section. We choose the maximum probability point as the initial points. In theory, we can obtain the optimal solution of X in a certain number of steps. Through experimentation, we find that the time needed for convergence is very long. The reason for this is that the posture is various, leading to a very huge prior probability space.

▲**Figure 2. Examples of localization in LFW database.**

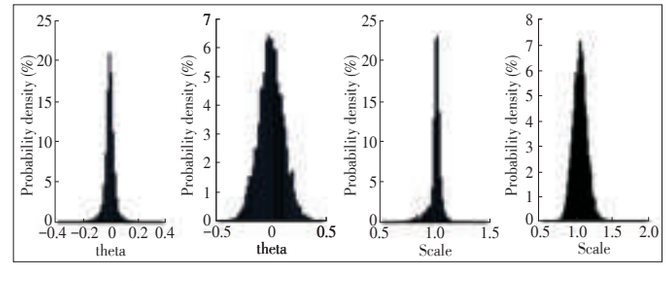
We align the key points to a template by procrustes analysis method [22] and then determine the prior model according to the aligned points. Thus, the probability space is compressed. **Fig. 3** shows a comparison between the two conditions. We find that the probability space is small after the alignment process. Then, we make samples in the probability space by the Gibbs Sampling algorithm. Finally, the optimal X^* is computed using (1).

3.2.2 Independently SVM Detector

This detector consists of eight independent SVM classifiers for each landmark. For training, LIBSVM [23] (the SVM tool package) is used. For each individual landmark, the training set is constructed. This set contains examples of the positive and negative class. We use the gray value of pixel patch as the features. That is, for each point, we choose the pixel value in the neighborhood as its feature. The positive samples are generated by patches cropped around the ground truth positions of the respective components. The negative samples are patches cropped from the image with a certain distance to the ground truth positions. The distance between the negative samples and the ground truth points satisfies the following condition:

$$\left\|x_j-\tilde{x_j}\right\|>\beta^*K(X) \tag{12}$$

where x_j is the ground truth location of jth key point and $\tilde{x_j}$



▲**Figure 3. Comparison of the probability distribution of scale and theta.**

is the negative samples location, we set $\beta = 0.1$ in the experiment.

To make the comparison meaningful, we also use Fast LLSVM. The SVM regularization constant C is also set as 50 in order to minimize the classification error computed on the validation part of the LFW database. The parameter setting is the same as that used in our proposed method.

For testing, we use the well-trained classifier of all components to predict the results of points in the test image. For each facial landmark, we choose the point with the maximum response of the classifier as the predicted position.

3.2.3 Active Shape Models

ASMs [4], [6] are statistical models of the shape of objects that are iteratively deformed to fit to an example of the object in a new image. The ASM algorithm has been widely used to analyze facial and medical images. Some extensions to this algorithm have been proposed. For example, Constrained Local Models [5] use PCA to model the landmark's appearance, and Boosted Regression Active Shape Models [24] use boosting to predict a new location for each point, given the patch around the current position. Stasm [25] is a C++ software library that is also based on ASM. We compare our proposed method with Stasm.

3.2.4 Detector Proposed by Everingham *et al.*

Everingham *et al.* [9] handle a wider range of pose, lighting, and expression by modeling the joint probability of the location of nine fiducials relative to the bounding box with a mixture of Gaussian trees. The local appearance model is learned by a multiple instance variant of the AdaBoost algorithm with Haar-like features used as the weak classifiers. The deformation cost is expressed as a mixture of Gaussian trees, and the parameters in this mix are learned from examples. This landmark detector is publicly available and we compare it with our detector, which is trained in a database of consumer images. To compare this detector, we consider only the relevant landmarks for our detector.

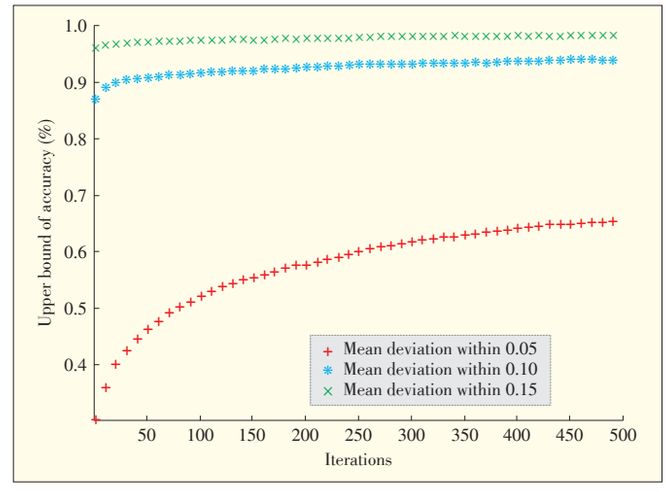
3.2.5 Flandmark Detector Proposed by Uricar *et al.*

Uricar *et al.* [10] jointly optimize appearance similarity and deformation cost with a parameterized scoring function. The parameters in this function are learned from training instances rather than validation instances using the structured output SVM classifier.

3.3 Analysis and Improvement

3.3.1 Sampling Analysis

We evaluate the sampling performance by the corresponding minimum-error sample. The upper-bound of location accuracy is then calculated. The accuracy is defined as the proportion of samples whose location errors are within a certain value. By continually increasing the sampling iterations, we depict the upper-bound curve of location accuracy (**Fig. 4**). As we can see, within 200 iterations the upper bound of accuracy when



▲**Figure 4. The upper-bound of accuracy (cumulative histograms for the mean deviation) increases with the Gibbs Sampling iterations augment.**

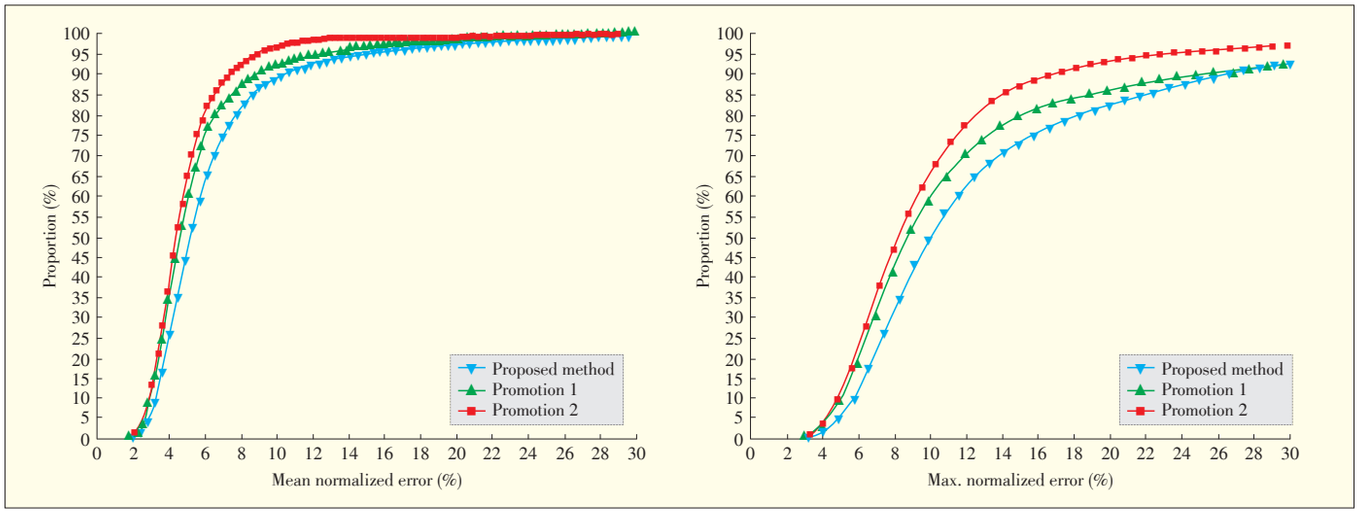
mean deviation within 0.10 is 92.45%, and appears higher with more samplings, but it is almost stable after 300 itertions. But the upper bound of accuracy when mean deviation within 0.05 keeps grow with more samplings.

One problem is inaccuracy because it is very hard to only sample at the ground truth position. We propose two schemes to solve this problem. The two schemes are denoted as method Promotion 1, method Promotion 2. First, we filter the likelihood score through a Gaussian filter to make the score plane smooth. In this way, we avoid very low ground truth due to little noise. In the other solution, "hard" sampling is changed to "soft" sampling. We sample a soft constraint for every point. That is, we construct a Gaussian Window centered in the each sampling point as the soft weight, and then find the maximum response in the window after weighted. **Fig. 5** shows the obvious effect of the two strategies to the proposed methods.

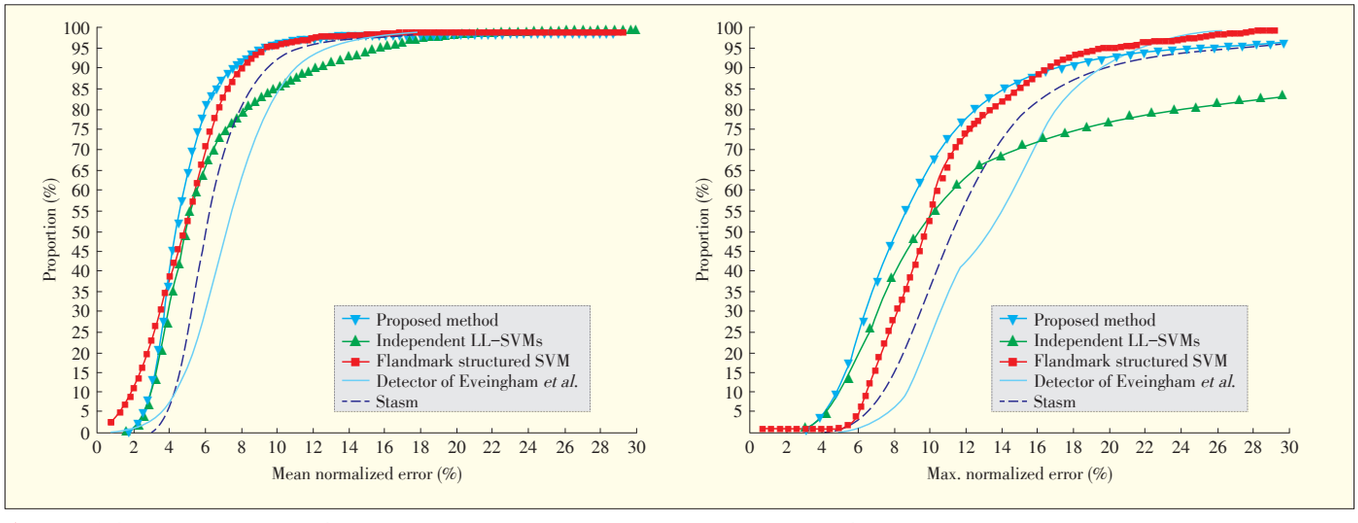
We also propose a method to improve the efficiency of Gibbs Sampling. The method like restarting is used. After every few iterations, we restart the sampling procedure with the last optimal sample. In our experiment, this method can largely reduce the sampling iterations.

3.3.2 Accuracy Comparison

We present the accuracy comparison in **Fig. 6**. The evaluation criteria are the mean normalized error and maximum normalized error. **Table 1** shows the percentage of examples from the test part of the LFW database with the mean/maximal normalized deviation less or equal to 10%. The curves of the Flandmark detector and detector of Eveingham et al. are estimated according to the results in [10]. The proposed method estimates more than 97% of the images with the mean normalized



▲Figure 5. Cumulative histograms for the mean and the maximal normalized deviation shown for the proposed method and promotion methods.



▲Figure 6. Cumulative histograms for the mean and the maximal normalized deviation shown for all compared detectors.

▼Table 1. Percentage of examples from the test part of the LFW database with the mean/maximal normalized deviation less or equal to 10%

	Mean	Maximal
Independent LL-SVMs	85.58%	52.68%
Flandmark detector	96.59%	53.23%
Eveingham *et al.*	85.28%	22.93%
ASM	92.43%	35.00%
Proposed method	97.20%	65.26%

deviation less than 10%. This is similar to the Flandmark detector and far better than the other methods. Our method estimates more than 65% of the images with the max normalized deviation less than 10%. This is far better than the Flandmark detector, which is 53.23%.

Independent LL-SVMs only detect the key points in local area. They do not utilize the shape constraint. The Flandmark detector [10] jointly optimizes appearance similarity and deformation cost with a parameterized scoring function using the structured output SVM classifier. The detector proposed by Eveingham et al. uses an ensemble of weak classifiers as a local discriminative classifier. The deformation cost is expressed as a mixture of Gaussian trees whose parameters are learned from examples. ASM also uses PCA to model the face appearance, while its "profile model" is less powerful then LL-SVM. This "profile model" looks for strong edges or uses the Mahalanobis distance to match a model template for the point.

The proposed Gibbs Sampling method is a novel method that combines the local discriminative information with the global constraint. We use PCA model to constrain the shape whereas existing methods always design out sophisticated formula. The main benefit of this algorithm is its powerful local discriminative classifier and its simple theory to utilize the global constraint. Such a simple method can achieve even bet-

ter results.

4 Conclusion

In this paper, we propose a novel method for facial landmarks detection. We use the MAP criterion to localize the landmarks and LL-SVM to get the likelihood function of every key point. PCA is used to reduce the dimensionality and learn the prior distribution. The posterior probability is optimized by Gibbs sampling. Various experiments on LFW database have shown that this method is efficient. The problem is that proposed method is still too slow to apply to real time system.

References

[1] W. Deng, J. Hu, J. Guo, W. Cai, and D. Feng, "Robust, accurate and efficient face recognition from a single training image: a uniform pursuit approach," *Pattern Recognition*, vol. 43, no. 5, pp. 1748–1762, May 2010. doi:10.1016/j.patcog.2009.12.004.

[2] W. Deng, J. Hu, J. Lu, and J. Guo, "Transform-invariant PCA: a unified approach to fully automatic face alignment, representation, and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1275–1284, Jun. 2014. doi: 10.1109/TPAMI.2013.194.

[3] W. Deng, J. Hu, and J. Guo, "Extended SRC: undersampled face recognition via intraclass variant dictionary," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1864–1870, Sept. 2012. doi: 10.1109/TPAMI.2012.30.

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001. doi: 10.1109/34.927467.

[5] D. Cristinacce and T. Cootes, "Automatic feature localisation with constrained local models," *Pattern Recognition*, vol. 41, no. 10, pp. 3054–3067, Oct. 2008. doi: 10.1016/j.patcog.2008.01.024.

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Comput. Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995. doi: 10.1006/cviu.1995.1004.

[7] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," *Int. J. Comput. Vision*, vol. 107, no. 2, pp. 177–190, Apr. 2014. doi: 10.1007/s11263-013-0667-3.

[8] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *IEEE Conf. Comput. Vision Pattern Recognition*, Providence, USA, Jun. 2012, pp. 2879–2886. doi: 10.1109/CVPR.2012.6248014.

[9] M. Everingham, J. Sivic, and A. Zisserman, "Hello! my name is... buffy—automatic naming of characters in TV video," *17th BMVC*, Edinburgh, UK, Sept. 2006, pp. 899–908. doi:10.5244/C.20.92.

[10] M. Uřičář , V. Franc, and V. Hlaváč , "Detector of facial landmarks learned by the structured output SVM," in *7th Int. Conf. Comput. Vision Theory Appl.*, Rome, Italy, pp. 547–556.

[11] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, "Real-time facial feature detection using conditional regression forests," in *IEEE Conf. Comput. Vision Pattern Recognition*, Providence, USA, Jun. 2012, pp. 2578–2585. doi: 10.1109/CVPR.2012.6247976.

[12] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2930–2940, Dec. 2013. doi: 10.1109/TPAMI.2013.23.

[13] J. Hu, Y. Li, W. Deng, J. Guo, and W. Xu, "Locating facial features by robust active shape model," in *2nd IEEE Int. Conf. Network Infrastructure Digital Content*, Beijing, China, Sept. 2010, pp. 196–200. doi: 10.1109/ICNIDC.2010.5657840.

[14] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust face landmark estimation under occlusion," in *IEEE Int. Conf. Comput. Vision*, 2013, pp. 1513–1520. doi: 10.1109/ICCV.2013.191.

[15] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *IEEE Conf. Comput. Vision Pattern Recognition*, Portland, USA, Jun. 2013 , pp. 3476–3483. doi: 10.1109/CVPR.2013.446.

[16] X. Xiong and F. De la Torre Frade, "Supervised descent method and its applications to face alignment," in *IEEE Conf. Comput. Vision Pattern Recognition*, Portland, USA, Jun. 2013 , pp. 532–539. doi: 10.1109/CVPR.2013.75.

[17] Z. Zhang, L. Ladicky, P. Torr, and A. Saffari. (2011). Learning anchor planes for classification. in *Neural Inform. Process. Syst. Conf.*, Granada, Spain, 2011, pp. 1611–1619.

[18] L. Ladicky and P. Torr, "Locally linear support vector machines," in *28th Int. Conf. Mach. Learning*, Bellevue, USA, 2011, pp. 985–992.

[19] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984. doi: 10.1109/TPAMI.1984.4767596.

[20] D. S. Bolme, B. A. Draper, and J. R. Beveridge, "Average of synthetic exact filters," in *IEEE Conf. Comput. Vision Pattern Recognition*, Miami, USA, Jun. 2009, pp. 2105–2112.

[21] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. (2007). Labeled faces in the wild: a database for studying face recognition in unconstrained environments. [Online]. Available: http://vis-www.cs.umass.edu/lfw/lfw.pdf

[22] D. G. Kendall, "A survey of the statistical theory of shape," *Statist. Sci.*, vol. 4, no. 2, pp. 87–99, 1989.

[23] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, article 27, Apr. 2011. doi: 10.1145/1961189.1961199.

[24] D. Cristinacce and T. Cootes. (2007). Boosted regression active shape models. *BMVC* [Online]. Available: http://www.dcs.warwick.ac.uk/bmvc2007/proceedings/CD-ROM/papers/paper-131.pdf

[25] S. Milborrow and F. Nicolls. (2014). Active shape models with sift descriptors and mars. [Online]. Available: http://www.milbo.org/stasm-files/active-shape-models-with-sift-and-mars.pdf

Manuscript received: 2014-08-20

Biographies

Bofei Wang (wang.bofei@zte.com.cn) received his BE degree in electronic information engineering and MS in Communication and information system from Huazhong University of Science and Technology (HUST), China in 2003 and 2007. He is a senior video and image algorithm engineer of ZTE Corporation. His research interests include video and image processing, pattern recognition, and computer vision.

Diankai Zhang (zhang.diankai@zte.com.cn) received his BE degree in electronic information engineering and MS degree in signal and information processing from Nanjing University of Posts and Telecommunications (NUPT), China in 2006 and 2009. He is a senior video and image algorithm engineer of ZTE Corporation. His research interests include video and image processing, pattern recognition, and computer vision.

Chi Zhang (zhangchi2013@bupt.edu.cn) received his BE degree in Electronic Information Engineering from NUPT in 2013, and is currently a master student in School of Information and Telecommunications Engineering of Beijing University of Posts and Telecommunications (BUPT), China. His research interests include pattern recognition, machine learning, and computer vision.

Jiani Hu (jnhu@bupt.edu.cn) received her BE degree in telecommunication engineering from China University of Geosciences in 2003, and PhD degree in signal and information processing from BUPT in 2008. She is currently a lecturer in School of Information and Telecommunications Engineering, BUPT. Her research interests include information retrieval, statistical pattern recognition, and computer vision.

Weihong Deng (whdeng@bupt.edu.cn) is an associate professor in School of Information and Telecommunications Engineering, BUPT. His research interests include statistical pattern recognition and computer vision, with a particular emphasis on face recognition. He has published over 40 papers in international journals and conferences, including a technical comment on face recognition in *Science* magazine. He also serves as the reviewer for several international journals, such as *IEEE TPAMI*, *IJCV*, *IEEE TIP*, *IEEE TIFS*, *PR*, and *IEEE TSMC-B*. His dissertation titled "Highly accurate face recognition algorithms" was awarded the Outstanding Doctoral Dissertation by Beijing Municipal Commission of Education in 2011. He has been supported by the program for New Century Excellent Talents by the Ministry of Education of China since 2013.

DOI: 10.3969/j. issn. 1673-5188. 2014. 04. 005
http://www.cnki.net/kcms/detail/34.1294.TN.20141212.0940.002.html, published online December 12, 2014

Angle-Based Interference-Aware Routing Algorithm for Multicast over Wireless D2D Networks

Qian Xu[1], Pinyi Ren[1], Qinghe Du[1], Gang Wu[2], Qiang Li[2], and Li Sun[1]

(1. Department of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China;
2. Microelectronics Institute Algorithm Design Department, ZTE Corporation, Shenzhen 518057, China)

Abstract

Wireless device-to-device (D2D) communications sharing the spectrum of cellular networks is important for improving spectrum efficiency. Furthermore, introducing multicast and multi-hop communications to D2D networks can expand D2D service functions. In this paper, we propose an angle-based interference-aware routing algorithm for D2D multicast communications. This algorithm reuses the uplink cellular spectrum. Our proposed algorithm aims to reduce the outage probability and minimize the average hop count over all multicast destinations (i.e., multicast receivers), while limiting interference to cellular users to a tolerable level. In particular, our algorithm integrates two design principles for hop-by-hop route selection. First, we minimize the distance ratio of the candidate-to-destination link to the candidate-to-base-station link, such that the selected route advances closer to a subset of multicast receivers. Second, we design the angle-threshold based merging strategy to divide multicast receivers into subsets with geographically close destinations. By applying the two principles for selection of each hop and further deriving an adaptive power-allocation strategy, the message can be more efficiently delivered to destinations with fewer branches when constructing the multicast tree. This means fewer duplicated data transmissions. Analyses and simulations are presented to show the impact of system parameters on the routing performances. Simulation results also demonstrate the superiority of our algorithm over baseline schemes in terms of outage probability and average hop count.

Keywords

device-to-device communications; multicast; interference-aware routing; cellular networks

This work is supported by National Natural Science Foundation of China under Grant No. 61102078, ZTE Industry-Academic-Research Cooperation Funds, and the Fundamental Research Funds for the Central Universities.

1 Introduction

With the popularity of applications such as social networks and local broadcast, which involve direct interaction between end users/nodes, wireless device-to-device (D2D) communication has aroused great interest in the research community [1]–[10]. Wireless D2D communication reusing cellular network spectrum have shown great potential to relieve spectrum scarcity issues. Typically, D2D communications employ a non-orthogonal spectrum sharing approach [3]–[10], which needs specifically designed sharing strategies to control the cross-interferences between users.

Recently, research on coexistence of cellular and D2D networks has mainly addressed interference management strategies [3], [4], transmission mode selection [5], power control [6], routing design [7], resources allocations [8], [9], and network coding [10]. However, the majority of research on D2D networks has focused on one-hop point-to-point transmission between device nodes, which limits the functions and applications of D2D networks. First, D2D users need to lower transmission power to avoid causing intolerable interference to cellular users, and the transmission range is limited. Therefore, multi-hop D2D communications are highly desirable. Second, multiple users often require the same content, such as software download, online gaming, and video streaming. Unicast-based approaches for such services would waste considerable spectrum. Thus, supporting multicast functions in D2D networks is important to better utilize precious wireless resources. To address these issues, we concentrate on the routing design for multicast transmissions over D2D networks.

Much research effort has been dedicated to multicast routing over diverse wireless networks [11]–[19]. However, we still meet new challenges introduced by the unique features of wireless D2D networks. In particular, the routing scheme needs to be aware of interference caused to cellular users. Cellular interferences to D2D links also significantly affect route-selection strategies. These interferences are called inter-network interferences. Moreover, as the multicast receivers are geographically independent, the multicast tree might expand with multiple branches. As a consequence, intra-network interferences often exist across different multicast branches. Joint handling of these two types of interferences for efficient multicast D2D routing remains an open problem that has not been thoroughly studied.

Aiming at supporting efficient multi-hop multicast over D2D networks, we propose an angle-based interference-aware routing algorithm for multicast D2D transmissions. Our proposed algorithm aims at lowering the outage probability for routing and minimizing the average hop count. Specifically, we select the route for each hop by applying the distance ratio minimization and angle-threshold based merging principles. We design the power allocation strategies to the selected routers for each

hop to not only avoid causing intolerable interference to cellular users but also lower the outage probability. Simulation results show that our proposed routing algorithm outperforms the baseline schemes in terms of outage probability and average hop count.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the system model. Section 4 proposes our angle-based interference-aware routing algorithm. Section 5 conducts analyzes and discussions on our proposed routing algorithm. Section 6 evaluates the performances on our proposed algorithm through simulations. The paper concludes with section 7.

2 Related Work

Recent research on wireless D2D networks has not properly addressed the multicast routing issue. In contrast, multicast routing schemes have been proposed in other types of networks, which provide useful information and valuable references for our design in D2D networks. We first review existing research outcomes on multicast routing in the cognitive networks [13]–[15] as cognitive and D2D networks reveal similar features in interference control and protection for prior users. Then we discuss some multicast routing schemes in other networks, such as mobile ad-hoc networks and wireless sensor networks [16]–[19].

The authors of [13] jointly considered scheduling and routing. They formulated the problem by a mixed-integer linear program and developed a polynomial-time algorithm accordingly to identify a near optimal solution. In [14], an on-demand multicast routing and channel-allocation algorithm called OMRA was proposed. The OMRA algorithm sent a signaling packet to build the multicast tree. The authors of [15] investigated the routing problem, where directional antenna is applied in contrast to the generally used omnidirectional antennas.

