信息通信领域产学研合作特色期刊
第三届国家期刊奖百种重点期刊 | 中国科技核心期刊

ISSN 1009-6868
CN 34-1228/TN

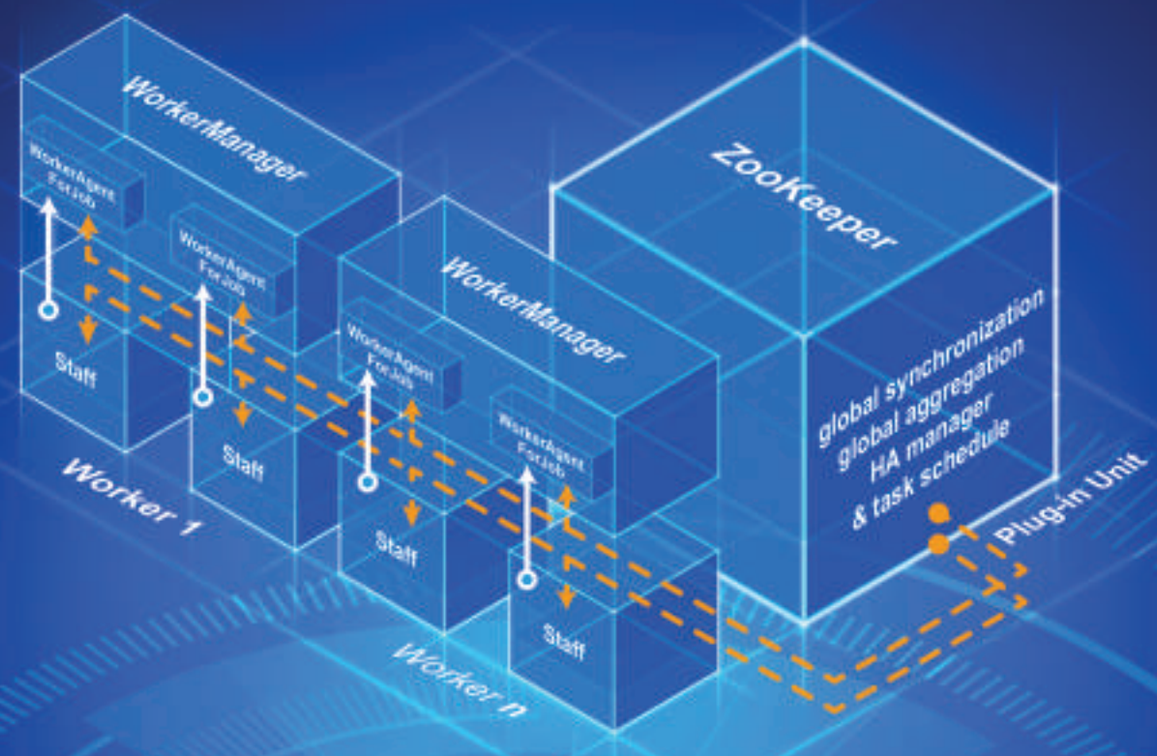
中兴通讯技术

ZTE TECHNOLOGY JOURNAL

www.zte.com.cn/magazine

2016年4月 • 第2期

专题：大数据分析处理与应用



9 771009 686168 04>

中兴通讯技术 ZTE TECHNOLOGY JOURNAL

二〇一六年四月 第二十二卷第二期 总第一二七期

《中兴通讯技术》第7届编辑委员会委员名单

信息通信领域产学研合作特色期刊
第三届国家期刊奖百种重点期刊
中国科技核心期刊
工信部优秀科技期刊
中国五大文献数据库收录期刊
ISSN 1009-6868
CN 34-1228/TN
1995年创刊

办刊宗旨

以人为本，荟萃通信技术领域精英；
迎接挑战，把握世界通信技术动态；
立即行动，求解通信发展疑难课题；
励精图治，促进民族信息产业崛起。

目次

中兴通讯技术 总第127期 第22卷 第2期 2016年4月

专题：大数据分析处理与应用

专家论坛

企业视界

综合信息

期刊基本参数：CN 34-1228/TN*1995*b*16*64*zh*P* ¥ 20.00*15000*15*2016-04

Contents

ZTE TECHNOLOGY JOURNAL Vol. 22 No. 2 Apr. 2016

Special Topic: Analysis Processing and Applications of Big Data

Expert Forum

Enterprise View

敬告读者

2016年第1—6期专题

1 网络空间安全
杨义先 北京邮电大学 教授
杨 庚 南京邮电大学 教授

2 大数据分析处理与应用
郑纬民 清华大学 教授

3 5G技术与业务创新
王 京 清华大学 教授
向际鹰 中兴通讯股份有限公司 博士

4 天地一体化信息网络
张乃通 中国工程院 院士
顾学迈 哈尔滨工业大学 教授

5 工业互联网与智慧工厂技术
邬贺铨 中国工程院 院士
王耀南 湖南大学 教授

6 SDN/NFV的实践与规模应用
蒋林涛 中国信息通信研究院 教授

专题栏目策划人

郑纬民

清华大学计算机科学与技术系教授、博士生导师；长期从事计算机系统结构、大规模数据存储、高性能计算等领域的科研教学工作；主持并完成了“973”、“863”、自然科学基金等科研项目36项，负责或参与工程项目11项；获国家科技进步一等奖1次，奖获国家科技进步奖二等奖2次，获国家技术发明奖二等奖1次；发表论文500余篇，著作10部。

专家论坛栏目策划人

李 涛

2004年7月获美国罗彻斯特大学计算机科学博士学位；现任美国佛罗里达国际大学计算机学院教授、博导，同时担任南京邮电大学计算机学院、软件学院院长，南京邮电大学大数据研究院院长；2006年获得美国国家自然科学基金委颁发的杰出青年教授奖，2009年获得佛罗里达国际大学最高学术研究奖，2010年获得IBM大规模数据分析创新奖；发表文章250余篇。

导读

随着信息和通信技术的飞速发展，人类社会已经步入了一个激动人心的时代——万物互联，虚实（cyber-physics）共长。在这一演化过程中，人们在实验、理论和计算之外，发展出了认识世界和改造世界的第4种手段——数据科学，而大数据则是数据科学在大众中间的一种通俗的说法。

简单来说，大数据描述了这样的一种事实和诉求：数据的巨量、快速增长和多样化，也就是通常人们所提到的大数据的3V（Volume、Velocity和Variety），这方面已经有众多的例子，我们就不再赘述；而如何从数据中提取出价值（Value），丰富人们对于世界的认识，增加人们改造世界的手段，这是大数据的核心所在，最近震撼整个人类社会的引力波确认事件就是一个典型的例子。

大数据蕴含着大机遇，正是看到了大数据的潜在价值，麦肯锡才将其作为创新、竞争和生产力的下一个前沿，世界各国政府纷纷推出相关的计划或战略予以支持。

天下没有免费的午餐，大的机遇也意味着众多的挑战。大数据面临的挑战可以用奥运会的口号——更快、更高、更强——来描述：为了在激烈的全球化竞争中获得优势，为了更好解决当前人类社会面临的种种挑战，人们需要更快地对数据进行分析处理，从中提取出更高的价值（而不仅仅像之前那样只是简单的给出一些统计结果），来更加科学地做出各种决策；要实现上述目标，人们需要站在更高的角度上高屋建瓴的审视整个行业，开发出更强的技术，既包括数据收集、存储和处理相关的各种软件技术，也包括传感器、存储器件以及高性能计算装置等各种硬件设备。

大数据时代，机遇与挑战并存。如何面对这些挑战和机遇，推动整个大数据行业的发展，进而带动整个社会经济和管理、服务水平的提升乃至人类自身的进步，是摆在各行各业人们面前的公共议题。本期专题就是从信息技术从业人员的视角，对大数据发展与应用中的一些问题，特别是技术问题进行了具体地探讨。

另外，在大数据时代，数据来源于应用，实际的应用是根本与目标。数据的产生、收集和管理是基础，数据挖掘（知识发现）是工具和手段，而数据安全和隐私保护贯穿应用的整个过程。大数据的发展为数据挖掘的研究和应用带来了新的特点和需求，同时也为数据安全和隐私保护带来了更加严峻的挑战。本期论坛栏目则对大数据时代的数据挖掘和安全及隐私保护进行了探讨，提出了一些创新的观点，如大数据的核心和本质是应用、算法、数据和平台4个要素的有机结合；信息安全企业未来的发展前景是以底层大数据服务为基础，并通过构建安全大数据，逐步形成大数据的安全生态环境等。

这些论文作者既有来自学术界的资深研究人员，也有来自工业界的前沿领军人物。希望这些探讨能够给读者带来有益的启示与参考，对于推动大数据的发展与应用能够起到一份绵薄之力。

郑纬民　李 涛

2016年2月18日

DOI:10.3969/j.issn.1009-6868.2016.02.001
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20160229.1014.002.html

大数据的开放式创新
Open Innovation of Big Data

吴甘沙/WU Gansha
(驭势科技有限公司,北京 100080)
(UISEE Co.Ltd., Beijing 100080, China)

中图分类号:TP393 文献标志码:A 文章编号:1009-6868 (2016) 02-0002-005

摘要: 大数据是社会从网络化演进到智能化的技术基础,更是未来数据经济的基础资产和货币。认为目前大数据的创新主要局限在技术栈和组织内部,数据的可获得性、处理和分析技术的缺乏以及封闭系统的数据思维成为制约创新的因素。提出大数据开放式创新的要素:通过开放数据及基于数据安全流通和定价的数据市场解决数据供给,开放基础设施及社会化分析服务实现技术共享,最后通过跨领域的开放数据思维获得数据创意。认为开放式创新重构了数据生态,将改变大数据的竞争格局。

关键词: 大数据;开放创新;匿名化;数据定价

Abstract: Big data is the technical foundation of an evolving society, from the networking to intelligent age, and plays the role of critical assets and currencies of future data economy. However, today big data innovation is limited to technical stacks and within the organizations, and suffers from unavailability of data, lack of processing and analytics technologies, and closed-world thinking. This paper discusses key factors of open innovation for big data: unleash the data supply via open data and data marketplaces with secure exchange and pricing, democratize the technologies through open data infrastructure and socialized analytics services, and finally harvest innovative data ideas by "crossover" thinking. Open innovation restructures the data ecosystem and will reshape the competitive landscape of big data.

Key words: big data; open innovation; anonymization; data valuation

大数据创新的最高境界是用构建数据生态来改变竞争格局——数据源解决数据供给,数据创意者从数据中创造价值,而这又有赖于大数据处理和分析技术。在开放式创新的体系中,5种元素扮演3种角色。

- 数据源:开放数据,基于数据安全流通和定价的数据市场;
- 大数据分析和处理技术:开放的基础设施,以及开放的社会化分析服务;
- 数据创业者/应用服务:跨越领域界限的开放数据思维。

它们五行相生,互相作用,形成价值的涌现。

1 开放数据的发展及问题

数据开放的主体首先是政府和科研机构,即把非涉密的政府数据,以及纳税人支持的一些科研数据开放出来。越来越多国家推出了统一的政府开放数据门户。中国在2015年也推出了《促进大数据发展行动纲要》,将开放数据作为工作重点。在开放数据运动的风起云涌之下,现在更多的企业也开始开放数据,实现数据的价值化,并建构生态系统和护城河。

收稿时间:2016-01-12
网络出版时间:2016-02-29

万维网之父Tim Berners Lee提出了数据开放的五星标准[1],以保证数据质量:一星是开放授权的格式,比如说PDF;二星是结构化,把数据从文件变成了像Excel这样的表;三星是开放格式,如CSV;四星是能够通过统一资源标识符(URI)定位每一个数据项;五星是能够跟其他数据链接,形成一个开放的数据图谱。

数据开放与开源软件也形成了共振。主流的数据开放门户,像data.dov,都基于开源软件。Data.gov用WordPress做数据内容呈现,用CKAN做数据目录,甚至data.gov的整个架构也在GitHub开源了。英特尔在麻省理工学院的大数据科研中心研发了开源的DataHub系统,支持对开放数据的多人协作分析,具有数据版本管理和多编程语言交互的能力。

数据开放中会碰到很多问题。

(1)数据权属的问题。数据属于谁?属于采集人?还是属于生产人?抑或是属于被观察的客体?在特定情况下,拥有权如何分割(比如离婚)或者转移(比如继承)?

(2)敏感数据的界定。比如位置信息数据在欧洲属于敏感数据,而在日本不属于敏感数据。另外各个不同行业有进一步规定,比如美国的《健康保险便利和责任法案》对个人

健康信息的隐私性、机密性和完整性做了规定；而在征信领域则有《公平信用报告法》对个人信用方面的信息做了规定。敏感数据需要法律和行业法规的界定。

（3）敏感数据的脱敏。如果开放数据中具有敏感数据，就要做数据的脱敏。脱敏最简单的做法是去标识，但是去标识未必能够彻底脱敏。美国研究显示：即使把姓名、地址等标识信息拿掉，只要有邮政编码、性别、生日等3项信息，就有60%～90%的可能性锁定个人。即使去标识很彻底，仍有“阿喀琉斯之踵（致命弱点）”。一种攻击的方法是通过多数据源的比对来缩小搜索范围，重新标识；另一种方法是基于统计的攻击，比如根据两个打分再加上一定的时间范围约束，还是有接近70%的可能性锁定个人。

（4）防止隐私攻击的匿名化技术。比较典型的如k-anonymity和L-diversity等，但在敏感属性不够多样化，或攻击者具有背景知识时，这两种技术仍不够鲁棒。目前最好的一种技术叫差分隐私，即把噪声加入到数据集中，但仍保持它的一些统计属性，支持特定的机器学习算法。

这些困难和挑战都不能阻挡开放数据运动的深入人心。在数据（尤其是商业数据）仍然无法充分流通的今天，开放数据无疑能够让具有数据思维和分析能力的创意者点石成金，把死的、消耗成本的数据变活、创造利润。

2 基于数据安全流通和定价的数据市场

数据之于数据社会，就如同水之于城市或血液之于身体——城市因河流而诞生，也受其滋养；血液一旦流动停滞，身体就有危险。所以，在数据化生存的今天，一定要让数据流动起来。数据开放更多适用于政府公共数据和纳税人资助的科研数据，而更多涉及私人隐私或企业机密的数据无法通过简单的开放获得。如果把数据看作一座冰山，公开的只是露出海面的一点点，绝大多数藏在暗黑的海面以下。

数据拥有者不愿意把数据拿出来，有两个原因：担心数据被偷窃；对自己并无好处。所以，解决时该问题时需要把握两点：保障数据的安全流通；对数据的使用进行定价，而实现这两个关键的载体是数据市场。

数据市场并非是新概念。早年的综合数据市场多进行原始数据集的下载交易，由于数据容易复制，版权保护困难，这种形态逐渐被几种新的形态取代：

（1）为特定用户定向采集或加工数据，如某公司从事人脸分析技术，委托第3方采集各类、各种姿态和光照条件的人脸数据，或某公司具有大型数据集，需要特定的服务来做标注。

（2）专业领域的数据服务，如交通领域的Inrix或金融领域美国三大征信公司。

（3）不给出整个数据集，只能基于查询或应用程序接口（API）提供数据的受控访问，中国出现的数据交易市场多为此类型。

（4）不给出原始数据，只交易加工信息，这是之前大数据时代的主流，有些公司（如彭博社）甚至提供专门的终端保证信息服务。

随着数据生态的完善，数据市场的形态将更为丰富。首先，上述形态多为数据提供者与数据请求者的简单交易关系，而未来市场的参与者可能同时是提供者与请求者。其次，交易将不仅是简单的“给”和“得”，而是融合、使用从而产生新的衍生价值。因此，数据的定价不是那些比特的固有价值，而是在这一次“使用”中产生的当前价值。数据市场应该是使用和买卖一站式服务，并且是先使用再买卖。

Steven Johnson的TED演讲《伟大创意的诞生》是从咖啡馆说起，它创造了一个安全的空间，让不同的人做思想碰撞，创造新的想法。数据何尝不需要这样一个咖啡馆，让各方的数据能够产生“化学作用”。“数据咖啡馆”项目[2]基于多方安全计算，试图解决3个问题：安全可控的开放；数据市场和云计算的一体化；数据定价的问题。

然而，绝大多数数据的价值是不确定的，这正是数据的外部性。这种属性决定了数据与石油本质上的区别：石油的价值在燃烧的一瞬间实现并消失了，但数据能够反复使用，产生不可预期的新价值。基于Moody的信息估值七律，可以衍生出数据估值七律：

（1）数据可以被无限次共享，可以产生更大的总体价值，但多次复制会使所有权复杂化，增加成本；

（2）数据用得越多，价值越大；

（3）数据价值会随时间衰变；

（4）数据越精确，价值越大；

（5）多个独立数据源的融合为1+1>2；

（6）更多的数据不见得能带来更多的价值；

（7）数据不会损耗，反而会越用越多。

这些基本原则对数据的定价具有指导意义——数据的使用频度、新鲜度、质量、外部性等都是重要变量。Glue Reply公司据此提出了基于使用的估值模型。

另一方面，Gartner分析师Doug Laney——大数据3V的提出者，把信息和数据的估值模型分成非金融模型和金融模型。

我们期待未来的数据市场有灵活的数据定价模型，该模型既考虑数据的使用历史和时间嬗变所形成的基础价值，又能计量当前的这次租用中可量化的价值，计算出这次交易的数据定价。同时，如果这次使用有多方数据参与，根据各方在计算中贡献的大小，对其数据分别进行定价。

数据的安全流通和定价将鼓励

数据拥有者将其数据参与流通，对其数据价值化、货币化和资产化，从而形成“收集-使用-价值化-更多收集-更多使用”的正向反馈，为开放式创新提供更广泛的原材料供给。

3 开放的基础设施

笔者的同事 Eric Dishman 罹患肾癌 23 年，尝试了各种治疗方案，甚至换肾，一直没有进展，直到他选择了基于基因分析的精准治疗。整个测序和锁定致病基因片段的过程花了 3 个月；接着，数 TB 的基因数据被拷到硬盘里，在美国东西岸传来递去，颠簸了 4 个月以后方才形成了治疗方案。虽然他现在已经恢复健康，但 7 个月的等待对于任何一个病人来说都是煎熬。

原因很简单，对于专业的医疗健康和生命科学机构来说，计算和存储的基础设施并不是他们所擅长。要知道，就连大数据领域内部也是隔行如隔山，做数据分析的人很难理解分布式的存储和处理系统。事实上系统部署的困难已经成为目前拦在大数据产业前面的一座大山。

要致富，先修路（基础设施）。在现实生活中的这个朴素道理也适用于大数据。基于云计算的公共基础设施，特别是大数据系统作为平台服务，是搬走这座大山的希望所在。在其他的一些国家，很多以数据思维见长的小型创新企业已经开始受益于这一趋势。

Decide.com 是笔者一直关注的一家创业公司（后被 Ebay 收购）。它每天吸入几十万条商品价格数据以及相关的新闻（这也是开放数据），分析后告诉顾客买什么牌子、型号以及预测何时买最划算。在其神奇的背后，只有 4 个博士精心调制算法，他们不用担心基础设施的问题，因为亚马逊已经把计算和存储能力作为基础设施开放出来了。

Prismatic 是另一家创造神奇的公司，它能读懂用户关心什么，发掘用户新的兴趣，实时地、个性化地推荐阅读。这家公司在很长一段时间内只有 4 个员工，3 个是学生，然而估值已经达到好几亿美金。之所以能够把神奇的数据思维变成现实，同样要感谢亚马逊的云计算把脏活累活都干了。

把大数据系统装在云上是第 1 代大数据奋斗者的梦想。早在 2007 年，Hadoop 解决方案的领导者 Cloudera 成立伊始，就已经在憧憬这一愿景（从 Cloudera 这个名字可以看出）。然而，这条道路并不顺利。

首先，把 Hadoop 这样的重型系统跑在虚拟机里是很大的挑战，大数据这样的输入输出（IO）密集型应用与虚拟化技术有点“水土不服”，性能下降严重。经过业界和社区多年的努力，这如今已经不是问题。而像 Spark 这样的新贵是生在云里，长在云里，与云相得益彰。

其次，对于大数据的早期用户来说，把数据放在云里是有疑虑的，一来大数据的搬动太过困难；二来数据安全没有保障。这些年来，云计算的积累效应悄然间改变了数据生态，越来越多的数据一生下来就在云里。而对于初尝云滋味的客户，亚马逊甚至专门设计了容量达 50 TB 的、可托运小型存储设备帮助他们把数据搬到云里。Spark 的商业化推动者 Databricks 也顺势与亚马逊结盟，在其 AWS 云服务上部署 Databricks 云，可以利用大量已经存在于亚马逊云的数据，这真是一个妙招。

而数据安全的保障有赖法律法规、行业自律和技术推动三箭齐发。目前关于大数据权利的立法已在酝酿之中，行业规范更是走在前列（如第 1 节所述）。在行业自律上，我们看到了阿里云发起的《数据保护倡议》。然而，没有技术推动，法律法规和行业自律会制约大数据的云部署。本小节开始讲的基因数据在磁盘里周游世界的故事，还是会一再重演，因为美国的《美国健康保险便利和责任法案》对数据在网络上的传输施加了很多限制。

Eric Dishman 的癌症经历引起了计算机科学家的深思。男性有一半的几率罹患癌症，女性的几率也达到 1/3。相比之下，过去 50 年癌症的治愈率只提升了 8%，在各种疑难重症中进步最小。究其原因，癌症作为一种长尾病症，需要足够多的数据样本才能有所突破，而《美国健康保险便利和责任法案》等法规对于数据共享的限制使得各大科研机构只能各自为战，相对较少的数据样本制约了生命科学技术的发展。

想象一下，如果第 2 节中所谈的多方安全计算技术能够使数据在法规允许的范围内共享和互通，癌症研究将大不一样。鉴于此，英特尔和俄勒冈健康科学大学等科研机构开始陆续推动基于安全多方计算的协作癌症云。

我们预计：随着云观念越来越深入人心，大数据和高性能计算在云中的部署将呈现加速之势。这时候，云作为一种开放基础设施的优势将得到充分展现。

还是回到 Eric Dishman 的案例。历时 7 个月的诊断过程固然有数据磁盘在路上的延误，另一个重要原因是计算基础设施的缺乏。在生命科学领域中（尤其是生命信息学），非常罕见地呈现了高性能计算和大数据分析齐头并进的态势，寻常的科研院所无法维护完美支持两种运算的基础设施。

可以想见：未来的几年中融合高性能计算和大数据分析能力的云基础设施将变得普及。我们有一个雄心勃勃的愿景：到 2020 年，像 Eric Dishman 这样的患者，一天之内就能完成全基因组测序，锁定致病基因，且形成个性化用药和修复方案。相比起他 7 个月的经历来说，计算能力与时俱进的开放基础设施能缩短数百倍的等待时间。另一个非常热门的领域——脑科学研究如今也面临

计算力有不逮的局面，一次功能性核磁共振对大脑的完整数据采集将获得500～600 GB左右的数据，而对其进行完整的分析耗时6 h。我们期待2020年这个工作将在1 s内完成，也就是说，能够对脑部活跃成像做一些实时的分析，这对脑科学和类脑计算的研究来说将打开一扇前所未有的大门。

4 开放的社会化分析服务

《哈佛商业评论》说数据科学家是21世纪最性感的职业。而麦肯锡认为:2018年前美国这类人才的缺口达到数十万，特别是能够做深度分析的分析师有50%～60%的缺口。也难怪，一个合格的数据科学家必须精通数理统计和计算机科学，对数据敏感，对业务理解。现有的计算机科学或数学的教育体系，无法批量生产这样的人才。我们看到基于慕课（MOOC）的数据科学课程获得了数百万学生的参与，很多大学开始推出在线数据科学课程和学位，相信基于互联网的新型教育体系将在人才供给中扮演更重要的角色。但是，短期内人才饥渴是非常现实的问题，这对于矢志立于大数据潮流之巅的企业来说，不免英雄气短。

与此同时，一股轰轰烈烈的资源革命在互联网卷过，共享经济充分利用互联网将闲散资源与需求对接，解决了供需失衡的问题。设想数据科学家的技能和时间也是一种资源（克莱·舍基将其称作“认知盈余”），应该也能够在这一框架下提高使用效率。这就是所谓的开放的社会化分析服务。

这种服务对我们的社会来说并不陌生。某种意义上，这是一种古老智慧“悬赏”和现代“众包”思维的合体。1714年，英国议会悬赏20 000英镑的“经度”大奖促使一个钟表匠发明了航海天文钟，完全改变了航海史和征服史。18世纪，拿破仑悬赏12 000法郎征集储存食物的方法，促使一个商人之子发明了罐头。近现代史上这样的悬赏还有很多，比如跨大西洋飞行、月球车、宇航员手套等。另一方面，众包完全改变了当代知识的生成和解决问题的方式，比如维基百科。

那么，开放的社会化分析服务该如何工作呢？下面我讲几个故事。

Netflix在2006—2009年之间向大众发起数据分析挑战赛，希望能够通过预测用户星级评分来提升推荐引擎的效率，目标是提升10%，为此设了百万美金大奖，吸引了全世界180多个国家4万多支团队来参加。非常可惜的是Netflix没有采用第1名的算法。那这个比赛是否没有价值呢？不然，大数据生态系统中最受关注的Spark平台正是因为这个比赛形成了灵感和最早的原型。大赛的价值往往不在赛场里。

第2个故事关于休利特基金会。它征集一个对学生的短论文进行自动化评分的算法，因此设立了10万美元奖金的Automated Student Assessment Prize。第1轮大赛先向十多家专业的教育科研机构开放，而第2轮则是在Kaggle平台上向社会开放。Kaggle坐拥数十万具有专业知识和自由时间的分析师，而具有数据分析需求的企业只要把数据和挑战赛规则放到网上，分析师们就可以八仙过海、各显神通、一较高低。结果出人意料，这些业余爱好者搞出来的算法，远胜于专业机构的算法。更让人大跌眼镜的是前3名获得者分别是美国一位机械工程专业的本科生，斯洛文尼亚一位计算机系的博士生，和新加坡一位39岁的保险精算师。第1、3名获奖者刚刚从Coursera慕课平台上学完了斯坦福机器学习的课程，刚刚学完去参赛，就摘得桂冠，这是非常颠覆的。Netflix大赛的获奖团队都是高大上的科研人员，包括两个AT&T的研究主管，而这次竟然让几个初通机器学习门径的学生拿到了大奖。竞赛改变了学生的命运，第1名转向了数据科学专业，而斯洛文尼亚和新加坡的两位优胜者在美国找到了职业发展的巨大空间。

第3个故事是关于一家很小的初创公司Jetpac，它在IPAD上做一个关于旅游的应用。这个公司非常小，做技术的两个人，一个CTO，另一个是程序员，他们希望有一个自动化的算法在很多照片中筛选出最好的照片。但两个人学识有限，于是他们在Kaggle平台上搞了一个比赛，因为资金有限，就出了5 000美金，没想到还是吸引到了400多支团队参赛，最终他们确实选到了一个合适的算法，让这个应用脱胎换骨。Jetpac马上就拿到了240万美金的风险投资，他们的精明之处在于：利用社会的资源为其贡献才智，换来资本的青睐。

对于当前的“大众创业、万众创新”，数据科学的专业性门槛必然导致洛阳纸贵；而这样的思想众包平台将解决数据智慧的短缺，提升众创的成功率。

大家试想，Kaggle这个平台，也就数十万注册用户，咱们中国毕业生每年都是千万，学科学工程专业的也有好几百万，在中国可资利用的社会化分析力量一定更为强大。

鉴于此，中国计算机学会大数据专家委员会主办了“中国好创意”全国青年大数据创新大赛。首先，它是学生学习数据科学，切磋数据分析技术的平台；第二，像中国好声音一样，它一定是年轻人展现自己的平台，就像吴晓波所言，这个时代是无名山丘崛起为峰的时代，这个时代需要这么一个平台；第三，操作系统BSD的发明人Bill Joy提出了Joy定律：在这个时代，无论公司再牛，世界上最聪明的绝大多数人都是为其他人工作的。那么最好的办法就是打开组织的边界，让组织虚拟化，让世界上成千上万的人帮忙你解决难题。同时，对于数据科学家/工程师来说，数据分析能力将成为其行走江湖的独特品牌，纵横于不同企业之间，最大化

其价值。

5 跨领域数据思维

2013年，一种病毒在上海和安徽爆发，国家派出了很多工作组，前往各个现场采样，对10 000个样本进行分析。他们寻找的是H7N9禽流感病毒。笔者当时在想，我们的生物科技人员要是有大数据思维多好！早在2005年，Craig Ventor——这位被称为“科学界Lady Gaga”的奇人，已经在对纽约的空气做全集的基因组测序。如果对源头菜市场的空气做全集的检测，不正是大数据全集思维相对于采样的优势吗？Ventor的跨界思维并不止于此。2014年，他的创业公司“人类长寿”从Google挖走了顶级计算机科学家，谷歌翻译首席科学家Franz Och。在这里，Franz将运用大数据去解密人类基因组的奥秘[3-5]。

同样，生物科学的思维也能帮助大数据。百度首席科学家吴恩达，曾经一度迷惘人工智能走进了死胡同：识别杯子需要一种算法，识别人脸又是一种算法，识别汽车还要一种算法，似乎永远无法穷尽人的智能。直到有一天，神经科学方面的最新进展让他大开眼界：科学家把大脑皮层负责听力的区域与听力器官的神经连接剪断，连到视网膜，过了一段时间，这部分区域竟然能够形成视觉理解了；同样，负责触觉的区域也可以被训练成具有视觉功能。吴恩达获得了顿悟：原来人脑只有一套算法实现各种认知功能，从此他走上了深度神经网络的研究之路。

Farecast.com是人工智能学者Oren Etzioni开的一个创业公司（后被微软的Bing收购），他携数据思维切入了航空公司白热化的价格竞争之中。通过洞悉机票随季节、燃油价格、天气状况甚至特定事件的变化趋势，他推出了机票价格预测服务。如果到此为止，这不失为一个精彩的跨界数据思维案例，但真正使其成为经典的是：在预测服务后Farecast.com增加了10美元的“Fareguard”保险服务，如果购买后一周内价格下跌，公司将补足差价。

前文的另一个案例Decide.com帮助顾客预测某个商品何时买最划算。同样，Decide.com对于某些商品提供价格保险，如果消费者购买后一段时间内商品降价，那么公司会补偿差价。

The Climate Corporation把气候学和农艺学揉在一起，告诉农民播种的时机，或为恶劣天气做好准备。真正天才的创意在于：他们把保险业引入到三角关系中——通过微气象建模预测异常气候的发生，帮农民办理保险，并在气象灾害发生后，自动理赔、打款。当气候学、农艺学和金融学以一种全新的方式组合在一起，造就了一家10亿美元的公司。

读者从上述的3个例子能够读出什么？

大数据的预测分析和保险是完美搭档，创造了新的商业模式。推而广之，大数据的预测分析与金融也能产生很多新的商业机会，因为金融本质上就是跨越时空的价值交换，而大数据则能够发现时空之间的价值剪刀差。这毫无疑问也要拜跨界思维之赐。

相比信息，数据的价值有很高的外延空间。信息的意义是明确的，价值也是确定的。而数据有外部性，它因为某种目的被采集，又可以无限服务于新的目的。克强指数采用的3个数据——耗电量、铁路货运量和贷款发放量，都不是为衡量经济运行状况而设计的，然而总理跨界的数据思维使其能够反映中国的经济全貌（必须指出，这些数据反映的更多是重工业运行情况）。同样，智能电表采集的社会用电情况不经意间反映了房屋空置比例。数据的这一奇特特性亟需跨界思维去挖掘。

在大数据的开放式创新中，不只是需要技术的开源，更需要思想的开源。如果能够把世界各地、各行各业的跨界数据思维及其实践内容档案化，加入检索功能，数据智慧就能得到积累和传播，真正让大数据之光普照大众、惠及我们的地球和城市。

6 结束语

文章从5个方面阐述了大数据的开放式创新。我们期待通过开放式创新，中国能够出现一万个、十万个甚至百万个数据思维公司，他们如群星般璀璨，秉持知行合一，或净化环境，或改善民生，或推动产业转型升级，或提升社会治理，形成一股巨大的力量，实现大数据在中国的繁荣！

参考文献

[1] Linked Data [EB/OL]. (2016-07-27)[2009-06-18]. http://www.w3.org/DesignIssues/LinkedData.html

[2] 吴甘沙.大数据技术发展的十个前沿方向[J/OL].大数据,2015(2) [2015.08.28]. http://www.j-bigdataresearch.com.cn/CN/10.11959/j.issn.2096-0271.2015023

[3] MOODY D, WALSH P. Measuring the Value Of Information: An Asset Valuation Approach [C]// Proceedings of Seventh European Conference on Information System (ECIS'99), Copenhagen Business School, Frederiksberg, Denmark, 1999

[4] Reply. The Valuation of Data as an Asset: A Consumption-Based Approach[EB/OL]. [2014-04-22]. https://www.reply.eu/Documents/13903

[5] LANEY D. Why and How to Measure the Value of Your Information Assets [EB/OL]. [2015-08-04]. https://www.gartner.com/doc/3106719/measure-value-information-assets

作者简介

吴甘沙，曾任英特尔中国研究院院长，现任驭势科技有限公司CEO；研究方向为大数据系统架构、隐私保护/多方安全计算、数据货币化。

DOI:10.3969/j.issn.1009-6868.2016.02.002
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20160302.1129.002.html

试论大数据之“大”

A Commentary on the “Big” of Big Data

李廉/LI Lian
(合肥工业大学 计算机与信息学院,安徽 合肥 230009)
(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)

中图分类号:TP393 文献标志码:A 文章编号:1009-6868 (2016) 02-0007-004

摘要: 认为大数据提供了一种全新的认知世界的角度和方法。与熟知的数学和大部分物理学的基本认知规律不同,大数据分析原则上是一种基于观察和归纳的经验主义认知,这种方法曾一度被现代实证主义的研究模式边缘化。随着近年来大数据产生与分析的技术进步,这一古老方法正在重新焕发活力,并赋予大数据新的内容和形式。在这个意义上,给出了关于大数据4V的新解释。同时通过一个NP问题的例子,探讨了大数据对于复杂问题解决的新方法和新思路。

关键词: 大数据;观察归纳;概率近似正确;数据分布;数据清洗;数据价值;例证法

Abstract: Big data provides a brand-new angle and method of perceiving the world. Like mathematics and physics, big data analysis is, in principle, a methodology based on observation and empirical induction, which has been marginalized in recent times by positivism in research models. As techniques for big data creation and analysis have developed, this methodology has blossomed. We give a new explanation of the “four Vs” of big data: state the four Vs here. We also discuss an example of an NP problem to explore new methods for solving complex.

Key words: big data; observation and induction; probability approximately correct; data distribution; data cleaning; data value; exemplification method

1 大数据的应用目的

毫无疑问,对于大数据的分析与处理,目的是要获取知识,或者说认知结论。那么,通过大数据来获取知识,与大数据时代之前获取知识有什么不同吗?为此,我们需要回顾人类直接从自然界获取知识的两种手段:观察和实验。

早期人们获取知识的手段是观察,通过对于自然现象的仔细观察,得到关于自然规律的认知。由于观察本身没有干预自然的运行,因此可能会受到众多因素的干扰而影响认知的质量,甚至得到不正确的知识。16世纪之后,由伽利略等逐步开创了现代实证主义研究的手段,这种研究需要预设因果关系,然后在实验室里进行现象重建。由于在实验条件下,干扰因素被抑制到最小,因此可以准确重现现象之间的因果。实验与观察的区别是:实验需要预先假定一种或者多种因果现象,然后在实验室设计适当的实验来重现这些现象,从而证实因果关系。实验并不特别依赖研究人员的直观经验,而且具有很强的说服力。观察是需要在众多的现象之间,找出其中的因果关系。这里面并没有什么统一的方法和标准,因此通过观察得到结论需要直观和经验,同时说服力往往也不够。在实证主义的研究体系建立之后,观察研究就让位于实验,除了少数的学科(例如宇宙学),在绝大多数自然学科中,实验成为形成结论的标准手段,任何结论必须在实验室里面被验证,仅仅在自然界被观察到是不够的。究其原因,还是因为历史上由于观察手段的不足,难以获得大量数据,而建立在小数据基础上的观察,往往是不准确的,得到的结论也缺乏说服力。例如通过观察,人们最容易得到的结论是地球中心论,这种学说统治了科学界1 500多年。只是到了开普勒、哥白尼时代,随着观察数据的增加,才能够颠覆以前的结论,重新建立新的学说。这说明:观察研究这种人类最基本的研究手段,其结论的可靠性依赖于是否有足够的观察数据,当数据多到一定程度时,所获取的结论才具有可靠性。因此一个重要的问题出现了:对于一个具体的观察对象,数据量达到多大时,我们才能采信所获取的结论呢?

既然过去是受限于数据的不足,使得人们研究自然问题主要依赖于实证主义的实验方法。那么现在随着信息技术的发展,获取数据的能力有了极大提高,进入了大数据时代。我们是否可以重新回到先辈那里,采用观察的方法来研究问题,获取知

收稿时间:2016-01-17
网络出版时间:2016-03-02
基金项目:自然科学基金(61370219);广东省佛山市创新团队项目(2015IT100095)

识？这个不是可能不可能的问题，而是已经在我们身边发生的事实。在人文科学、社会科学、自然科学等领域已经开始采用大数据来进行研究，产生新的知识，这些新知识极大地丰富了我们对于自然和社会的认知，有许多成果是依赖试验方法无法想象的，其中最典型的例子可能是图像识别和语音分析，在基本无法通过实验来重构现象的人文社科领域更是如此。通过观察设备（传感器）作用于各种自然现象、社会活动和人类行为，产生了大量的数据，分析和处理这些数据就是对这些观察结果的归纳和提炼；因此通过大数据来认知各种自然的、社会的和人文的规律，是传统意义上对于观察研究的新提升和新表现。人们研究科学的手段又重新回到了观察这个最原始和最基本的手段，但是这一次的回归是螺旋式上升，比起张衡和托勒密时代的观察完全不在一个层面上。从古代依靠人的感官来观察现象，到现在依靠传感器来观察现象，数据的密度、广度、准确性和一致性已经不能同日而语了，因此观察这种研究手段在信息时代换发了新的生命力，成为新时代的科学研究方法。

2 大数据的定量化

大数据是与观察研究密不可分的，大数据分析和处理的目标是获取知识，得到结论。那么怎样从大数据得到的结论呢？在小数据时代，这需要经验和直观。在大数据时代，需要应用计算机来进行分析和处理。一般来说，大数据分析是一种归纳的方法，因此必然具备归纳方法的普遍特点，即通过大数据获取的结论具有某种不确定性，这就是数据分析理论中常说的概率近似正确（PAC）[1]。确切地说，一个结论概率近似正确，是指该结论能够以$1-\delta$的概率获取，并且具有误差ε（类似于机器学习里说的泛化误差）。也就是说：我们通过大数据来获取知识，不能保证每次都能够正确获取，而且获取的知识也不能保证绝对正确。δ和ε这两个数，反映了使用大数据获取知识的能力和精度。这是所有归纳分析的共同特点，也是观察研究的固有性质。这一点既可以说是优点，又可以说是缺陷。优点是这样可以保证我们至少获得一个接近真理的结论；缺点是我们不能期待获取绝对正确的结论。如文献[2]中所说：“当我们掌握了大量新型数据时，精确性就不那么重要了，我们同样可以掌握事情的发展趋势。大数据不仅让我们不再期待精确性，也让我们无法实现精确性。然而，除了一开始会与我们的直觉相矛盾之外，接受数据的不精确和不完美，我们反而能够更好地进行预测，也能够更好地理解这个世界。”

但是问题到此远没有结束，反而是刚刚开始。和古代的科学家不同，在大数据时代，我们需要回答这样一个问题：给定任意的δ和ε，为了在大于$1-\delta$的概率下得到一个误差小于ε的结论，我们需要多少数据？如果能够回答这个问题，哪怕是在某种程度上回答了这一问题，我们就超越了古代科学家凭经验和直观做出结论的限制，真正把获取结论的过程建立在客观和科学的基础上，这样得到的结论自然也就有了很强的说服力。

为了更加仔细考察从大数据获取的知识的过程，从中得到方法论的一些结果，我们需要明确一些概念。

第1个概念是样本和分布。从观察现象得到的数据并从中来获取知识，首先需要解决的问题是得到的数据不可能是所有的数据，我们能够得到的数据永远是客观上整体数据的一部分。显而易见，只有明确知道样例数据与整体数据之间满足的分布假设，从样例来获取知识才具有可靠性和准确性。其中最受关注的就是样例集合与整体数据之间具有何种分布状态，同分布自然是理想状态，但是也已经发展了一些方法来讨论非同分布的情况[3-4]。

第2个概念是数据的清洗。观察是现象的记录，并且从记录的数据来获取结论。数据都是具有属性的，如果属性与期望的结论之间没有可关联的关系，那么数据只是一堆随机的噪声而已。在小数据时代，我们主要靠直觉和经验来筛选属性和处理数据，使得从处理后的数据能够有效地得到结论。在现代大数据分析和处理过程中，发展了一些自动或者半自动的方法来进行处理。

第3个概念是获取结论的成本。从计算机科学的角度，是指获取结论所花费的时间复杂度和数据空间复杂度，主要是时间复杂度。

综上所述，在大数据背景下获取结论，与数学和大部分物理学的结论形式不同，采用了概率近似正确的概念，并由此建立结论的获取方法和标准。实际上，由于观察得到的数据总是局部的和不完整的，所以通过观察得到的结论原则上都是PAC形式。