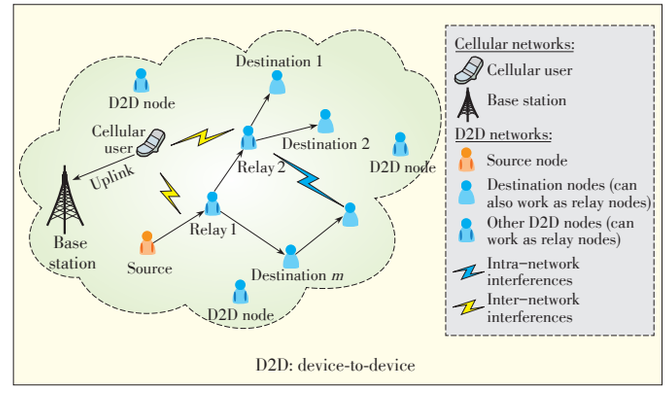
Multicast routing for wireless sensor networks has been studied in [16], [17] and [18]. The authors in [16] first built a heuristic virtual Euclidean Steiner multicast tree by using a specifically designed metric termed reduction ratio. Then, each forwarding node divides destinations into subsets based on the virtual tree and selects respective next hop for each subset. The authors of [17] proposed a geographic multicast routing (GMR) algorithm for wireless sensor networks. The GMR algorithm makes a tradeoff between multicast tree cost and effectiveness of data distribution. The authors of [18] proposed a routing scheme that avoids getting close to the interfering source. In particular, each forwarding node uses the minimum required transmission power to represent the interference strength level and then makes a routing decision. The authors of [19] proposed a position-based multicast (PBM) routing scheme for mobile ad-hoc networks. The proposed PBM algorithm does not need to build or expand the data distribution structure such as multicast tree and/or mesh grid. Next-hop selection is only based on the positions of the forwarding node, associated neighboring nodes, and destinations. All these algorithms have their respective merits but cannot be directly applied to the D2D networks because of the unique features in interference management. In the following sections, discuss the design of an angle-based interference-aware multicast routing scheme.

3 System Model

3.1 System Descriptions

We consider a scenario where the D2D network and cellular network coexist (**Fig. 1**). For the cellular network, we consider one base station and focus on one cellular user's uplink transmission with bandwidth equal to B Hz. The D2D network contains U D2D device nodes (indexed by 1,2,...,U), where we further define the D2D node set by $\aleph \triangleq \{1,2,...,U\}$. All D2D nodes can establish direct communications links with each other by reusing the cellular user's uplink spectrum.

In the D2D network, there is one multicast session where one source node M_{sr} ($M_{sr} \in \aleph$) attempts to multicast a common message to m destinations (multicast receivers) $M_1, M_2, ...M_m \in \aleph$ through multi-hop connections. As shown in Fig. 1, other than source and destination nodes, the rest of D2D nodes can work as the relay nodes (also called forwarding nodes) to help forward the multicast message towards destinations. The destination nodes themselves can function as the relay nodes. Following this setup, the source node, destination nodes, and relay nodes together form a multicast tree. As in Fig. 1, a multicast tree often expands multiple branches. Therefore, there might be multiple D2D nodes transmitting packets simultaneously, which causes intra-network interference between simultaneous transmissions of different D2D links. On the other hand, D2D links reuse the spectrum of cellular users, and there is interference between the D2D links and cellular links. This results in inter-network interference. For multicast routing over D2D networks, we need to take both the inter- and



▲**Figure 1. System model for D2D multicast communications coexisting with a cellular network.**

intra-network interferences into account.

3.2 Multicast Tree

Following the system model introduced in section 3.1, we next present several rigorous definitions to precisely describe the multicast tree structure.

Definition 1 (D2D Multicast tree): A D2D multicast tree κ is a tree structure indicating the paths for packets to be delivered to multicast destinations. Generally, a tree comprises root node, internal nodes, and leaf nodes. For multicast tree, the root node is the source node. All leaf nodes in the tree are destination nodes for multicast; however, destination nodes may not be leaf nodes because they can also function as internal nodes. A direct connection between two nodes is called a hop or an edge, where a hop connecting two nodes a and b, and $a,b\in\aleph$, is written as (a,b). Furthermore, for the hop (a,b), a and b are termed transmitting and receiving node, respectively.

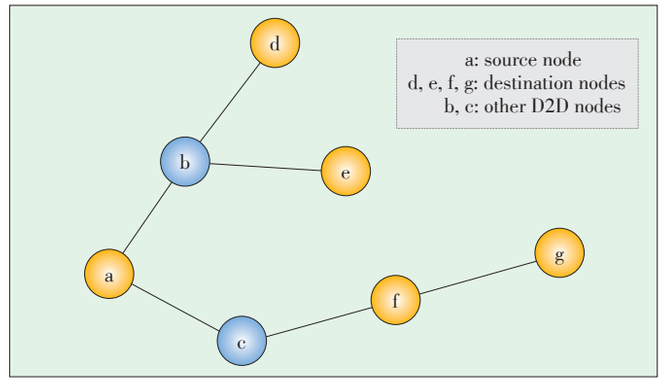
Definition 2 (Level): The level of a node in the multicast tree is defined by one plus the number of edges from the source (root) to this node. Accordingly, the source node M_{sr} is at the first level.

Definition 3 (SL-path): Source-to-leaf path (SL-path) in a multicast tree represents a sequence of nodes and hops (edges) connecting the source node to a leaf node. Assume that there are total Y SL-paths in a multicast tree and then we use $\mathbf{W}_y$, $y=1,2,..,Y$, to denote the set of all edges on yth SL-path. We then have $Y\leqslant m$. We also define $\mathbf{Q}_y$, $y=1,2,..,Y$, as the set of nodes on the yth SL-path.

Definition 4 (xth hop): We define the xth hop belonging to an SL path as the hop connecting the node at the xth level (also called the xth node of this SL path) and that at the (x+1)th level. The xth hop′s receiving node is the (x+1)th hop′s transmitting node. Furthermore, we use H_{xy} to denote the xth hop *of* yth SL-path and T_{xy} and R_{xy} to denote the transmitting node and receiving node of hop H_{xy}, respectively, where $y=1,2,\cdots,Y$ and $x=1,2,\cdots,|\mathbf{W}_y|$. Here Y is the total number of SL paths, and $|\mathbf{W}_y|$ is the number of hops on the yth SL path, both of which depend on how the multicast tree is built, The cardinality of the set is denoted $|\cdot|$.

Definition 5 (xth hop set): We define the xth hop set $\mathbf{H}_x$ as the set of the xth hop belonging to each SL path, i.e., $\mathbf{H}_x=\bigcup_{y=1}^{Y}\{H_{xy}\}$. In addition, we use $\mathbf{T}_x$ and $\mathbf{R}_x$ to denote the set of transmitting nodes and receiving nodes, respectively, of hops in $\mathbf{H}_x$.

Fig. 2 shows a multicast tree example, with a source node $M_{sr}=a$, four destination nodes $M_1=d$, $M_2=e$, $M_3=f$, and $M_4=g$, and two other D2D nodes b and c, where a through g are different elements in the D2D node set $\aleph$. As shown in Fig. 2, there are three leaf nodes d, e, and g, and three internal nodes b, c, and f, where f is also a destination node. We can see from Fig. 2 that there are three SL-paths, denoted by $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$, where $\mathbf{W}_1=\{(a,b),(b,d)\}$, $\mathbf{W}_2=\{(a,b),(b,e)\}$, and $\mathbf{W}_3=\{(a,c),(c,f),(f,g)\}$. Using the first SL path as a typical example, we get the transmitting and receiving nodes as $T_{11}=a$, $R_{11}=b$, $T_{21}=b$, and $R_{21}=d$. Concentrating on the 2nd hop, we can obtain $H_{21}=(b,d)$, $H_{22}=(b,e)$, and $H_{23}=(c,f)$, forming the 2nd hop set $\mathbf{H}_2=\{H_{21},H_{22},H_{23}\}$ for all SL-paths. In addition, we get $\mathbf{T}_2=\{b,c\}$ and $\mathbf{R}_2=\{d,e,f\}$.



▲**Figure 2. A multicast tree.**

3.3 Problem Formulation

Our target is to develop an interference-aware routing algorithm. We need to construct the multicast tree according to the inter- and intra-network interferences. Then, we discuss the transmission and interference model for our framework and formulate the multicast routing problem. The transmission and interference model in this paper is similar to that in [7], which includes unicast routing over D2D networks. However, multicasting involves concurrent transmissions by different branches, and the interference model needs to be modified accordingly.

We use P_0 to denote the transmission power of the cellular user, which is a constant. If D2D node i (i = 1, 2, ... , U) is a transmitting node, it uses power P_i, which is regulated according to the interference statuses. All D2D node transmission power is characterized by a vector $\mathbf{P}=\{P_1,P_2,\ldots,P_U\}$. If a D2D node does not transmit data during the multicast transmission, its power is zero. The distance between the BS and cellular user is denoted d_0; the distance between node i and BS is denoted D_i; we further use Δ_j to denote the distance between node j and cellular user while using $d_{i,j}$ to denote the distance between node i and node j. While the D2D transmissions reuse the cellular user′s uplink spectrum, the resulted signal-to-Interference ratio (SIR) at the BS cannot exceed a tolerable level, denoted ρ_{th}. On the other hand, in order to guarantee the quality of service (QoS) of the D2D communications, the data rate of each hop for D2D multicast, is required to be no less than a specified threshold R_{th}.

The transmissions at each hop for D2D multicast are performed in a time-division fashion. Beginning from the source node, the transmission for the xth hop occupies the xth time slot. When all destinations have received the data packet, the

source node begins to multicast a new packet for all destinations. Future D2D networks may work under the coordination and control of the BS [1], [2]. So, we assume that the BS coordinate the entire multicast process and share information such as the positions and transmission power among D2D nodes.

Following the above principles, we now take the xth hop's transmissions to establish the mathematical framework. In this paper, we mainly consider the path loss for the transmission, where small-scale fading is assumed to be averaged by applying diversity technologies. Correspondingly, the useful signal power received by BS, represented by G_0, is written by

$$G_0 = \alpha P_0 d_0^{-\eta}, \tag{1}$$

where η represents the path-loss exponent and α is a constant decided by antenna gain and/or other factors.

Without loss of generality, we concentrate on the xth hop in the multicast transmission. Then, within the xth time slot, the SIR received at the BS station, denoted by SIR_x^{BS}, is derived by

$$SIR_x^{BS} = \frac{G_0}{\sum_{i \in T_x} \alpha P_i D_i^{-\eta}} = \frac{P_0 d_0^{-\eta}}{\sum_{i \in T_x} P_i D_i^{-\eta}}, \tag{2}$$

where $\mathbf{T}_x$ is the transmitting node set of the xth hop. In order to satisfy the interference constraint, i.e., $SIR_x^{BS} \geqslant \rho_{\text{th}}$, we get

$$\sum_{i \in T_x} P_i D_i^{-\eta} \leqslant \frac{P_0 d_0^{-\eta}}{\rho_{\text{th}}} \tag{3}$$

For any transmitting node in set $\mathbf{T}_x$, the D2D transmission is affected by the cellular user's signal and also the signals from other transmitting nodes in set $\mathbf{T}_x$. The physical signals transmitted by multicast nodes can be different whereas the carried information is the same. Furthermore, it is hard to synchronize multicast nodes. Accordingly, other transmitting nodes cause interference as the cellular user does. Hence, the maximum achievable transmission rate from transmitting node $i \in \mathbf{T}_x$ to its receiving node j, denoted by $\Re_{i,j}$, can be written as

$$\Re_{i,j} = B \log\left(1 + \frac{\alpha P_i d_{i,j}^{-\eta}}{N_0 + \alpha P_0 \Delta_j^{-\eta} + \sum_{k,k \in T_x \setminus \{i\}} \alpha P_k d_{k,j}^{-\eta}}\right), \tag{4}$$

Following (4), we can obtain the maximum transmission rate of the xth hop in the yth SL-path (i.e., hop H_{xy}), denoted by $\Re_{t,r}$, where $t = T_{xy}$ & $r = R_{xy}$, as

$$\Re_{t,r} = B \log\left(1 + \frac{\alpha P_t d_{t,r}^{-\eta}}{N_0 + \alpha P_0 \Delta_r^{-\eta} + \sum_{k \in T_x \setminus \{t\}} \alpha P_k d_{k,r}^{-\eta}}\right). \tag{5}$$

We design a routing algorithm that reduces the average hop count and lowers the outage probability. We define the hop count for a particular destination as the number of hops for it to receive a multicast packet from the source. Further, denoting the hop count for the vth destination by L_v, the average hop count over all multicast destinations is calculated by

$$L = \frac{1}{m} \sum_{v=1}^{m} L_v. \tag{6}$$

Then we formulate a hop-count minimization problem as follows:

$$\begin{aligned}
&\min_{\kappa, P} \left\{ \frac{1}{m} \sum_{v=1}^{m} L_v \right\} \\
&s.t.: \quad \mathbf{T}_1 = \{M_{sr}\}; \\
&\quad \sum_{i \in T_x} P_i D_i^{-\eta} \leqslant \frac{P_0 d_0^{-\eta}}{\rho_{\text{th}}}, \quad \forall\, x = 1,2,\ldots,\max_y \{|\mathbf{W}_y|\}; \\
&\quad \Re_{T_{xy}, R_{xy}} \geqslant R_{\text{th}}, \quad \forall\, y = 1,2,\ldots,Y \; x = 1,2,\ldots,|\mathbf{W}_y|; \\
&\quad R_{ky} \in \{M_1, M_2, \ldots, M_m\}, \quad k = |\mathbf{W}_y|, \quad \forall\, y = 1,2,\ldots,Y; \\
&\quad \exists y \in \{1,2,\ldots,Y\} \; such \; that \; M_v \in \mathbf{Q}_y, \quad \forall\, v = 1,2,\ldots,m.
\end{aligned} \tag{7}$$

where κ denotes the multicast tree structure (see Definition 1) and $\mathbf{P}$ characterizes all D2D nodes' transmission power. The first constraint indicates the root node must be multicast source, the second constraint limits the interference caused to the cellular users' signal, the third constraint requires all hops for multicast can achieve the rate beyond R_{th}, the fourth constraint implies that all leaf nodes be multicast receivers, and the last constraint suggests that any multicast receiver needs to be on some SL-path.

4 Angle-Based Interference-Aware Routing Algorithm for D2D Multicast Transmissions

The hop-count minimization problem formulated in section 3 involves mixed integer-nonlinear programming, the optimal solution to which is unrealistic to track. In this section, we propose a heuristic algorithm, called angle-based interference-aware multicast routing algorithm. Our algorithm has two main parts. One is the receiving-node selection for each hop and the other is the power allocation for receiving nodes, which will be respectively detailed in the following sections. Note that we assume that all location information for D2D nodes and cellular user is known to the BS. The BS will be responsible for the route selection and power allocation for the multicast session.

4.1 The Selection of Receiving Nodes for Each Hop

We take the following crucial factors into account for our routing design. First, the routes from the source need to advance towards each specific multicast destination. Second, the destinations are usually divided into multiple groups lying on different branches of the multicast tree. Thus, each transmitting node often seeks multiple receiving nodes within its trans-

mission range, expanding into different branches. Third, the route selection needs to be regulated according to group characteristics of multicast destinations with close positions and/or similar directions. Following the above guidelines, we introduce two principles for receiving node selections, namely, distance ratio minimization principle and angle-threshold based merging principle.

Assume that the first $(x-1)$ hops have been determined. Thus, we can obtain the set $\mathbf{T}_x$ of transmitting nodes for the xth hop. Of all the transmitting nodes in $\mathbf{T}_x$, we focus on the ith transmitting node in $\mathbf{T}_x$, denoted by s_i^x, to describe our strategy for hop selections. We use $\mathbf{N}_i$ to denote the node set whose elements satisfy the rate constraint for the link connecting them to s_i^x. Then, we call $\mathbf{N}_i$ the neighbor set of s_i^x. The nodes in the first $(x-1)$th hop should not been included in $\mathbf{N}_i$. Therefore, $\mathbf{N}_i$ can be calculated by

$$\mathbf{N}_i=\left\{u\,|\,u\in\aleph\backslash\bigcup_{z=1}^{x}\mathbf{T}_z,\Re_{s_i^x u}\geqslant R_{\text{th}}\right\} \tag{8}$$

The multicast tree has multiple branches. Consequently, the transmitting node s_i^x is responsible for delivering packets to only some of the destinations, the target destination set, associated with s_i^x and denoted $\tilde{\mathbf{M}}_i^x$. An interesting and convenient setup for $\tilde{\mathbf{M}}_i^x$ in our algorithm is that each element of $\tilde{\mathbf{M}}_i^x$ is a target destination subset, which includes destinations with close locations and similar directions. In particular, we use $\tilde{M}_{ij}^x$, $1\leqslant j\leqslant J_i$, to represent the jth element in $\tilde{\mathbf{M}}_i^x$. Each element in $\tilde{M}_{ij}^x$ is a multicast destination node instead of a destination subset in $\tilde{\mathbf{M}}_i^x$.

4.1.1 Distance Ratio Minimization Principle

The destinations in each target destination subset $\tilde{M}_{ij}^x$ are in a similar direction or close to each other whereas destinations in different subsets are in much different directions. Therefore, we select a receiving node for each $\tilde{M}_{ij}^x$. Assume there are K_j elements in $\tilde{M}_{ij}^x$, which is denoted $\tilde{M}_{ij}^x=\left\{\hat{M}_1^{ij},...,\hat{M}_t^{ij},...,\hat{M}_{K_j}^{ij}\right\}$, where $\hat{M}_t^{ij}$ is a multicast destination (receiver). To select the receiving node for $\tilde{M}_{ij}^x$, we consider two factors. First, we should select the node which makes large advance towards destinations in $\tilde{M}_{ij}^x$. Second, as shown in (3), the maximum transmission power of the selected node is quite limited if the node gets too close to BS. Accordingly, the receiving node needs to stay away from the BS, which is similar to the unicast routing in [7].

To jointly consider the above two factors, we design a distance ratio minimization (DRM) principle. The ratio here means the distance ratio of the candidate-to-destination link to the candidate-to-base-station link, where the candidate is one of the nodes in $\mathbf{N}_i$. We denote the distance between candidate u and destination $\hat{M}_t^{ij}$ (an element of $\tilde{M}_{ij}^x$), $1\leqslant t\leqslant K_j$ $d_{u,t}$, and we denote the distance between u and BS $d_{u,BS}$. The final receiving node for $\tilde{M}_{ij}^x$, denoted r_{ij}^x, can be selected from all candidates by

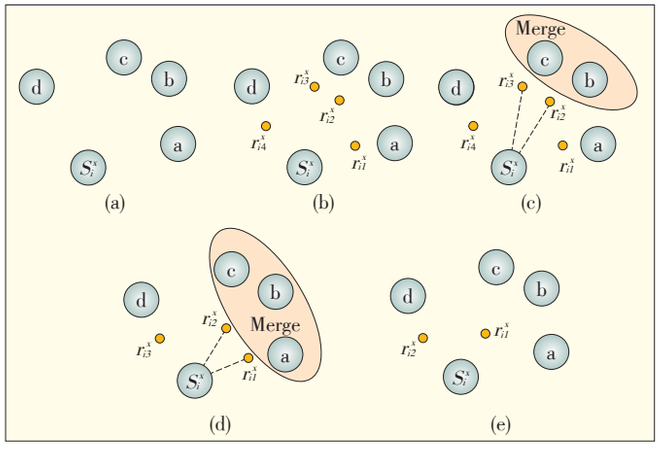
$$r_{ij}^x=\arg\min_{u\in N_i}\left\{\frac{1}{K_j}\sum_{t=1}^{K_j}\frac{d_{u,t}}{d_{u,BS}}\right\} \tag{9}$$

As in (9), we choose the node whose average distance ration over all destinations in $\tilde{M}_{ij}^x$ is the minimum. Thus, the routes will not only advance towards destinations but also stay away from BS.

4.1.2 Angle-Threshold Based Merging Principle

In the last section, we described how to select a receiving node for each $\tilde{M}_{ij}^x$ but did not discuss how to group all the destinations corresponding to s_i^x into J_i subsets, namely, $\tilde{M}_{ij}^x$, $1\leqslant j\leqslant J_i$, which form the final target destination set $\tilde{\mathbf{M}}_i^x$. Destination nodes in different subsets are in much different directions for routes whereas nodes in the same subset are in the similar direction. In particular, we use a specifically designed metric hop-angle to evaluate the difference in directions between any two different destination subsets $\tilde{M}_{ip}^x$ and $\tilde{M}_{iq}^x$. The term "hop-angle" here means the angle $\angle r_{ip}^x-s_i^x-r_{iq}^x$, where r_{ip}^x and r_{iq}^x is the receiving node for $\tilde{M}_{ip}^x$ and $\tilde{M}_{iq}^x$, respectively. Then, we set an angle threshold θ_{th} to determine whether two subsets should be merged into a set or not. The two node groups(subsets) $\tilde{M}_{ip}^x$ and $\tilde{M}_{iq}^x$ will be merged together only when $\angle r_{ip}^x-s_i^x-r_{iq}^x<\theta_{\text{th}}$. If the receiving nodes for $\tilde{M}_{ip}^x$ and $\tilde{M}_{iq}^x$ are the same; that is, $r_{ip}^x=r_{iq}^x$, the hop-angle is zero and the two subsets merge. By iteratively applying this merging principle, we can finally group all the destination nodes corresponding to s_i^x into J subsets, which are called target destination subset. Then, we can obtain the target destination set $\tilde{\mathbf{M}}_i^x$.

Fig. 3 shows a merging process example. There are four destinations a, b, c, and d, corresponding to s_i^x (Fig. 3a). We select a receiving node for each destination according to the DRM principle. These receiving nodes are denoted r_{i1}^x, r_{i2}^x, r_{i3}^x and r_{i4}^x (Fig. 3b). We sort the six hop-angles formed by line segment $s_i^x-r_{ip}^x,p=1,2,3,4$ and line segment $s_i^x-r_{iq}^x,q=1,2,3,4,q\neq p$ in ascending order. As the minimum angle $\angle\left(r_{i2}^x-s_i^x-r_{i3}^x\right)$ is less than θ_{th}, we merge b and c into a set (Fig. 3c). Then, we select a receiving node for destination subsets $\{a\}$, $\{b,c\}$ and $\{d\}$, respectively. As the angle $\angle\left(r_{i1}^x-s_i^x-r_{i2}^x\right)$ is less than θ_{th}, we merge $\{a\}$ and $\{b,c\}$ into a set $\{a,b,c\}$ (Fig. 3d). Similarly, we select two receiving nodes r_{i1}^x and r_{i2}^x for $\{a,b,c\}$ and $\{d\}$, respectively. Because the angle $\angle\left(r_{i1}^x-s_i^x-r_{i2}^x\right)$ is larger than θ_{th}, we stop the merging process. Finally, we obtain two target destination subset, $\tilde{M}_{i1}^x=\{a,b,c\}$ and $\tilde{M}_{i2}^x=\{d\}$, which form the target destination



▲Figure 3. A merging process.

set $\tilde{\mathbf{M}}_i^x=\left\{\tilde{M}_{i1}^x,\tilde{M}_{i2}^x\right\}=\left\{\{a,b,c\},\{d\}\right\}$. The selected receiving nodes for the two subsets are r_{i1}^x and r_{i2}^x (Fig. 3e).

4.2 Power Allocation for Receiving Nodes

Once the receiving nodes for each s_i^x, $i=1,2,...,\left|\mathbf{T}_x\right|$, have been determined, they perform as the transmitting nodes of (x+1)th hop. To avoid causing intolerable interference to cellular user's transmission, the transmission power of these nodes need to satisfy (3). However, following (3), we cannot get the specific power for each node r_{ij}^x. Consequently, we need a power allocation strategy to not only limit the interference to cellular user but also lower the outage probability for routing. A multicast tree has multiple branches expanding to different destinations. The branches close to the BS or cellular user are usually the bottleneck for tree building because of their high outage probability caused by limited transmission power or strong inter-network interference. Therefore, we need to allocate more power to the transmitting nodes on these bottleneck branches. Generally, the maximum transmission rate from a transmitting node r_{ij}^x (transmitting node for the (x+1)th hop as well as receiving node for the xth hop) to its *target destination* $\hat{M}_t^{ij}$ (element of $\tilde{M}_{ij}^x$), denoted $\Re_{r_{ij}^x,\hat{M}_t^{ij}}$, can be used to represent the outage probability of the branch expanding towards this destination. Therefore, we design a power allocation strategy to maximize the minimum $\Re_{r_{ij}^x,\hat{M}_t^{ij}}$, $\forall i,\forall j\forall t$. This power allocation problem can be formulated as

$$\max_{P_{r_{ij}^x}} \min \Re_{r_{ij}^x,\hat{M}_t^{ij}}$$

$$s.t.\Re_{r_{ij}^x,\hat{M}_t^{ij}}=B\log\left(1+\frac{\alpha P_{r_{ij}^x}d_{r_{ij}^x,\hat{M}_t^{ij}}^{-\eta}}{N_0+\alpha P_0\Delta_{\hat{M}_t^{ij}}^{-\eta}+\sum\limits_{\substack{(u,z)\\(u,z)\neq(i,j)}}\alpha P_{r_{uz}^x}d_{r_{uz}^x,\hat{M}_t^{ij}}^{-\eta}}\right),\ \forall t=1,2,...,K_j$$

$$\sum_i\sum_j P_{r_{ij}^x}D_{r_{ij}^x}^{-\eta}=\frac{P_0d_0^{-\eta}}{\rho_{\text{th}}} \tag{10}$$

To obtain the near optimal solution, we traverse all possible solutions by fixed step-size. Assume there are only two transmitting nodes s_1^x and s_2^x for the xth hop, two receiving nodes for s_1^x, i.e., r_{11}^x, r_{12}^x, and one receiving node for s_2^x, i.e., r_{21}^x. The value range of $P_{r_{11}^x}$ is $\left[0,\frac{P_0d_0^{-\eta}D_{r_{11}^x}^{\eta}}{\rho_{\text{th}}}\right]$. Once the value of $P_{r_{11}^x}$ is determined, the value range of $P_{r_{12}^x}$ is $\left[0,\left(\frac{P_0d_0^{-\eta}}{\rho_{\text{th}}}-P_{r_{11}^x}D_{r_{11}^x}^{-\eta}\right)D_{r_{12}^x}^{\eta}\right]$. After the values of $P_{r_{11}^x}$ and $P_{r_{12}^x}$ are both determined, the value of $P_{r_{21}^x}$ is determined, i.e., $\left(\frac{P_0d_0^{-\eta}}{\rho_{\text{th}}}-P_{r_{11}^x}D_{r_{11}^x}^{-\eta}-P_{r_{12}^x}D_{r_{12}^x}^{-\eta}\right)D_{r_{21}^x}^{\eta}$. We try to obtain the near optimal solution by several rounds. In the first round, we set $P_{r_{11}^x}=0$ and traverse the value of $P_{r_{12}^x}$ in the interval $\left[0,\left(\frac{P_0d_0^{-\eta}}{\rho_{\text{th}}}-P_{r_{11}^x}D_{r_{11}^x}^{-\eta}\right)D_{r_{12}^x}^{\eta}\right]$ by step-size $\Delta P_{r_{12}^x}$. Once the value of $P_{r_{12}^x}$ exceeds the maximum value allowed, we quit the first round and start the second round. In the second round, we set $P_{r_{11}^x}=\Delta P_{r_{11}^x}$ and traverse the value of $P_{r_{12}^x}$ by step $\Delta P_{r_{12}^x}$ as before. In the third round, we set $P_{r_{11}^x}=2\Delta P_{r_{11}^x}$ and in the next round, set $P_{r_{11}^x}=3\Delta P_{r_{11}^x}$. Repeat updating the value of $P_{r_{11}^x}$ until its value exceeds $\frac{P_0d_0^{-\eta}D_{r_{11}^x}^{\eta}}{\rho_{\text{th}}}$. Then, we choose the solution which maximizes $\min\Re_{r_{ij}^x,\hat{M}_t^{ij}}$ as the final result of power allocation.

4.3 Routing Protocol

After describing the two principles for the selection of receiving nodes and the strategy for power allocation, we now concentrate on the whole multicast routing protocol. The pseudo code for our proposed routing algorithm is listed and explained as follows.

---

00. **Initialization:**
01. $\mathbf{T}_1:=\{M_{sr}\}$; $x:=1$; set $P_{M_{sr}}$ based on Eq. (3); ! Initialize source for 1st hop
02. $\mathbf{D}:=\{M_1,M_2,\cdots,M_m\}$; $\tilde{\mathbf{M}}_1^1:=\{\{M_1\},\{M_2\},\cdots,\{M_m\}\}$; ! Destination set
03. **Routing Algorithm:**
04. while ($D\neq\varnothing$) { ! Routing until all destinations are reached
05. for ($i:=1;i\leqslant\left|\mathbf{T}_x\right|;i++$) { ! ith transmitting node s_i^x at xth hop
06. $\mathbf{N}_i:=\{u|u\in\aleph\backslash\bigcup_{z=1}^{x}T_z,\Re_{s_i^x u}\geqslant R_{\text{th}}\}$; ! Find neighbors for s_i^x
07. $\tilde{\mathbf{M}}_i^x:=\tilde{\mathbf{M}}_i^x\backslash\mathbf{N}_i$; $\mathbf{D}:=\mathbf{D}\backslash\mathbf{N}_i$; ! $\tilde{\mathbf{M}}_i^x$ includes all target destination subsets for s_i^x

08. if $\tilde{\mathbf{M}}_i^x = \varnothing$ then continue;
09. else{while (1) {
10. get s_i^x ´s receiving node r_{ij}^x towards $\tilde{M}_{ij}^x$ via Eq. (9) $\forall j$;
! $\tilde{M}_{ij}^x$ is the *j*th element (target destination subset) of $\tilde{M}_i^x$;
11. $(\gamma,\lambda) := \arg\min_{(j,t),j\neq t}\left\{\angle r_{ij}^x - s_i^x - r_{it}^x\right\}$;
$\theta_{\min} := \angle r_{i\gamma}^x - s_i^x - r_{i\lambda}^x$;
! min. hop angle of s_i^x . $\theta_{\min}$ is set to ∞ if $|\tilde{\mathbf{M}}_i| < 2$.
12. if $\theta_{\min} < \theta_{\text{th}}$ then
$\tilde{\mathbf{M}}_i^x := \left(\tilde{\mathbf{M}}_i^x \backslash \left\{\tilde{M}_{i\gamma}^x, \tilde{M}_{i\lambda}^x\right\}\right) \cup \left\{\tilde{M}_{i\gamma}^x \cup \tilde{M}_{i\lambda}^x\right\}$;
! Merge destination subsets
13. else { $r_i^x := \{r_{ij}^x | j = 1,2,...,\left|\tilde{\mathbf{M}}_i^x\right|\}$; Update $\mathbf{H}_x$; break; } } }
! Finalize all receiving nodes associated with s_i^x , update hop set $\mathbf{H}_x$;
14. $\Psi_i^{x+1} := \left((r_{i1}^x, \tilde{M}_{i1}^x), (r_{i2}^x, \tilde{M}_{i2}^x), \cdots, (r_{iJ}^x, \tilde{M}_{iJ}^x)\right)$, where $J = \left|\tilde{\mathbf{M}}_i^x\right|$; }
! Map next - hop transmitting nodes & their destination subset
15. $\mathbf{T}_{x+1} := \bigcup_{i=1}^{|T_x|} \mathbf{r}_i^x$; Denote s_i^{x+1} the *i*th element of $\mathbf{T}_{x+1}$;
! Update (*x*+1)-th hop´s transmitting node set
16. construct $\tilde{\mathbf{M}}_i^{x+1}$ based on the mapping given by Ψ_i^{x+1} for all *i*;
17. allocate power based on $\mathbf{T}_{x+1}$ & Ψ^{x+1} by using the algorithm in Section 4.2;
18. $\mathbf{D} := \mathbf{D} \backslash \mathbf{T}_{x+1}$; ! Update the overall destination set
19. $x := x + 1$; } ! Next hop
20. Construct multicast tree κ based on all $\mathbf{T}_x$ and $\mathbf{H}_x$.

Lines 00-02 start the routing by initializing the variables for first hop. Lines 04-19 form a loop that ends until all the multicast destinations (receivers) have correctly received packets. Lines 05-14 also form a loop which selects the receiving nodes for each s_i^x by the two principles given in section 4.1. Line 06 finds the neighbor set for s_i^x and line 07 excludes destinations which are within the transmission range of s_i^x from $\tilde{\mathbf{M}}_i^x$ and unreached destination set $\mathbf{D}$. Line 08 identifies the case where we start the selection process for s_{i+1}^x . Lines 09-13 form a loop which keeps updating $\tilde{\mathbf{M}}_i^x$ until $\theta_{\min} \geq \theta_{\text{th}}$ and then set $\mathbf{r}_i^x$, namely, the set of all receiving nodes for s_i^x . Line 14 records the mapping between each r_{ij}^x and $\tilde{M}_{ij}^x$. Line 15 updates (*x*+1) th hop´s transmitting node set and line 16 constructs $\tilde{\mathbf{M}}_i^{x+1}$ based on the mapping given by line 14. Power allocation for receiving nodes is given by line 17. Lines 18-19 start the routing for (*x*+1)-th hop. Finally, the completed multicast tree κ is built by line 20.

5 Algorithm Analyses

5.1 Computation Complexity

In this section, we analyze the computation complexity of our proposed algorithm and compare it with the complexity of two baseline algorithms in [17] and [19]. We use the same power-allocation strategy in the two baseline algorithms. In our proposed routing algorithm, described in section 6.1., we only discuss the complexity of selecting receiving nodes for each s_i^x .

Assume there are *n* nodes in neighbor set $\mathbf{N}_i$ and *D* destination nodes in $\tilde{\mathbf{M}}_i^x$. Assume $\tilde{\mathbf{M}}_i^x$ has been partitioned into J_i subsets in a step, i.e., $M_{i1}^x, M_{i2}^x, ..., M_{iJ_i}^x$. For each subset, we need to check all of the *n* nodes in the neighbor set to determine r_{ij}^x according to (9). Thus, we need $n*J_i$ comparisons to find nodes for J_i subsets. In the worst case scenario, the selected node for each subset is different and we obtain J_i selected nodes. Then, some subsets can be merged according to the principle in section 4.1. To determine the minimum hop angle from angles formed by J_i selected nodes, we need $C_{J_i}^2 = \frac{J_i(J_i-1)}{2}$ comparisons. If the minimum hop angle is less than θ_{th} , we merge the corresponding two subsets and get a new partition result $M_{i1}^x, M_{i2}^x, ..., M_{i(J_i-1)}^x$. Therefore, for the case with J_i subsets, $n*J_i + \frac{J_i(J_i-1)}{2} = \frac{1}{2}J_i^2 + \left(n-\frac{1}{2}\right)J_i$ comparisons are needed to get a new partition result. The maximum value of J_i is obtained as follows. If $D < n$, the number of subsets will be, in the worst case scenario, size *D*. If $D > n$, the number of subsets will be *n* at most. Thus, $0 < J_i \leq \min(D,n)$. In the worst case scenario, we need to merge exactly two subsets in each iteration. Then, we need $\sum_{J_i=2}^{\min(D,n)}\left[\frac{1}{2}J_i^2 + \left(n-\frac{1}{2}\right)J_i\right]$ comparisons in the worst case scenario. If $D < n$, the computation complexity is $O\left(D^3 + nD^2\right)$. If $D > n$, the computation complexity is $O\left(n^3\right)$. From [17] we know that the computation complexity of PBM algorithm is $O\left(2^n\right)$ and the computation complexity of GMR algorithm in the worst case scenario is $O\left(Dn\min(D,n)^3\right)$. Compared with these two baseline algorithms, our proposed algorithm is not complex.

5.2 Outage Probability

For a dynamic changing multicast tree, it is difficult to analyze the outage probability for routing because the routing nodes and transmission power are both uncertain. In this section, we analyze the outage probability for a hop. For a transmitting node *i* in $\mathbf{T}_x$, the maximum transmission rate from

node i to its receiving node j, denoted $\Re_{i,j}$, is given by (4). If we set $\Re_{i,j}=R_{\text{th}}$, we will get a closed curve. Nodes inside the region enclosed by the curve form the neighbor set of i. If there are no actual nodes in the region, the routing breaks off. We denote the region σ and the acreage of σ S_σ, which is hard to calculate. Assume U D2D nodes are evenly distributed in a sector area with acreage S_0. Then, the probability that there are no actual nodes in region σ, i.e., the probability for routing outage, is

$$P_{out}=\left(1-\frac{S_\sigma}{S_0}\right)^U \tag{11}$$

From (11) we know that the outage probability will be lower if the region σ or the amount of D2D nodes is larger. The acreage of σ is mainly determined by the transmission power and the strength of interference. Low transmission power or strong interference will reduce the acreage of σ, which increases the outage probability.

5.3 The Impact of Angle Threshold

The angle threshold is a very important for tree-building. It can determine whether two subsets should be merged or not. **Fig. 4** is an example of different multicast routes resulted from different angle thresholds with the same network topology. The threshold ranges from 0° to 180°. When the threshold is close to 0°, the multicast tree expands into multiple branches earlier (Fig. 4a). If the threshold is close to 180°, the multiple branches will not be generated until the rest destinations are too far from each other (Fig. 4b).

6 Simulation Evaluations

6.1 Two Modified Baseline Algorithms

In this section, we describe the two baseline algorithms used in the following simulation. One is the GMR routing algorithm in [17], the other is the PBM routing algorithm in [19]. For the selection of receiving nodes for s_i^x, the original two algorithms both only consider the number of receiving nodes and further advance to destinations. However, the GMR algorithm involves a merging process of destinations like our proposed algorithm whereas the PBM algorithm traverses all the possible subsets of $\mathbf{N}_i$ to select one as the set of receiving nodes. As the original two algorithms do not take into account the power constraint in (3), they cannot be directly used in the D2D network. Therefore, we modify the two algorithms to make them suitable for our simulation scenario. The two modified algorithms are called modified-GMR (M-GMR) routing algorithm and modified-PBM (M-PBM) routing algorithm. The only modification we make is to introduce the same allocation strategy described in section 4.2 to allocate power to each r_{ij}^x selected by the two original algorithms.

6.2 Simulation Settings

In the simulation, we consider a sector area with a central angle equal to $2\pi/3$ and radius 500 m, as illustrated in Fig. 4. The transmission power P_0 used by cellular user is 23 dBm and the threshold ρ_{th} of SIR at BS is 8 dB. The path-loss exponent η is 3. The constant α is set such that the signal-to-noise ratio after 500 m transmissions is 0 dB. The positions of cellular user, D2D source node and D2D destination nodes are shown in Fig. 4. But other D2D nodes are generated randomly in the sector and the topology follows uniform distribution. The following simulation results are obtained by averaging over 1000 randomly generated topologies.

6.3 Simulation Results

Fig. 5a shows the outage probability versus the number U of D2D nodes. As described in [19], the PBM algorithm has a parameter λ ranging from 0 to 1, which also exists in the M-PBM algorithm. We have thus run the same routing task 1000

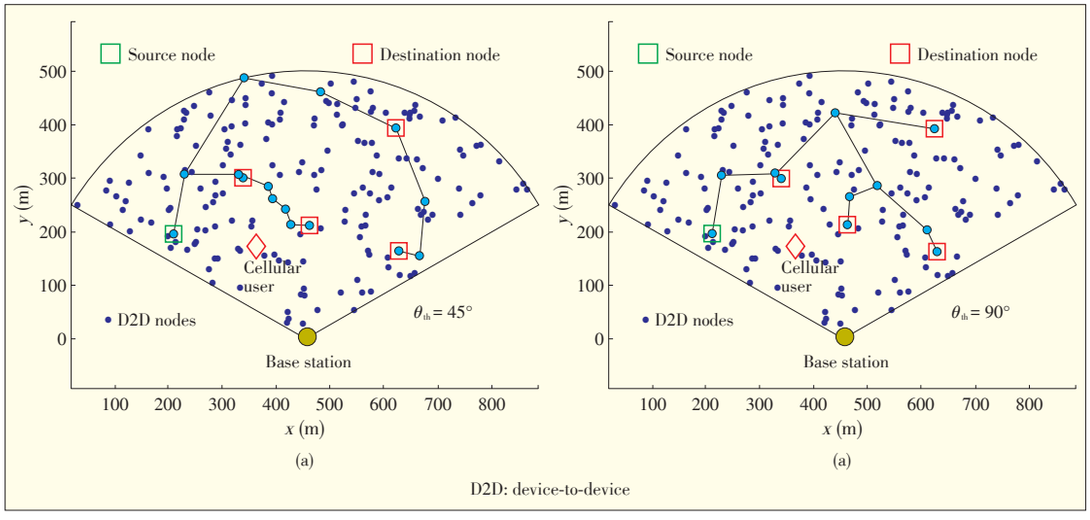


Figure 4. Routing examples with different angle thresholds: (a) $\theta_{\text{th}}=45$ and (b) $\theta_{\text{th}}=90$.

times. Only the best one $\lambda = 0.8$ is included in Fig. 5. Here, we set $R_{th} = 2Mb/s$ and $\theta_{th} = 110^{\circ}$. As in Fig 5a, M-GMR and M-PBM have much higher outage probability than our proposed algorithm because the routes generated by our proposed algorithm do not go near the BS and region with strong interference whereas the others only consider providing advance to destinations. We can also observe from Fig. 5a that the probability of all three algorithms decreases as U increases. This phenomenon verifies the outage probability analysis in section 5.2.

Fig. 5b shows the average hop-count result, where we set $\lambda = 0.8$, $R_{th} = 2Mb/s$ and $\theta_{th} = 110^{\circ}$. The average hop-count of our proposed one is the least among three algorithms (Fig. 5b). Meanwhile, the performance of M-GMR and M-PBM are getting closer when U becomes larger. This is because they both concentrate on finding nodes providing largest advance to destinations. Thus, the two algorithms are inclined to choose the same routing path when U is large enough.

Fig. 6a shows the outage probability versus the angle threshold θ_{th}, where we set $R_{th} = 2Mb/s$ and $U = 200$. The outage probability is high when the threshold is small because we generate too many target destination subsets and corresponding receiving nodes. As in (3), the transmission power and transmission range of each node will be quite limited, which will increase the outage probability. The outage probability decreases when θ_{th} increases and trends towards a stable value. This is because we merge all destinations into one target subset and the multicast tree has hardly any expanded branches. Thus, the outage probability is mainly decided by the number of D2D nodes in the network.

Fig. 6b shows the average hop-count versus the angle threshold θ_{th}. The small threshold leads to more hops because we generate too many next-hop nodes. Their transmission ranges are quite limited, as discussed in the last paragraph. Thus, the average hop-count is high. The hop-count decreases as θ_{th} increases. When θ_{th} is larger than 85°, the hop-count increas-

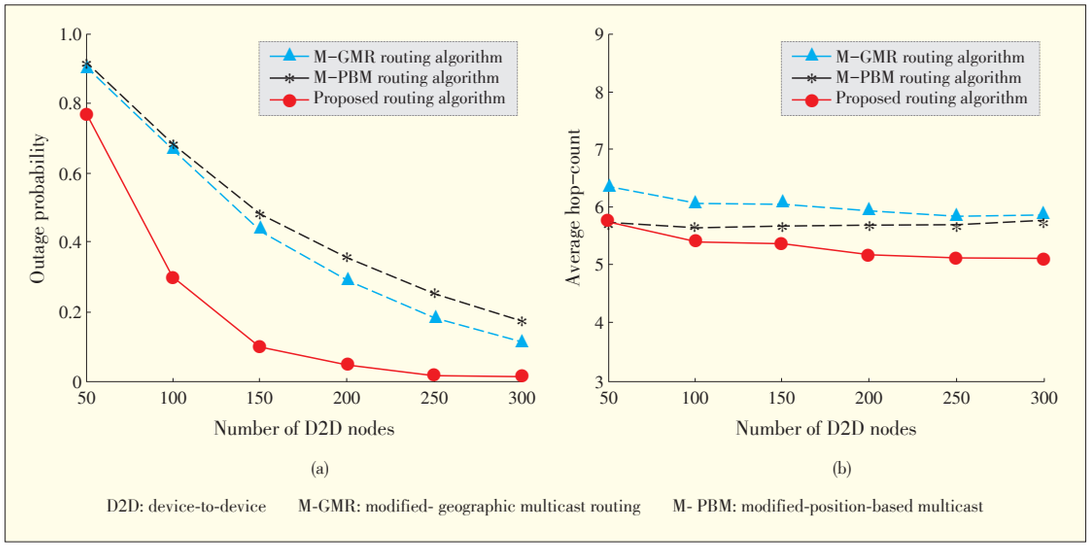


Figure 5. Performance comparisons for three routing algorithms: (a) Outage probability versus the number U of D2D nodes, and (b) Average hop-count versus the number U of D2D nodes.

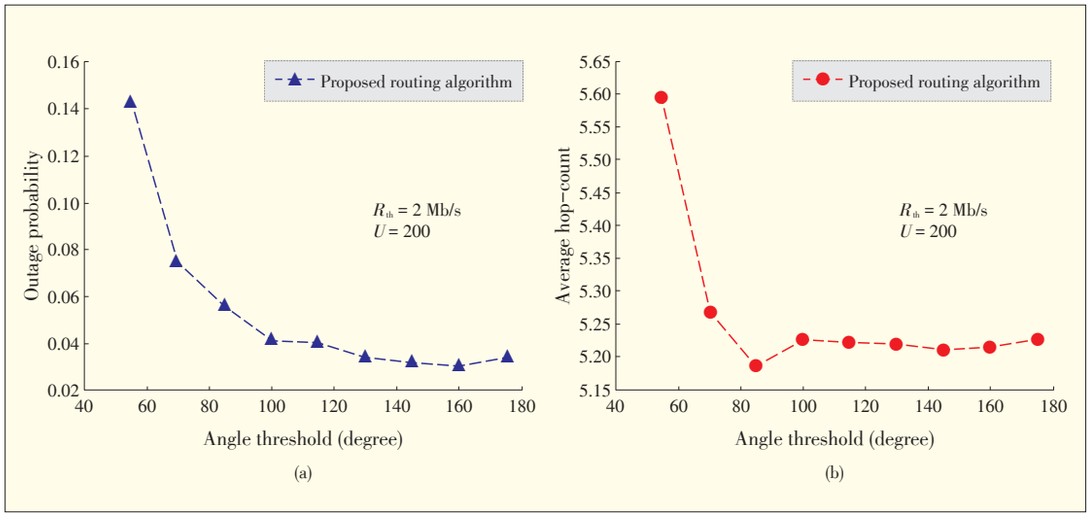


Figure 6. Performance evaluations for our proposed algorithm: (a) Outage probability versus angle threshold, and (b) Average hop-count versus angle threshold.

es again and trends to a stable value. Therefore, 85° is the best threshold for the specific positions of cellular user, D2D source node and D2D destination nodes given in Fig. 4.

7 Conclusion

In this paper, we proposed an angle-based interference-aware routing algorithm for multicast over wireless D2D networks. Our proposed algorithm aims to lower the outage probability and minimize average hop-count. By utilizing the DRM principle and the angle-threshold based merging principle, the packets can effectively progress towards destinations. Our proposed algorithm has low computational complexity. Simulation results show that our proposed algorithm outperforms the given baseline schemes in terms of outage probability and average hop count.

References

[1] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Commun. Mag.*, vol. 47, no. 12, pp. 42–49, Dec. 2009. doi: 10.1109/MCOM.2009.5350367.

[2] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklos, and Z. Turanyi, "Design aspect of network assisted device-to-device communications," *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 170–177, Mar. 2012. doi: 10.1109/MCOM.2012.6163598.

[3] H. Min, J. Lee, S. Park, and D. Hong, "Capacity enhancement using an interference limited area for Device-to-Device uplink underlaying cellular networks," *IEEE Trans. Wireless. Commun.*, vol. 10, no. 12, pp. 3995–4000, Dec. 2011. doi: 10.1109/TWC.2011.100611.101684.

[4] H. Wang and X. Chu, "Distance-constrained resource-sharing criteria for device-to-device communications underlaying cellular networks," *Electron. Lett.*, vol. 48, no. 9, pp. 528–530, Apr. 2012. doi: 10.1049/el.2012.0451.

[5] H. Min, W. Seo, J. Lee, S. Park, and D. Hong, "Reliability improvement using receive mode selection in the device-to-device uplink period underlaying cellular networks," *IEEE Trans. Wireless. Commun.*, vol. 10, no. 2, pp. 413–418, Feb. 2011. doi: 10.1109/TWC.2011.122010.100963.

[6] Y. Cheng, Y. Gu, and X. Lin, "Combined power control and link selection in device-to-device enabled cellular systems," *IET Commun.*, vol. 7, no. 12, pp. 1221–1230, Aug. 2013. doi: 10.1049/iet-com.2012.0769.

[7] P. Ren, Q. Du, L. Sun, "Interference-aware routing for hop-count minimization in wireless D2D networks," in *IEEE/CIC ICCC 2013 Workshop Internet of Things*, Xi'an, China, 2013, pp. 65–70. doi:10.1016/j.adhoc.2008.02.004.

[8] J. Wang, D. Zhu, C. Zhao, J. C. F. Li, and M. Lei, "Resource sharing of underlaying device-to-device and uplink cellular communications," *IEEE Commun. Lett.*, vol. 17, no. 6, pp. 1148–1151, Jun. 2013. doi: 10.1109/LCOMM.2013.042313.130239.

[9] C.-H. Yu, K. Doppler, C. B. Ribeiro, and O. Tirkkonen, "Resource sharing optimization for device-to-device communication underlaying cellular networks," *IEEE Trans. Wireless. Commun.*, vol. 10, no. 8, pp. 2752–2763, Aug. 2011. doi: 10.1109/TWC.2011.060811.102120.

[10] P. Pahlevani, M. Hundebøll, M. V. Pedersen, D. Lucani, H. Charaf, F. H. P. Fitzek, H. Bagheri, and M. Katz, "Novel concepts for device-to-device communication using network coding," *IEEE Commun. Mag.*, vol.52, no.4, pp.32–39, Apr. 2014. doi: 10.1109/MCOM.2014.6807944.