现在我们可以讨论一个有意义的问题：预设一个目标结论以后，需要多少数据量才能以PAC的方式得到该结论。这个问题无疑是大数据研究中最重要的内容之一。在小数据时代，对于这个问题并没有特别关注，因为通过数据来获取结论是借助直观和经验的，数据量的多少对于能够得到结论没有直接的联系，一个聪明人只要少数的几个例子就可以“猜”到结论，而对于一般的人来说，再多的例子也无法从中得到结论。但是在大数据时代，由于是通过设计算法，借助计算机进行数据分析，因此数据量的多少自然会对于结论的产生和结论的正确性具有直接的关系。由于大数据的研究才仅仅起步，对于这个问题目前上没有一般的结果。但是在附加一些不太苛刻的条件之后，却有一个出乎意料的结果，这就是Blumer等在1989年得到的一个定理。

定理1（Blumer定理）[5]：设D是实例的集合，S是样例的集合，H是目标

函数，A是算法，如果：

（1）S与D具有相同的分布；

（2）H是一个二分类函数；

（3）H在算法A的假设空间中。

那么，对于任意给定的δ和ε，当数据量N满足

$$N \geqslant \frac{1}{\varepsilon}\left(4\log_2\frac{2}{\delta}+8VC(\aleph)\log_2\frac{13}{\varepsilon}\right) \quad (1)$$

可以在期望$1-\delta$内，得到函数G，并且G与H的误差不超过ε，即以PAC的模式得到函数G。其中VC($\aleph$)是算法A的假设函数空间$\aleph$的VC维数。

我们经常说大数据有4个V，即体量（Volume）、高速（Velocity）、多态（Variety）和价值（Value）。这些V反映了大数据的特点，但是究竟达到什么程度才叫做大数据，需要有一个量化的讨论，否则大数据就仅仅是一个笼统的概念。

结合前面的讨论和定理，我们尝试给出一种大数据的量化的解释。首先要指出的是：数据量大不大是依据所要得到的结论性质而言。对于一个工厂的产品检验来说，可能几百个抽样（观察）数据就足够了，但是对于暗物质的探测，可能几个P的数据量也未必够用[7]。这说明谈论数据量之大小，脱离了目标是无意义的。

定理1指出：在给定目标（包括预设的结论形式和精度，即δ和ε）的前提下，当数据量达到一定程度后，就可以按照PAC模式得到结论。因此我们可以把Blumer定理中的N的倒数$1/N$定义为数据的价值密度，这就给出了4个V中Value的量化定义。在数据平等的前提下，每一个数据相对于期望结论与相应算法，它的价值就是$1/N$。同样的数据对于不同的期望结论和算法，其价值是不同的。同时根据该定理，可以定义N为解决问题所需要的最小数据体量，即Volume。当数据量达到N时，就可以称为关于期望结论和相应算法的大数据。由于这个数量的巨大，因此如何存储和处理海量数据是重要的技术问题。对于另外两个V：Velocity是指需要有快速存储技术和计算技术来接纳和处理高速涌入的数据，但是也可以看作是最小数据体量与问题解决时间要求的比值，这个值决定了数据处理的最低速度；Variety是指数据的来源和类型很多，对于问题解决而言，这种多态性取决于数据清洗的质量。

一般来说，数据的多态性越丰富，越是会有利于数据的整理和表现，也越会容易得到结论，对机器学习的语言来说，越容易保证目标函数在假设集合中。当然，数据的多态性会增加数据获取和整理的难度，因此需要在数据处理的成本和效率之间加以折中[8-10]。

3 1个NP复杂类的例子

上面已经讨论了如何通过大数据来获取结论，以及获取结论的精确性和可靠性问题。在这一节，我们继续通过1个例子来说明这个问题。

一个NP问题是指一台非确定图灵机在多项式时间可以解决的问题。NP问题能否具有确定的多项式算法是一个长期以来未能解决的重要问题。现在我们通过大数据的思维方式来探讨此类问题，寻求新的解决问题思路。

定理2：对于任意的NP语言类L，以及给定的n、δ和ε，则存在一个算法A，当随机抽取的样例个数超过了

$$N=\frac{1}{\varepsilon}\left(4\log_2\frac{2}{\delta}+f^2(n)\log_2\frac{13}{\varepsilon}\right)$$

时，可以期望$1-\delta$获取一个确定的函数，该函数对每一个长度等于n的x，计算$x \in L$？误差不超过ε。并且N多项式（实际上是平方）依赖于n，$1/\delta$和$1/\varepsilon$。

这个定理只是一个理论上的结果，因为即使当n=100，δ=0.05，ε=0.01时，需要的样例个数也达到了8 000万这样的数量级。对于这么多的样例，需要进行标注，即一个个注明它们是否属于L，本身就是一项十分费力的事情。但是该定理却表现了通过大数据分析获取结论一些规律。首先该结果表明了通过一些例子的分析，就可以得到一般性的结论（具有一定的误差）。对于非确定语言L而言，不需要去构造相应的图灵机，只需要计算一定数量的样例，同样可以某种概率得到一个判断函数H，在误差ε的范围内判断是否$x \in L$？大数据给我们带来的一个重要方法论正是在这个意义上的，通过对大量的观察数据的分析和处理，可以得到原来只有实验验证和逻辑推理才能得到的结论。这种模式在古代就存在，但是后来被更先进的实证主义的研究方法所取代，而大数据的出现重新召回了它的灵魂。

通过例子来证明问题，这个方法在80年代就被洪加威等研究过[11]，称为例证法。在小数据时代，例证法需要经过仔细挑选的特殊例子，在大数据时代，可以通过大量的数据来取代这个苛刻的条件，因此大数据的出现将例证法推到了几乎可以在所有领域应用的地步。这对于过去只靠实验和逻辑证明问题而言自然是开创了一个新时代。

4 结束语

大数据提供了认识世界的新方法和新角度。有别于我们习惯的实验验证和逻辑推理方法，大数据定义了通过观察和样例获取结论的模式，这种模式古已有之，而且是人类研究自然的最古老的方法。大数据的出现使得这一方法重新焕发活力，并且赋予了新的内容和形式。由于大数据本质上是通过观察来获取结论，因此和所有采用观察方法研究问题（无论是否采用大数据分析）具有相通之处，所获取的结论具有某种不确定。在当前讨论的大数据分析方法中，这种不确定性主要表现在两个方面：一个是获取结论的可能性，一个是结论本身的可靠性。同时，获取结论的不确定性可以在某些条件下任意逼近

确定性。正如舍恩伯格所说：这种不确定性不是表示大数据分析不如物理学和数学，而是说明大数据提供了一种新的认知世界的模式。

大数据分析并不排斥传统的物理学和数学的研究模式，相反，大数据分析建立的关联关系可以为因果关系和逻辑关系的研究提供佐证和启示。

参考文献
[1] MITCHELL T. Machine Learning [M]. 曾华军, 译. 北京: 机械工业出版社, 2008
[2] SCHONBERNER V. Big Data：A Revolution that Will Transform How We Live, Work and Think [M]. 周涛, 译. 杭州: 浙江人民出版社, 2013
[3] FAKOOR R , LADHAK F , NAZI A , et al. Using Deep Learning to Enhance Cancer Diagnosis and Classification[C]// Proceedings of the 30 th International Conference on Machine Learning. USA: ICML, 2013：211–218
[4] WANG A, AN N, YANG J, et al. Alterovitz, Incremental Wrapper Based Gene Selection with Markov Blanket[C]//ASE BioMedCom Conference. USA. USA: ASE, 2014：106–108
[5] BLUMER A, EHRENFEUCHT A, HAUSSLER D, et al. Learnability and the Vapnik–Cherbonenkis Dimension [J]. Journal of the ACM, 1989: 36(4): 929–965
[6] 罗军舟. AMS大数据处理的挑战[R]. 合肥: 中国计算机大会, 2015
[7] 周志华, 李武军, 张利军. CCF2014–2015中国计算机科学技术发展报告[M].北京: 机械工业出版社, 2015
[8] TOPOL E. The Creative Destruction of Medicine [M]. 张南, 等译. 北京: 电子工业出版社, 2014
[9] CHO K. A Brief Summary of the Panel Discussion at DL Workshop of ICML[EB/OL]. [2015–07–13]. http://deeplearning.net/2015/07/13/a–brief–summary–of–the–panel–discussion–at–dl–workshop–icml–2015
[10] 洪加威. 能用例证法来证明几何定理吗?[J]. 中国科学A辑, 1986(3): 234–242
[11] LASZLO BARABASI A. Bursts: The Hidden Pattern Behind Everything We Do [M]. 马慧, 译. 北京: 人民出版社, 2012

作者简介

李廉，合肥工业大学计算机与信息学院教授，中国计算机学会理论计算机科学专业委员会主任，教育部高等学校大学计算机课程教学指导委员会主任；主要研究方向为机器学习、计算机网络、无线传感器网络等；获国家教学成果二等奖1项，安徽省教学成果特等奖1项。

综合信息

2015年中国发明专利申请量首次突破百万件

国家知识产权局发布的数据显示：2015年，国家知识产权局共受理发明专利申请110.2万件，同比增长18.7%，连续5年位居世界首位。

随着中国科技水平发展，专利创新越来越重要。2015年中国专利数量有显著增长，专利申请量超过200万件，发明专利申请受理量首次超过100万件。

其中，中国发明专利授权26.3万件，比2014年增长了10万件，同比增长61.9%。而同为专利大国的美国，2015年专利申请数量却少有地迎来了授权专利数量下降。美国商业专利数据库IFICLAIMS发布的2015年专利统计数据显示：全年授权专利数约为29.8万件。虽然同比下降不到1%，但这是自2007年以来授权专利数量首次下跌。

美国专利在十年间维持了小幅的上涨，申请量年均30万～40万件。中国专利申请量一直呈增长态势，在2005年以前，数量上已经超越美国。兰德公司发布的中国的专利与创新报告指出：中国十年来专利爆发性增长，主要是专利激励政策和市场力量推动的结果。在政策指导下，国家鼓励个人和企业创业。

激烈的市场竞争也促进企业进行专利技术研发。以手机行业为例，近十年间，中国手机品牌崛起，加入以往被国外手机厂商所垄断的市场，手机品牌数量大幅度增长。对于专利技术的投入能够帮助这些品牌在市场中占据一席之地。

（转载自《中国信息产业网》）

通信行业支出将再次增长 设备厂商应密切关注需求变化

自Ovum的最新研究报告称：传统的电信行业正在萎缩。2015年全球电信运营商的收入预计为1.78万亿美元，相较2014年下降5%，2012—2014年这3年间全球CSP收入一直处于持平状态。

对于设备厂商来说，一些全新的客户正在从OTT风潮中出现。包括所有这些供应商在内，行业资本支出预计将会逐渐上升，从2014年的4 050亿美元增至2020年的4 660亿美元。

Ovum表示：全球电信运营商的资本支出仍旧受到严格限制，2014年这一数字为3 390亿美元，CSP资本支出在2015—2017年将出现下滑，之后到2020年将再次出现增长，而这主要是受到早期5G支出的推动。

从总体来看：ICP的资本支出正在稳步增长，到2020年将达到1 100亿美元。这一数字与所有固定电信运营商2020年的网络资本支出差不多。许多ICP自身拥有强大的内部技术研发队伍，并会通过ODM厂商来制造定制化的设备。

虽然ICP代表着一种机会，但企业和政府客户也正在建设日益复杂的网络，他们也需要帮助。设备厂商需要调整他们的解决方案，并与互补企业进行合作，同时大力投入销售和营销，才能赢得市场。简单地将那些针对电信运营商的技术转售给这些市场是无法赢得客户认可的。

（转载自《中国信息产业网》）

DOI:10.3969/j.issn.1009-6868.2016.02.003
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20160225.1736.004.html

大数据分析平台——从扩展性优先到性能优先

Big Data Analytic Platforms: Changing the Priority from Scalability to Performance

郑纬民/ZHENG Weimin
陈文光/CHEN Wenguang
(清华大学 计算机科学与技术系,北京 100084)
(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

中图分类号:TP393 文献标志码:A 文章编号:1009-6868 (2016) 02-0011-003

摘要: 认为现有以MapReduce/Spark等为代表的大数据处理平台在解决大数据问题的挑战问题方面过多考虑了容错性,忽视了性能。大数据分析系统的一个重要的发展方向就是兼顾性能和容错性,而图计算系统在数据模型上较好地考虑了性能和容错能力的平衡,是未来的重要发展方向。

关键词: 大数据;分布与并行处理;并行编程;容错;可扩展性

Abstract: Existing big data analytic platforms, such as MapReduce and Spark, focus on scalability and fault tolerance at the expense of performance. We discuss the connections between performance and fault tolerance and show they are not mutually exclusive. Distributed graph processing systems are promising because they make a better tradeoff between performance and fault tolerance with mutable data models.

Key words: big data; distributed and parallel processing; parallel programming; fault tolerance; scalability

随着信息化技术的发展,人类可以产生、收集、存储越来越多的数据,并利用这些数据进行决策,从而出现了大数据的概念。大数据的定义很多,比较流行的定义是Gartner公司提出的简称为3V的属性,即数据量大(Volume),到达速度快(Velocity)和数据种类多(Variety)。大数据分析利用数据驱动的方法,在科学发现、产品设计、生产与营销、社会发展等领域具有应用前景。

由于大数据的3V属性,需要在多台机器上进行分布与并行处理才能满足性能要求,因此传统的关系型数据库和数据挖掘软件很难直接应用在大数据的处理分析中。传统的超级计算技术,虽然具有很强的数据访问和计算能力,但其使用的MPI编程模型编程较为困难,对容错和自动负载平衡的支持也有缺陷,主要运行在高成本的高性能计算机系统上,对于主要在数据中心运行的大数据分析不是非常适合。

为了解决大数据的分析处理所面临的编程困难,负载不平衡和容错困难的问题,业界发展出了一系列技术,包括分布式文件系统、数据并行编程语言和框架以及领域编程模式来应对这些挑战。以MapReduce[1]和Spark[2]为代表的大数据分析平台,是目前较为流行的大数据处理生态环境,得到了产业界的广泛使用。

但是在文章中,我们通过分析认为:MapReduce和Spark系统将容错能力作为设计的优先原则,而在系统的处理性能上做了过多的让步,使得所需的处理资源过多,处理时间很长,这样反而增加了系统出现故障的几率。通过进一步分析性能与容错能力的关系,我们提出了一种性能优先兼顾扩展性的大数据分析系统构建思路,并以一个高性能图计算系统为例,介绍了如何用这种思路构建大数据分析系统。

1 以MapReduce/Spark为代表的大数据分析平台

现有的大数据分析平台主要基于开源的Hadoop系统,该系统使用Hadoop分布式文件系统(HDFS),通过多个备份的方法保证大量数据的可靠存储和读取性能,其上的Hive[3]系统支持数据查询,Hadoop MapReduce则支持大数据分析程序的开发。

与传统的并行编程方法MPI[4]相比,MapReduce是近年来并行编程领域的重要进展。尽管Map和Reduce

收稿时间:2016-01-14
网络出版时间:2016-02-25
基金项目:国家重点基础研究发展("973")计划(2014CB340402);国家自然科学基金(61525202)

在函数语言中早已被提出，但将其应用于大规模分布并行处理应归功于Jeff Dean和Ghemewat Sanjay。在MapReduce并行编程模型中，用户仅需要编写串行的Map函数体和Reduce函数体，MapReduce框架就可以完成并行的计算，并实现了自动容错和负载均衡。这对于数据中心中采用的异构服务器、低成本服务器集群是非常重要的。MapReduce开始仅能在使用通用中央处理器（CPU）的分布式系统上运行，但后来被移植到图形处理器（GPU）和多种加速器上。

MapReduce需要将中间结果保存到磁盘中，从而大大影响了性能，美国加州伯克利大学提出的Spark系统可以看做是基于内存的MapReduce模型，通过将中间结果保存在内存中，大大提高了数据分析程序的性能，类似思路的系统还包括HaLoop[5]和Twister[6]等。

Spark和MapReduce在大数据领域取得了巨大的成功，已经成为事实上的大数据处理标准。它们与分布式文件系统HDFS、查询系统Hive都集成在Hadoop系统中，为大数据的存储、查询和处理提供了相对完整的解决方案。这一系统也具有完整的开源社区支持和商业公司支持，HortonWorks和Cloudera提供Hadoop的发行版和服务，DataBricks为Spark提供发行版和服务。IBM于2016年宣布将投入10亿美元开发Spark。

2 大数据分析平台性能的重要性

尽管以Spark/MapReduce为代表的大数据分析平台已经得到了广泛应用，然而，其性能方面的问题也日益暴露出来。一些研究表明：对一些大数据分析问题来说，使用Spark在几十台机器上的性能甚至不如在某些优化过的程序在单机上的性能，例如对Twitter数据集来说，Spark在128个处理器核上需要857 s，而优化良好的单线程程序完成同样的处理功能仅需要300 s的时间[7]，即在中小规模数据集上Spark的性能功耗比比单线程程序要差2个数量级，甚至在绝对处理时间上也比单线程程序要慢。

Spark/MapReduce的性能问题，根源在于其设计理念上陷入了一个误区：即以容错能力为优先的设计目标，忽视了处理性能。例如，MapReduce和Spark都采用只读数据集的概念，这一方面大大方便了系统进行容错，但也使得系统在处理相当一部分应用时，性能会受到严重影响。例如，对于广泛使用的广度优先图搜索问题，需要记录哪些结点被访问过，这个数据集如果是只读的，就只能在每次遍历迭代时生成新的数据集，这会大大增加所需的内存复制操作和内存容量需求，使得性能大大下降。

而实际上处理性能的提高，对提高系统的容错能力也是有正面意义的。一个数据分析任务的总执行时间，可以按如式(1)估算（为描述方便，公式中略有简化）：

总执行时间 = 无故障执行时间①+无故障时容错机制开销②+故障发生概率*无故障执行时间*单次故障恢复时间③ (1)

Spark的设计主要对②进行优化，即通过只读数据集简化无故障容错机制的开销，却大大增加了①的无故障执行时间，而③实际是与①正相关的，即相同机器数，执行时间越长，出故障的概率越大，所需故障恢复时间也就越长。

从上面的分析可以看出：Spark的设计理念，即使对容错本身来说，也很难说是合理的，因为如果性能损失太大，无故障执行时间增加太多，会使得在②减少的开销被③抵消甚至超越[8]。

因此，我们认为：大数据分析系统的一个重要的发展方向就是兼顾性能和容错性。我们需要进一步在编程模型和框架上开展研究，在保持自动负载平衡和一定容错能力的基础上，提供优化的系统性能。

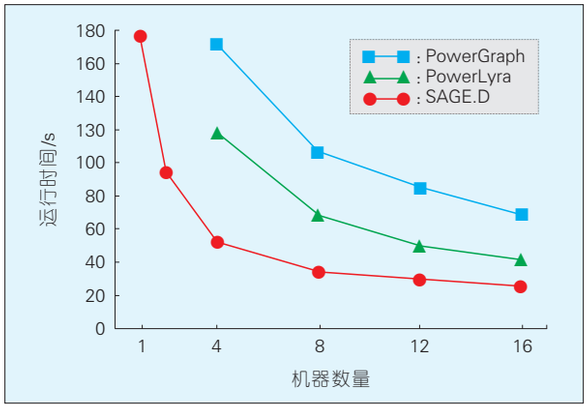
以Pregel[9]和GraphLab[10]等的图计算编程框架是这一类工作的代表，这些编程模型主要提供了基于图结点（vertex）的编程抽象，并沿着图的边进行通信，与Map-Reduce相比，这类图编程框架在处理图数据（如社交网络、航运网络和生物网络等）时比Map-Reduce/Spark的表达更加自然，所获得的性能也要好得多。这方面的工作引起了全球研究者和工业界的广泛关注，这些工作针对图计算中的负载不均衡、随机访问多、同步和异步等问题提出了解决方案。PowerGraph[11]和PowerLyra[12]系统是在GraphLab上改进后的图计算系统，其性能比GraphLab又有显著提高。GridGraph[13]提出了利用二维混洗的数据结构对图计算进行优化，可以有效减少图计算中的随机内存访问，提高处理性能。基于GirdGraph的分布式图计算系统SAGE.D其性能比PowerLyra进一步又提高了1倍左右。如图1所示：SAGE.D可以在16台机器上以30 s的时间内完成Twitter数据集的20次PageRank迭代，性能比Spark提高了接近30倍。

我们可以看到：在某些分析任务上，基于图计算系统的性能比基于Spark的分析系统快1～2个数量级。这意味着基于图计算系统在执行期间内发生错误的机会仅为Spark的1/10以下，从而不仅在执行性能方面，在容错能力方面也优于Spark。

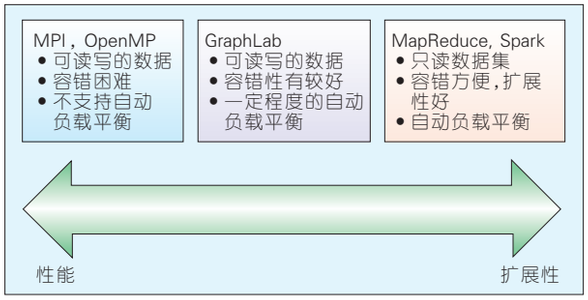
3 大数据问题展望

未来的大数据问题会呈现两种趋势：

（1）具有较小上限的大数据问题。以社交网络的分析问题为例，目前Facebook有约10亿活跃用户，用户之间的关注关系大约有1 000亿个，大约需要几个TB的内存容量。社交网络的结点是用户，地球上只有几十亿人口，社交网络的分析问题其上限



◀图1
在Twitter数据集上图计算系统完成20轮PageRank算法迭代的时间



◀图2
不同并行编程模型在设计理念和运行时支撑方面的差异

就是将全部人口数作为网络结点。

随着摩尔定律的持续作用，我们今天已经可以很容易地买到内容容量为TB量级的服务器，今后可望达到几十甚至数百TB。不断增长的硬件能力与较小上限的大数据问题相遇的结果，就是把今天的大数据问题变为明天的小数据问题，把今天需要数十、数百服务器解决的问题变为今后只需要几台甚至单台服务器就可以解决的问题。

针对这类应用，显然性能优化的大数据分析处理平台能够获得更好的性价比。

（2）具有较大上限的大数据问题。高性能计算中的很多问题规模具有非常大的上限，例如气候模拟，需要将空间分成网格、时间分片，显然空间上和时间上的进一步细分都会导致计算量和存储量的大幅度增加，人类已有的计算能力还远远无法满足高精度气候模拟的要求。针对这类应用，性能优化的大数据分析处理平台能够通过减少运行时间，提高系统的处理效率和处理规模。图2展示了不同并行编程模型在设计理念和运行时支撑方面的差异。

综上所述，现有以Spark为代表的大数据处理平台在解决大数据问题的挑战问题方面过多考虑了容错性，忽视了性能。我们认为图计算系统在数据模型上较好地考虑了性能和容错能力的平衡，是未来的重要发展方向。

参考文献

[1] DEAN, JEFFREY, SANJAY G. MapReduce: Simplified Data Processing on Large Clusters [J]. Communications of the ACM, 2008, 51 (1): 107–113. DOI: 10.1145/1327452.1327492
[2] ZAHARIA M, CHOWDHURY M, DAS T, et al. Resilient Distributed Datasets: A Fault–Tolerant Abstraction for In–Memory Cluster Computing[C]// Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. USA: USENIX Association, 2012:15–28
[3] THUSOO A, SARMA S J, JAIN N, et al. Hive: A Warehousing Solution over a Map–Reduce Framework [J]. Proceedings of the VLDB Endowment, 2009, 2(2): 1626–1629. DOI: 10.14778/1687553.1687609
[4] GROPP W, LUSK E, DOSS N, et al. "A High–Performance, Portable Implementation of the MPI Message Passing Interface Standard [J]. Parallel Computing, 1996, 22(6): 789–828. DOI: 10.1016/0167–8191(96)00024–5
[5] BU Y, HOWE B, BALAZINSKA M, et al. HaLoop: Efficient Iterative Data Processing on Large Clusters [J]. Proceedings of the VLDB Endowment, 2010, 3(1): 285–296. DOI: 10.14778/1920841.1920881
[6] EKANAYAKE, JALIYA. Twister: A Runtime for Iterative Mapreduce [C]//Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing. USA: ACM, 2010: 810–818
[7] FRANK M, MICHAEL I, MURRAY D G. Scalability! But at what COST [C]//5th Workshop on Hot Topics in Operating Systems (HotOS XV). USA: USENIX Association, 2015
[8] KWAK, HAEWOON. What is Twitter, A Social Network or A News Media? [C]/ Proceedings of the 19th International Conference on World Wide Web. USA: ACM, 2010: 591–600
[9] MALEWICZ, GRZEGORZ. Pregel: A System for Large–Scale Graph[C]// Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. USA: ACM, 2010: 135–146
[10] LOW, YU C. Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud [J].Proceedings of the VLDB Endowment, 2012, 5(8): 716–727
[11] GONZALEZ, Joseph E. PowerGraph: Distributed Graph–Parallel Computation on Natural Graphs [J]. OSDI, 2012, 12(1): 23–27
[12] CHEN R. Powerlyra: Differentiated Graph Computation and Partitioning on Skewed Graphs[C]//Proceedings of the Tenth European Conference on Computer Systems. USA: ACM, 2015: 1–15
[13] ZHU X, HAN W, CHEN W. GridGraph: Large–Scale Graph Processing on a Single Machine Using 2–Level Hierarchical Partitioning[C]//Proceedings of the Usenix Annual Technical. USA: ASM, 2015: 375–386

作者简介

郑纬民，清华大学计算机科学与技术系教授、博士生导师；长期从事计算机系统结构、大规模数据存储、高性能计算等领域的科研教学工作；主持并完成了“973”、“863”、自然科学基金等科研项目36项，负责或参与工程项目11项；获国家科技进步一等奖1次，国家科技进步奖二等奖2次，国家技术发明奖二等奖1次；发表论文500余篇，著作10部。

陈文光，清华大学计算机科学与技术系教授，现为中国机算机学会杰出会员、副秘书长、杰出讲者，青年科技论坛（YOCSEF）荣誉委员，ACM中国理事会副主席等；获国家科技进步二等奖1次，部级科技一等奖2次；发表文章50余篇。

DOI:10.3969/j.issn.1009-6868.2016.02.004
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.tn.20160223.1346.002.html

典型大数据计算框架分析

Typical Big Data Computing Frameworks

赵晟/ZHAO Sheng
姜进磊/JIANG Jinlei
(清华大学 计算机科学与技术系,北京 100084)
(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

中图分类号:TP393 文献标志码:A 文章编号:1009-6868 (2016) 02-0014-005

摘要: 认为大数据计算技术已逐渐形成了批量计算和流计算两个技术发展方向。批量计算技术主要针对静态数据的离线计算,吞吐量好,但是不能保证实时性;流计算技术主要针对动态数据的在线实时计算,时效性好,但是难以获取数据全貌。从可扩展性、容错性、任务调度、资源利用率、时效性、输入输出(IO)等方面对现有的主流大数据计算框架进行了分析与总结,指出了未来的发展方向和研究热点。

关键词: 大数据分类;大数据计算;批量计算;流计算;计算框架

Abstract: Big data computing technologies have two typical processing modes: batch computing and stream computing. Batch computing is mainly used for high-throughput processing of static data and does not produce results in real time. Stream computing is used for processing dynamic data online in real time but has difficulty providing a full view of data. In this paper, we analyze some typical big data computing frameworks from the perspective of scalability, fault-tolerance, task scheduling, resource utilization, real time guarantee, and input/output (IO) overhead. We then points out some future trends and hot research topics.

Key words: big data; big data computing; batch computing; stream computing; computing framework

近年来,随着互联网进入Web 2.0时代以及物联网和云计算的迅猛发展,人类社会逐渐步入了大数据时代。根据维基百科的描述,所谓的大数据,是指所涉及的数据量规模巨大,无法通过人工在合理时间内达到截取、管理、处理、并整理成为人类所能解读的信息。大数据在带来发展机遇的同时,也带来了新的挑战,催生了新技术的发展和旧技术的革新。例如,不断增长的数据规模和数据的动态快速产生要求必须采用分布式计算框架才能实现与之相匹配的吞吐和实时性,而数据的持久化保存也离不开分布式存储。

图1展示了大数据应用的一般架构,其中的核心部分就是大数据计算框架和大数据存储。大数据存储提供可靠的数据存储服务,在此之上搭建高效、可扩展、可自动进行错误恢复的分布式大数据计算框架,计算依赖存储,两者共同构成数据处理的核心服务。由于文献[1]已经对大数据存储进行总结,详述了文件系统、数据库系统、索引技术,因此文中将重点对大数据计算框架进行分析。

1 大数据计算技术面临的问题与挑战

大数据计算技术采用分布式计算框架来完成大数据的处理和分析任务。作为分布式计算框架,不仅要提供高效的计算模型、简单的编程接口,还要考虑可扩展性和容错能力。作为大数据处理的框架,需要有高效可靠的输入输出(IO),满足数据实时处理的需求。当前大数据处理需要

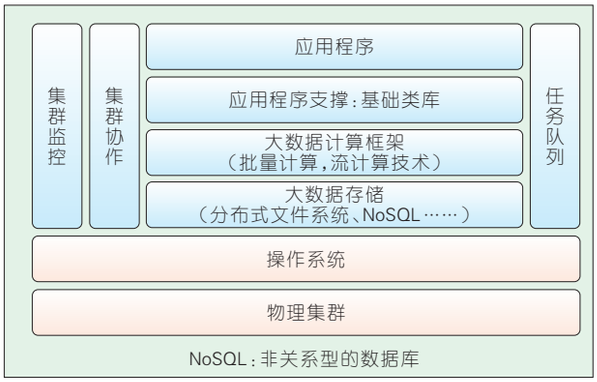


图1 大数据应用的一般性架构

收稿时间:2016-01-10
网络出版时间:2016-02-23
基金项目:国家高技术研究发展("863")计划(2013AA01A213);国家自然科学基金 (61572280、61433008、U1435216、61373145)

解决如下问题和挑战，这些问题和挑战也是对大数据计算框架进行分析的重要指标。

（1）可扩展性：计算框架的可扩展性决定可计算规模，计算并发度等指标。现有计算框架通常采用主从模式的架构设计，便于集群的管理和任务调度，但主节点会成为系统的性能瓶颈，限制了可扩展性。另外，在现有弹性计算集群部署中，不断动态添加、删除计算节点，快速平衡负载等也对系统可扩展性提出挑战。

（2）容错和自动恢复：大数据计算框架需要考虑底层存储系统的不可靠性，支持出现错误后自动恢复的能力。用户不需要增加额外的代码进行快照等中间结果的备份，只需要编写相应的功能函数，就可以在有输入的条件下得到预期的输出，中间运行时产生的错误对使用人员透明，由计算框架负责任务重做。

（3）任务调度模型：大数据计算平台中往往存在多租户共同使用，多任务共同执行的情况。既要保证各用户之间使用计算资源的公平性，又要保证整个系统合理利用资源，保持高吞吐率，还要保证调度算法足够简单高效，额外开销小。因此调度器设计需要综合大量真实的任务运行结果，从全局的角度进行设计。

（4）计算资源的利用率：计算资源的利用率代表机器能够实际创造的价值。数据中心运转时，能耗问题非常突出，设备和制冷系统都在消耗能源。由于不合理的架构设计，导致集群中非计算开销大，计算出现忙等待的现象时有发生。高效的计算框架需要和硬件环境共同作用达到更高的计算资源利用率。

（5）时效性：数据的价值往往存在时效性，随着时间的推移，新数据不断产生，旧数据的利用价值就会降低。离线批量处理往往导致运算的时间长，达不到实时的数据处理。流计算方案减少了响应的时间，但是不能够获得数据的全貌。因此增量计算的方法是当今的一个解决思路。

（6）高效可靠的IO：大数据计算中，IO开销主要分为两部分，序列化反序列化时数据在硬盘上读写的IO开销，不同节点间交换数据的网络IO开销。由于硬盘和网络的IO读写速率远远低于内存的读写速率，导致整个任务的执行效率降低，计算资源被浪费。在现有的计算机体系结构下，尽可能使用内存能够有效提高处理的速度，但是预取算法的合理性和内存的不可靠性都是需要考虑的问题。

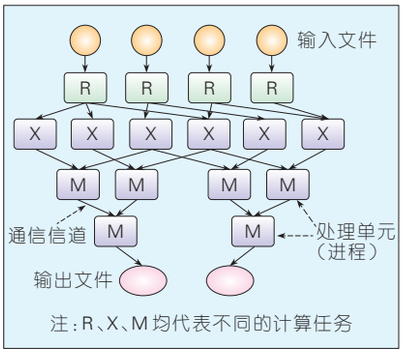
2 大数据批量计算技术

大数据批量计算技术应用于静态数据的离线计算和处理，框架设计初衷是为了解决大规模、非实时数据计算，更加关注整个计算框架的吞吐量。MapReduce低成本、高可靠性、高可扩展的特点降低了大数据计算分析的门槛，自Google提出以来，得到了广泛应用。在此基础上，人们设计出众多的批处理计算框架，从编程模型、存储介质等角度不断提高批处理的性能，使其适应更多的应用场景。

（1）MapReduce计算框架：MapReduce计算框架通过提供简单的编程接口，在大规模廉价的服务器上搭建起一个计算和IO处理能力强大的框架，并行度高，容错性好，其开源项目Hadoop已经形成完整的大数据分析生态系统，并在不断改进。可扩展性方面，通过引入新的资源管理框架YARN，减轻主节点的负载，集群规模提高，资源管理更加有效。任务调度方面，提出如公平调度[2]、能力调度[3]、延迟调度[4]等调度器，更加关注数据中心内资源使用的公平性、执行环境的异构性和高吞吐的目标。另外也采用启发式方法进行预测调度，能够实时跟踪节点负载变化，提供更优的执行序列和资源分配方案。容错性方面，MapReduce框架本身支持任务级容错，任务失败后会重新计算，但是对于Master节点的容错一直忽略，现有的解决方法采用备份的方式解决，通过共享存储同步数据，采用网络文件系统（NFS）或者Zookeeper的方式来支持共享存储。另外，MapReduce也已经添加了多平台支持，可以部署在图像处理单元（GPU）等高性能计算环境中。

（2）Dryad计算框架：Dryad是构建微软云计算基础设施的核心技术。编程模型相比于MapReduce更具一般性——用有向无环图（DAG）描述任务的执行，其中用户指定的程序是DAG图的节点，数据传输的通道是边，可通过文件、共享内存或者传输控制协议（TCP）通道来传递数据，任务相当于图的生成器，可以合成任何图，甚至在执行的过程中这些图也可以发生变化，以响应计算过程中发生的事件。图2给出了整个任务的处理流程。Dryad在容错方面支持良好，底层的数据存储支持数据备份；在任务调度方面，Dryad的适用性更广，不仅适用于云计算，在多核和多处理器以及异构集群上同样有良好的性能；在扩展性方面，可伸缩于各种规模的集群计算平台，从单机多核计算机到由多台计算机组成的集群，甚至拥有数千台计算机的数据中心。Microsoft借助Dryad，在大数据处理方面也形成了完整的软件栈，部署了分布式存储系统Cosmos[5]，提供DryadLINQ编程语言，使普通程序员可以轻易进行大规模的分布式计算。

（3）Spark计算框架：Spark是一种高效通用的分布式计算框架，采用基于DAG图的编程模型，提供了丰富



▲图2 Dyrad计算框架的任务处理流程

的编程接口。不同于MapReduce只能通过串联多个任务实现复杂应用，Spark可以在DAG图中划分不同的阶段，完成复杂应用的定义。在计算效率方面，Spark将结果以及重复使用的数据缓存在内存中，减少了磁盘IO带来的开销，更适用于机器学习等需要迭代计算的算法；在容错性方面，Spark表现突出，数据以弹性分布式数据集（RDD）[6]的形式存在，依靠Lineage的支持（记录RDD的演变），能够以操作本地集合的方式来操作分布式数据集。当RDD的部分分区数据丢失时，它可以通过Lineage获取足够的信息来重新运算和恢复丢失的数据分区。通过记录跟踪所有RDD的转换流程，可以保证Spark计算框架的容错性。资源管理及任务调度方面，Spark借助Mesos或者YARN来进行集群资源的管理，部署在集群中使用。Spark发展至今，已经形成了完整的软件栈，在Spark的上层，已经能够支持可在分布式内存中进行快速数据分析的Shark[7]、流计算Spark Streaming、机器学习算法库Mllib、面向图计算的GraphX等。

（4）GraphLab计算框架：图计算框架GraphLab的提出是为了解决大规模机器学习问题。相比于信息传递接口（MPI），GraphLab提供了更简单的编程接口，抽象的图模型使用户不必关注进程间的通信。相比于MapReduce计算框架，GraphLab更适合处理各数据之间依赖程度强、数据与数据之间需要频繁计算和信息交互的场景。GraphLab提出的图计算理论和方法不仅解决了集群中图处理的扩展问题，也解决了单机系统中大规模的图计算问题，可形成完整的面向机器学习的并行计算框架。但是，对于大规模自然图的处理，GraphLab仍然存在负载极不均衡、可扩展性差等缺点，因而GraphLab团队进一步提出了PowerGraph[8]。PowerGraph并行的核心思想是根据边的规模对顶点进行分割并部署在不同的机器上，由于不需要将同一个节点所对应的所有边的信息载入单机的内存中，因而消除了单机内存的约束。在系统的容错性方面，PowerGraph采用检查点技术，未来也考虑使用节点的副本冗余来提高容错性，能够在提高计算效率的同时完成快速恢复。

3 大数据流计算技术

大数据批量计算技术关注数据处理的吞吐量，而大数据流计算技术更关注数据处理的实时性，能够更加快速地为决策提供支持。大数据的流计算技术是由复杂事件处理（CEP）发展而来的，现在流计算的典型框架包括Storm、S4、Samza、Spark Streaming等。

（1）Storm计算框架：Storm提供了可靠的流数据处理，可以用于实时分析、在线机器学习、分布式远程过程调用（RPC），数据抽取、转换、加载（ETL）等。Storm运行用户自定义的拓扑，不同于MapReduce作业，用户拓扑永远运行，只要有数据进入就可以进行相应的处理。Storm采用主从架构，如图3所示，主节点中部署Nimbus，主要负责接收客户端提交的拓扑，进行资源管理和任务分配。从节点上运行监控器，负责从节点上工作进程也就是应用逻辑的运行。可扩展性方面，Storm借助于Zookeeper很好地解决了可扩展性的问题，集群非常容易进行横向扩展，便于统一的配置、管理和监控。容错性方面，使用ZeroMQ传送消息，消除了中间的排队过程，使得消息能够直接在任务之间流动，其注重容错和管理，实现了有保障的消息处理，保证每一个元组都会通过整个拓扑进行处理，未处理的元组，它会自动重放，再次进行。Storm的缺点是：集群存在负载不均衡的情况；任务部署不够灵活，不同的拓扑之间不能相互通信，结果不能共用。

（2）S4计算框架：S4是Yahoo发布的开源流计算平台，它是通用的、可扩展性良好、具有分区容错能力、支持插件的分布式流计算平台。S4采用分散对称的架构，没有中心节点，计算框架更加易于部署和维护。在计算过程中，每个计算节点都在本地内存中进行处理，避免了IO给计算带来的巨大瓶颈。S4的核心思想是将整个任务处理分为多个流事件，抽象成为一个DAG图，每个事件对应DAG图中的一条有向边，并用（K，A）的形式表示，其中K和A分别表示对应事件的键和属性。这种表示类似MapReduce中的键/价值设计，适合多个处理进行连接。S4的结构如图4所示，它采用Zookeeper来管理集群，提高了集群的可扩展性。在通信层，提供备用节点，如果有节点失败，处理框架会自动切换到备用节点，但是在内存中的数据会发生丢失。主从备份虽然在一定程度上提高了容错能力，但是相对较弱。同时通信层还使用一个插件式的架构来选择网络协议，通过TCP和用户数据报协议

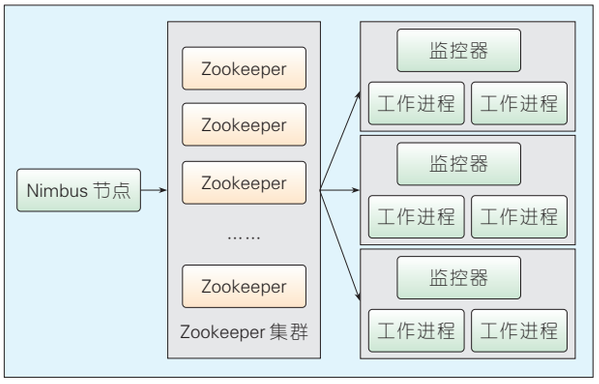
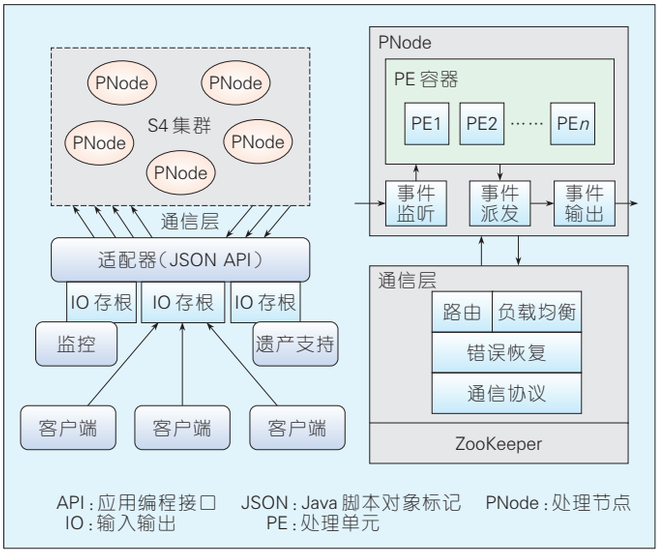