[11] J. Luo, D. Ye, L. Xue, and M. Fan, "A survey of multicast routing protocols for mobile Ad-Hoc networks," *IEEE Commun. Surveys. Tutorials*, vol. 11, no. 1, pp. 78–91, Mar. 2009.

[12] D. D. Le, M. Molnár, and J. Palaysi, "Multicast routing in WDM networks without splitters," *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 158–167, Jul. 2014. doi: 10.1109/MCOM.2014.6852098.

[13] C. Gao, Y. Shi, Y. T. Hou, H. D. Sherali, and H. Zhou, "Multicast communications in multi-hop cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 784–793, Apr. 2011. doi: 10.1109/JSAC.2011.110410.

[14] H. M. Almasaeid, T. H. Jawadwala, and A. E. Kamal, "On-demand multicast routing in cognitive radio mesh networks," in *IEEE GLOBECOM 2010*, Miami, USA, pp.1–5. doi: 10.1109/GLOCOM.2010.5683665.

[15] Z. Chen, D. Jiang, Z. Xu, Y. Han, H. Xu, and P. Zhang, "A multicast routing algorithm in cognitive ad hoc networks," in *ICCP 2010*, Lijiang, China, pp. 284–288.

[16] S. Wu and K. S. Candan, "GMP: distributed geographic multicast routing in wireless sensor networks," in *ICDCS 2006*, Lisboa, Portugal, pp. 1–9. doi: 10.1109/ICDCS.2006.44.

[17] J. A. Sanchez, P. M. Ruiz, I. Stojmenovic, "GMR: geographic multicast routing for wireless sensor networks," in *SECON 2006*, Reston, USA, pp. 20–29. doi: 10.1109/SAHCN.2006.288405.

[18] J. Cha, J. Jeon, J. Kim, and Y. Kwon, "Location-based multicast routing algorithms for wireless sensor networks in presence of interferences," in *ICCS*, Singapore, 2010, pp. 41–45. doi: 10.1109/ICCS.2010.5686104.

[19] M. Mauve, H. Füβler, J. Widmer, T. Lang, "Position-based multicast routing for mobile ad-hoc networks," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 7, no.3, pp. 53–55, Jul. 2003. doi: 10.1145/961268.961288.

Manuscript received: 2014-08-25

Biographies

Qian Xu (xq1216@stu.xjtu.edu.cn) received her BS degree in information engineering from Xi'an Jiaotong University, China, in 2014. She is currently working towards the PhD degree in communication and information system at the same university. Her research interests include D2D networks and wireless communications.

Pinyi Ren (pyren@mail.xjtu.edu.cn) received his BS, MS and PhD degrees from Xi'an Jiaotong University, China. He is currently a prefessor and the department head of Information and Communications Engineering Department, Xi'an Jiaotong University. His current research interests include cognitive radio networks, MIMO systems, game theory in wireless communications, wireless relay, routing, and signal detection. He has published more than 80 technical papers in international Journals and conferences. He received the Best Letter Award of IEICE Communications Society in 2010. He has more than 10 authorized Chinese Patents. Prof. Ren serves as an editor for the *Journal of Xi'an Jiaotong University*, and served as the leading guest editors for the special issue of *Mobile Networks and Applications* on "Distributed Wireless Networks and Services" and that of *Journal of Electronics* on "Cognitive Radio".

Qinghe Du (duqinghe@mail.xjtu.edu.cn) received his BS and MS degrees both from Xi'an Jiaotong University, China, and his PhD degree from Texas A&M University, USA. He is currently an assistant professor of Information and Communications Engineering Department, Xi'an Jiaotong University, China. His research interests include mobile wireless communications and networking with emphasis on mobile multicast, statistical QoS provisioning, and cognitive radio networks. He has published more than 30 technical papers. He received the Best Paper Award in IEEE GLOBECOM 2007. He serves as an associate editor of *IEEE Communications Letters*.

Gang Wu (wu.gang26@zte.com.cn) received his PhD degree from Southeast University, China in 2002. He is currently a R&D expert at ZTE Corporation. His R&D interests include wireless communication system and terminal chipset with emphasis on algorithm, and system design and standardization. He has published more than 20 technical papers, 40 international patents and 10 3GPP standardization proposals. He is leading a research task of National Science and Technology Major Project of China.

Qiang Li (li.qiang8@zte.com.cn) received his PhD degree in communications & information systems from Southeast University, China. He is currently the head of Algorithm Design Department of ZTE Corporation, and responsible for the research and design of telecommunication baseband algorithms.

Li Sun (lisun@mail.xjtu.edu.cn) received his BS and PhD degrees in Information Engineering from Xi'an Jiaotong University, China, in 2006 and 2011. He is currently an assistant professor at the School of Electronic and Information Engineering, Xi'an Jiaotong University, China. His research interests include cooperative relaying networks and wireless communications.

DOI: 10.3969/j. issn. 1673-5188. 2014. 04. 006
http://www.cnki.net/kcms/detail/34.1294.TN.20141224.0955.001.html, published online December 24, 2014

Digital Signal Processing for Optical Access Networks

Jianjun Yu
(Optics Labs, ZTE (TX) Inc., NJ 07960, USA)

Abstract

In this paper, we investigate advanced digital signal processing (DSP) at the transmitter and receiver side for signal pre-equalization and post-equalization in order to improve spectrum efficiency (SE) and transmission distance in an optical access network. A novel DSP scheme for this optical super-Nyquist filtering 9 Quadrature Amplitude Modulation (9-QAM) like signals based on multi-modulus equalization without post filtering is proposed. This scheme recovers the Nyquist filtered Quadrature Phase-Shift Keying (QPSK) signal to a 9-QAM-like one. With this technique, SE can be increased to 4 b/s/Hz for QPSK signals. A novel digital super-Nyquist signal generation scheme is also proposed to further suppress the Nyquist signal bandwidth and reduce channel crosstalk without the need for optical pre-filtering. Only optical couplers are needed for super-Nyquist wavelength-division-multiplexing (WDM) channel multiplexing. We extend the DSP for short-haul optical transmission networks by using high-order QAMs. We propose a high-speed Carrierless Amplitude/Phase-64 QAM (CAP-64 QAM) system using directly modulated laser (DML) based on direct detection and digital equalization. Decision-directed least mean square is used to equalize the CAP-64QAM. Using this scheme, we generate and transmit up to 60 Gbit/s CAP-64QAM over 20 km standard single-mode fiber based on the DML and direct detection. Finally, several key problems are solved for real time orthogonal-frequency-division-multiplexing (OFDM) signal transmission and processing. With coherent detection, up to 100 Gbit/s 16 QAM-OFDM real-time transmission is possible.

Keywords

digital signal processing; high spectrum efficiency; super-Nyquist; coherent optical transmission

This work is supported by the High Technology Research and Development Program of China ("863" Program) under Grant No. 2012AA011303 and 2013AA010501 and National Nature Science Foundation of China under Grant No. 61325002.

1 Introduction

In long-haul backbone networks and short-haul access networks, bandwidth demand has increased by 30% to 60% annually due to the rapid development of cloud computing, social media, and mobile data services. The trend of increasing service bandwidth requires a lower cost per bit; thus, high-speed optical transmission interfaces and high spectrum efficiency (SE) technologies have become more important. For example, a successful solution for the 100 Gbit/s long-haul system is to combine the single-carrier Polarization-Division Multiplexing Quadrature Phase Shift Keying (PDM-QPSK) modulation format and the digital signal processing (DSP) based coherent detection. Transmission technologies providing high SE have also been widely investigated. These technologies fall into two main categories: those that reduce spectrum bandwidth and those that increase the modulation order. The former uses spectrum shaping technologies of optical or electrical domain filtering, which are also called Nyquist or Super-Nyquist technologies. The latter uses higher-order modulation formats, such as 32 Quadrature Amplitude Modulation (32-QAM), 64-QAM, or even the QAMs that have a higher order [1]–[26]. However, the two technologies depend on the advanced DSP of transmitters or receivers, and there are various restrictions such as high sensitivity to laser frequency offset, phase noise, and inter-symbol interference (ISI). There is also a variety of intra-channel and inter-channel impairments.

Higher-order modulation is the simplest solution to high SE. This solution, however, causes impairment, requires high receiver sensitivity and only covers a short distance. Compared with Quadrature Phase-Shift Keying (QPSK) signals, 16-QAM signals require 6 dB higher Optical Signal to Noise Ratio (OSNR), and the OSNR requirement increases exponentially with the increase of constellation points. As for optical fiber transmission, the nonlinearity of optical fibers restricts the launch power and significantly limits the OSNR. In addition, the Euclidean distance of constellation points in high SE modulation is shorter, and there is even lower tolerance for the nonlinearity of optical fibers. In the latest experiment [27], the OSNR at BER = 10^{-3} for 16-QAM has a penalty of 8 dB whereas for QPSK only a penalty of 1 dB. Therefore, it is very important to study the advanced DSP algorithms for higher-order QAM. Unlike QPSK, higher-order QAM needs new solutions to the polarization de-multiplexing, frequency offset and phase recovery of signals. In field tests, dual-carrier 16-QAM signals are transmitted at 512 Gbit/s for a maximum distance of 734 km in dispersion-compensated fibers, while at the same time non-return-to-zero (NRZ) signals are co-propagated at 10 Gbit/s within a 200 GHz bandwidth [28]. These results show that it is challenging to use 16-QAM and 64-QAM to increase the SE. Due to smaller electrical bandwidth at the same bit rate, high SE transmission based on higher-order QAM can bring a better system performance in a short-haul optical network that re-

quires a higher OSNR.

With the development of high SE coherent detection and DSP, spectrum shaping technologies of Nyquist Wavelength-Division Multiplexing (N-WDM) and Super-Nyquist WDM (SN-WDM) have become hot topics in the field of 100 Gbit/s long-haul transmission. Current research shows that QPSK modulation formats provide the best balance between SE and transmission distance. Therefore, using spectrum shaping technologies to implement N-WDM or SN-WDM for increasing the SE of the PDM-QPSK system is a promising and highly efficient solution for long-haul large-capacity optical transmission networks [11]–[21]. However, filter shaping and DSP may cause ISI, inter-channel crosstalk and noise amplification, all of which seriously affect system performance [11]–[16]. When a linear equalization algorithm, such as the Constant Modulus Algorithm (CMA), is used, high-frequency noise and inter-channel crosstalk in the signal spectrum is enhanced. To compensate for the impairment, extra processing is needed for noise suppression and multi-symbol detection decision. In [11]–[16], a delay-and-add post-filter is used to suppress the enhanced noise. In addition, a 1-bit Maximum Likelihood Sequence Estimation (MLSE) is introduced to overcome the ISI impairment. However, some problems still exist.

1) Although the post-filter combined with Constant Modulus Equalization (CMEQ) algorithm has been widely applied to 100 Gb/s and higher optical Nyquist and Super-Nyquist transmission [14]–[16], some DSP modules, including carrier recovery, may still be affected by noise and crosstalk.
2) A Wavelength Selective Switch (WSS), which is costly and difficult to integrate, cannot be easily integrated with a traditional optical transceiver, especially in a multi-channel system.
3) The instability of the filter center window may cause serious deterioration of the system performance.

To solve these problems, we use a digital-to-analog converter (DAC) with a high sampling rate and high analog bandwidth. This DAC shapes the spectrum through a digital filter with hundreds of taps. This solution does not need any extra device and parameters can be reset easily. It can be combined with other functions (such as tilt correction) of a transmitter, and WDM channels can also be multiplexed by optical couplers.

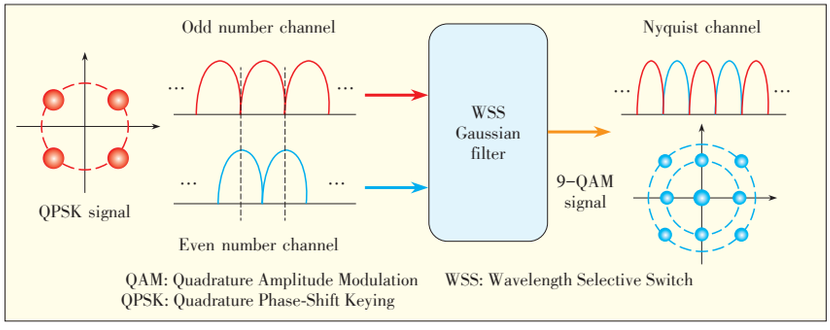
In this paper, we study DSP algorithms for high SE and long-haul transmission in optical access networks. We describe post-processing at the receiver end and pre-processing at the transmitter end in an SN-WDM system. We propose a 9-QAM-like Multi-Modulus Equalization (MMEQ) DSP scheme based on optical super-Nyquist filtering. By using the Cascaded Multi-Modulus Equalization algorithm (CMMA), this scheme directly restores QPSK signals from 9-QAM-like signals [29]–[30]. In addition, for Quadrature Duobinary (QDB) signals, the system performance of post-filter CMEQ and MMEQ schemes are compared for different filter bandwidths, carrier spacing, and transmission distances. We transmit QPSK signals with an SE of 4 b/s/Hz. We also propose a novel digital Super-Nyquist signal generation scheme that reduces Nyquist signal bandwidth and channel crosstalk without using optical pre-filtering. The spectrum of Super-Nyquist 9-QAM signals generated by using this scheme is more effectively compressed than that of regular Nyquist QPSK signals. Only an optical coupler is needed to implement WDM channel multiplexing. After the 20% soft-decision Forward Error Correction (FEC) overhead is removed, this scheme still achieves a net SE of 4 b/s/Hz. By using higher-order QAM technologies, we also apply DSP in short-haul optical transmission networks. We propose a high-speed Carrierless Amplitude/Phase-64 QAM (CAP-64-QAM) system based on Directly Modulated Laser (DML), direct detection, and digital equalization. CAP-64-QAM signal equalization is implemented with the Decision Directed Least Mean Square (DD-LMS) algorithm. By using this scheme, we generate high-speed CAP-64-QAM signals based on DML and direct-detection technologies. We transmit the signals over a 20 km Standard Single-Mode Fiber (SSMF) at a record rate of 60 Gbit/s. This paper also describes the latest research progress of the DSP-based real-time coherent system.

The remainder of this paper is organized as follows. In section 2, we describe the MMEQ-based post-transmission DSP algorithms used in the Super-Nyquist system. In section 3, we discuss a novel digital Super-Nyquist signal generation scheme. This scheme reduces channel crosstalk without using optical pre-filtering. In section 4, we show the 64-QAM signal transmission experiments based on Carrierless Amplitude/Phase (CAP) algorithms. We also discuss the DSP-based real-time coherent system in section 5, and conclude the paper in section 6.

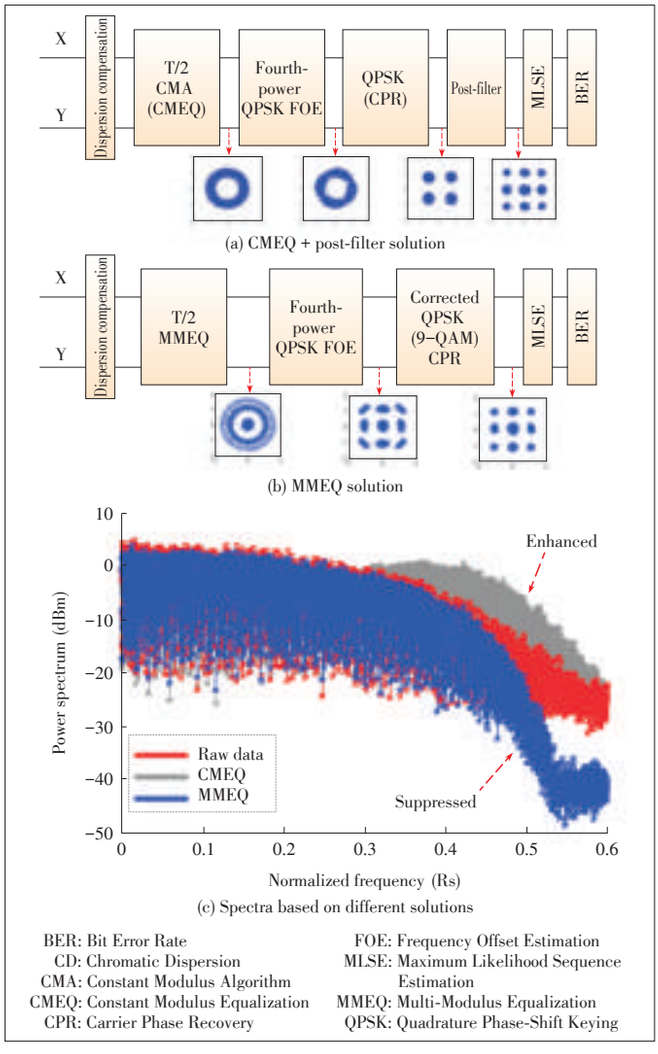
2 MMEQ-Based Post-Processing DSP Algorithms Used in Optical Super-Nyquist Channels

Optical super-Nyquist shaping can be implemented by 4-order super-Gaussian narrowband filtering, such as WSS [11]–[16]. For PDM-QPSK signals with a symbol rate of *R*s, we use a filter shaping device with 3 dB bandwidth less than or equal to *R*s in order to implement QDB spectrum shaping. Due to filtering effect, 4-point QPSK signals can be changed into 9-QAM-like signals from the viewpoint of constellation points. Compared with QPSK signals, QDB signals have a narrower spectrum and side lobe suppression. In general, conventional Nyquist signals are generated by a raised cosine function that has a bandwidth equal to symbol rate. We propose a solution to reach the limit of the Super-Nyquist SE by adopting a filter with a 3 dB bandwidth less than signal baud rate. **Fig. 1** shows the principle for the generation of WDM channels of Super-Nyquist filtering 9-QAM-like signals from QPSK signals, which is based on optical Gaussian filtering.

Fig. 2 shows the DSP module process. Figs. 2a and 2b show



▲Figure 1. Generation of WDM channels of super-Nyquist filtering 9-QAM-like signals from QPSK signals.



▲Figure 2. DSP Module Process.

the DSP module processes of two different processing schemes. In [11]–[16], the authors use the CMA and post-filterCMEQ algorithm. [29] and [30] introduce the MMEQ scheme which we have proposed recently. For the post-filter CMEQ, received signals are first restored to QPSK signals and then converted into 9-QAM-like signals by delay-and-add post-filtering to suppress noise. In the MMEQ scheme, however, we use the CMMA algorithm to restore QDB signals to 3-modulus 9-QAM-like signals and then obtain 9-QAM signals by using an improved carrier phase recovery (CPR) algorithm. The detailed DSP algorithm is discussed in [29], and Fig. 2 shows the signal constellations after the processing of each DSP module. The main advantage of using the MMEQ algorithm to process QDB filtering signals is that the frequency response of adaptive MMEQ taps has a compression effect on the high-frequency components, which, compared with CMEQ, avoids performance deterioration caused by noise and crosstalk. Fig. 2c shows the signal spectra of different schemes when a 3 dB 22 GHz QDB filter is used. Noise and crosstalk are improved, and the high-frequency components near $\pm Rs/2$ are restored by the CMEQ algorithm. However, noise and crosstalk are suppressed by the processing of the CMMA-based MMEQ algorithm. This suppressed noise is mainly the noise inside the channel near the high-frequency part and includes Amplified Spontaneous Emission (ASE) noise and ISI. Therefore, with the CMEQ-based scheme, it is necessary to add a post-filter after the CPR process in order to suppress noise and crosstalk [11]–[16]. However, some DSP modules are still affected by noise and crosstalk during CPR. Compared with the CMEQ algorithm, the MMEQ algorithm better suppresses noise and crosstalk during the initial phase of DSP, and improves system performance.

To compare the filter tolerance for noise and crosstalk of CMEQ and MMEQ algorithms, we design an 8-channel PDM-QPSK experiment with 28-Gbaud QDB filtering. The transmission rate of the system is 8 × 112 Gbit/s, and the channel spacing is 25 GHz. Single-mode fiber-28 (SMF-28) is used, and the circulating fiber loop is divided into ten 88 km spans. The average loss of each span is 18.5 dB and chromatic dispersion (CD) is 17 ps/km/nm. An EDFA is added before each 88 km fiber span to compensate for the fiber loss. In addition, a programmable WSS is introduced into the fiber loop to suppress ASE noise as an optical bandpass filter (BPF). The WSS has a 4-order Gaussian spectral feature with a 3 dB bandwidth of 2.2 nm. At the receiver end, a tunable BPFat 3 dB bandwidth of 0.34 nm is used to select the desired subchannel. Polarization- and phase-diversity homodyne coherent detection is also adopted. The External Cavity Laser (ECL) functioning as the Local Oscillator (LO) in the transmitter or receiver has a linewidth of about 100 kHz. Each Balanced Photodiode (BPD) is with 3 dB bandwidth of 42 GHz. The average input optical power of each photodiode in BPD ranges from -20 dBm to 13 dBm. The power of received signals is 3 dBm, and the power of the pre-amplified LO before an optical mixer is 20 dBm. A digital sampling oscilloscope with a sampling rate of 80 GSa/s and bandwidth of 30 GHz is used for analog-to-digital conversion (ADC). The

crosstalk of adjacent channels is suppressed after the ADC and there is no need to add an extra filter in offline processing.

The result shows that MMEQ provides a better BER than CMEQ with post-filtering because the former more effectively suppresses noise and crosstalk. When the filter bandwidth is 20.1 GHz, the OSNR corresponding to BER = 1 x 10^{-3} in the MMEQ scheme is about 16.5 dB, which improves by 1 dB compared with the post-filtering CMEQ scheme. In addition, the MMEQ scheme improves the filter tolerance for noise and crosstalk, and the maximum transmission distance of 25 GHz QDB signals can reach 2640 km. For post-filtering CMEQ, however, the maximum transmission distance at the BER below the FEC limit is about 2000 km. Therefore, compared with the post-filtering CMEQ scheme, the MMEQ scheme provides better transmission and increases transmission distance by 32% when the BER is 3.8×10^{-3}.

3 Pre-Processing Algorithm for Generation and Processing of Digital Super-Nyquist Signal

Fig. 3 shows the difference between the generation of DAC-based Super-Nyquist 9-QAM signals and that of regular Nyquist QPSK signals. For the filtering of regular Nyquist signals, only Square-Root-Raised-Cosine (SRRC) filter is needed to generate Nyquist pulses. However, when the channel spacing is less than the symbol rate, the bandwidth beyond the channel spacing may cause serious crosstalk (Fig. 3). To achieve Super-Nyquist transmission, we add a low pass filter (LPF) to generate super-Nyquist pulses. In this way, the signal spectrum is further compressed to reduce channel crosstalk. In our scheme, the LPF can be implemented by the QDB delay-and-add, and the z-transform of the related transfer function is

$$H_{\mathrm{QDB}}(z)=1+z^{-1} \quad (1)$$

Thus, the LPF can be implemented by a 2-tap FIR filter, which converts QPSK signals into 9-QAM signals [11]–[16]. By cascading QDB and SRRC filters, the Super-Nyquist digital filter in the time domain can be expressed as

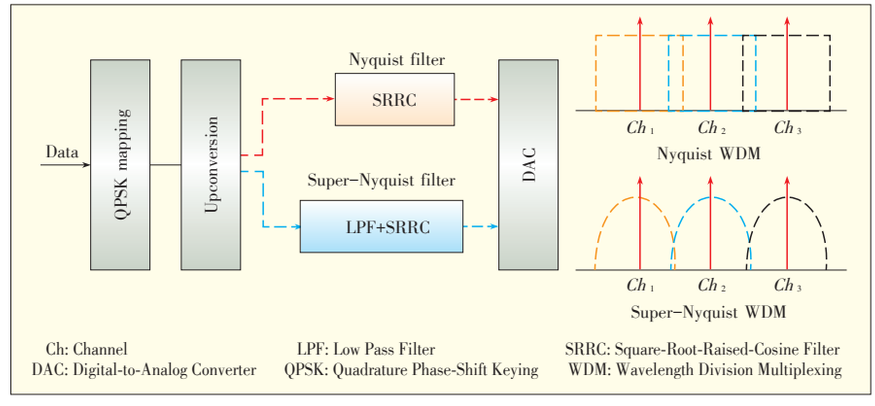
$$h_{\mathrm{SN}}(t)=h_{\mathrm{QDB}}(t)\oplus h_{\mathrm{srrc}}(t) \quad (2)$$

where $h_{\mathrm{srrc}}(t)$ is the time domain pulse response of the SRRC filter [17]–[20], and $h_{QDB}(t)$ is the impulse response of the QDB filter in (1).