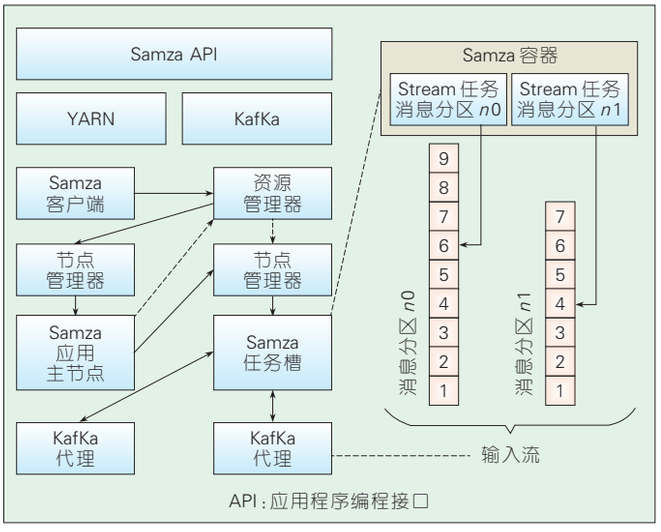
图3▶ Storm计算框架的架构



◀图 4
S4 计算框架的架构

(UDP)之间的权衡来提高网络 IO 的速率。S4 框架的主要缺点是:持久化相对简单,数据存在丢失的风险;节点失败切换到备份节点之后,任务都需要重做;缺乏自动负载均衡的相关能力。

(3) Samza 计算框架:Samza 是 Linkedin 开源的分布式流处理框架,其架构如图 5 所示,由 Kafka[9]提供底层数据流,由 YARN 提供资源管理、任务分配等功能。图 5 也给出了 Samza 的作业处理流程,即 Samza 客户端负责将任务提交给 YARN 的资源管理器,后者分配相应的资源完成任务的执行。在每个容器中运行的流任务相对于 Kafka 是消息订阅者,负责拉取消息并执行相应的逻辑。在可扩展性方面,底层的 Kafka 通过 Zookeeper 实现了动态的集群水平扩展,可提供高吞吐、可水平扩展的消息队列,YARN 为 Samza 提供了分布式的环境和执行容器,因此也很容易扩展;在容错性方面,如果服务器出现故障,Samza 和 YARN 将一起进行任务的迁移、重启和重新执行,YARN 还能提供任务调度、执行状态监控等功能;在数据可靠性方面,Samza 按照 Kafka 中的消息分区进行处理,分区内保证消息有序,分区间并发执行,Kafka 将消息持久化到硬盘保证数据



◀图 5
Samza 计算框架的架构及其作业处理流程

安全。另外,Samza 还提供了对流数据状态管理的支持。在需要记录历史数据的场景里,数据实时流动导致状态管理难以实现,为此,Samza 提供了一个内建的键/价值数据库用来存储历史数据。

(4) Spark Streaming 计算框架:Spark 是当前迭代式计算的典型代表,在前面的批量计算中已经介绍了 Spark 在大数据计算、数据抽象和数据恢复等方面的成果。如今 Spark 也在向实时计算领域发展,2013 年发表于顶级会议 SOSP 上的论文介绍了 Spark 在流计算中取得的最新成果。Spark Streaming 是建立在 Spark 上的应用框架,利用 Spark 的底层框架作为其执行基础,并在其上构建了 DStream 的行为抽象。利用 DStream 所提供的应用程序编程接口(API),用户可以在数据流上实时进行 count、join、aggregate 等操作。Spark Streaming 的原理是将流数据分成小的时间片断,以类似批量处理的方式来处理这小部分数据。DStream 同时也是 Spark Streaming 容错性的一个重要保障。

4 框架比较

随着数据的爆炸式增长,大数据计算平台在数据分析和处理中扮演着越来越重要的角色。本文分析了现有大数据处理面临的挑战和问题,详细分析了批处理和流处理相关计算框架的特点和重点解决的问题,结合存储、应用介绍了它们的核心创新点和软件栈,这些框架的总结和对比如表 1 所示。

5 结束语

应用推进了技术的发展和革新,目前业界在不断提高大数据计算框架的吞吐量、实时性、可扩展性等特性以应对日益增长的数据量和数据处理需求,大数据计算框架依然是现在以及未来一段时间内的研究热点。未来的发展趋势是:随着商业智

▼表 1 典型大数据计算框架的对比

计算框架	计算效率（实时性）	容错性	特点	适用场景
MapReduce	低	任务出错重做	编程接口简单，计算模型受限	文本处理、log 分析、机器学习
Spark	高	RDD 的 Lineage 保证	内存计算，通用性好，更适合迭代式任务	迭代式离线分析任务、机器学习
Dryad	较高	任务出错重做	针对 Join 进行了优化，允许动态优化调度逻辑（修改 DAG 拓扑）	机器学习、微软技术栈
GraphLab	较高	检查点技术	机器学习图计算专用框架	机器学习、大图计算
Storm	高	Worker 重启或分配到新机器，任务重做	通用性好，消息传递可靠，支持热部署，主节点可靠性差	通用的实时数据分析处理
S4	高	部分容错，检查点技术	通用性较好，通信在 TCP 和 UDP 之间权衡，持久化方式简单	实时广告推荐、容忍数据丢失
Samza	高	任务出错重做	可扩展性好，兼容流处理和批处理	在线和离线任务相结合的场景
Spark Streaming	低	RDD 和预写日志（Write Ahead Logs）	通用性好，容错性好，通过设置短时间片实现实时，应用较为局限	历史数据和实时数据相结合的分析

DAG：有向无环图　RDD：弹性分布式数据集　TCP：传输控制协议　UDP：用户数据报协议

能和计算广告等领域的发展，更强调实时性的流计算框架将得到更加广泛的关注；能够缩短批量计算处理时间的 Percolator[10]、Nectar[11]、Incoop[12] 等增量计算模式将获得进一步的发展和应用；批量计算和流计算模式将进一步融合以减少框架维护开销。而实际上，现在的 Spark 计算框架除了支持离线批处理任务以外，已经能够通过 Spark Streaming 支持在线的实时分析。

除了计算框架本身的改进，消除磁盘 IO 和网络 IO 对计算效率的影响也是重要的研究方向之一。这方面，内存计算已经取得了不错的应用效果。例如，PACMan[13] 用内存缓存输入数据，从而加速了 MapReduce 的执行；文中提到的 Spark 也是尽可能将所有的数据放在内存中，从而减少磁盘随机读写的 IO 开销，提高任务的执行效率。由于内存资源始终是宝贵的，难以满足大数据的存储需求，因此在很多情况下将仍然充当缓存的角色，需要进一步探究更为高效的缓存管理算法。另一方面，非易失性存储（NVM）器件的发展将对计算机系统的存储架构带来革命性的变化，如何充分利用新存储介质及其存储架构的特性提升计算效率也将是未来大数据计算框架研究的重要话题。

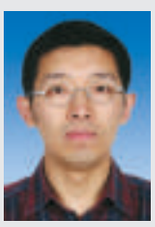
总之，应用的推动和技术的进步将会产生新的问题。作为大数据应用的核心，对于挖掘数据价值起着重要作用的计算框架将会面临更多的挑战，亟待解决。

参考文献

[1] 孟小峰, 慈祥. 大数据管理: 概念, 技术与挑战 [J]. 计算机研究与发展, 2013, 50(1): 146–169
[2] ZAGARIA M, BORTHAKUR D, SARMA J S, et al. Job Scheduling for Multi–User MapReduce Clusters [R]. USA: EECS Department, University of California, 2009
[3] Hadoop.[EB/OL].[2013–08–24]. http://hadoop.apache.org/docs/r1.2.1/capacity_scheduler.html# Overview
[4] ZAHARIA M, KONWINSKI A, JOSEPH A D, et al. Improving MapReduce Performance in Heterogeneous Environments[C]// 8th USENIX Symposium on Operation Systems Design and Implementation(OSDI). USA: ASM, 2008：7
[5] CHAIKEN R, JENKINS B, LARSON PÅ, et al. SCOPE: Easy and Efficient Parallel Processing of Massive Data Sets [J]. Proceedings of the VLDB Endowment, 2008, 1(2): 1265–1276
[6] ZAHARIA M, CHOWDHURY M, DAS T, et al. Resilient Distributed Datasets: A Fault–Tolerant Abstraction for in–Memory Cluster Computing[C] //Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. USA: USENIX Association, 2012：2–2
[7] XIN R S, ROSEN J, ZAHARIA M, et al. Shark: SQL and Rich Analytics at Scale[C]// Proceedings of the 2013 ACM SIGMOD International Conference on Management of data. USA: ACM Press,2013: 13–24
[8] GONZALEZ J E, LOW Y, GU H, et al. PowerGraph: Distributed Graph–Parallel Computation on Natural Graphs[C]// Proceedings of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI). USA: USENIX Association, 2012: 17–30
[9] KREPS J, NARKHEDE N, RAO J. Kafka: A Distributed Messaging System for Log Processing[C]//Proceedings of the 6th International Workshop on Networking Meets Databases (NetDB). USA: ACM Press, 2011
[10] PENG D, DABEK F. Large–Scale Incremental Processing Using Distributed Transactions and Notifications[C]// Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI). USA: USENIX Association, 2010: 1–15
[11] GUNDA P K, RAVINDRANATH L, THEKKATH C A, et al. Nectar: Automatic Management of Data and Computation in Datacenters[C]// Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI). USA: USENIX Association, 2010:75–88
[12] BHTOTIA P, WIDER A, RODRIGUES R, et al. Incoop: MapReduce for incremental computations[C]//Proceedings of the 2nd ACM Symposium on Cloud Computing. USA: ACM, 2011：7
[13] ANANTHANARAYANAN G, GHODSI A, WANG A, et al. PACMan: Coordinated Memory Caching for Parallel Jobs[C]//9th USENIX Symposium on Networked Systems Design and Implementation(NSDI). USA: USENIX Association, 2012

作者简介

赵晟，清华大学计算机科学与技术系硕士研究生；研究方向为分布式存储与计算。



姜进磊，清华大学计算机科学与技术系副教授；研究方向为分布式计算与系统、云计算、大数据和虚拟化等；先后主持和参加国家自然科学基金、“863”计划、“973”计划等项目 10 余项；获国家技术发明奖二等奖 1 项；已发表论文 50 多篇，其中被 SCI/EI 检索 40 余篇。

DOI:10.3969/j.issn.1009-6868.2016.02.005
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20160224.1702.004.html

分布式数据处理系统内存对象管理问题分析

In-Memory Data-Object Management in Distributed Data Processing System

张雄/ZHANG Xiong
陆路/LU Lu
石宣化/SHI Xuanhua
(华中科技大学 大数据技术与系统湖北省工程实验室,湖北 武汉 430074)
(Big Data Technology and System Lab, Huazhong University of Science and Technology, Wuhan 430074, China)

中图分类号:TP393 文献标志码:A 文章编号:1009-6868 (2016) 02-0019-004

摘要: 通过从程序语言的特性、垃圾回收机制、内存对象的序列化机制到基于区域的内存管理机制分析了内存对象的管理存在的问题,并分析了内存对象的生命周期在内存对象管理中能发挥的作用。提出了基于内存对象的生命周期对内存进行区域化管理的思路,可以从根本上解决垃圾回收问题。

关键词: 大数据;内存对象管理;分布式数据处理系统

Abstract: Through analysis of the characteristics of program languages, mechanism of garbage collection, serialization of in-memory data-objects and region-based memory management, most existing problems with memory management are exposed. What's more, the lifetime of in-memory data objects can be key factor in memory management. Thus a solution is region-based memory management combined with the lifetime of in-memory data objects, which can solve the garbage collection problem.

Key words: big data; in-memory data-object management; distributed data processing system

以MapReduce为代表的分布式数据处理系统使得人们可以以增加硬件资源的方法来处理海量数据。已有的大量研究集中在多核或分布式环境下的可扩展性和容错性。最新的研究工作表明:这类系统计算执行效率是一个被忽视的重要问题。导致执行效率低下的一个重要原因是代表性的开源系统,如Hadoop和Spark,都使用带有托管执行环境的高级语言开发,从而降低分布式环境下的部署和调试的难度。托管环境提供内建的高级功能,比如自动内存管理和并发模型,使得其对象模型的底层实现非常复杂,带来额外的内存和中央处理器(CPU)开销。工业级托管环境的现代即时编译优化很大程度上解决了中间代码(IR)的解释执行效率问题,但是难以解决复杂对象模型实现带来的对象管理开销问题。以Java虚拟机(JVM)为例:(1)每个对象在JVM中会有一个头结构保存元数据,头结构除了记录对象类型,还要支持垃圾收集和并发加锁优化;(2)所有(非原生类型)对象都在堆中创建,因此每个存活对象至少要有一个额外变量保存其引用;(3)泛型容器的元素如果是基本类型,必须首先被装箱为对象类型;(4)主流垃圾收集算法都是基于对象追踪的,因此堆中有大量的存活对象时,垃圾收集器需要耗费大量CPU周期来标记存活对象[1]。随着处理数据量的增大,内存对象越来越多,尤其是长驻内存对象的存在[2],内存对象管理会带来严重的内存膨胀和CPU开销问题。内存膨胀会间接影响执行性能:如果内存足够,更大的内存占用会导致更频繁的垃圾回收;如果内存不足,缓存的数据需要部分丢弃或者换出到磁盘,导致额外的重计算或输入输出(IO)开销。

最初用来解决这类问题的方法是垃圾收集优化。垃圾收集优化分为两个方面:一方面是通过参数[3]调优,避免频繁垃圾收集;另一方面是通过优化垃圾收集算法实现来提高垃圾收集的性能[4]。垃圾收集优化的方法只是减缓了垃圾收集操作的影响,实际上内存对象管理所存在的问题仍然存在。后续的性能优化方案逐渐从垃圾收集的优化深入到内存对象管理本身,针对数据对象在内存中的存储进行优化,主要包括序列化存储[5],基于区域的内存管理[6]。这种解决方案从根本上解决了对象对内存资源的占用,但是仍然无法避免对

收稿时间:2016-01-23
网络出版时间:2016-02-25
基金项目:国家自然科学基金(61433019、61370104)

象的存在。在对象存储优化的基础上，目前分布式系统的一些上层特定应用，例如 Spark 结构化查询语言（SQL），利用特定的数据结构解决了对象存储优化的不足，从根本上消除对象。当然，由于是特定的应用系统，使用范围窄。

1 垃圾收集优化

垃圾收集问题是内存对象管理中最重要的问题之一，也是影响系统性能的关键因素。因此，最初针对内存对象管理问题的解决方法都是从垃圾收集入手。

1.1 垃圾收集调优

垃圾收集调优是最传统的垃圾收集优化技术，也是一些长时间运行的低延迟 Web 服务所推荐的方法。一些开源分布式数据处理系统，例如 Cassandra 和 HBase 都使用以延迟为中心的方法来避免长时间垃圾收集开销[7]。以上所述垃圾收集调优方法的关键在于：用标记清除算法（CMS）的垃圾收集控制器代替原有以吞吐量为中心的垃圾收集；调整标记清除算法的垃圾收集控制器参数以降低垃圾收集开销。

1.2 垃圾收集算法优化

目前的垃圾收集算法有引用计数法、标记清除算法、拷贝收集算法等，不同的垃圾收集算法让内存对象管理有更多的选择来处理对象的回收。垃圾收集算法的优化性能要到达最优一般有特定场景，例如非统一内存访问（NUMA）感知的垃圾回收器[8]。并且，垃圾收集算法的优化只是掩盖了内存对象管理的问题，内存对象的自动化管理存在的问题仍然存在，频繁调用垃圾收集的本质因素没有解决。

1.3 程序语言优化

分布式数据处理系统使用高级面向对象语言进行开发会导致内存对象管理的问题，而传统的面向机器的语言，如 C、C++，则不存在内存对象管理存在的问题。为了追求性能上的优势，一部分企业机构会选择用这些传统的语言来改写目前的分布式数据处理系统，但是失去了高级语言特性的系统开发难度非常之大，并且不利于系统的更新。

2 对象存储的优化策略

垃圾收集的优化并没有考虑内存对象管理所存在的问题的本质，即内存中的对象仍然是自动化管理的。所以一些系统将对象用序列化的方式存放到内存以减少内存的占用来防止频繁垃圾收集，或者将常规的内存对象管理方法替换为针对对象标记回收的区域内存管理方法来消除垃圾收集。这一类方法从内存对象管理的本质上考虑了性能问题。

2.1 序列化存储

目前的分布式数据处理系统，如 Hadoop 和 Spark，都支持将中间数据对象序列化为 byte 数组，从而减少对象在内存中存储的占用。Hadoop 系统中的对象大部分都是临时的数据对象，因此 Hadoop 仅将 Map 的输出数据序列化成 byte 数据，存放到磁盘，然后通过 Shuffle 传输给 reduce task。尽管不存在内存对象管理的问题，但是序列化机制确实对分布式数据处理系统有重要作用。Spark 系统不仅在 Shuffle 阶段提供序列化机制，还在 Cache 时提供了序列化选择，Cache 时 Spark 会将弹性分布式数据集（RDD）中的数据保存到内存，用户可以选择是否采用序列化保存数据。Spark 之所以支持非序列化保存，是因为序列化机制存在序列化和反序列化的开销。一般来说，序列化机制能够有效降低内存对象的占用，但是要在 Cache 数据对象时执行序列化操作，而在使用对象时执行反序列化操作。

序列化存储降低了内存对象的占用，但是应用仍然基于对象执行的。因此在序列化和反序列化的基础上，内存对象管理仍然需要考虑中间对象的管理，当数据量大时，对象的回收仍然会影响系统的性能[9]。

2.2 基于区域的内存管理

无论是基于垃圾收集调优还是序列化存储，都是由内存管理机制自动标注对象的生命周期，始终存在对象的操作，就必然会需要内存管理机制根据标注回收无需再使用的对象，也就必然会导致垃圾收集。从内存对象相反的一个方向分析，C、C++等语言完全手动的标注内存中使用的对象，手动的回收对象。基于区域的内存管理综合了自动化和完全手动标注内存对象的两种策略，采取了绕过垃圾收集的策略，将一部分对象统一标记后直接存储到堆外区域，整块回收区域内的对象，从而消除频繁垃圾收集，解决内存对象管理的问题。

FACADE[10]系统是基于区域的内存管理的典型实例。FACADE 以程序分析为基础，在程序代码中由用户标识需要转换的 Java 对象。FACADE 会首先识别用户标注的 Java 对象，将其转换为轻量级的 FACADE 对象并通过 byte 形式保存 FACADE 到堆外内存，极大地减少了对象对内存的占用，而 FACADE 相比序列化更加进一步消除对象之处在于它同时转换了 Java 对象的操作代码。用户自定义的操作函数是基于 Java 对象的，FACADE 转换 Java 对象为 FACADE 对象后，同时将操作函数转换为基于 FACADE 对象的操作函数，完全实现了对象的消除。FACADE 的内存对象管理采取了整块分配和整块回收的原则，一方面配合 FACADE 对象的存储方式；一方面减少了垃圾收集开销，相比传统的内存对象管理取得了非常好的效果。

尽管 FACADE 的内存对象管理已经从很大程度上解决了对象管理的垃圾收集问题，但是它基于一个很强烈的假设：在整块分配和整块回收的

操作间隔内的所有对象在内存中的存活时间都是相同的。在一些分布式数据处理系统中，例如 Spark 和 Flink，将作业划分为有向无环图，按照每个阶段执行。这类系统中的数据对象在内存中的存活时间就非常复杂，如果有用户将数据 Cache 到内存，数据对象的存活时间持续整个作业执行期；如果是 Shuffle 阶段的数据对象，数据对象可能会在多个阶段的执行期内都存活在内存中。所以，FACADE 在内存对象的管理上忽略了内存对象的生命周期。

Broom[11]综合考虑了基于区域的内存管理的特点以及内存对象的生命周期的特点，Broom 以 NET CLR 平台为研究对象，将内存对象的存放区域进一步区分为：可转移区域，用来存放操作的传递消息，同一时间只有一个操作可以访问该区域；操作所需区域，针对某个操作私有的对象存储区域，该区域内的对象的生命周期与相应的操作生命周期相同；临时区域，存放一些临时的数据对象。由于 Broom 基于闭源的系统实现而且作为 short paper 对系统实现提及较少，所以能够获取的信息只在于基于区域的内存管理与内存对象生命周期特点的结合是消除垃圾收集所必须考虑的两个因素。

3 特定应用领域的优化

大多优秀的开源分布式数据处理系统，如 Hadoop 和 Spark，都基于底层的系统实现了上层应用领域的生态系统[12]，例如 Spark 生态系统包括 Spark SQL、Spark Streaming、Spark GraphX 和 Spark MLlib。特定应用领域的分布式数据处理系统在底层数据系统的基础上定义了特定的计算结构，从而可以实现更加复杂的内存对象管理机制。在最初 Java 等高级语言被应用在数据分析系统时，除了受益于高级语言的特性，一些系统也意识到其在内存对象管理上的不足，因此结合特定的结构进行优化[13]，例如 SQL。Shark 等针对 data-intensive 的数据库管理系统，将 Java 对象转化为 Telegraph 数据流，在内存分配上整存整取，绕过 Java 的内存管理方法[14]。Shark 使用基于列的内存存储和动态查询优化来提高 SQL 查询的性能。Apache 项目 Tungsten 基于 Spark SQL 实现，将传统的关系表结果转换为以列为结构的字节序列，同时将 SQL 操作全部转换为基于字节序列的操作。

分布式数据处理系统内存对象管理的解决方案各具优点和缺点，具体见表 1。

4 结束语

分布式数据处理系统按照数据流的路径可以分为控制路径和数据路径，控制路径由系统框架支持和实现，数据路径由用户定义和实现。我们发现：控制路径的编程实现更多的从对象模型的高级特性中获益，包括类型的运行时动态识别、并发同步操作的偏向锁优化和自动对象内存管理等。数据路径的编程实现很少使用语言的高级特性。数据路径的实现由用户自定义，具体包括用户自定义类型（UDT）和用户自定义方法（UDF）。UDT 定义的是数据路径中实际操作的对象类型，而 UDF 定义了对这些数据类型执行的操作。UDT 通常是基本类型的浅层组合和常用方法的实现封装，很少会使用复杂的继承层次和多态。

比如，通过定义接口来抽象化模块之间的交互，进而通过工厂模式，依赖注入来支持灵活的模块和插件加载，而这些设计模式依赖于多态和反射等语言特性。框架负责任务的并行化执行和同步机制，因此依赖于托管环境的并发执行模型，比如线程创建和加锁操作。由于并发执行由框架负责，UDF 本身都是串行代码，在 UDF 内部使用加锁操作通常没有意义，更不可能会在 UDT 数据对象上加锁。因此，UDT 并不依赖于一个复杂的对象模型实现。比如：（1）当没有多态导致的虚方法派发时，也没有使用反射时，不需要在对象头部记录对象的运行时类型；（2）当对象没有加锁操作时，也不需要在头部存储偏向锁状态；（3）UDT 数据对象的生命周期具有很强的规律性，如果能够跳过 JVM 的内存管理，不仅消除了垃圾收集的 CPU 开销，也不需要在对象头部存储垃圾收集所需的状态信息。

最重要的一点是：数据对象的生命周期具有很强的规律性。我们发现：在以 Spark 为代表的新一代通用数据并行系统中，作业执行时通常有几类数据容器会持有数据对象，而数据对象的生命周期和持有它的数据容器的生命周期有很强的关联性：

（1）UDF 变量。包括 UDF 对象的字段和 UDF 局部变量，前者的生命周

▼表 1 3种内存对象管理问题的解决方案对比分析

解决方案		特点	优点	缺点
垃圾收集优化	垃圾收集调优	调整系统运行参数，减少垃圾收集频率	适用范围广，目前系统均适用	调节能力有限，不可避免垃圾回收
	垃圾收集算法优化	优化垃圾收集算法，加快垃圾收集过程	一种优化算法可以适用多种系统	没有从系统层面解决内存对象的管理问题
	程序语言优化	利用语言特性手动释放占用的内存	从语言层面根本解决了内存中对象的管理问题	面向对象语言的特性不复存在，系统开发难度大
对象存储优化	序列化存储	利用序列化器减少内存中对象的占用	精简了内存中的对象形式，一定程度上缓解了垃圾收集的影响	数据量大而内存有限时，仍然无法解决内存对象的管理问题
	基于区域的内存管理	堆外区域整块标记、分配和回收	综合了自动回收内存空间和面向对象的自动内存管理的优点，减少了频繁了垃圾回收	需要用户手动标记，影响用户程序编写；区域的生命周期复杂时适用程度有限
特定领域优化		利用上层特定系统的特定数据结构优化内存中的对象	特定的数据结构可以极大程度上精简内存中的对象存储，也非常适合进一步在流程上优化	优化策略的通用性有限，只适用于一类特定的系统

期刚好为一个任务的执行时间，其存活时间内时所持有的对象生命周期由该字段的赋值操作决定，但最长不超过任务的执行时间；后者的生命周期为一次 UDF 方法的调用，因此其持有的数据对象最多存活一次方法调用，可以视为临时对象。

（2）缓存数据集。缓存数据集的生命周期由应用程序显示决定，其持有的数据对象的生命周期具有与该 RDD 等同的生命周期。

（3）Shuffle 缓冲区。不考虑溢出到磁盘的情况，Shuffle 缓冲区的生命周期也刚好为一个任务的执行时间。其持有的数据对象的生命周期较为复杂，在聚合计算的过程中，Shuffle buffer 仅为一个 Key 数据对象保存一个数据对象作为当前聚合的结果。

一种可行的方法是通过自动转换数据处理应用程序来减少程序运行时创建的 UDT 数据对象的数量。转换工作对应用开发人员完全透明，不会限制数据并行编程模型的表达能力和灵活性。

致谢

本研究得到华中科技大学金海教授的指导，谨致谢意！

参考文献

[1] JONES R, HOSKING A, MOSS E. The Garbage Collection Handbook : the Art of Automatic Memory Management [M]. USA: CRC Press, 2012
[2] ZAHARIA M, CHOWDHURY M, DAS T, et al. Resilient Distributed Datasets: a Fault–Tolerant Abstraction for in–Memory Cluster Computing [C]//Proceeding in 9th USENIX Conference on Networked Systems Design and Implementation (NSDI). USA: USENIX Association, 2012：141–146
[3] Cassandra Garbage Collection Tuning, Find and Fix Long GC Pauses [EB/OL]. [2013–11–14]. http://aryanet.com/blog/cassandra–garbage–collector–tuning
[4] Laboratory for Web Algorithmic [EB/OL]. [2014–10–12].http://law.di.unimi.it/datasets.php
[5] CARPENTER B, FOX G, KO S H, et al. Object Serialization for Marshalling Data in a Java Interface to MPI[C]//ACM Java Grande Conference. USA: ASM, 1970：66–67
[6] MADS T, JEAN–PIERRE T. Region–Based Memory Management [J]. Information & Computation, 1997, 132(2):109–176
[7] The Garbage Collector and Apache Hbase [EB/OL]. (2016–02–18)[2014–03–22].http://hbase.apache.org/book.html#gc
[8] 浦云明. JAVA 垃圾收集器算法分析及垃圾收集器的运行透视[J]. 计算机系统应用, 2003 (11): 39–41
[9] MILLER H, HALLER P, BURMAKO E, et al. Instant Pickles: Generating Object–Oriented Pickler Combinators for Fast and Extensible Serialization [J]. ACM Sigplan Notices, 2013, 48(10):183–202
[10] NGUYEN K, WANG K, BU Y, et al. FACADE: A Compiler and Runtime for (Almost) Object–Bounded Big Data Applications [J]. ACM Sigplan Notices, 2015, 50(4):675–690. DOI: 10.1145/2775054.2694345
[11] GOH I, GICEVA J, SCHWARZKOPF M, et al. Broom: Sweeping Out Garbage Collection from Big Data Systems [C]//15th Workshop on Hot Topics in Operating Systems (HotOS XV). USA: ACM, 2015
[12] 胡俊, 胡贤德, 程家兴. 基于 Spark 的大数据混合计算模型[J]. 计算机系统应用, 2015(4): 214–218
[13] SHAH M A, FRANKLIN M J, MADDEN S, et al. Java Support for Data–Intensive Systems: Experiences Building the Telegraph Dataflow System [J]. Sigmod Record, 2001, 30(4):103–114
[14] XIN R. S, ROSEN J, ZAHARIA M, et al. Shark: SQL and Rich Analytics at Scale[C]// Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. USA: ASM, 2013：13–24. DOI: 10.1145/2463676.2465288

作者简介

张雄，华中科技大学在读硕士；主要研究领域为分布式数据处理系统。

陆路，华中科技大学在读博士；主要研究领域为分布式数据处理系统。

石宣化，华中科技大学教授；主要研究领域为并行计算与分布式系统。

综合信息

光纤到户堡垒拆除：网速直通 10G

最近，英国伦敦大学设计并测试了一种新型光接收机，有望大大降低光纤网络直达家庭用户的成本，使每个家庭直接与全球互联网相连。

据物理学家组织网 14 日报道，FTTH 通常只到交接箱，还远不及终端用户。所谓的“最后 1 公里”，即家庭用户通过交接箱与全球互联网的连接，大多是用铜缆，但能读取光信号的光接收机非常昂贵，许多家庭难以负担。即使在 FTTH 技术领先的日本、韩国等国家，FTTH 连接也不足 50%，在英国还不到 1%。

限制 FTTH 的主要原因是成本，要实现它不仅要把光缆铺到每个家庭，还要提供用户负担得起的光接收机。英国伦敦大学光网团队和其他团队共同合作开发的新型光接收机，保留了传统光接收器的诸多优点，但体积更小，只有原来 75%~80%的组件，显著降低了制造成本和维修成本，而且它的灵敏度能与现有的网络相匹配。

（转载自《中国信息产业网》）

DOI:10.3969/j.issn.1009-6868.2016.02.006
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20160303.1005.002.html

Spark计算引擎的数据对象缓存优化研究

Data Object Cache in Spark Computing Engine

陈康/CHEN Kang
王彬/WANG Bin
冯琳/FENG Ling
(清华大学 计算机科学与技术系,北京 100084)
(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

中图分类号:TN929.5 文献标志码:A 文章编号:1009-6868 (2016) 02-0023-005

摘要: 研究了Spark并行计算集群对于内存的使用行为,认为其主要工作是通过对内存行为进行建模与分析,并对内存的使用进行决策自动化,使调度器自动识别出有价值的弹性分布式数据集(RDD)并放入缓存。另外,也对缓存替换策略进行优化,代替了原有的近期最少使用(LRU)算法。通过改进缓存方法,提高了任务在资源有限情况下的运行效率,以及在不同集群环境下任务效率的稳定性。

关键词: 并行计算;缓存;Spark;RDD

Abstract: In this paper, Spark parallel computing cluster for memory is studied. Its main work is about modeling and analysis of memory behavior in the computing engine and making the cache strategy automatic. Thus, the scheduler can recognize a valuable data object to be cached in the memory. A new cache replacement algorithm is proposed to replace least recently used (LRU) and have better performance in some applications. Thus, the performance and reliability of the Spark computing engine can be improved.

Key words: parallel computing; cache; Spark; resilient distributed dataset(RDD)

大数据处理的框架在现阶段比较有影响力的是基于Google所发明的MapReduce[1]方法及其Hadoop[2]的实现。当然,在性能上传统的消息传递接口(MPI)[3]会更好,但是在处理数据的方便使用性、扩展性和可靠性方面MapReduce更加适合。使用MapRecuce可以专注于业务逻辑,不必关心一些传统模型中需要处理的复杂问题,例如并行化、容错、负载均衡等。

由于Hadoop通过Hadoop分布式文件系统(HDFS)[4]读写数据,在进行多轮迭代计算时速度很慢。随着需要处理的数据越来越大,提高MapReduce性能变成了一个迫切的需求,Spark[5]便是在此种背景下应运而生。

Spark主要针对多轮迭代中的重用工作数据集(比如机器学习算法)的工作负载进行优化,主要特点为引入了内存集群计算的概念,将数据集缓存在内存中,以缩短访问延迟。

Spark编程数据模型为弹性分布式数据集(RDD)[6]的抽象,即分布在一组节点中的只读对象集合。数据集合通过记录来源信息来帮助重构以达到可靠的目的。RDD被表示为一个Scala[7]对象,并且可以从文件中创建它。

Spark中的应用程序可实现在单一节点上执行的操作或在一组节点上并行执行的操作。对于多节点操作,Spark依赖于Mesos[8]集群管理器。Mesos能够对底层的物理资源进行抽象,并且以统一的方式提供给上层的计算资源。通过这种方式可以让一个物理集群提供给不同的计算框架所使用。

Spark使用内存分布数据集,除了能够提供交互式查询外,它还可以优化迭代工作负载。使用这种方法,几乎可以将所有数据都保存在内存中,这样整体的性能就有很大的提高。然而目前Spark由于将缓存策略交由程序员在代码中手动完成,有可能会引起缓存的低效甚至程序出错,主要原因如下:

(1)程序员如果缓存无用的数据,将会导致内存未被充分利用,降低内存对程序性能的提升;

(2)错误的缓存甚至会产生内存溢出等严重后果,直接导致程序崩溃出错;

(3)程序中有具有缓存价值的数

收稿时间:2016-01-16
网络出版时间:2016-03-03
基金项目:国家高技术研究发展("863")计划(2013AA01A213);国家自然科学基金(61433008、61373145、61170210、U1435216);国家核高基重大专项(2013zx01039-002-002)

据得不到缓存，将使程序不能达到最高效率。

随着项目变大，代码量增加，这个问题会变得越来越严重。如果使用自动分析的方法，自动完成缓存的工作，无疑会降低程序员负担以及避免上述的问题。下面将对这方面进行初步研究，通过分析与建模，目的是使内存的使用更加智能有效，并加速任务的运行速度。

1 Spark中缓存优化研究

我们分3个方面对Spark中的缓存进行研究：一个是缓存自动化方法；一个是缓存替换方法改进；最后是程序执行调度顺序与缓存的关系。

1.1 Spark中的缓存

Spark通过将RDD数据块对象缓存在内存中这一方式对MapReduce程序进行性能上的提升改进。以经典的PageRank[9]算法为例，Spark比Hadoop快3倍左右。

以逻辑回归算法实验代码为例，如图1所示。

可以看出：在使用Spark时，从第2轮迭代计算开始，points的数据可以从缓存中直接读取出来，因此获得了极高的加速比。

1.2 数据对象的自动缓存

对于Spark来说，并不是每个RDD都要缓存到内存当中，需要进行筛选保留有价值的RDD存入内存。目前Spark中这种筛选的工作都交由程序员手动完成。以PageRank作为例子，如图2所示。

代码中有3个变量，且3个变量都是RDD类型，其中第1行最后的cache操作，会导致links被缓存到内存，在循环中可以从内存中直接读取，而ranks和contribs则没有被缓存，这就是Spark当前的缓存机制。

将缓存的工作交由程序员手动完成，对于系统本身的实现来讲，是简化了许多，但是对于程序员来讲则是极大的挑战。对于复杂操作，程序员通常很难直接分析出具有缓存价值的数据对象进行缓存。sortByKey是用来对数据依据其指定的键值进行排序的一个常用操作，具体如图3所示。

```
val points = spark.textFile(...).map(parsePoint).cache()
var w = Vector.random(D) // current separating plane
for (i <- 1 to ITERATIONS) {
    val gradient = points.map(p =>
    (1 / (1 + exp(-p.y*(w dot p.x))) -1) * p.y * p.x).reduce(_ + _)
    w -= gradient
}
println("Final separating plane: " + w)
```

▶图1 逻辑回归算法的Spark实现代码

```
val links = Spark.textFile("HDFS:...").map(...).cache()
val ranks = // RDD of (URL, rank) pairs
for (i <- 1 to ITERATIONS) {
            val contribs = links.join(ranks).flatmap {
            (url, (links, rank)) =>
            links.map(dest => (dest, rank/links.size))
    }
            ranks = contribs.reduceByKey((x,y) => x+y).
      mapValues(sum => a/N + (1-a)*sum)
}
ranks.save("HDFS:...")
```

▶图2 PageRank的Spark实现代码

可以看出，代码中的3个RDD均只使用了一次，在缓存links与不缓存links两种情况下，理论上两个时间应该相近，因为该RDD对象并未被再次利用，没有缓存的价值。然而实际中缓存links比不缓存快了近1倍。通过分析源码发现sortByKey的具体实现过程中有两个隐性任务，而这两个任务有数据相关性，通过缓存可以获得性能提升。这种数据相关性对于不了解sortByKey实现细节的程序员来说，是很难被察觉到的。

因此，将缓存的工作交由程序员手动完成，会让代码运行的效率随着程序员的水平不同而差别巨大，低效的缓存策略会让程序变慢，错误的缓存策略甚至会导致程序出错这一严重后果，由程序通过智能分析自动完成缓存策略则变得尤为重要。我们提出了一种缓存策略的自动化实现策略：

（1）在Spark源码中插入监听代码，当程序运行时记录程序中的关键信息，得到代码的无回路有向图（DAG）[10]，其中DAG图中的点对应程序中的RDD，边对应函数操作；

（2）统计DAG图中每个点的出度，即每个RDD将被使用到的次数；

（3）选出出度大于1的点，并将其对应的RDD进行缓存。

可以看出该方法需要运行2次程序：第1次为了分析出有价值的RDD进行缓存；第2次才是真正的作业运行。因此可以修改Spark的调度器，使其第1次运行时使用kB级别的小数据集，这样可以在很短的时间内运行完毕，相对于第2次真正运行时的大数据集（通常是GB或TB级别），额外一次运行不会影响性能。

1.3 数据对象缓存的调度策略

相对于MapReduce的计算框架，Spark的优势在于能将有价值的数据缓存到内存中，这样在迭代计算时能

```
val file = spark.textFile("hdfs:...")
val links = file.map(parseArctile _)
val sortlinks = links.sortByKey()
sortlinks.saveAsTextFile("hdfs:...")
```

▲图3 sortByKey的Spark实现代码

直接从内存读取数据，减少磁盘输入输出（IO），提高读写速度。然而实际生产中，由于内存大小有限，不可能缓存全部有价值的数据，只能部分缓存，在这种情况下，缓存替换算法的效率将极大影响作业运行效率。

目前，Spark 缓存替换算法使用的是近期最少使用算法（LRU）。基本方法是：当内存空间不足需要替换时，将缓存中距离当前时间最久没有被用到的数据替换出去；但在缓存空间不足时，Spark 的缓存调度器由于无法准确预测数据将来的使用顺序，导致LRU 替换算法效果欠佳。

实际上，针对缓存空间不足的情况，采用寄存器分配（RA）算法[11]会有更好的效果。已知内存容量的大小、RDD 的个数和 RDD 的访问顺序（由此可以得出 RDD 的生命周期以及每个 RDD 生命周期之间的关系），可以使用 RA 模型将 RDD 分配到内存的不同位置中。具体实施如下：

（1）通过分析程序得到每个 RDD 的生命周期，即开始使用时间和结束使用时间，并将该时间区间按照开始时间递增的顺序存入到 list 列表中；

（2）顺序遍历 list 列表，从缓存区中尝试为每个 RDD 分配可用空间；

（3）在遍历过程中持续维护一个列表，列表存放遍历的 RDD 的生存区间，并按照 RDD 的结束使用时间递增排序，同时删除当前时间已结束使用的 RDD，将其占用的空间释放返还到缓存区；

（4）如果缓存区已满，则将当前待分配空间的 RDD 加入上一步维护的列表中，并将结束使用时间最晚的 RDD 从列表中移除，将其占用的空间释放分配给当前待分配的 RDD；若需要移除的是当前 RDD，则不为其分配空间。

在 Spark 作业中，通过存储 RDD 的元数据信息来到达容错的目的。在执行到真实的数据读取与计算操作即 Action 操作之前，代码中计算的都只是 RDD 的元数据信息，并不会改变真实数据。只有遇到 Action 操作时才会进行真正的数据计算，并根据存储的 RDD 的元数据信息向前回溯，直至找到所需要的数据。因此，RDD 的访问顺序实质上是根据 Action 的顺序决定的。

低效的 Action 顺序，会导致低效的运行效率，以图 4 为例。图中的 R 为计算过程中产生的元数据 RDD，A 为触发真实数据的读写与计算的行为 Action。如果 Action 顺序为：A1、A4、A2、A5、A3，则 RDD 的访问顺序为：A、B、A、C、[A]、B、[A]、C、[A]、B、[A]、C。