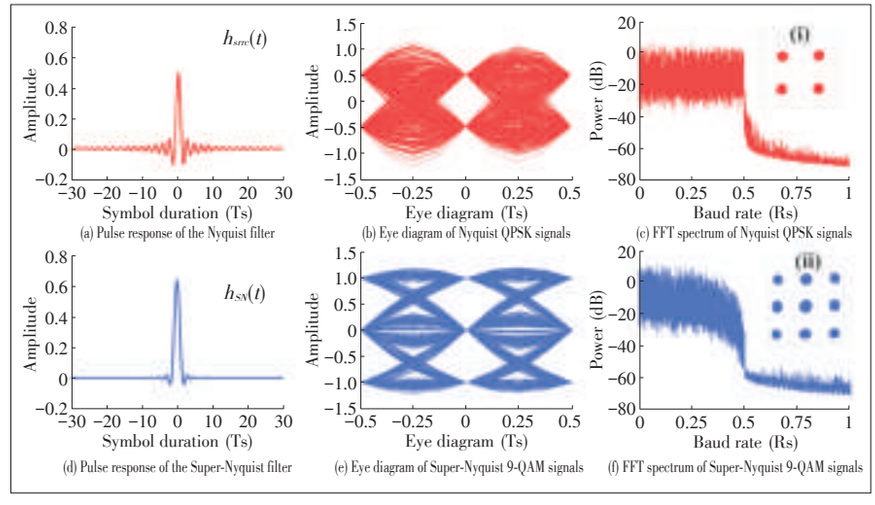
Figs. 4a and **4d** show the time domain pulse responses of an SRRC-based regular Nyquist filter and a super-Nyquist filter based on cascaded QDB and SRRC filters. In the figure, the SRRC roll-off factor is set to 0. The Super-Nyquist digital filter has a smaller resonance and faster convergence compared with the conventional Nyquist filter. **Figs. 4b** and **4e** are the eye diagrams of a conventional Nyquist QPSK 2-level baseband signal and a Super-Nyquist 9-QAM 3-level baseband signal. **Figs. 4c** and **4f** show the electrical power spectra of Nyquist QPSK and Super-Nyquist 9-QAM signals. The power spectrum of the Super-Nyquist signal is more severely compressed; its spectrum side lobe is greatly suppressed; and its 3 dB bandwidth is less than one half of the baud rate.

4 CAP-64-QAM Short-Haul Transmission Using Direct Detection and Advanced Digital Equalization Technologies

Because of the smaller electrical bandwidth at a given bit rate, high SE transmission based on higher-order QAM can bring about better system performance in a short-haul optical



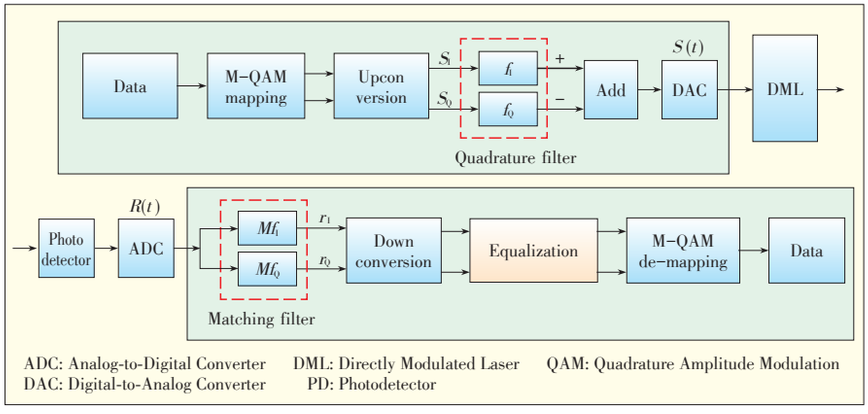
▲**Figure 3. DAC-based Nyquist and Super-Nyquist 9-QAM signal generation.**



▲**Figure 4. Comparison of Nyquist and super-Nyquist.**

network that requires a higher OSNR. On the other hand, with the rapid growth in the demand for short-haul communication bandwidth of optical links between access networks and data centers, to increase the transmission capacity is a hot topic [5], [6]. Considering cost and complexity, intensity modulation and direct detection (IM/DD) using higher-order modulation formats is a universally feasible solution [5], [6], [31]–[41]. IM/DD-based modulation technologies such as QAM-subcarrier modulation (SCM) [5], [6], pulse amplitude modulation (PAM) [31], Discrete Multi-Tone (DMT) or orthogonal frequency division multiplexing (OFDM) [32], [33], and CAP modulation [34]–[41] have been proposed.

Studies have shown that the IM/DD-based CAP structure reduces complexity and ensures good performance. It is still able to provide a high data transmission rate by using only a DML, a Vertical-Cavity Surface-Emitting Laser (VCSEL), a photoelectric device with a limited bandwidth, or other cheap components [34]–[42]. Compared with QAM-SCM [5], [6] and OFDM [32], [33], CAP does not require complex-to-real conversion in the electrical domain, complex mixers, RF sources or optical in-phase/quadrature (I/Q) modulators. CAP also eliminates the discrete Fourier transform (DFT) used during OFDM signal modulation and demodulation [40]. Various CAP-based optical communication systems have been discussed in [35]–[42]. In [38], the authors describe how multiband CAP-QAM can increase the bandwidth in short-haul communication. G. Stepniak et al. [42] propose a system based on CAP-16-QAM and CAP-64-QAM. However, the bit rates of these systems are only 2 Gbit/s and 2.1 Gbit/s, respectively. In [40], a CMMA-based digital equalizer is used to equalize the ISI of CAP-16-QAM signals, and this improves performance. However, higher-order modulation CAP systems, such as CAP-64-QAM with a rate up to tens of gigabit per second, have not been demonstrated, and the corresponding digital equalization technologies also have not been deeply investigated. Thus, we propose and experimentally demonstrate a high-speed CAP-64-QAM system based on DML, direct detection, and digital equalization technologies.



▲**Figure 5. CAP M-QAM transmitters and receivers using DML, direct detection, and digital equalization.**

Fig. 5 shows the principles of CAP M-QAM transmitters and receivers that use DML, direct detection, and digital equalization. CAP Bell Labs first proposed CAP, a multi-level multi-dimensional modulation format suitable for short-haul communications [34]–[43]. This modulation format is similar to QAM but does not require an RF source. Two-dimensional CAP can be implemented by the two orthogonal filters f_I and f_Q in Fig. 5. The original bit sequence is first mapped to the complex symbol of M-QAM, where M is the level of the QAM. In order to match the sampling rate of the shaping filter, the mapped complex symbol is then up-sampled. The sampling rate of the shaping filter is jointly determined by the data baud rate and DAC sampling rate. After the outputs of the two filters are combined, DAC processing is performed to form $S(t)$, which drives DML. The receiver uses direct detection. After the ADC processing, the signals are sent to two matching filters to separate the in-phase and quadrature components. After down-sampling, linear equalization and M-QAM demodulation, the original bit sequence is obtained.

$f_I(t)$ and $f_Q(t)$ represent a pair of quadrature matching filters, and $Mf_I(t)$ and $Mf_Q(t)$ are their corresponding shaping filters. These two pairs of filters make up a Hilbert pair between the transmitter and receiver. The two quadrature filters can be established by multiplying the SRRC of sine and cosine functions [43]. Therefore, the relationships between the matching filters are $Mf_I^n(t) = f_I^n(-t)$ and $Mf_Q^n(t) = f_Q^n(-t)$. Because of the orthogonality of the filters, in-phase and quadrature data can be obtained with the quadrature matching filters. In order to accurately recover the in-phase and quadrature data, synchronization during CAP demodulation is very important. Time errors of matching Finite Impulse Response (FIR) filters introduce serious ISI [40]–[42]. The most suitable sampling point is difficult to determine; therefore, the deviation of the sampling time point causes subsequent signals to be severely affected by ISI and I/Q crosstalk, and this results in blurring and phase rotation of constellation points.

Therefore, it is necessary to use a linear equalizer after down-sampling to process complex signals, and the original signals can be obtained by QAM decoding. In our system, both quadrature filters and matching filters are implementedby digital FIR filters, and the tap lengths are T-OFL and R-MFL, respectively. The tap length of a FIR filter determines its time domain pulse shape and frequency response [40]. Its impact on system performance is also considered in the experiment.

In the previous work, CAP signals are equalized by a two-stage equalization scheme that combines ISI equalization and phase recovery algorithms. CMA is used first for pre-convergence, and then the CMMA algorithm is used for ISI equalization. For higher-order

CAP-QAM signals, however, CMMA equalization is not ideal because the intervals in a QAM ring are generally smaller than the minimum inter-symbol interval. Previous studies have shown that DD-LMS results in better SNR than CMMA for higher-order QAM signals [22]. On the other hand, the convergence of CMMA is based on the modulus value of a symbol, and this algorithm is independent of the phase. Therefore, phase recovery is additionally needed after CMMA in order to equalize crosstalk.

We propose a novel DSP algorithm for equalizing the ISI and crosstalk in CAP-QAM signals. After CMA pre-convergence, a DD-LMS based 1-level equalizer is used to adjust the tap coefficient of the FIR filter. **Figs. 6a** and **6b** show the structure and principle of the DD-LMS algorithm. The FIR filter used for CAP signal equalization is a butterfly-configured adaptive digital filter with a $T/2$ interval. Unlike DD-LMS used in a coherent optical system, the four time-domain tap coefficients of the FIR filter are all real numbers. $Z_I(n)$ and $Z_Q(n)$ respectively indicate the in-phase and quadrature signal outputs after the Nth equalization of the filter. $D_I(n)$ and $D_Q(n)$ are the decision results of in-phase and quadrature signals. Although the in-phase and quadrature signal inputs are independent, each output is related to both inputs. The error function of DD-LMS can be expressed as

$$e_{I,Q}(n)= D_{I,Q}(n)- Z_{I,Q}(n) \quad (3)$$

where $e_I(n)$ and $e_Q(n)$ are the error functions of in-phase and quadrature signals, respectively, and the four real-value FIR filters h_{ii}, h_{iq}, h_{qi}, and h_{qq} are updated by the error function after decision:

$$h_{ii}(n)=h_{ii}(n-1)+\mu e_I(n)r_I(n) \quad (4)$$
$$h_{qi}(n)=h_{qi}(n-1)+\mu e_I(n)r_Q(n) \quad (5)$$
$$h_{iq}(n)=h_{iq}(n-1)+\mu e_Q(n)r_I(n) \quad (6)$$
$$h_{qq}(n)=h_{qq}(n-1)+\mu e_Q(n)r_Q(n) \quad (7)$$

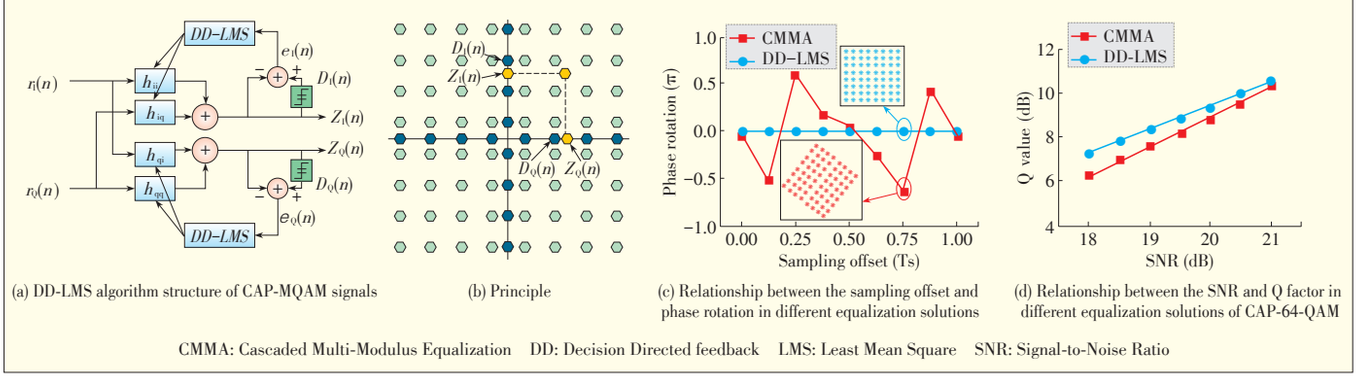
This means that the ISI and crosstalk of in-phase and quadrature signals can be eliminated.

We compared the CMMA and DD-LMS schemes by simulation. **Figs. 6c** and **6d** show the CAP-64-QAM signal equalization results. Fig. 6c shows the relationship between the sampling deviation and phase rotation in the CMMA and DD-LMS equalization schemes, and the up-sampling rate is 8 Sa/symbol. The phase rotation introduced by the clock deviation cannot be compensated by CMMA. Extra phase recovery processing is necessary after CMMA. When the DD-LMS algorithm is used, however, the phase of received signals can be correctly restored, because DD-LMS is very sensitive to phase information. Fig. 6d shows the relationship between the Q value and SNR of received signals in different equalization schemes. The results show that the Q value of CAP-64-QAM signals is higher in the DD-LMS algorithm than in the CMMA algorithm, because for CAP-64-QAM signals, the error function of DD-LMS is based on the symbol interval whereas that of CMMA is based on the ring interval. For QAM, the ring interval is smaller than the minimum symbol interval in most cases; thus, the DD-LMS algorithm performs better.

The aforementioned results verify that our proposed CAP-64-QAM system, which adopts DML, direct detection, and improved DD-LMS equalization, is very feasible.

5 DSP-Based Real-Time Coherent System

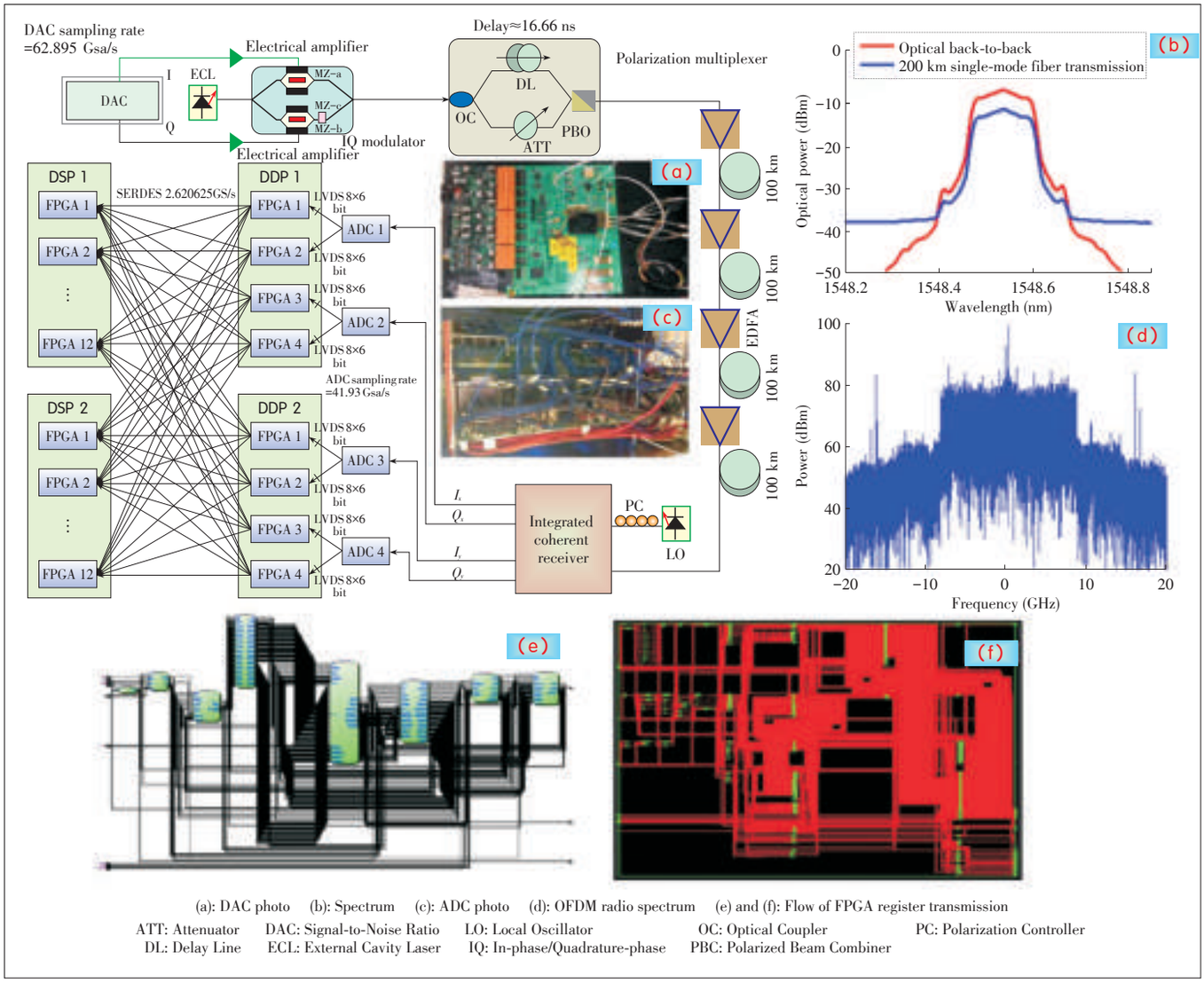
We have, for the first time, constructed a 100 Gbit/s single-band real-time coherent optical 16-QAM-OFDM transmission system [44]. The OFDM signal has a high SE and its spectral resources can be dynamically allocated. It also can effectively resist chromatic dispersion (CD) in optical fiber transmission. Therefore, it is an advanced modulation format that has been continuously studied in the industry. **Fig. 7** shows the experiment setup. The laser in the experiment has an operating wavelength of 1548.53 nm and a linewidth less than 100 kHz. After electrical amplification, 16-QAM-OFDM signals generated by DAC are used to drive the optical I/Q modulator. The DAC sampling rate is 62.895 GSa/s. In OFDM modulation, the FFT size is 1024, of which 256 subcarriers are used to carry data, 8 subcarriers are used to carry pilot signals, the first subcarrier



(a) DD-LMS algorithm structure of CAP-MQAM signals (b) Principle (c) Relationship between the sampling offset and phase rotation in different equalization solutions (d) Relationship between the SNR and Q factor in different equalization solutions of CAP-64-QAM

CMMA: Cascaded Multi-Modulus Equalization DD: Decision Directed feedback LMS: Least Mean Square SNR: Signal-to-Noise Ratio

▲**Figure 6. Simulation comparison between CMMA and DD-LMS solutions.**



(a): DAC photo (b): Spectrum (c): ADC photo (d): OFDM radio spectrum (e) and (f): Flow of FPGA register transmission

ATT: Attenuator DAC: Signal-to-Noise Ratio LO: Local Oscillator OC: Optical Coupler PC: Polarization Controller
DL: Delay Line ECL: External Cavity Laser IQ: In-phase/Quadrature-phase PBC: Polarized Beam Combiner

▲Figure 7. 100 GHz 16-QAM-OFDM real-time system.

is zero, and the other 759 subcarriers also are all set to zero. In the experiment, the DFT-spread technology is used to evenly distribute SNR and reduce the peak-to-average power ratio (PAPR) in signal subcarriers. In addition, intra-symbol frequency-domain averaging (ISFA) is used to eliminate the influence of noise in the optical channels in channel estimation. Thus, the BER of the system can be improved. After the inverse fast Fourier transform (IFFT), 24 sampling points are used as cyclic prefix. At the receiver end, the ADC sampling rate is 41.93 GSa/s, and the bandwidth is 16 GHz. DAC and ADC resolutions are 8 bits and 6 bits, respectively. The FPGA chip models are Altera EP4S100G and Xilinx 6VSX475 FPGA [44]. Fig. 7 shows the photos of DAC and ADC, optical spectrum and electrical spectrum of signals, and the flow chart of FPGA register transmission. In the absence of electronic dispersion compensation (EDC), the BER of 100 Gbit/s polarization division multiplexing 16-QAM-OFDM is less than 3.8×10^{-3} after 200 km transmission.

6 Conclusion

In this paper, we studied the high SE and long-haul transmission DSP algorithms in optical access networks. In order to improve the SE of QPSK signals, we proposed two Super-Nyquist WDM algorithms based on pre-processing at the transmitter end and post-processing at the receiver end. We also proposed and verified an MMEQ DSP scheme for optical super-Nyquist filtering 9-QAM-like signals. By using the CMMA algorithm, this scheme can directly restore QPSK signals from 9-QAM-like signals. For QDB signals, we compared the system performance of post-filter CMEQ and MMEQ schemes for different filter bandwidths, carrier spacing, and transmission dis-

tances. In addition, we proposed a novel digital super-Nyquist signal generation scheme that further compresses Nyquist signal bandwidth and reduces channel crosstalk without the need for optical pre-filtering. The spectrum of super-Nyquist 9-QAM signals is more compressed than that of regular Nyquist QPSK signals, and only an optical coupler is needed to implement super-Nyquist WDM channel multiplexing, with a net SE up to 4 b/s/Hz (after removing the 20% soft-decision FEC overhead). We also expanded the higher-order QAM DSP algorithm to short-haul optical transmission networks and proposed and experimentally verified high-speed CAP-64-QAM systems based on DML, direct detection, and digital equalization technologies. DD-LMS is used for CAP-64-QAM signal equalization. By using this scheme, we achieved 60 Gbit/s CAP-64-QAM transmission over 20 km SSMF based on DML and direct detection. We first achieved a 100 Gbit/s single-band polarization division multiplexing 16-QAM-OFDM real-time coherent optical transmission system. For the DSP algorithms of a real-time system, we proposed a novel solution, which simplifies the complex floating point multiplication into simple XOR operations by the comparison of only the symbol bits of signals. Thus, the algorithm complexity of time-domain synchronization and frequency offset estimation can be greatly reduced. In the experiment, distortion-less DFT-spread technology is used to reduce the PAPR of OFDM signals. In addition, ISFA is used to eliminate the influence of noise in the optical channels in channel estimation. In this way, the BER of the system can be improved. Our research provides a reliable alternative for real-time transmission in a local area network (LAN) with a channel rate up to 100 Gbit/s.