可以看出，该访问顺序导致 RDD 频繁替换的概率要大出许多，因此 Action 顺序的优化变得尤为关键。

针对 Action 顺序优化问题，可以使用一种贪心算法来计算 Action 的顺序，具体如下：

（1）将初始的 Action 集合根据依赖关系进行聚类，形成新的 Action 集合 A，并建立一个空集合 R，一个空序列 S；

（2）从集合 A 中随机选出一个 Ai 加入 S 末尾，同时从 A 中剔除 Ai，并将 Ai 对应的所有 RDD 加入集合 R；

（3）遍历集合 A，选出当 Aj 对应的 RDD 集合与 R 交集最大时的 Action Aj，将 Aj 加入 S 末尾，同时从 A 中剔除 Aj，并将 Aj 对应的所有 RDD 加入集合 R；

（4）重复上述步骤，直至集合 A 为空，计算结束，序列 S 即为 Action 的优化后顺序序列。

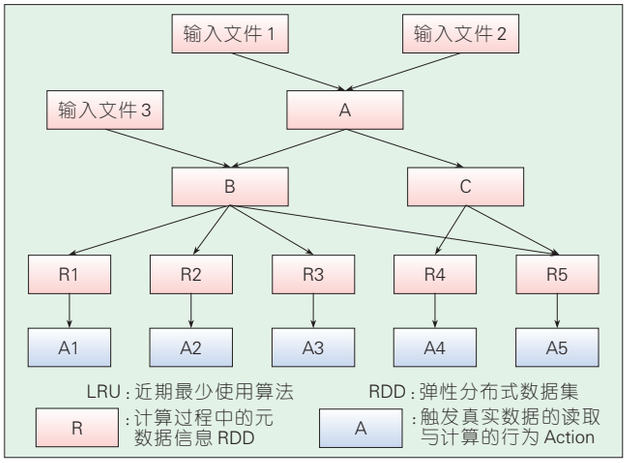
2 Spark 中缓存优化实验

2.1 实验环境

我们使用两套不同的集群配置运行不同的任务。第 1 部分使用 3 台节点搭建的小集群，配置如下：16 核中央处理器（CPU），48 G 内存，750 G 硬盘，网络带宽 1 000 M，操作系统为 64 位 CentOS 5.4；第 2 和第 3 部分实验使用 16 台节点配置的集群，配置如下：12 核 CPU，48 G 内存，880 G 硬盘，1 000 M 网络带宽，操作系统为 64 位 RedHat 6。使用的软件及版本为：Spark 0.5、Mesos 0.9.0、Hadoop 0.20.205。

2.2 实验 1——逻辑回归算法

我们以 1.1 节中的代码进行验证，该代码采用的是逻辑回归算法，是对数据的一种划分方式。计算方法为：通过多轮迭代计算来不断修正初始值 w，直至 w 的值在某轮计算后的改变小于设定阈值或迭代计算的次数达到设定次数上限。这里使用的是达到设定迭代次数上限的方式来停止计算。其中需要缓存的 RDD 为变量 points。实验主要观察在不同的数据大小的情况下，缓存 points 和不缓存 points 的运行时间的区别，实验结果如表 1 所示。



▶图 4 LRU 行为解析示例

▼表1 逻辑回归算法实验结果

数据大小/ G	迭代轮数	运行时间(缓存)/s	运行时间(未缓存)/s
2.8	迭代1	30	30
	迭代2～5	0.7	8
5.5	迭代1	66	72
	迭代2～5	0.9	12
14.0	迭代1	191	228
	迭代2～5	1.8	30

由实验结果可以发现：第1轮由于需要从HDFS中读取数据，缓存不能命中，因此使用缓存和不使用缓存对运行时间没有影响，二者时间基本相同；但从第2轮开始，由于可以从缓存中读取points数据，使用缓存将使性能极大提高，可以获得10～20倍左右的加速比。该实验证明Spark中的缓存机制在数据密集型计算中有巨大的提升效果。

2.3 实验2——sortByKey

我们以1.2中sortByKey的代码进行验证。上文中我们已经分析过，理论上两个时间应该相近，结果如图5所示。

使用缓存使运行时间缩短了一半。通过对实验过程的进一步细化拆分，我们发现整个任务又可以拆分成3个子任务，运行时间分别如图6所示。

可以发现：对于任务2和任务3，使用缓存使任务时间极大地缩短，任务2获得近200倍的加速比，任务3也有近5倍。sortByKey采用了sample的方法，因此两个数据相关的隐性任务2和任务3封装在代码当中。对于类似sortByKey这样的复杂操作，很容易出现这种具有隐藏的数据相关的情况，如果不了解这些复杂操作的具体实现细节，很难发现这种相关性加以利用。

2.4 实验3——不同缓存策略比较

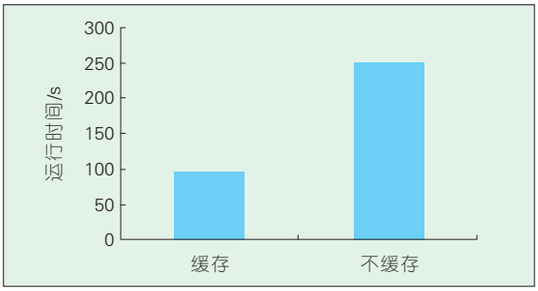
在该实验中，我们主要对比不同的缓存策略对任务执行效率的影响，使用的是图1所示的代码，一个PageRank的实现。下面我们将对比LRU替换算法和RA替换算法的运行时间的情况。代码中有多个RDD需要缓存，采用两种替换策略，在不同的缓存大小情况下，任务的运行时间差别很大，主要测试的是：在缓存空间极少（仅能缓存1个RDD）、较少（能够缓存一半的RDD）以及充足（能够缓存所有RDD）的情况下，在LRU算法和RA算法两种不同替换策略下，任务运行时间的变化，实验结果如图7所示。

从图7中我们可以看出：当所有RDD都缓存后，使用LRU替换算法与RA替换算法的运行时间都随可用缓存空间的增大而降低，当缓存空间足够大到能够缓存所有运行时的RDD时，两者的运行时间基本相同，均为120 s。但是对于缓存空间不足的情况，RA算法明显比LRU算法的性能要好：在缓存空间较少，仅能缓存运行时产生的一半的RDD的情况下，LRU算法运行时间为460 s，而RA算法运行时间只有350 s。RA替换算法更能适应不同的情况，在大多情况下都比LRU替换算法运行效率更高。

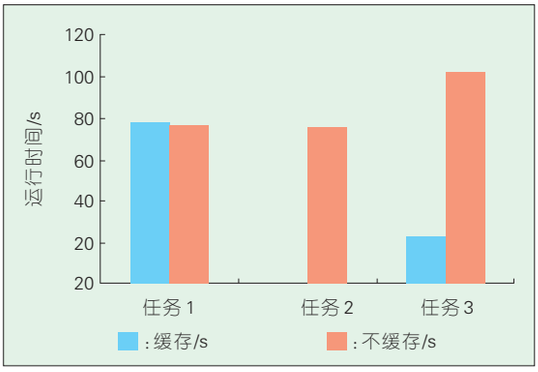
2.5 实验4——Action顺序效果实验

该实验的目的是验证Action的顺序对于任务运行时间的影响。实验代码的核心部分代码如图8所示，其中1为原始代码，2为做Action顺序优化后的代码。

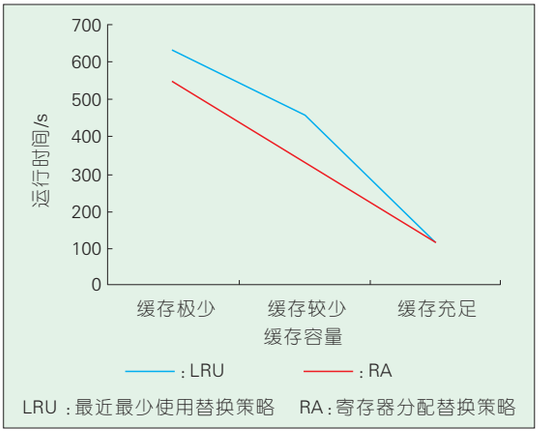
实验结果如图9所示：测试了在替换算法相同时，在内存极少以及内存较少的情况下，不同Action顺序的运行时间。从图中我们可以看出：Action顺序优化后的代码的运行时间在内存极少和内存较少的情况下均比优化前代码运行时间短，尤其在内



▲图5 sortByKey实验结果1



▲图6 sortByKey实验结果2



▲图7 两种缓存策略的结果比较

存较少的情况下，可以达到近4倍的加速比。Action顺序优化的作用十分明显。

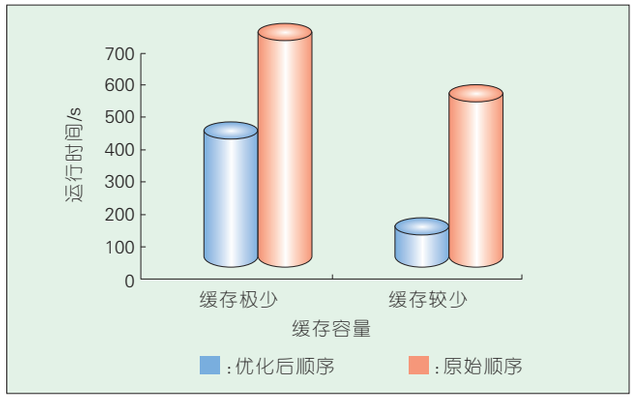
3 结束语

通过对并行计算框架Spark进行深入研究，并对Spark中的内存使用模型进行系统分析后，我们从多个角度对Spark中的缓存系统进行改进，使得系统具有更强的鲁棒性，并且在内存空间不足、执行的作业复杂等恶劣情况下依然有较高的性能。首先提出了自动化缓存策略的制定，将程序员从手动制定缓存策略中解放出来，降低程序员的编程时间以及出错的可能性；其次，对Spark原有的RDD缓存替换策略进行优化，使得新的替换策略在内存空间不足的情况下比原有策略性能更好，提高作业运行效率；最后，研究了Action顺序优化对作业运行效率的影响。通过这一系列的建模与改进工作，可以使Spark变得更有效率，最后也在实验的结果验证了这一观点。

理论建模是文中所涉及工作的重点内容，但为了体现出工作的实际效果，还需要大量的工程实践，将改进后的系统真正完善起来，才能真正投入应用，从而发挥出实际价值。

```
[1]
for(i <- 1 to 5) {
  links1.count
  links2.count
}
[2]
for(i <- 1 to 5){
  links1.count
}
for(i <- 1 to 5) {
  links2.count
}
```

▲图8 两种Action顺序的实现代码



◀图9 Action顺序结果

参考文献
[1] DEAN J, GHEMAWAT S. Mapreduce: Simplified Data Processing on Large Clusters [J]. Communications of the ACM 50th Anniversary Issue, 2008, 51(1): 107–113. DOI: 10.1145/1327452.1327492
[2] GABRIEL E, FAGG G, BOSILCA G, et al. Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation[C]// Proceedings European PVM/MPI Users' Group Meeting. Germany: Springer Berlin Heidelberg, 2004: 97–104. DOI: 10.1007/978–3–540–75416–9_35
[3] GHEMAWAT S, GOBIOFF H, LEUNG S T. The Google File System[C]//Proceedings of the Nineteenth ACM Symposium on Operating Systems. USA: ACM, 2003: 29–43
[4] CHANG F, DEAN J, GHEMAWAT S, et al. Bigtable: A Distributed Storage System for Structured Data[C]//Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation–Volume 7. USA: USENIX Association, 2006: 15–15
[5] JIANG Y. HBase Administration Cookbook [M]. UK: Packt Publishing, 2012
[6] ZAGARIA M, CHOWDURY M, DAS T, et al. Resilient Distributed Datasets: A Fault–Tolerant Abstraction for In–Memory Cluster Computing[C]//Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. USA: USENIX Association, 2012:2–2
[7] OLIVEIRA B C, GIBBIONS J. Scala for Generic Programmers[C]//Proceedings of the ACM SIGPLAN workshop on Generic programming. USA: ACM, 2008: 25–36
[8] HINDMAN B, KONWINSKI A, ZAHARIA M, et al. Mesos: A Platform for Fine–Grained Resource Sharing in the Data Center[C]// Proceedings of the 8th Usenix Conference on Networked Systems Design and Implementation. USA: USENIX Association, 2011: 22–23
[9] ZHANG J, ZHOU H, CHEN R, et al. Optimizing Data Shuffling in Data–Parallel Computation by Understanding User–Defined Functions[C]//Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. USA: USENIX Association, 2012: 22–23
[10] OLSTON C, REED B, SILBERSTEIN A, et al. Automatic Optimization of Parallel Dataflow Programs[C]//Proceedings of USENIX 2008 Annual Technical Conference on Annual Technical Conference. USA: USENIX Association, 2008: 267–273
[11] SMITH M D, RAMSEY N, HOLLOWAY G. A Generalized Algorithm for Graph–Coloring Register Allocation[C]//Proceedings of the ACM Sigplan 2004 Conference on Programming Language Design and Implementation.USA: ACM, 2004: 277–288

作者简介

陈康，清华大学计算机科学与技术系副教授；研究方向为分布式系统、存储系统等；先后主持和参加“863”、“973”等项目10余项；获得1项科研成果奖；已发表论文20多篇，其中被SCI/EI检索10余篇。

王彬，清华大学计算机科学与技术系硕士研究生；研究方向为分布式计算。

冯琳，清华大学计算机科学与技术系硕士；研究方向为分布式计算。

DOI:10.3969/j.issn.1009-6868.2016.02.007
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20160222.1016.002.html

大数据存储系统中负载均衡的数据迁移算法

Load Balanced Data Migration Algorithm for Big Data Storage Systems

李甜甜/LI Tiantian
王智/WANG Zhi
宋杰/SONG Jie
(东北大学,辽宁 沈阳 110819)
(Northeastern University, Shenyang 110819, China)

中图分类号:TP393 文献标志码:A 文章编号:1009-6868 (2016) 02-0028-005

摘要: 认为在大数据时代,数据迁移已成为以数据为中心的挖掘分析操作的基础环节。通过对大数据存储系统中的数据迁移进行需求分析,首先提出了数据迁移模型,并分析了影响迁移性能的因素;然后基于上述模型,从作业层面提出一种负载均衡的数据迁移算法。该算法能够规避数据访问热点,提高数据迁移效率。

关键词: 大数据;数据迁移;负载均衡

Abstract: In the big data era, data migration has become the basis for data centric mining analysis. In this paper, we analyze the requirements of data migration in big data storage systems and propose the corresponding migration model. We then analyze the factors affecting the migration performance. Then, using our proposed model, we design a load balanced migration algorithm from the aspect of job level, which can efficiently improve migration performance through avoiding data retrieving hotspots.

Keywords: big data; data migration; load balance

随着云计算、物联网等的发展,各行业产生的数据爆炸性增长,人类已经进入大数据时代。互联网数据中心(IDC)曾在2012年的报告中指出:全球数据量每两年翻一番,至2020年将增至40 ZB[1]。大数据时代,能否从海量数据中快速获取知识来指导业务发展很大程度上决定了企业的竞争力,如何将企业各业务沉淀的数据进行全面汇总并快速返回分析结果是大数据分析亟待解决的问题。

海量数据分析往往依赖于分布式环境,然而由于业务数据类型各异,数据迁移汇总将是首要任务。能否高效、稳定地将源数据迁移到目标存储系统很大程度上决定了数据的分析效率,因此一个高效的数据迁移方法亟待发现。现有迁移技术按照应用对象不同大致可分为两类:面向虚拟机和面向存储。面向虚拟机[2-4]主要解决整个虚拟系统在不同物理环境之间的迁移问题,是整个逻辑系统或计算容器的迁移,它更多关注于实时(服务不间断)和无缝(迁移之后切换对用户透明)两个特性,与文中关注的分布式存储系统间的数据迁移不同。面向存储主要解决数据在存储系统之间或同一存储系统的不同实例之间的迁移问题,系统之间的迁移重点考虑数据的存储格式、传输路径、网络状况等因素[5-8],实例之间的迁移重点考虑存储系统的存储形式、接口性能等一系列因素(如数据库[9-10]、VxVM[11]、独立冗余磁盘阵列[12-13](RAID)、云数据库[14-17])。

文中我们研究分布式系统间的数据迁移方法,属于面向存储的数据迁移,主要关注数据的迁移性能。虽然也有类似文献同样关注于迁移性能,但大部分文献仅考虑了集群单方面的均衡,而忽略了集群间迁移作业的负载均衡性。文中我们首先结合现有需求给出数据迁移的抽象模型并分析影响迁移性能的因素;其次,基于上述模型从作业层面提出一种负载均衡的数据迁移算法;最后,通过大量实验模拟分析了该算法的均衡效果。

1 数据迁移模型

我们将数据源构成的集合记作$DS=\{D_1, D_2, \cdots, D_M\}$,其中$M$代表数据源的个数,$|D_i|$为数据源的规模;将迁移作业构成的集合记作$JS=\{J_1, J_2, \cdots, J_N\}$,其中$N$代表迁移作业的个数;将每个迁移作业需要访问的数据源构成的集合记作$DS_j=\{D_1^j, D_2^j, \cdots, D_M^j\}$,其

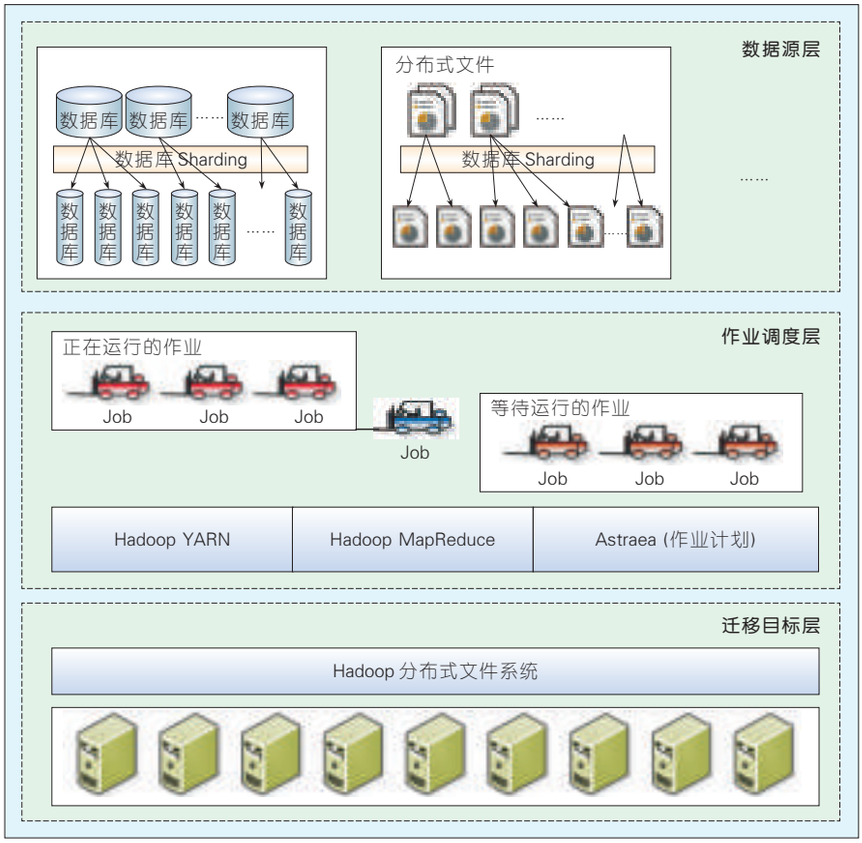
收稿时间:2016-01-23
网络出版时间:2016-02-22
项目基金:国家自然科学基金(61433008、61502090)

中 M_j 代表 J_j 需要访问的数据源个数，$DS_j \subseteq DS$，迁移作业的规模 $|J_j|=\sum|D_i|$；将 J_j 从 D_i 拉取数据的迁移任务记作 $P(D_i, J_j)$，所用时间记为 T_{ij}。

迁移策略好坏最直观的衡量方式是性能，而时间是性能的最好表征，我们首先给出迁移性能的评估函数，见式(1)。其中，CJS 是包含所有串行执行的作业集合中执行最慢的那些迁移作业，CDS_j 是 J_j 中包含最后结束的 $P(D_i, J_j)$ 的、串行运行的数据集集合，T_j 表示 J_j 的运行时间。

$$T=\sum_{J_j\in CJS} T_j,$$
$$T_j=\sum_{D_i\in CDS_j} T_{ij} \tag{1}$$

从性能评估函数中可以看出：影响迁移性能的因素有 CJS、CDS_j 和 T_{ij}，而并发度和负载均衡又是影响这三者的因素。迁移并发度越高，$|CJS|$ 和 $|CDS|$ 越小，同时迁移的数据量就越多，T_{ij} 越短；数据迁移的负载越均衡，任务相互等待的时间越短，从而降低迁移作业的运行时间。