References

[1] B. Swanson and G. Gilder, "Estimating the exaflood—the impact of video and rich media on the internet—a zettabyte by 2015," Discovery Institute, 2008.
[2] R. W. Tkach, "Scaling optical communications for the next decade and beyond," *Bell Labs Technol.*, vol. 14, no. 4, pp. 3–9, 2010.
[3] R. Essiambre and R. W. Tkach, "Capacity trends and limits of optical communication networks," *Proc. IEEE*, vol. 100, no. 5, pp. 1035–1055, May 2012. doi: 10.1109/JPROC.2012.2182970.
[4] C. V. N. Inde, "Forecast and Methodology 2010–2015," Cisco Systems, 2011.
[5] K. Szczerba, B. E. Olsson, P. Westbergh, *et al.*, "37 Gbit/s transmission over 200 m of MMF using single cycle subcarrier modulation and a VCSEL with 20 GHz modulation bandwidth," *36th Eur. Conf. Exhibition Opt. Commun.*, Torino, Italy, 2010, pp. 1–3. doi: 10.1109/ECOC.2010.5621209.
[6] A. S. Karar and J. C. Cartledge, "Generation and detection of a 56 Gbit/s signal using a DML and half-cycle 16-QAM Nyquist-SCM," *IEEE Photon. Technol.*, vol. 25, no. 8, pp. 757–760, 2013.
[7] K. Roberts, M. O'sullivan, K. T. Wu, *et al.*, "Performance of dual-polarization QPSK for optical transport systems," *Lightwave Technol.*, vol. 27, no. 16, pp. 3546–3559, 2009.
[8] J. Renaudier, O. Bertran-Pardo, H. Mardoyan, *et al.*, "Performance comparison of 40G and 100G coherent PDM-QPSK for upgrading dispersion managed legacy systems," *OFC 2009*, San Diego, USA.
[9] C. Xie, G. Raybon, and P. J. Winzer, "Transmission of mixed 224-Gbit/s and 112-Gbit/s PDM-QPSK at 50-GHz channel spacing over 1200-km dispersion-managed LEAF® spans and three ROADMs," *Lightwave Technol.*, vol. 30, no. 4, pp. 547–552, 2012.
[10] J. X. Cai, C. R. Davidson, A. Lucero, *et al.*, "20 Tbit/s transmission over 6860 km with sub-Nyquist channel spacing," *Lightwave Technol.*, vol. 30, no. 4, pp. 651–657, 2012.
[11] J. Li, Z. Tao, H. Zhang, *et al.*, "Spectrally efficient quadrature duobinary coherent systems with symbol-rate digital signal processing," *Lightwave Technol.*, vol. 29, no. 8, pp. 1098–1104, 2011.
[12] J. Li, E. Tipsuwannakul, T. Eriksson, M. Karlsson, and P. A. Andrekson, "Approaching Nyquist limit in WDM systems by low-complexity receiver-side duobinary shaping," *Lightwave Technol.*, vol. 30, no. 24, pp. 1664–1676, 2012.
[13] Z. Jia, J. Yu, H. Chien, Z. Dong, and D. Huo, "Field transmission of 100 G and beyond: multiple baud rates and mixed line rates using Nyquist-WDM technology," *Lightwave Technol.*, vol. 30, no. 24, pp. 3793–3804, 2012.
[14] D. Dong, J. Yu, Z. Jia, *et al.*, "7×224 Gbit/s/ch Nyquist-WDM transmission over 1600-km SMF-28 using PDM-CSRZ-QPSK modulation," *IEEE Photon. Technol. Lett.*, vol. 24, no. 13, pp. 125–129, 2012.
[15] J. Yu, Z. Dong, H. Chien *et al.*, "Transmission of 200 G PDM-CSRZ-QPSK and PDM-16 QAM with a SE of 4 bit/s/Hz," *Lightwave Technol.*, vol. 31, no. 5, pp. 515–522, 2013.
[16] H. Chien, J. Yu, Z. Jia, Z. Dong, and X. Xiao, "Performance assessment of noise-suppressed Nyquist-WDM for terabit superchannel transmission," *Lightwave Technol.*, vol. 30, no. 24, pp. 3965–3971, 2012.
[17] J. Wang, C. Xie, and Z. Pan, "Generation of spectrally efficient Nyquist-WDM QPSK signals using DSP techniques at transmitter," *OFC 2012*, Los Angeles, USA.
[18] B. Pardo, J. Renaudier, P. Tran, *et al.*, "Submarine transmissions with spectral efficiency higher than 3 bit/s/Hz using Nyquist pulse-shaped channels," *OFC 2013*, Anaheim, USA.
[19] Q. Juan, B. Mao, N. Gonzalez, N. Binh, and N. Stojanovic, "Generation of 28GBaud and 32GBaud PDM-Nyquist-QPSK by a DAC with 11.3GHz analog bandwidth," *OFC 2013*, Anaheim, USA.
[20] J. Wang, C. Xie, and Z. Pan, "Generation of spectrally efficient Nyquist-WDM QPSK signals using digital FIR or FDE filters at transmitters," *Lightwave Technol.*, vol. 30, no. 24, pp. 3679–3686, 2012.
[21] J. X. Cai, "100G transmission over transoceanic distance with high spectral efficiency and large capacity," *Lightwave Technol.*, vol. 30, no. 24, pp. 3845–3856, 2012.
[22] X. Zhou, L. E. Nelson, P. Magill P, *et al.*, "High spectral efficiency 400 Gbit/s transmission using PDM time-domain hybrid 32–64 QAM and training-assisted carrier recovery," *Lightwave Technol.*, vol. 31, no. 21, pp. 999–1005, 2013.
[23] X. Zhou, J. Yu, M. F. Huang, and Y. Shao, "64-Tbit/s, 8 bit/s/Hz, PDM-36QAM transmission over 320 km using both pre- and post-transmission digital signal processing," *Lightwave Technol.*, vol. 29, no. 11, pp. 571–577, 2011.
[24] Y. Koizumi, K. Toyoda, M. Yoshida, and M. Nakazawa, "1024 QAM (60 Gbit/s) single-carrier coherent optical transmission over 150 km," *Opt. Express*, vol. 20, no. 21, pp. 12508–12514, 2012.
[25] K. Toyoda, Y. Koizumi, T. Omiya, *et al.*, "Marked performance improvement of 256 QAM transmission using a digital back-propagation method," *Opt. Express*, vol. 20, no. 21, pp. 19815–19821, 2012.
[26] Y. Koizumi, K. Toyoda, T. Omiya T, *et al.*, "512 QAM transmission over 240 km using frequency-domain equalization in a digital coherent receiver," *Opt. Express*, vol. 20, no. 21, pp. 23383–23389, 2012.
[27] P. J. Winzer, A. H. Gnauck, S. Chandrasekhar, *et al.*, "Generation and 1,200-km transmission of 448-Gbit/s ETDM 56-Gbaud PDM 16-QAM using a single I/Q modulator," *OFC 2010*, San Diego, USA.
[28] F. Buchali, K. Schuh, D. Rosener, *et al.*, "512-Gbit/s DP-16-QAM field trial over 734 km installed SSMF with co-propagating 10 Gbit/s NRZ neighbors incorporating soft-FEC decoding," *OFC 2010*, San Diego, USA.
[29] J. Zhang, J. Yu, N. Chi, *et al.*, "Multi-Modulus Blind Equalizations for Coherent Quadrature Duobinary Spectrum Shaped PM-QPSK Digital Signal Processing," *Lightwave Technol.*, vol. 31, no. 21, pp. 1073–1078, 2013.
[30] J. Zhang, B. Huang, and X. Li, "Improved quadrature duobinary system performance using multi-modulus equalization," *IEEE Photon. Technol. Lett.*, vol. 25, no. 16, pp. 1630–1633, 2013.
[31] R. Rodes, M. Müeller, B. Li B, *et al.*, "High-speed 1550 nm VCSEL data transmission link employing 25 GBd 4-PAM modulation and hard decision forward error correction," *Lightwave Technol.*, vol. 31, no. 21, pp. 689–695, 2013.
[32] T. Tanaka, M. Nishihara, T. Takahara, *et al.*, "50 Gbit/s class transmission in single mode fiber using discrete multi-tone modulation with 10G directly modulated laser," *OFC 2012*, Los Angeles, USA.
[33] R. P. Giddings, X. Q. Jin, E. Hugues-Salas, *et al.*, "Experimental demonstration of a record high 11.25 Gbit/s real-time optical OFDM transceiver supporting 25 km SMF end-to-end transmission in simple IMDD systems," *Opt. Express*, vol. 18, no. 6, pp. 5541–5555, 2010.
[34] J. L. Wei, J. D. Ingham, D. G. Cunningham, R. V. Penty, and I. H. White, "Per-

formance and power dissipation comparisons between 28 Gbit/s NRZ, PAM, CAP and optical OFDM systems for data communication applications," *Lightwave Technol.*, vol. 30, no. 26, pp. 3273–3280, 2012.
[35] J. D. Ingham, R. Penty, I. White, and D. Cunningham, "40 Gbit/s carrierless amplitude and phase modulation for low - cost optical data communication links," *OFC 2012*, Los Angeles, USA.
[36] R. Rodes, M. Wieckowski, T. T. Pham, *et al.*, "Carrierless amplitude phase modulation of VCSEL with 4 bit/s/Hz spectral efficiency for use in WDM - PON," *Opt. Express*, vol. 19, no. 27, pp. 26551–26556, 2011.
[37] M. B. Othman, X. Zhang, L. Deng L, *et al.*, "Experimental investigations of 3D/4D-CAP modulation with DM-VCSELs," *IEEE Photon. Technol. Lett.*, vol. 24, no. 22, pp. 2009–2012, 2012.
[38] M. I. Olmedo, T. Zuo, J. B. Jensen, *et al.*, "Towards 400GBASE 4-lane solution using direct detection of multiCAP signal in 14 GHz bandwidth per lane," *OFC 2013*, Anaheim, USA.
[39] J. Wei, L. Geng, D. G. Cunningham, R. V. Penty, and I. White, "100 Gigabit Ethernet transmission enabled by carrierless amplitude and phase modulation using QAM receivers," *OFC 2013*, Anaheim, USA, pp. 2065–2067.
[40] L. Tao, Y. Wang, Y. Gao, *et al.*, "Experimental demonstration of 10 Gbit/s multi -level carrier-less amplitude and phase modulation for short range optical communication systems," *Opt. Express*, vol. 21, no. 5, pp. 6459–6465, 2013.
[41] J. Zhang, J. Yu, F. Li, *et al.*, "11 × 5 × 9.3Gbit/s WDM-CAP-PON based on optical single-side band multi-level multi-band carrier-less amplitude and phase modulation with direct detection," *Opt. Express*, vol. 21, no. 5, pp. 18842–18848, 2013.
[42] G. Stepniak and J. Siuzdak, "Transmission beyond 2 Gbit/s in a 100 m SI POF with multilevel CAP modulation and digital equalization," *OFC 2013*, Anaheim, USA.
[43] G. H. Im, D. D. Harman, G. Huang, et al., "51.84 Mbit/s 16-CAP ATM LAN standard," *IEEE J. Sel. Areas Comm.*, vol. 13, no. 4, pp. 620–632, 1995.
[44] X. Xiao, F. Li, J. Yu, *et al.*, "100-Gbit/s single-band real-time coherent optical DP-16-QAM-OFDM transmission and reception," *OFC 2014*, San Francisco, USA.

Manuscript received: 2014-10-10

Biography

Jianjun Yu received his PhD degree in electrical engineering from Beijing University of Posts and Telecommunications in 1999. He works for ZTE Corporation as the chief scientist on high - speed optical transmission and director of optics labs in North America. He is also a chair professor at Fudan University and adjunct professor and PhD supervisor at the Georgia Institute of Technology, Beijing University of Posts and Telecommunications, and Hunan University. He has authored more than 100 papers for prestigious journals and conferences. Dr. Yu holds 8 U.S. patents with 30 others pending. He is a fellow of the Optical Society of America. He is editor -in-chief of *Recent Patents on Engineering* and an associate editor for the *Journal of Lightwave Technology* and *Journal of Optical Communications and Networking*. Dr. Yu was a technical committee member at IEEE LEOS from 2005 to 2007 and a technical committee member of OFC from 2009 to 2011.

Roundup

Domestic 4G Mobile Phone Shipments Reached 31.6 Million

December 12, 2014—China Academy of Telecommunication Research of MIIT has announced its analysis report of domestic mobile phone market of November 2014.

In November 2014, the overall shipments of mobile phones reached 44.543 million units in China. Among which 2G mobile phone shipments was of 5.742 million, 3G mobile phone was of 7.2 million and 31.601 million for 4G ones. 4G mobile phone shipments continued to increase rapidly, which was more than 4 times of 3G mobile phone shipments.

During January to November in 2014, mobile phone shipments accumulated to 407 million in China. Among which 2G mobile phone shipments was of 53.935 million, 3G and 4G mobile phones were of 213 million and 140 million, respectively.

In November 2014, 122 new mobile phone models was available in the market. In which 2G mobile phones are of 29 models, 3G mobile phones are 19 and 4G are 74. During January to November in 2014, there are 1952 new models in total. Among which the new 2G phone models are 364, 3G new phone models are 870 and 4G are 718. (source: c114)

MIIT Plans to Open Broadband Access Market

November 28, 2014—Recently, MIIT issued the "Opinions On Open Access To Broadband Market (Draft)", which regulates that private enterprises should be encouraged to participate in the construction and operation of broadband access network infrastructure; encourage the private enterprises to participate in the relevant investment and carry out cooperation with infrastructure companies and provide broadband resale services etc. The main three telecom operators shall not sign exclusive agreements with private enterprises and dynamic adjustment mechanism of prices should be established.

The first batch of pilot cities include: Taiyuan, Shenyang, Harbin, Shanghai, Nanjing, Hangzhou, Ningbo, Xiamen, Qingdao, Zhengzhou, Wuhan, Changsha, Guangzhou, Shenzhen, Chongqing and Chengdu. The pilot time is 3 years. (source:c114)

DOI: 10.3969/j. issn. 1673-5188. 2014. 04. 007
http://www.cnki.net/kcms/detail/34.1294.TN.20141218.1521.001.html, published online December 18, 2014

Influence on Multimode Rectangular Optical Waveguide Propagation Loss by Surface Roughness

Chuanlu Deng[1], Li Zhao[2], Zhe Liu[2], Nana Jia[1], Fufei Pang[1], and Tingyun Wang[1]

(1. Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Shanghai University, Shanghai 200093, China;
2. Manufacture Technology Research Dept., ZTE Corporation, Shenzhen 518057, China)

Abstract

Optical scattering loss coefficient of multimode rectangular waveguide is analyzed in this work. First, the effective refractive index and the mode field distribution of waveguide modes are obtained using the Marcatili method. The influence on scattering loss coefficient by waveguide surface roughness is then analyzed. Finally, the mode coupling efficiency for the SMF-Optical-Waveguide (SOW) structure and MMF-Optical-Waveguide (MOW) structure are presented. The total scattering loss coefficient depends on modes scattering loss coefficients and the mode coupling efficiency between fiber and waveguide. The simulation results show that the total scattering loss coefficient for the MOW structure is affected more strongly by surface roughness than that for the SOW structure. The total scattering loss coefficient of waveguide decreases from 3.97 × 10-2 dB/cm to 2.96 × 10-4 dB/cm for the SOW structure and from 5.24 × 10-2 dB/cm to 4.7 × 10-4 dB/cm for the MOW structure when surface roughness is from 300nm to 20nm and waveguide length is 100cm.

Keywords

optical interconnect; surface roughness; optical waveguide; scattering loss coefficient

This work is supported by the Project of Shanghai Committee of Science and Technology under Grant No. 10511500500 and ZTE Industry-Academia-Research Cooperation Funds.

1 Introduction

In the past decade, optical printed circuit boards (PCBs) interconnection based on waveguide theory has become a research focus [1], [2]. There are many unique advantages about optical interconnection, including high transmission rate, no electromagnetic interference, high density integration, and low power consumption. Optical PCBs interconnection will be widely applied in broadband communication [3], high-performance computing [4], and large data centers [5].

Improving transmission performance of waveguide is important research [1], [6], [7]. Transmission loss of 0.05 dB/cm has been achieved [1]. Surface roughness can bring about guided mode to radiation mode, thereby greatly increasing transmission loss. The authors of [8] and [9] established theoretical model of surface roughness and derived the theoretical expression of scattering loss coefficient of planar waveguide. Kevin K. Lee [10] modified the theoretical expression of scattering loss coefficient for rectangular waveguide. E. Jaberansary *et al.* [11] reported a method based on Fourier integral and finite-difference time-domain (FDTD) in analyzing surface roughness, and the results conformed to the above works, which mainly focused on the single-mode waveguide. For the multimode waveguide, there are few research achievements of surface roughness. A multimode waveguide contains a large number of transmission modes, each of which is influenced by surface roughness. D. Lenz [12] analyzed the influence on multimode rectangular waveguide modes by surface roughness using radiation mode theory.

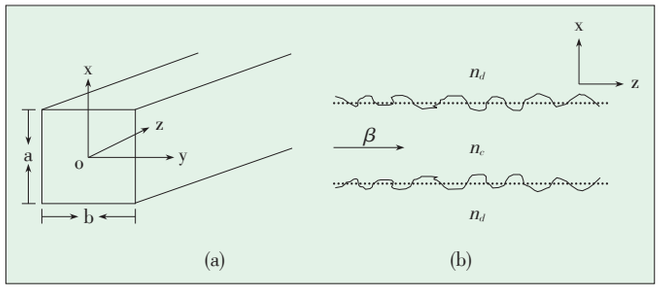
Based on the theoretical model reported by Kevin K. Lee [10], this paper discusses influence on scattering loss coefficients of multimode rectangular waveguide modes by surface roughness, deeply studies the coupling efficiencies between guided modes of fiber (single-mode fiber, single-mode fiber (SMF) and multimode fiber, multi-mode fiber (MMF)) and multimode rectangular waveguide modes, and further calculates total scattering loss coefficient of multimode rectangular waveguide by surface roughness based on SMF-Optical-Waveguide (SOW) structure and MMF-Optical-Waveguide (MOW) structure.

2 Theory of Transmission Loss Induced by Surface Roughness

Surface roughness along a waveguide is a random variable that reflects the degree of smoothness of the core surface. The geometry and surface roughness of multimode rectangular waveguide are represented in **Fig. 1**. The core $n_c = 1.51$ is surrounded by the cladding $n_{cl} = 1.48$, and the width a and height b is 50 μm and 50 μm. The transmission wavelength λ is 850 nm. Surface roughness causes variations of modes effective refractive index. It brings about guided modes to radia-

Influence on Multimode Rectangular Optical Waveguide Propagation Loss by Surface Roughness

Chuanlu Deng, Li Zhao, Zhe Liu, Nana Jia, Fufei Pang, and Tingyun Wang



▲Figure 1. (a) rectangular waveguide, and (b) surface roughness.

tion modes, and thereby affects the transmission loss. The scattering loss coefficient describes the influence on transmission loss of waveguide by surface roughness and is given by

$$\alpha = \varphi^2(d)\left(n_c^2 - n_{cl}^2\right)^2 \frac{k_0^3}{4\pi n_c}\int_0^{\pi} R\left(\beta - n_{cl}k_0\cos\theta\right)d\theta \quad (1)$$

where $\varphi(d)$ is a modal function only depending on waveguide geometrical parameters, $k_0 = 2\pi/\lambda$ is the wave number in vacuum, and $\beta = k_0 n_{eff}$ is propagation constant of mode. $R(\Omega)$ is the power spectrum function, where $\Omega = \beta - n_{cl}k_0\cos\theta$ with θ as the scattering angle relative to the waveguide axis. The integral term from (1) takes into account all the spatial frequencies Ω induced by surface roughness [10]. $R(\Omega)$ is linked to the autocorrelation function $r(u)$ through a Fourier transform:

$$R(\Omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} r(u)\exp(i\Omega u)\mathrm{d}u \quad (2)$$

where $r(u)$ is an average correlation between one position along the waveguide at a distance u. In general, exponential statistic or Gaussian statistic describes the autocorrelation function. The exponential statistic is well suited to characterizing surface roughness of multimode waveguides. This surface roughness described by exponential statistic is given by

$$r(u) = \sigma^2 \exp\left(-\frac{u}{L_c}\right) \quad (3)$$

where L_c is the correlation length, and σ is the standard deviation. Scattering loss coefficient in decibel per centimeter can be written as [10]

$$\alpha = 4.34\frac{\sigma^2}{\sqrt{2}\,k_0 d^4 n_c} g(V) f(x,\gamma) \quad (4)$$

The function of $g(V)$ is determined by waveguide geometry

$$g(V) = \frac{U^2V^2}{1+W} \quad (5)$$

with the normalized coefficients $U = k_0 d\sqrt{n_c^2 - n_{eff}^2}$, $V = k_0 d\sqrt{n_c^2 - n_{cl}^2}$, and $W = k_0 d\sqrt{n_{eff}^2 - n_{cl}^2}$.

The function $f(x,\gamma)$ describes the integral over the spectral density function, which depends on the surface roughness. It can be expressed as

$$f(x,\gamma) = \frac{x\sqrt{1 - x^2 + \sqrt{\left(1+x^2\right)^2 + 2x^2\gamma^2}}}{\sqrt{\left(1+x^2\right)^2 + 2x^2\gamma^2}} \quad (6)$$

with the normalized coefficients $x = \frac{WL_c}{d}$, $\gamma = \frac{n_{cl}V}{n_c W\sqrt{\Delta}}$, and $\Delta = \frac{n_c^2 - n_{cl}^2}{2n_c^2}$.

3 The Effective Refractive Index

In order to obtain scattering loss coefficients, we need to obtain the effective refractive indices. There are two modes E_{mn}^x and E_{mn}^y for multimode rectangular waveguide (m and n are model number to x and y direction).

The effective refractive indices of rectangular waveguide modes can be calculated using the Marcatili method. **Table 1** shows the effective refractive indexes of E_{mn}^x mode and E_{mn}^y mode, where the model number is arranged according to the value of the effective refractive indexes. When we compare E_{13}^x mode with E_{13}^y mode, we find that the effective refractive indices are very close, which means that E_{mn}^x mode and E_{mn}^y mode degenerate at the same model number. Thus only E_{mn}^x mode is analyzed in this paper.

Fig. 2 shows the three-dimensional filed distribution of E_{22}^x mode and E_{33}^x mode respectively. It indicates that E_{22}^x mode is asymmetrical, and E_{33}^x mode is symmetrical.

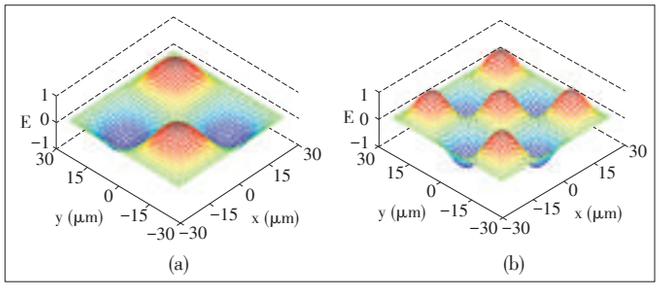
4 Influence on Transmission Loss by Surface Roughness

4.1 Scattering Loss Coefficient of Waveguide Modes

The rectangular waveguide based on multimode structure in this paper contains a large number of guided modes, all of

▼Table 1. Effective refractive index of E_{mn}^x mode and E_{mn}^y mode

E_{mn}^x				E_{mn}^y			
Order	Number		n_{eff}	Order	Number		n_{eff}
	x	y			x	y	
1	1	1	1.50995380	1	1	1	1.50995380
2	1	2	1.50988456	2	2	1	1.50988456
3	2	1	1.50988446	3	1	2	1.50988446
4	2	2	1.50981521	4	2	2	1.50981521
5	1	3	1.50976914	5	3	1	1.50976914
6	3	1	1.50976888	6	1	3	1.50976888



▲**Figure 2. Mode field distribution: (a) E^x_{22}; (b) E^x_{33}.**

which are influenced by surface roughness. Different order modes have different scattering loss coefficients.

When σ is 20 nm, L_c is 4 μm and λ is 850 nm, **Fig. 3** shows the distribution of scattering loss coefficient of the fundamental mode and high order modes. From Fig. 3 the scattering loss coefficients of modes increase as the model number increases. The scattering loss coefficient is on the m, n ($m=n$) symmetry basically, for example, scattering loss coefficient of E^x_{13} mode and E^x_{31} mode is 1.77588×10^{-3} dB/cm and 1.77944×10^{-3} dB/cm, respectively.

4.2 Total Scattering Loss Coefficient of Waveguide

In multimode rectangular waveguide, each mode carries a certain proportion of optical power determined by the coupling efficiency between the excited modes (the guided modes of fiber) and the transmission modes of the waveguide. Therefore, the total scattering loss coefficient is not only related to the scattering loss coefficient of each mode but also depends on the coupling characteristics of the joint configurations (fiber-waveguide) (**Fig. 4**).

4.2.1 Mode Coupling Efficiency

The coupling efficiency between the guided mode of SMF and E^x_{mn} mode of multimode rectangular waveguide in SOW is

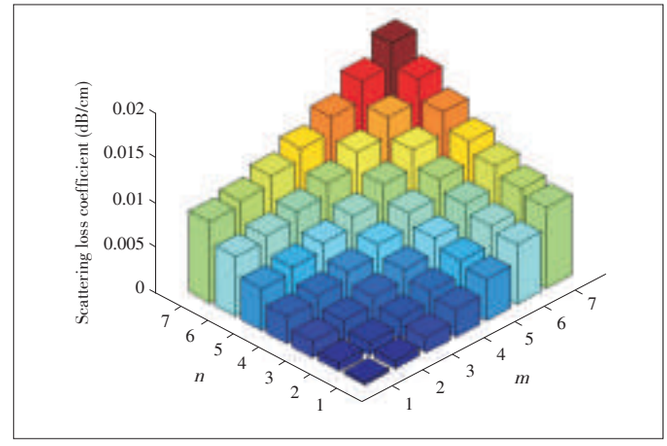
$$|\eta_{mn}|^2 = \frac{\left|\int_{-\infty}^{\infty}\psi_{in}\varphi^*_{mn}\mathrm{d}x\mathrm{d}y\right|^2}{\left|\int_{-\infty}^{\infty}\psi_{in}\psi^*_{in}\mathrm{d}x\mathrm{d}y\right|\left|\int_{-\infty}^{\infty}\varphi_{mn}\varphi^*_{mn}\mathrm{d}x\mathrm{d}y\right|} \quad (7)$$

where ψ_{in} is the guided mode field distribution of SMF and regarded as a Gauss mode field with mode field radius (ω). ω is calculated using the 2nd Petermann method [13] as 3.748 μm. The guided mode field distribution of SMF is expressed as

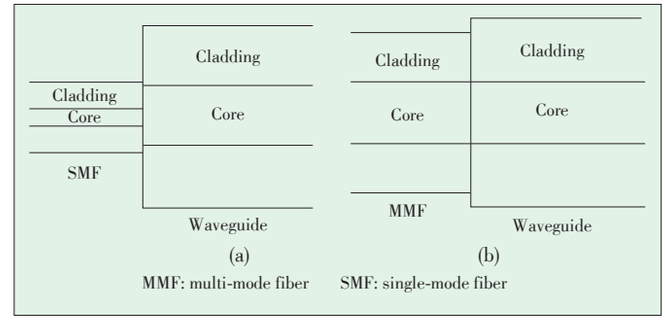
$$\psi_{in} = \exp(-\frac{x^2}{\omega^2})\exp(-\frac{y^2}{\omega^2}) \quad (8)$$

φ_{mn} is the field distribution of E^x_{mn} mode of multimode rectangular waveguide.

However, for the MOW structure, the coupling efficiencies between the guided modes of MMF and E^x_{mn} mode of rectangular waveguide are given by



▲**Figure 3. Scattering loss coefficient for different model numbers.**



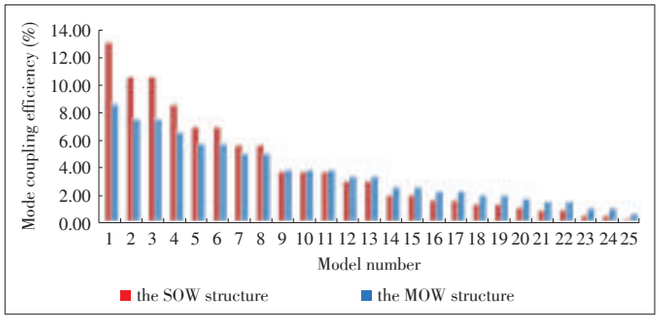
▲**Figure 4. Optical PCBs interconnection coupling structures: (a) SOW; (b) MOW.**

$$|\eta_{mn}|^2 = \sum_{i=1}^{M}\frac{\left|\int_{-\infty}^{\infty}\psi_{0i}\varphi^*_{mn}\mathrm{d}x\mathrm{d}y\right|^2}{\left|\int_{-\infty}^{\infty}\psi_{0i}\psi^*_{0i}\mathrm{d}x\mathrm{d}y\right|\left|\int_{-\infty}^{\infty}\varphi_{mn}\varphi^*_{mn}\mathrm{d}x\mathrm{d}y\right|} \quad (9)$$

where ψ_{0i} is LP_{0i} mode field distribution of MMF, and M is the total mode number of MMF.

The coupling efficiencies between the guided modes of fiber and E^x_{mn} mode of rectangular waveguide with even model number is quite small due to different odd-even characteristic of its filed distribution, which can be neglected. The coupling characteristic means that the energy of input field from fiber is mostly transferred to the odd number modes of waveguide. Thus, numerical calculation following considers E^x_{mn} mode of rectangular waveguide with odd model number only.

Fig. 5 shows the histogram of mode coupling efficiency of the SOW structure and MOW structure. The coupling efficiencies decrease as the model number increases. The coupling efficiency of E^x_{mn} mode with model number 9 ($m=1, n=7$) below for the SOW structure is higher than that of E^x_{mn} mode for the MOW structure, but it is opposite for E^x_{mn} mode with model number 9 ($m=1, n=7$) above. The coupling efficiency dis-



▲Figure 5. Mode coupling efficiency for SOW and MOW.

tribution characteristics imply the different transmission loss for the SOW structure and the MOW structure.

The mode coupling efficiency for the SOW structure and the MOW structure are shown in **Table 2**, which proves the coupling characteristics previously discussed.

4.2.2 Total Scattering Loss Coefficient

For the SOW and MOW structures, the total scattering loss coefficient is given by [12]

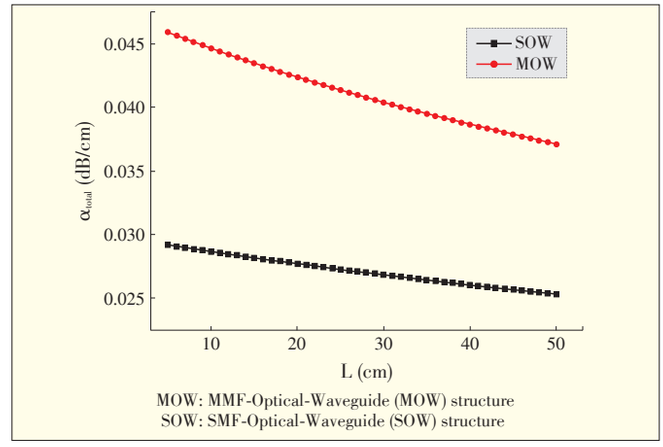
$$\alpha_{total} = \frac{1}{10L}\lg\left(\frac{\sum|\eta_{mn}|^2}{\sum|\eta_{mn}|^2 \exp(-\alpha_{mn}L)}\right) \text{ dB/cm} \qquad (10)$$

where L is the length of waveguide. 10 indicates that the total scattering loss coefficient is related to the mode coupling efficiency and the individual scattering loss coefficient of different order modes. It also changes with waveguide length.

Fig. 6 shows how the total scattering loss coefficient changes with the waveguide length for the SOW and MOW structures when σ = 200 nm and L_c = 4 μm. Overall, the total scattering loss coefficient decreases linearly as L increases; however, the total scattering loss coefficient of the MOW structure is larger than that of the SOW structure. For example,

▼Table 2. Mode coupling efficiency of SOW and MOW

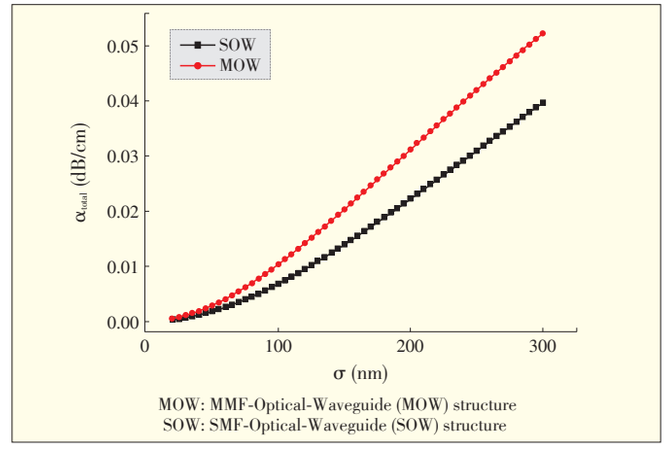
E^x_{mn}									
Number	Model Number		$\|\eta_{mn}\|^2$ (%)		Number	Model Number		$\|\eta_{mn}\|^2$ (%)	
	m	n	SMF	MMF		m	n	SMF	MMF
1	1	1	12.99	8.50	9	1	7	3.61	3.74
2	1	3	10.49	7.41	10	5	5	3.60	3.74
3	3	1	10.49	7.41	11	7	1	3.60	3.74
4	3	3	8.47	6.46	12	3	7	2.91	3.26
5	1	5	6.84	5.64	13	7	3	2.91	3.26
6	5	1	6.84	5.63	14	5	7	1.90	2.48
7	3	5	5.53	4.91	15	7	5	1.90	2.48
8	5	3	5.52	4.91	16	1	9	1.54	2.16



▲Figure 6. Total scattering loss coefficient changes with waveguide length.

when L = 50 cm, the total scattering loss coefficient is 3.71 × 10^{-2} dB/cm for the MOW structure and 2.53 × 10^{-2} dB/cm for the SOW structure. The reason is that the coupling efficiencies of waveguide high order modes for the MOW structure are larger than those for the SOW structure as shown in Fig. 5, and that scattering loss coefficients of waveguide high order modes are also larger relatively as shown in Fig. 3.

When L_c is 4 μm and L is 100 cm, **Fig. 7** shows the total scattering loss coefficients with σ for SOW and MOW structure. The total scattering loss coefficient increases as σ increases for the two coupling structures. When σ is less than 120 nm, the loss coefficient is in 10^{-3} dB/cm order of magnitude, and when σ is greater than 120 nm, it is in 10^{-2} dB/cm order of magnitude. Figs. 6 and 7 shows that the total scattering loss coefficient for the MOW structure is also larger than that for the SOW structure with the same σ. Also, the difference in total scattering loss coefficient between the two coupling structure is larger when σ is larger, but the difference is very small for $\sigma = 20\,nm$. Given σ is from 20 nm to 300 nm, the total scattering loss coefficient increases from 2.96 × 10^{-4}



▲Figure 7. Total scattering loss coefficient of waveguide as different surface roughness.

dB/cm to 3.97×10^{-2} dB/cm for the SOW structure and from 4.7×10^{-4} dB/cm to 5.24×10^{-2} dB/cm for the MOW structure. In other words, the scattering loss of waveguide is 3.97 dB for the SOW structure and 5.24 dB for the MOW structure when σ is 300 nm and L is 100 cm. This means that surface roughness is a disadvantage for the transmission characteristic of a waveguide.

In light of current preparation technology of waveguides, the scattering loss has a large decreasing space. If L = 100 cm and surface roughness decreases from 300 nm to 20 nm by technological optimization, the scattering loss of the waveguide decreases from 3.97 dB to 0.0296 dB for the SOW structure and from 5.24 dB to 0.047 dB for the MOW structure. However, for the multimode rectangular waveguide that has been made, what it is discussed above also produces other idea that the SOW structure instead of the MOW structure may be a better method in order to further decrease scattering loss. From Fig. 7, the scattering loss of waveguide for the SOW structure reduces about 0.5-1.5 dB than that for the MOW structure given σ is from 100 nm to 300 nm and L is 100 cm.

The total scattering loss coefficient for the SOW structure is affected by the alignment accuracy with 3 directions (r,θ,z) more greatly than that for the MOW structure in practical application, but it is the secondary factor and can be neglected. Our work theoretically supports improvement of the transmission characteristic of multimode rectangular waveguide.

5 Conclusion

The scattering loss coefficient is related to the mode scattering loss coefficients of multimode rectangular waveguide and depends on the coupling efficiency between the guided mode of fiber and the transmission mode of the waveguide. This paper introduces two kinds of coupling structures: SOW and MOW. The simulation results show that the total scattering loss coefficient of multimode rectangular waveguide for the MOW structure is affected more strongly by surface roughness than for the SOW structure. The total scattering loss coefficient of waveguide decreases from 3.97×10^{-2} dB/cm to 2.96×10^{-4} dB/cm for the SOW structure and from 5.24×10^{-2} dB/cm to 4.7×10^{-4} dB/cm for the MOW structure when σ is from 300 nm to 20 nm and L is 100 cm. This implies that optical PCBs interconnection based on the SOW structure performs better.

References

[1] R. Dangel, F. Horst, D. Jubin, *et al.*, "Development of versatile polymer waveguide flex technology for use in optical interconnects," *J. Lightwave Technol.*, vol. 31, no. 24, pp. 3915–3926, Dec. 2013. doi: 10.1109/JLT.2013.2282499.
[2] P. Rosenberg, S. Mathai, W. V. Sorin, *et al.*, "Low cost, injection molded 120 Gbps optical backplane," *J. Lightwave Technol.*, vol. 30, no. 4, pp. 590–596, Feb. 2012. doi: 10.1109/JLT.2011.2177813.
[3] J. Matsui, T. Yamamoto, K. Tanaka, *et al.*, "Optical interconnect architecture for servers using high bandwidth optical mid-plane," *OFC/NFOEC 2012*, Los Angeles, USA, pp. 1–3.
[4] M. A. Taubenblatt, "Optical interconnects for high-performance computing," *J. Lightwave Technol.*, vol. 30, no. 4, pp. 448–457, Feb. 2012. doi: 10.1109/JLT.2011.2172989.
[5] C. Kachris and I. Tomkos, "A survey on optical interconnects for data centers," *IEEE Commun. Surveys & Tutorials*, vol. 14, no. 4, pp. 1021–1036. doi: 10.1109/SURV.2011.122111.00069.
[6] N. Bamiedakis, A. Hashim, R. V. Penty, and I. H. White, "Regenerative polymeric bus architecture for board-level optical interconnects," *Opt. Express*, vol. 20, no. 11, pp. 11625–11636, 2012. doi: 10.1364/OE.20.011625.
[7] A. Sugama, K. Kawaguchi, M. Nishizawa, *et al.*, "Development of high-density single-mode polymer waveguides with low crosstalk for chip to-chip optical interconnection," *Opt. Express*, vol. 21, no. 20, pp 24231–24239, 2013. doi: 10.1364/OE.21.024231.
[8] J. P. R. Lacey and F. P. Payne, "Radiation loss from planar waveguides with random wall imperfections," *IEE Proc. J. Optoelectronics*, vol. 137, no. 4, pp. 282–288, Aug. 1990.
[9] F. P. Payne and J. P. R. Lacey, "A theoretical analysis of scattering loss from planar optical waveguides," *Opt. and Quantum Electron.*, vo. 26, no. 10, pp. 977–986, Oct. 1994. doi: 10.1007/BF00708339.
[10] K. K. Lee, D. R. Lim, H.-C. Luan *et al.*, "Effect of size and roughness on light transmission in a Si/SiO_2 waveguide: Experiments and model," *Appl. Physics Lett.*, vol. 77, no. 11, pp. 1616–1619, Sept. 2000. doi: 10.1063/1.1308532.
[11] E. Jaberansary, T. M. B. Masaud, M. M. Milosevic, *et al.*, "Scattering loss estimation using 2-D fourier analysis and modeling of sidewall roughness on optical waveguides," *IEEE Photonics J.*, vol. 5, no. 3, article no. 6601010, Jun. 2013. doi: 10.1109/JPHOT.2013.2251869.
[12] D. Lenz, D. Erni, and W. Bächtold, "Modal power loss coefficients for highly overmoded rectangular dielectric waveguides based on free space modes, *Opt. Express*, vol. 12, no. 6, pp. 1150–1156, 2004. doi: 10.1364/OPEX.12.001150.
[13] P. Ou., *Higher Optical Simulation*. Beijing, China: Beihang University press, 2011, pp. 122–125.

Manuscript received: 2014-11-02

Biographies

Chuanlu Deng (chuanludeng@163.com) received his BS degree from Ludong University, China in 2005 and master's degree from University of Shanghai for Science and Technology in 2009. From 2009 to 2013, he worked as an optical engineer at Shanghai Wei Qi for Photoelectric Science and Technology Co., Ltd. and then Shanghai Ya Ming Lighting Co., Ltd. He is currently a doctoral candidate of Shanghai University, with a major in Communication and Information Systems. His research interests include optical PCBs interconnection technologies.

Li Zhao (zhao.li8@zte.com.cn) received her BS and master's degrees from College of Microelectronics and Solid Electronics, University of Electronic Science and Technology of China (UESTC). She is currently a process engineer at ZTE Corporation.

Zhe Liu (liu.zhe@zte.com.cn) received his BS and master's degree from UESTC. He is currently a chief process engineer at ZTE Corporation.

Nana Jia (18817872809@163.com) received her BS degree from College of Information Technology and Communication, Qufu Normal University, China in 2012. She is currently a master candidate of Shanghai University, with a major in Communication and Information Systems. Her research interests include optical PCBs interconnection technologies.

Fufei Pang (ffpang@shu.edu.cn) received his PhD degree in optical engineering from Shanghai Institute of Optics and Fine Mechanics of Chinese Academy of Sciences in 2006. He is currently a professor of Shanghai University. His research interests include specialty fiber for optical sensing applications.

Tingyun Wang (tywang@mail.shu.edu.cn) received his BS degree in automatic engineering from Hebei Institute of Technology, China in 1983, MS degree in electrical engineering from Harbin University of Science and Technology, China in 1986, and PhD degree in electromagnetic measurement and instrumentation from Harbin Institute of Technology, China in 1998. He worked as a post doctorate fellow at Tsinghua University, China from 1998 to 2000. His major interests lie in specialty fiber optics, fiber optic sensors, nano-photonics and fiber devices and their systems. Dr. Wang is a member of Optical Society of America (OSA) and senior member of The Chinese Optical Association. He is also editors of *Journal of Opto-Electronics Laser* (in Chinese) and *Opto-Electronics Letters*.

DOI: 10.3969/j. issn. 1673-5188. 2014. 04. 008
http://www.cnki.net/kcms/detail/34.1294.TN.20141204.1406.002.html, published online December 4, 2014

An MAS Framework for Speculative Trading Research in Stock Index Futures Market

Junneng Nie and Haopeng Chen
(School of Software, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract

In this paper, we develop a futures trading simulation system to determine how speculative behavior affects the futures market. A configurable client is designed to simulate traders, and users can define trade strategies using different programming languages. A lightweight server is designed to handle large-scale and highly concurrent access requests from clients. HBase is chosen as the database to grantee scalability of the system. As HBase only supports single-row transaction, a transaction support mechanism is developed to improve data consistency for HBase. The HBase transaction support mechanism supports multi-row and multi-table by using two phase commit protocol. The experiments indicate that our system shows high efficiency in the face of the large scale and high concurrency access request, and the read/write performance loss of HBase introduced by the transaction support mechanisms is also acceptable.

Keywords

NOSQL; transaction management; isolation level; multi-agent system

1 Introduction

The index futures market is a complex adaptive system that allows the traders to submit orders to supply liquidity and to consume liquidity. The behavior of index futures market is an important consideration for both traders and market regulators for the reason that the orders needs to be placed and execute fluently and the market has to be functioned smoothly.

One way to estimating and researching the market could be focused on analyzing and fitting statistical models on past data. However, the future market conditions and scenarios may never have occurred before. So this approach is very restrictive. An more effective approach is the multi-agent system (MAS) as emphasized by Dawid and Fagiolo [1]. This is a specialization to economics of the basic complex adaptive systems paradigm [2].

Hence, inspired by previous studies like the Santa Fe Artificial Stock Market [1], we propose a distributed MAS for speculative trading strategy research and policy design in the CSI300 index futures market.

Our system is realistic. We build the configurable client on a distributed architecture. The whole system scales well when the number of clients increases. We also let the client's trading strategy be updated with another type written in several different programming languages without shutting down the system. The agents gossip with each other which results the pessimistic or optimistic mood among themselves affecting their trading strategy. Furthermore, the real market data is incorporated as the fundamental parameter which contributes to make the traders' behaviors more reliable and credible.

Every module in the server is lightweight, especially for FIX server, it only concerns the transmitting of the trade request. The business logic modules are implemented as simplified as possible. The state of orders and traders is transmitted by messages, which enhances the scalability of the system.

Besides, the system is required to process huge volume of trades concurrently, and the historical market information, including the intermediate trading messages and the action log of each agent, summing up results in the generation of significant data. Meeting up the massive concurrent access as well as providing efficient access to the various data are the major bottlenecks of the database. Furthermore, the various data include both structured data, like account ID, initial money, contracts held, and unstructured data like the action log which registered every move the agent made all the time.

In conclusion, there are two major obstacles, large scale and high concurrent access requests as well as the large scale data storage and processing, concerning the futures trading simulation system. The difficulties of the whole system are listed as follows. High efficiency has to be met while dealing with large and high concurrent access requests, as well as the strong consistency should also be satisfied while handling large scale data storage and processing.

2 Related Work

The basic economic structure of the market draws heavily on existing market setups such as Bray [3]. However, in recent years, agent-based simulations of market have been developing rapidly and are more widely accepted. The Santa Fe Institute Artificial Stock Market Model was one of the first agent-based models of a financial market, and was developed by Brian Arthur *et al.* The model is inhabited by a population of myopic, imperfectly rational, heterogeneous agents who make invest-

ment decisions by forecasting the future status of the market and who also learn from their experience over time. The model illustrates how simple interactions between such agents may lead to the appearance of the realistic structure itself.

Researchers at Santa Fe Institute later constructed a widely used multi-agent software platform called Swarm that simulates financial trading systems. Swarm [4] is an opensource toolkit with both Objective-C and Java bindings. It was originally developed as a software toolkit for the creation of simulation models in the field of Artificial Life. It provides modelers with a flexible, nested approach to modeling interactions. Minar [4] states that agent-based modeling toolkits or libraries are important in that they save scientists from wasting time on repetitive and error-prone programming. Object orientation is also useful because of its close association with agents.

Other agent-based simulation platforms are listed as follows. Repast [5] is a family of three free opensource agent-based modeling libraries. The three libraries are Java-based Repast J, C#-based Repast .NET, and NQP (Not Quite Python)-based Repast by StarLogo [5]. NetLogo [6] is a free opensource agent-based simulation environment that uses a modified version of the Logo programming language.

All these agent-based simulation platforms are bound directly to specific programming languages. It is common for people from business and financial communities are lack of coding proficiency. Therefore, it is of vital importance to keep high scalable ability of the Futures trading platform which means that the platform has to support several different languages implementing the trading algorithm.

Based on platforms discussed above, researchers have built many simulation systems for financial trading. Multi-agent automated intelligent shopping system (MAISS) [7] is a distributed system where human users (buyers and sellers) delegate their tasks to agents, which then shop on their behalf and present the results. Buyers (customers) and sellers (suppliers) may be organizations, companies or individuals. S. Mundle [8] introduces a multi-agent architecture that is designed and simulated using opensource Java Agent Development Framework (JADE). In this JADE-based multi-agent system for mobile computing in cellular networks, agents interact with each other to find the optimal threshold for call admission using distributed service architectures. It evaluates connection-level performance characteristics of dynamic and mobility based call admission control schemes using agents. A. Shemshandi [9] proposes implementing a multi-agent system for real-time coordination of a typical supply chain based on JADE technology. This paper addresses the preliminary approach towards process-oriented collaborative inventory management in supply chains, taking advantage of multi-agent technology in terms of modeling and simulation.

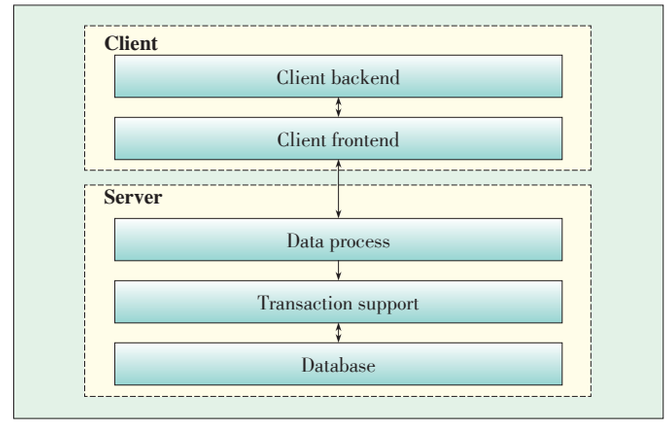
Almost all the simulation software previously mentioned use relational databases. However, a futures trading simulation system faces the challenge of requests on large scale and high concurrent access. This means that a NoSQL database [10] is a better choice. However, NoSQL databases can hardly provide full support for complicated transactions [11]. This leads to failure in keeping the strong consistency of the data. And that is an indispensable key technology in our system.

An open source project called Omid [12] provides transactional support for HBase. Omid uses a central server called Transaction Oracle to manage transactions. The API can be used in transaction clients. Transaction clients are in charge of connecting to the transaction oracle and perform the required operations in data-stores. The transaction oracle replicates transactional information to the transaction clients which just contact the transaction oracle when they want to start a transaction or commit it. HBASE-5229, one of HBase's updating issues, mentioned that it provides basic building blocks for "multi-row" local transactions, however, afterward it also says this feature is by design and is not available to the client via the HTable API [13].

Goolge proposed a technique called Percolator for BigTable [14]. Percolator chooses to use a column of itself to provide lock service and implements a two phase commit protocol to guarantee the synchronism of multi-row and multi-table. We implemented the multi-row and multi-table transactional mechanism based on HBase referencing percolator. Furthermore, we extended the multi-row and multi-table transaction mechanism, such as providing multi-level isolation level.

3 Framework and Flow

Fig. 1 shows the architecture of the futures trading simulation system. Trading strategies of all kinds of traders are stored in the client backend. Each client front-end stimulates a certain number of traders and communicates with the client backend. Trading requests generated according to corresponding strategies are sent to the server side using Financial Information Exchange (FIX) protocol. FIX is a communication and messaging protocol widely used to conduct securities transactions. After trading requests are received, they are processed



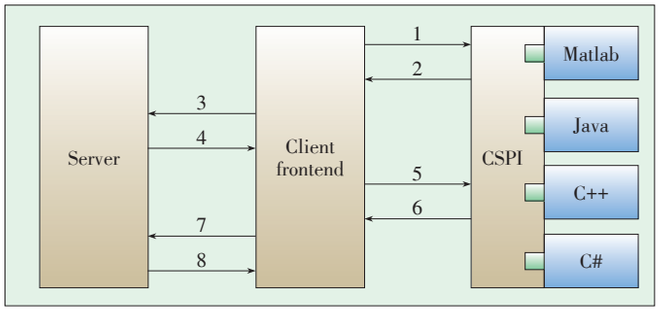
▲**Figure 1. Architecture of the futures trading simulation system.**

by the business logic adopting distributed computation methods. HBase, which is used as the database in our system, only supports single-row transactions. Therefore, we add a transaction support layer for multi-row and multi-table transactions. The detailed architecture of client and server will be discussed in the next section.

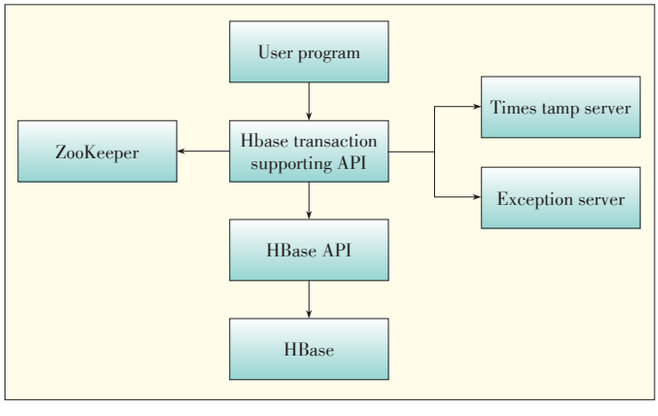
Fig. 2 shows a simulated trading process that involves the interaction between the three modules such as client frontend, client backend (CSPI) and server. Client frontend is in charge of generating and transmitting trading requests; client backend stores all trading strategies used by traders; server is responsible for managing and matching the trading requests. First, one of the traders that client front end simulates communicates with the corresponding trading strategies stored in client back end to decide what kind of data it needs. This data may be real-time and historical data of stock index futures. Then, the client front end collects the required information from the server and sends it back to the client backend. After client back end receives the data, trading orders are generated by running relevant strategy and then orders are sent to the server. Finally, the server processes the orders and sends feedback, that the order is successfully matched or failed, to client frontend.

Fig. 3 shows the transaction supporting mechanism of HBase. The HBase transaction supporting mechanism is the interlayer between the user software and HBase, and the HBase transaction supporting mechanism invokes the API of HBase to support multi-table transaction. The HBase transaction supporting mechanism can be divided into five sections:

1) User program, which uses HBase transaction supporting mechanism to execute the transaction, The user program is the entrance to the HBase transactions.
2) HBase transaction supporting API, which is the main body of the transaction supporting mechanism. This API also offers many of the transaction functions, and transactional operations such as create, update, and delete.
3) Timestamp Server, which provides unified the timestamp of the service for the HBase transaction supporting mechanism. The timestamp must be monotonically increasing. The transaction gets a time stamp from the Timestamp Server each time at the beginning and submitting.
4) ZooKeeper, which monitors and keeps track of the state of every node in the cluster.
5) Exception server, which is the handler of the abnormal transaction in HBase transaction supporting mechanism. Each initiator has to communicate with the Exception Server and the transaction information stored is used by Exception Server to process the abnormal transaction after a abnormal node in the cluster being detected.



▲Figure 2. A specific simulated trading process.



▲Figure 3. Architecture of transaction support mechanism for HBase.

4 Key Techniques

4.1 Configurable Client

In order to simulate real, complex futures trading, simulated traders depend on well-defined trading strategies and apply the data they get from the simulated market to make predictions and generate trading orders. There are three different kinds of traders: hedger, speculator and arbitrager. Each kind of trader has different trading strategies. The client is split into client front end module and client back end module. Client front end communicates mutually with a server, and the client back end provides the trading strategy for client front end. Detailed realization of the trading strategy is not our top priority. This is realized by financial professionals. The strategies in the client back end can be implemented using advanced programming language such as Java, C++, C#, and even Matlab. Also the uniform interface is provided to support different strategies, adopting the design pattern of adapter.

The scalability and usability of the system are improved by using the configuration settings to initialize the client front end and back end. The configuration settings for client front end contain the port number, the FIX protocol version, the server's IP address and port number, the client backend's IP address and port number, the interaction protocol with client back end, the number of simulated traders, the percentage of each kind of simulated trader, and the interface of the trading strategy. The configuration settings for the client back end contain the port number, the interaction protocol with client front end, the implementation of the adapter, and the interface of trading strategy. All configuration settings exist as cfg files that can be

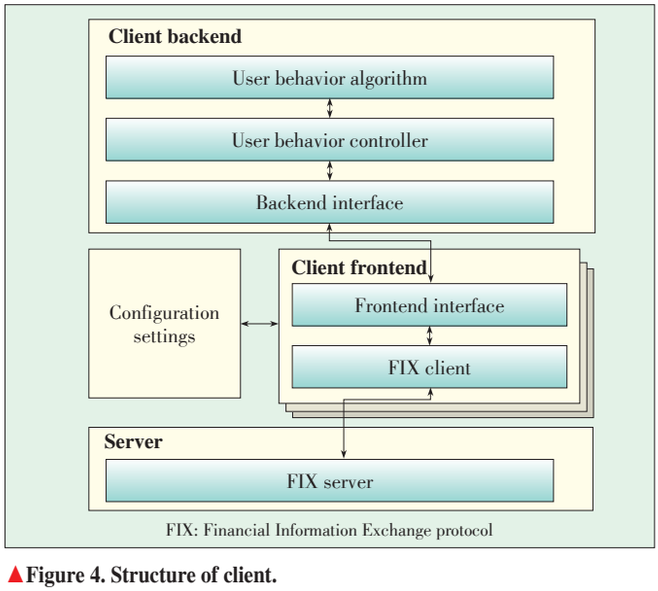
edited by text and graphic. The system also provides several different default configuration settings.

Fig. 4 shows the structure of the client. There are several client front end modules deployed in a cluster of nodes, and each of these modules runs a certain number of simulated traders. The client back end is deployed in a single node and communicates with all client front ends. There are multiple types of interaction between the client front end and backend. These include RMI and Web Service. The interaction between the client front end and server relies on FIX protocol. Message formats and mechanisms for sending and receiving message are defined in FIX protocol, and FIX protocol also includes the content of session layer management, application layer messages, and data dictionary. All FIX protocol versions and most part of the functions are implemented in FIX server module and FIX client module.

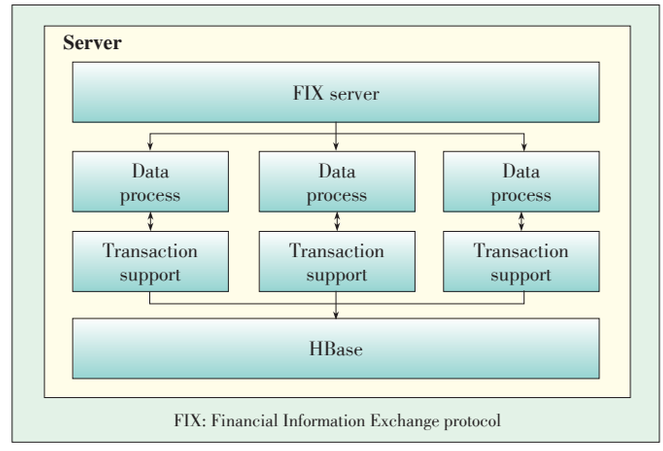
4.2 Lightweight Server

Fig. 5 shows the structure of the server. As soon as the FIX server receives the trade requests from clients, it transmits them to the nodes deployed in the cluster. Each node in the cluster runs two processes: data and transaction support. The data processing module handles the trade requests according to the business logic, and transaction support module provides multi-row and multi-table transaction support for HBase. As with the client, the server also uses the configuration settings, which can configure the port number of the FIX server, the FIX protocol version and the port number and IP address of the node in the cluster.

In order to improve system performance, we designed a lightweight server. First, every module in the server is lightweight, especially for FIX server, it only concerns the transmitting of the trade request. Also, we implement the business logic of other modules in the cluster to be as simple as possible. Second,



▲Figure 4. Structure of client.



▲Figure 5. Structure of server.

according to the actual operation condition, we implement different transaction isolation levels and use the level will be as low as possible to decrease the responding time of the database. For example, for the operation in the table of the trade request, we set the isolation level to be read-committed to guarantee consistency. For the operation in table of the real-time data of the market, we set the isolation level to be read-uncommitted in order to improve read performance. For the operation in table of the historical data of the market, we set no isolation level because it is a read-only table. Finally, the server is stateless. The state of orders and traders is transmitted by messages and this enhances the scalability of our system.

4.3 Non-relational Database Supporting Strong Consistency

HBase is deployed as the database in our system. Similar to BigTable, HBase supports only single-row transaction. Consider a situation where trader A transmits a purchase order to a server, and trader B sends an order selling the same futures at the same time. Afterwards, the requests are considered matching successfully, and two trade records are generated by the server. Then, the two records have to be written into the database. We need to ensure atomicity because they would both succeed or fail. However, this operation involves two different rows, and the above requirement cannot be satisfied by HBase. Therefore, in our system, a technique similar to Percolator is used to support multi-row and multi-table transaction. Beyond Percolator, we also implement three different transaction isolation levels for Hbase: read-uncommitted, read-committed, and repeatable-read.

4.3.1 The Design of The Distributed Transaction Locks

Our system uses a single row in HBase to represent lock for the following reasons:

- If a single row in HBase is used to represent lock, lock simply means writing the log data to a unit of the single-row representing the lock, while unlocking by means of deleting the

data. The atomicity of lock and unlock operation is ensured as the HBase provides a single row transaction.

- HBase is a distributed database, data stored in networks of nodes, and the throughput is related to the bandwidth of the system. That high throughput of HBase is ensured by storing data in distributed clusters result in the throughput of the distributed transaction locks implemented in the HBase.
- Region server in HBase automatically sends the complex load information to HMater, and HMaster assigns the region to the Region server according to the load of this server. Low latency is ensured by a load-balancing function provided by Hbase. This also lead to the low latency of the distributed transaction locks.
- HBase is built on top of the HDFS file system, and HDFS provides the backup function. Each file block in HDFS has three backups by default, which means that if a node in the cluster malfunctioned, there are still two other modes available. And the persistence of HBase results from the Backup function supported by HDFS. The distributed transaction locks implemented in HBase thereby possess the characteristic of persistence.
- Using sing-row of HBase as the distributed transaction lock benefits in a way that the read and write performance of the lock is the same of the read and write of HBase, so the performance of the distributed transaction lock won't be the bottleneck of the system.

Consider the requirements for the lock needed by transaction operation in HBase. The lock itself must be persistent, and the lock service has to provide high throughput and low latency. HBase runs on top of HDFS, which provides storage backup capability for HBase, so as to be persistent. As a kind of distributed database, HBase naturally has high throughput. Although HBase can do load balancing itself, it also has low latency. Therefore, HBase can provide lock service to itself, so an extra column is added into the table to illustrate the locks.

4.3.2 Synchronization of Distributed Transaction

The atomicity of multi-row and multi-table transaction in HBase is guaranteed by two phase commit protocol which involves coordinator and participator. The coordinator can be viewed as both the initiator and a participator of the transaction. Each row contained in the transaction corresponds to a participator. The two-phase commit protocol has pre-commit and actual-commit phases. In the phase of pre-commit, coordinator informs all participators to be prepared for committing or cancelling the transaction. And then the coordinator is notified of the decision made by the participators. While in the phase of actual-commit, coordinator will make the decision depending on the result of the voting in the previous phase. If and only if all participators agree on committing the transaction, all participators will be informed to commit the transaction. Otherwise, the transaction will be cancelled.

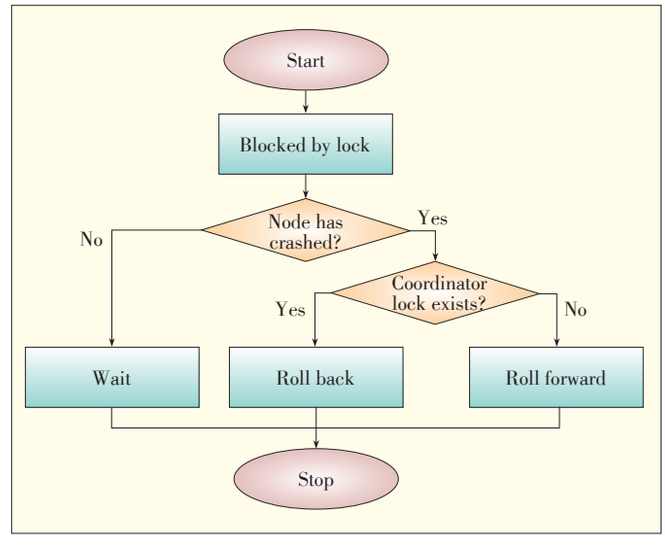
The traditional two-phase commit protocol is adjusted to be more pragmatic and practical:

1) Participants are no longer the traditional cluster nodes but rows in HBase. Because the single-row transaction is supported by HBase, multi-row transaction is the emphasis for the realization of transaction supporting mechanism in HBase. Thus, each row can be abstract as participants, voting on transaction submission respectively.
2) The result of the vote is submitted to the originator of the transaction instead of the coordinator. The entire transaction submission is completed by the user program invoking the transaction for the reason that the user program could collect the vote from the participants conveniently and the rows is not available to communicate with each other.
3) The coordinator is randomly selected, the first the row involved in the transaction is normally selected as the coordinator, and the others the participants. Communicating with participants is no longer the responsibility of the coordinator, which is in charge of recording the state of transaction submitting.

4.3.3 Exception Handling

If there is problem during a transaction, the node processing the transaction might crash, and the transaction cannot be properly completed. Therefore, a lock is left behind (**Fig. 6**). It is intolerable to the system that the left-behind lock is left unhandled and the row involved is locked and invisible.

The key to detecting the left-behind locks is to check whether the corresponding software has crashed or collapsed. Timestamp two, which is obtained from the timestamp server when the transaction started, is attached to the lock held by the transaction and facilitates the process of finding the transaction holding the lock. The Exception Server recodes the information of all running nodes and transactions. The state of the node is quickly checked though ZooKeeper. Furthermore, it is still not



▲**Figure 6. Exception handling process.**

efficient automatically detect the left-behind locks becuase the confliction is rare in the system. Therefore, a shortcut is taken to clean up the left-behind locks.

The left-behind locks are processed by the system in the manner of rolling back. If the transaction crashed before submission, the result of rows, being locked, participating the transaction hasn't been submitted, the main task of rolling back is to clear up the left-behind locks by means of deleting the lock locking the rows participating the transaction. If the transaction crashed during the submission, the results are partly submitted, the rollback operation has to clear the left-behind locks of uncommitted rows and recover the state of committed rows.

The previously mentioned coordinator's role is to record the state of transactions. Whether the transaction crashed before submission or during submission is distinguished according to the state of the coordinator. The transaction of coordinator is finally committed meaning that the lock of coordinator row is finally cleared. If the lock of coordinator row doesn't exist, there is no need to roll back due to complete transaction submission. The lock of the coordinator row contains the row key of other transaction, the other rows will be checked in turn whether the data is updated. If the data existed on the time stamp of a row, the row committed the transaction already and crashed during submission, otherwise the row crashed before submission.

4.3.4 Isolation Level

We implement three kinds of isolation level: read uncommitted, read committed and repeatable reads (**Table 1**).

Also, we use the snapshot isolation provided by HBase to reduce the happening of conflict and improve concurrency. Each transaction will get two timestamps from a timestamp server: one is at the beginning time of transaction and the other is at the committing time of transaction. The beginning timestamp of a transaction determines the data it can read and the committing timestamp determines the data other transactions can read.

5 Experiments and Results

5.1 Load Test

In this section, we discuss the performance of our system for large and highly concurrent access requests. Because clients and the process part of server are both distributed, the FIX server is most likely to be the system's bottleneck. We run a load test to measure the performance of FIX server. We simulated 100 traders, and by modifying the request frequency of each trader, we obtained seven results. **Fig. 7** shows the response time of FIX server. The horizontal axis in Figs. 7 and **8** represents the number of concurrent requests per second. In Fig. 7, when the concurrent number is below 500, the response time is short, and when the concurrent number grows to 1000, the response time increases significantly. However this problem can be solved by a larger HBase cluster.

Fig. 8 shows CPU utilization of FIX server. The CPU utilization rate is grows in a similar way. The linearly increasing of concurrency, resulting in computation scaling linearly, led to the CPU utilization increasing accordingly.

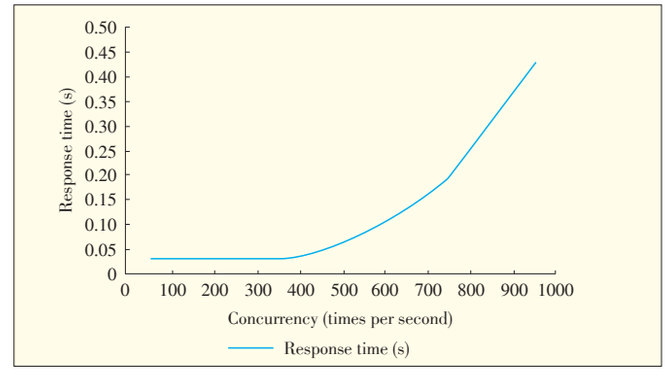
▼**Table 1. Used locks and detected locks in different isolation levels**

Isolation level	Write lock	Read lock	Range lock	Write operation	Read operation
Read uncommitted	V	Null	Null	Checking V and S	Null
Read committed	V	S	Null	Checking V and S	Checking V
Repeatable read	V	V	Null	Checking V and S	Checking V and S

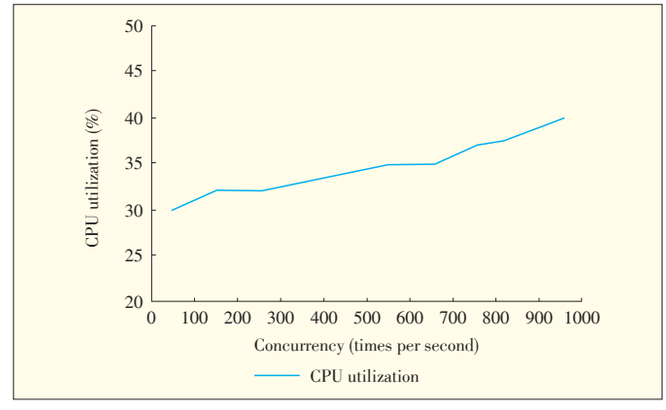
S: shared lock V: exclusive lock

5.2 The Influence Caused by Strong Consistency

As mentioned previously, we provide multi-row and multi-table transaction support for Hbas. In this section, we discuss the influence on the read/write performance of HBase caused by this. We designed four experiments: random write 5M rows in HBase; random read 5M rows in HBase; random write 5M transactions in HBase; random read 5M transactions in HBase. In the last two experiments, each transaction only includes one write or read operation, and this represents the worst case of



▲**Figure 7. The response time of FIX server.**



▲**Figure 8. The CPU utilization of FIX server.**

the transaction. **Tables 2 to 4** show the results.

After providing multi-row and multi-table transaction support, the random read/write of HBase performance decreases.

▼Table 2. Comparison in read uncommitted level

	HBase API	HBase supporting API	Correlation coefficients
Random write time (s)	3424	1041	0.30
Random read time (s)	2285	2202	0.96

▼Table 3. Comparison in read committed level

	HBase API	HBase supporting API	Correlation coefficients
Random write time (s)	3424	992	0.29
Random read time (s)	2285	784	0.34

▼Table 4. Comparison in repeatable read level

	HBase API	HBase supporting API	Correlation coefficients
Random write time (s)	3424	955	0.28
Random read time (s)	2285	786	0.34

Under the read uncommitted isolation level, for no addition operations in read operation, the performance difference between original HBase API and HBase supporting API is negligible. The subtle performance difference results from obtaining unified time stamp from the Timestamp Server after failure. On the contrary, a write operation involves an additional read and write operation, which creates a big performance gap of 0.3 between original HBase API and HBase supporting API.

While at the read-committed level and repeatable-read level, an extra read and two extra write is required in read operation. There is a difference of 0.34 between original HBase API and HBase supporting API in random read operation. There is also a big performance gap as 0.29 and 0.28 between original HBase API and HBase supporting API for an additional read and write involved in write operation.

6 Conclusion and Future Work

We have designed and implemented a futures trading simulation system that has been used by financial professionals to study how individual speculation affects the futures market. Our experiments show that the system is very efficient in the face of large-scale, high concurrency access requests. It is also very scalable and consistent in the face of the large scale-data storage and processing. Our experiments also show that the read/write performance loss of HBase introduced by the transaction support mechanisms is acceptable. We plan to deploy our system to a much bigger cluster and test the system more full-scale. Also in reality, the futures market and the spot market are influenced by each other. In order to make the study of how individual speculative behavior produces an effect on futures market more precise, we should concern the spot market. So implementing the spot market simulating system will be our future work.

References

[1] H. Dawid and G. Fagiolo, "Agent-based models for economic policy design: Introduction to the special issue," *Journal of Economic Behavior & Organization*, vol. 67, no. 2, pp. 351–354, 2008. doi: 10.1016/j.jebo.2007.06.009.
[2] L. Wei, W. Zhang, X. Xiong, and Y. Zhao, "A multi-agent system for policy design of tick size in stock index futures markets," *Systems Research and Behavioral Science*, vol. 31, no. 4, pp. 512–526, 2014. doi: 10.1002/sres.2292.
[3] M. M. Bray, "Futures trading, rational expectations, and the efficient markets hypothesis," *Journal of the Econometric Society*, vol. 49, no. 3, pp.575–596, May 1981.
[4] N. Minar, R. Burkhart, C. Langton, and M. Askenazi, "The Swarm simulation system: a toolkit for building multi-agent simulations," Santa Fe Institute, Santa Fe, USA, Paper 96-06-0421996, Jun. 1996.
[5] M. J. North, T. R. Howe, N. T. Collier, and J. R. Vos, "The repast simphony runtime system," in *Proc. Agent 2005 Conf. Generative Social Processes, Models, and Mechanisms*, Argonne, USA, 2005.
[6] S. Tisue and U. Wilensky, "Netlogo: a simple environment for modeling complexity," presented at Int. Conf. Complex Systems, Boston, USA, 2004.
[7] L. Yu, E. Masabo, L. Tan, and M. He, "Multi-agent automated intelligent shopping system (MAISS)," in *9th Int. Conf. Young Computer Scientists*, Hunan, China, 2008, pp. 665–670. doi: 10.1109/ICYCS.2008.35.
[8] S. Mundle, N. Giri, A. Ray, and S. Bodhe, "JADE based multi agent system for mobile computing for cellular networks," in *Proc. Int. Conf. Advances in Computing, Communication and Control*, 2009, pp. 467–473.
[9] A. Shemshadi, J. Soroor, and M. J. Tarokh, "Implementing a multiagent system for the real-time coordination of a typical supply chain based on the JADE technology," *System of Systems Engineering*, vol. 2, pp. 1–6, 2008.
[10] B. G. Tudorica and C. Bucur, "A comparison between several NoSQL databases with comments and notes," in *10th Roedunet Int. Conf.*, Iasi, Romania, Jun. 2011, pp. 1–5. doi: 10.1109/RoEduNet.2011.5993686.
[11] C. Strauch. (2012). *NoSQL Databases* [Online] Available FTP: http://www.christof-strauch.de/nosqldbs.pdf
[12] D. G. Ferro, F. Junqueira, I. Kelly, B. Reed, and M. Yabandeh,. "Omid: lock-free transactional support for distributed data stores", in *IEEE 30th Int. Conf. Data Engineering (ICDE)*, Chicago, USA, 2014, pp. 676–687.
[13] L. George, *HBase: The Definitive Guide*. Sebastopol, USA: O′Reilly Media, 2011.
[14] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A Distributed Storage System for Structured Data", *ACM Trans. Comput. Syst.*, vol. 26, no. 2, pp. 1–26, Jun. 2008. doi: 10.1145/1365815.1365816.

Manuscript received: 2014-09-10

Biographies

Junneng Nie (njn@sjtu.edu.cn) received his BS degree in software engineering from Shanghai Jiao Tong University. He is currently a MsC candidate at the School of Software, Shanghai Jiao Tong University. His research interests include massive data processing and software testing.

Haopeng chen (chen-hp@sjtu.edu.cn) is an associate professor of School of Software, Shanghai JiaoTong university. He received his PhD degree from Department of Computer Science and Engineering, Northwestern Polytechinal University in 2001. He has worked with School of Software, Shanghai Jiao Tong University since 2004 after he finished his two-year postdoctoral research job in Department of Computer Science and Engineering, Shanghai Jiao Tong University. In 2010, he studied and researched at Georgia Institute of Technology as a visiting scholar. His research group focuses on distributed computing and software engineering, and has kept studying Web Services, Web 2.0, Java EE, .NET, and SOA for several years. Recently, the group is also interested in cloud computing, including cloud federation, resource management, and dynamic scaling up and down.

ZTE Communications
Table of Contents, Volume 12, Numbers 1–4, 2014

Volume–Number–Page

SPECIAL TOPICS

Vehicular Networks

Software-Defined Networking

Wireless Body Area Networks for Pervasive Healthcare and Smart Environments

ZTE Communications
Table of Contents, Volume 12, Numbers 1–4, 2014

ZTE Communications Guidelines for Authors

• Remit of Journal

ZTE Communications publishes original theoretical papers, research findings, and surveys on a broad range of communications topics, including communications and information system design, optical fiber and electro-optical engineering, microwave technology, radio wave propagation, antenna engineering, electromagnetics, signal and image processing, and power engineering. The journal is designed to be an integrated forum for university academics and industry researchers from around the world.

• Manuscript Preparation

Manuscripts must be typed in English and submitted electronically in MS Word (or compatible) format. The word length is approximately 4000 to 7000, and no more than 6 figures or tables should be included. Authors are requested to submit mathematical material and graphics in an editable format.

• Abstract and Keywords

Each manuscript must include an abstract of approximately 150 words written as a single paragraph. The abstract should not include mathematics or references and should not be repeated verbatim in the introduction. The abstract should be a self-contained overview of the aims, methods, experimental results, and significance of research outlined in the paper. Five carefully chosen keywords must be provided with the abstract.

• References

Manuscripts must be referenced at a level that conforms to international academic standards. All references must be numbered sequentially in-text and listed in corresponding order at the end of the paper. References that are not cited in-text should not be included in the reference list. References must be complete and formatted according to IEEE Editorial Style www.ieee.org/documents/stylemanual.pdf. A minimum of 10 references should be provided. Footnotes should be avoided or kept to a minimum.

• Copyright and Declaration

Authors are responsible for obtaining permission to reproduce any material for which they do not hold copyright. Permission to reproduce any part of this publication for commercial use must be obtained in advance from the editorial office of *ZTE Communications*. Authors agree that a) the manuscript is a product of research conducted by themselves and the stated co-authors, b) the manuscript has not been published elsewhere in its submitted form, c) the manuscript is not currently being considered for publication elsewhere. If the paper is an adaptation of a speech or presentation, acknowledgement of this is required within the paper. The number of co-authors should not exceed five.

• Content and Structure

ZTE Communications seeks to publish original content that may build on existing literature in any field of communications. Authors should not dedicate a disproportionate amount of a paper to fundamental background, historical overviews, or chronologies that may be sufficiently dealt with by references. Authors are also requested to avoid the overuse of bullet points when structuring papers. The conclusion should include a commentary on the significance/future implications of the research as well as an overview of the material presented.

• Peer Review and Editing

All manuscripts will be subject to a two-stage anonymous peer review as well as copyediting, and formatting. Authors may be asked to revise parts of a manuscript prior to publication.

• Biographical Information

All authors are requested to provide a brief biography (approx. 150 words) that includes email address, educational background, career experience, research interests, awards, and publications.

• Acknowledgements and Funding

A manuscript based on funded research must clearly state the program name, funding body, and grant number. Individuals who contributed to the manuscript should be acknowledged in a brief statement.

• Address for Submission

magazine@zte.com.cn
12F Kaixuan Building, 329 Jinzhai Rd, Hefei 230061, P. R. China

ZTE COMMUNICATIONS

ZTE Communications has been indexed in the following databases:

- Cambridge Scientific Abstracts (CSA)
- China Science and Technology Journal Database
- Chinese Journal Fulltext Databases
- Inspec
- Norwegian Social Science Data Services (NSD)
- Ulrich's Periodicals Directory
- Wanfang Data—Digital Periodicals