▲图1 数据迁移系统架构

2 负载均衡策略

基于前面给出的数据迁移模型以及对性能影响因素的分析，我们设计了如图1所示的数据迁移系统，包括数据源层、作业调度层以及迁移目标层。

数据源层将不同类型的数据源水平扩展成分布式数据源；作业调度层使用 MapReduce 框架作为分布式程序的基础，使用 YARM 精确控制数据迁移使用的资源；迁移目标层使用分布式文件存储系统将目标集群虚拟为一个整体存储系统。

从上述架构中可以看出，提高迁移效率可从两方面着手：在数据源层和迁移目标层内部进行数据移动以充分利用闲置资源，从数据角度避免访问热点；在作业调度层设计实现负载均衡的迁移策略，从作业角度避免对同一块数据的热点访问。对于前者，一种称作间接迁移[18]的技术可用来实现集群内部的负载均衡，这种技术已被证明在闲置资源较多的环境中能够有效提高性能。此外，数据源层和迁移目标层的优化还可通过 Sharding 技术[19-20]直接保证数据的均衡分布，为高效率的数据迁移提供保证。因此，文章中我们着重研究第2种方法，通过合理调度作业尽可能错开访问同一数据源任务的执行时间。

2.1 问题定义

基于第1节给出的迁移模型，本节对迁移任务层的负载均衡问题进行形式化描述。

$$a_{ij}=\begin{cases}1, & D_i\in J_j\\ 0, & D_i\notin J_j\end{cases} \tag{2}$$

$$b_{jk}=\begin{cases}1, & J_j\in P_k\\ 0, & J_j\notin P_k\end{cases} \tag{3}$$

$$\forall i=1,2,\cdots,M;\forall k=1,2,\cdots,C$$
$$h_{ik}=\sum_{j=1}^{N} a_{ij}\times b_{jk}h_k=\max(h_{ik}) \tag{4}$$

对给定的 JS 和 DS，矩阵 $A=[a_{ij}]^{M\times N}$ 描述 D_i 和 J_j 的对应关系，矩阵 $B=[b_{jk}]^{N\times C}$ 描述 J_j 和 P_k 的对应关系，计算方法见式(2)和(3)；定义 P_k 的热度 h_k 为 P_k 中作业运行时需要访问的所有 D_i 中的热度最大值，D_i 的热度 h_{ik} 计算见式(4)。负载均衡的目标为寻找 JS 上的一个划分 $P\_JS=\{P_k\}$ 并且使其满足以下约束条件：$\forall k\in\{1,2,\cdots,C\},\sum_{J_j\in P_k}|J_j|\leqslant\Omega$，且 $\sum_1^C h_k$ 最小，其中 Ω 为目标集群的容量。

然而通过进一步分析我们发现：保证 P_JS 的热度总和最小并不能等价保证执行时间最短。这是因为热点访问会带来明显的性能下降。执行时间 T 与访问并发数 b 之间具有以下关联特征：T 随 b 的增大而增大；T 增长的速度随 b 的增大而增大。那么，必然存在一个临界点 b_0，使得当 $b>b_0$ 时并发执行的总时间大于串行

执行 b 次的总时间，也即 $T>b\times T_0$，其中 T_0 为 $b=1$ 时的执行时间。根据本文研究对象的特点，数据源的吞吐量是被严格限制的，又由于迁移作业的容器分配了足够的资源，因此 $b_0\leqslant 1$，这一结论在实际业务环境中得到了验证。

因此，P_JS 还需满足以下条件：P_k 包含的任意一个 J_j 涉及的 D_i 的热度为 1。由此可得 $h_{ik}=1$ 且 P_JS 的热度总和 $\sum_1^C h_k=C$。此外，鉴于迁移任务是高度并行的，且数据源又通过已有技术保证是均衡分布的，设单个迁移作业的执行时间为 T_0，则迁移作业的执行总时间 $T=C\times T_0$，其中 $C=|P\_JS|$。至此，优化目标最终等价转换为寻找 JS 的一个包含元素个数最小的划分 P_JS，该划分满足以下约束条件：

$$\begin{aligned}&\forall i=1,2,\cdots,M;j=1,2,\cdots,N;k=1,2,\cdots,C\\&\text{条件①}\ \sum_1^C b_{jk}=1\\&\text{条件②}\ \sum_{J_j\in P_k}|J_j|\leqslant\Omega\\&\text{条件③}\ h_{ik}=1\end{aligned}\tag{5}$$

P_JS 的最优解获取问题（记为 Q_0）可以归约到一个经典的 NP 完全（NPC）问题（均分问题），也即该问题至少是一个 NP-hard 问题，无法在可接受的时间范围内求解，因此文章提出 Astraea 近似求解算法。

2.2 近似求解

Q_0 的最优解需要满足式(5)中的 3 个条件，近似求解算法 Astraea 则满足了前 2 个条件，放宽了条件③的约束。为了量化解的近似性，文章基于数据源的访问热度与运行时间之间的关联关系给出一个近似评价函数 $Approxi(P\_JS)$，见式(6)。

$$\begin{aligned}&;\forall k=1,2,\cdots,C\\&h_{ik}=\sum_{j=1}^{N}a_{ij}\times b_{jk}h_k=\max(h_{ik})\\&Approxi(P\_JS)=-\sum_{k=1}^{C}h_k^{e-1}=-\sum_{k=1}^{C}\max(h_{ik})^{e-1}\end{aligned}\tag{6}$$

其中，h_{ik} 为 P_k 中作业涉及到的 D_i 的访问热度，h_k 为 P_k 的热度。这里，之所以选择（$e-1$）作为时间随热度的增速是因为它比线性增长快（与事实相符）又不至于过快而影响到解的准确性。

通过对问题 Q_0 的解空间进行分析，发现该问题并不适合采用绝大多数传统的启发式优化算法求解，因此我们提出了 Astraea 近似求解算法，它运用了贪婪算法的思想，利用每一步 Set Packing 结果的评价值对结果集进行过滤，并使用一个树形结构对每一次 Set Packing 的结果进行记录，不断迭代直至算法结束，再从树中取出最优结果。

Astraea 将 Q_0 看成 C 次 Set Packing 问题，每次求得矩阵 B 的一列。实际运行环境中，大部分情况下 $|DS|$ 比较大，但 $|JS|$ 比较小，因此我们采用简单的蛮力算法解决 Set Packing 问题。

运行过程中，Astraea 构造一个 B 的解空间搜索树，并对树中每个节点进行搜索。节点可以找到其父节点和子节点集合，通过序列号记录其所在层级（P_k 的下标），保存第 k 次 Set Packing 得到的 P_k（记作 $node.p$，即 $node.k$ 和 $node.p$ 构成了 P_k），记录其是否为最终节点及其评分。当前节点到根节点的路径上的所有 p 构成了当前解 X。

Astraea 的求解过程就是构造“B 的解空间搜索树”的迭代过程，包含以下 7 步：

（1）对于任意叶子节点，根据当前节点到根节点上的所有 p，构造当前调度计划 X；

（2）根据 X 和 $node.p$ 判断是否每个作业都被分配，若是则标明节点结束；否则，进入下一步；

（3）遍历所有可能的 P_k，并结合 X 判断其是否满足条件②，保留满足条件的 P_k，记为 $List\_P$。其中，P_k 遍历过程的复杂度已被充分降低，后文会详细解释，条件①也在这里满足；

（4）按公式(7)计算 P_k 评分，P_k 的热度越低越好，包含的作业个数越多越好；

$$Score(P_k)=-\max(h_{ik})+\sum_{j=1}^{N}X_{jk}\tag{7}$$

（5）采用加权贪婪策略，保留前 $\gamma\times\text{size}(List\_P)$ 个 $Score$ 最大的 P_k，其中 γ 是过滤参数；

（6）将剩余的 P_k 构造成新节点添加到当前节点的子节点中；

（7）返回第(1)步，对所有子节点进行迭代。

迭代过程结束时，Astraea 得到一棵 B 的解空间搜索树，其叶子节点全为结束状态。这时，任意叶子节点通往根节点的路径上的 P_k 都可以构造一个完整的调度计划 X。Astraea 对所有 X 进行评估，近似度最高的 X 就是所求解。

Astraea 的输入为 $A=[a_{ij}]^{M\times N}$、可行解的过滤比例 γ 以及目标集群能容纳的任务数量 $\mathbb{C}$；输出为最优的近似解 X。第（3）步的 P_k 遍历过程，是指在当前调度计划 X 的基础上，遍历下一个批次的作业集合 P_k 所有可能情况的过程。构造 $List\_P$ 实际上就是构造 P_k 所有可能的取值，并排除不满足条件的取值。一方面，条件①要求已分配的作业不能再次分配，因此 P_k 可以填入 1 的位置组成的集合可以通过对当前调度计划 X 的分析得到；另一方面，P_k 中 1 的总数就是本次调度的作业数量。由于资源限制，每次最多处理 $\mathbb{C}$ 个数据集，所以 Astraea 可以得到当前调度计划 X 下最少还需要运行 min 个批次的作业。min 个 P_k 中至少有一个包含的作业个数 $\leqslant|NS\_JS|/\min$，其中 NS_JS 为当前调度计划 X 下未被分配的作业集合。因为 B 各个列的顺序与最终运行时间无关，Astraea 可以将当前的作业批次包含的作业数量的上限设置为 $|NS\_JS|/\min$。

3 实验验证

针对第 2 节提出的 Astraea 近似求解算法，本节设计大量实验进行验证分析。

测试用例设计如表 1 所示，包含 3 个变量：作业个数 $|JS|$、数据源个数 $|DS|$ 以及算法中影响解的近似性的参

▼表 1 测试用例设计

$\|JS\|$	$\|DS\|=10$	…	$\|DS\|=50$	…	$\|DS\|=100$	γ
$\|JS\|=3$	10×3	…	50×3	…	100×3	γ=0.2、0.4、0.6、0.8、1.0
$\|JS\|=4$	10×4	…	50×4	…	100×4	
$\|JS\|=5$	10×5	…	50×5	…	100×5	
$\|JS\|=6$	10×6	…	50×6	…	100×6	
$\|JS\|=7$	10×7	…	50×7	…	100×7	
$\|JS\|=8$	10×8	…	50×8	…	100×8	

$|DS|$：数据源个数 $|JS|$：作业个数 γ：算法中影响解的近似性的参数

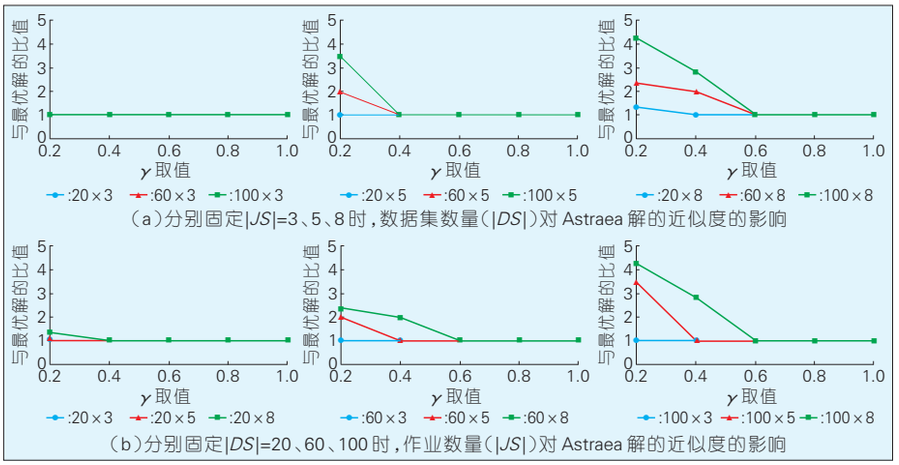
数 γ。其中，$|JS|$ 取值为 3、4、…、8；$|DS|$ 取值为 10、20、…、100；γ 取值为 0.2、0.4、0.6、0.8 和 1。$|JS|$、$|DS|$ 和 γ 均能够影响 Astraea 所求解的近似性以及算法性能，我们通过控制它们的变化来分析算法的有效性。

3.1 3个因素对解的近似性的影响

本实验主要分析 $|JS|$、$|DS|$ 和 γ 对 Astraea 所求解的近似性的影响。当 γ=1 时，需对整个解空间进行遍历，此时 Astraea 所求解即是最优解。文中我们采用近似解与最优解执行时间之间的比值来衡量解的近似度，该比值越小越近似。此外，鉴于实验数量较多，仅选取具有代表性的几组进行展示。

图 2 分别展示了 5 组 γ 设置下，$|DS|$ 和 $|JS|$ 对 Astraea 所求解的近似性的影响。在图 2（a），分别固定 $|JS|$ 的值为 3、5 和 8，研究 $|DS|$ 取值为 20、60 和 100 时对解的近似性的影响。从图中可以看出：当 $|JS|$ 较小时（取值为 3），无论 $|DS|$ 取值多少，Astraea 获取的解都是最优解（近似度为 1）；当 $|JS|$ 取值为 5 时，解的近似程度随着 $|DS|$ 的增大而变小，但最多都在 γ=0.4 的时候取得最优解；当 $|JS|$ 较大时（取值为 8），解的近似程度也随着 $|DS|$ 的增大而变小，但最多都在 γ=0.6 的时候取得最优解。在图 2（b），分别固定 $|DS|$ 的值为 20、60 和 100，研究 $|JS|$ 取值为 3、5 和 8 时对解的近似性的影响。从图中可以看出：当 $|DS|$ 较小时（取值为 20），解的近似程度随着 $|JS|$ 的增大而变小，但最多都在 γ=0.4 的时候取得最优解；当 $|DS|$ 取值为 60 和 100 时，解的近似程度也随着 $|JS|$ 的增大而变小，但最多都在 γ=0.6 的时候取得最优解。

综上所述，Astraea 算法获取的解与最优解之间的差距主要取决于 $|JS|$、$|DS|$ 和 γ3 个因素，当 $|JS|$ 和 $|DS|$ 较大时，要获取较优的近似解就需要设置更大的 γ。



（a）分别固定$|JS|$=3、5、8时，数据集数量（$|DS|$）对 Astraea 解的近似度的影响

（b）分别固定$|DS|$=20、60、100时，作业数量（$|JS|$）对 Astraea 解的近似度的影响

▲图 2 3个因素对 Astraea 解的近似性的影响

3.2 3个因素对算法性能的影响

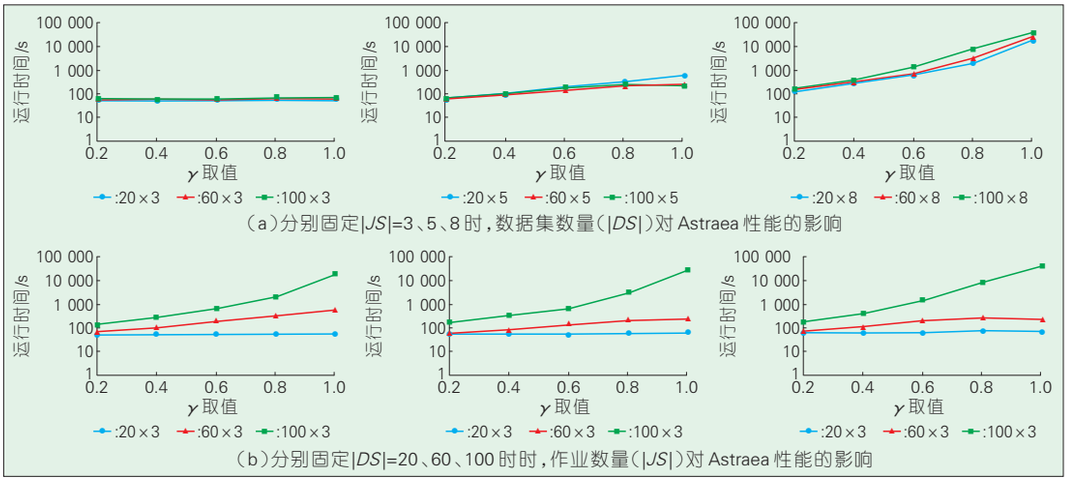
本实验主要分析 $|JS|$、$|DS|$ 和 γ 对 Astraea 算法性能的影响。衡量性能最常用的指标是时间，因此本节选取算法在不同实验设置下的执行时间来衡量 Astraea 的性能。同样，仅选取具有代表性的几组进行展示。

图 3 分别展示了 5 组 γ 设置下，$|DS|$ 和 $|JS|$ 对 Astraea 算法性能的影响，图中的纵坐标采取对数坐标轴。在图 3（a）中，分别固定 $|JS|$ 的值为 3、5 和 8，研究 $|DS|$ 取值为 20、60 和 100 时对 Astraea 算法性能的影响。从图中可以看出：算法的执行时间均随 $|DS|$ 以及 γ 的增加而增大，并且 $|JS|$ 越大，这种增长效果越明显。在图 3（b）中，分别固定 $|DS|$ 的值为 20、60 和 100，研究 $|JS|$ 取值为 3、5 和 8 时对 Astraea 算法性能的影响。从图中可以看出：算法的执行时间均随 $|JS|$ 以及 γ 的增加而增大，并且 $|DS|$ 越大，这种增长效果越明显。此处，γ 是影响解的近似性的重要因素，当对 Astraea 算法求解的近似性越高时，γ 的值就需要设置越大，进而执行的时间就越长。

综上所述，Astraea 算法的性能主要取决于 $|JS|$、$|DS|$ 和 γ3 个因素，$|JS|$ 和 $|DS|$ 较大时，获取较优的近似解就需要设置更大的 γ 值，需要更长的执行时间。

4 结束语

大数据时代，高效地从数据中发现知识已经成为企业重要的竞争力之一，而存储系统间的数据迁移则是知识发现的一个基础环节。我们首先给出数据迁移模型并对迁移性能影响因素进行分析，接着基于迁移模型从迁移任务层面提出负载均衡的优化方法，最后通过实验验证提出的优化方法的有效性。迁移任务层的优化方法主要通过调控迁移任务的执行顺序使得包含同一数据源的作



（a）分别固定|*JS*|=3、5、8时，数据集数量（|*DS*|）对Astraea性能的影响

（b）分别固定|*DS*|=20、60、100时时，作业数量（|*JS*|）对Astraea性能的影响

◀图3
3个因素对Astraea算法性能的影响

业尽可能避免在同一时间执行，尽可能避免迁移时的数据访问热点，从而提高迁移任务的并行性，改善迁移效率。作业调度的最优解获取问题可以证明是一个NP-hard问题，因此文中提出Astraea近似的求解算法来获取一个可接受范围内的问题解。

本中提出的负载均衡的数据迁移方法能够应用到很多系统中，例如淘宝商品搜索的索引数据存储处理系统，该系统的输入数据往往来自于多个存储系统，不同存储系统组成了一个拥有海量数据的数据源集群，需要该系统同步这些数据到Hadoop分布式文件系统（HDFS）、HBase等分布式存储系统。数据同步过程中，该系统一方面要保证数据的同步性能，同时还要尽可能降低对数据源系统查询性能的影响。文中提出的负载均衡的数据迁移方法能够成功地避开数据源的热点访问问题，从而实现系统的上述目标。

参考文献

[1] GANTZ J, REINSEL D. The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East [EB/OL]. [2012-03-22]. www.emc.com/leadership/digital-universe/index.htm
[2] AHMAD R W, GANI A, HAMID S, et al. A Survey on Virtual Machine Migration and Server Consolidation Frameworks for Cloud Data Centers[J]. Journal of Network and Computer Applications, 2015, 52: 11-25
[3] DERBEKO P, NATANZON A, EYAL A, et al. System and Method for Live Migration of a Virtual Machine with Dedicated Cache: US Patent 8,930,947[P], 2015
[4] FORSMAN M, GLAD A, LUNDERG L, et al. Algorithms for Automated Live Migration of Virtual Machines [J]. Journal of Systems and Software, 2015, 101: 110-126
[5] LI C. Transforming Relational Database into HBase: A Case Sudy [C]// 2010 IEEE International Conference on Software Engineering and Service Sciences (ICSESS). USA: IEEE, 2010: 683-687
[6] LIU C, FU Z, YANG Z, et al. General Research on Database Migration from RDBMS to Hbase[C]// 2015 International Symposium on Computers & Informatics. French: Atlantis Press, 2015: 124-237
[7] VUKOTIC A, FOX D, Partner J, et al. Neo4j in Action [M]. USA: Manning Publications, 2014
[8] SHIRAZI M N, KUAN H C, DOLATABADI H. Design Patterns to Enable Data Portability between Clouds Databases[J]. Computational Science and Its Applications (ICCSA), 2012: 117-120
[9] PANT P, THAKUR S. Data Migration Across the Clouds [J]. International Journal of Soft Computing and Engineering (IJSCE), 2013, 3 (2):14-21
[10] LONEY K. Oracle Database 10g: the Complete Reference [M]. USA: McGraw-Hill/Osborne, 2004
[11] MADELL T. Disk and File Management Tasks on HP-UX [M]. USA: Prentice-Hall, 1997
[12] ZHENG W, ZHANG G. FastScale: Accelerate RAID Scaling by Minimizing Data Migration [J]. FAST, 2011: 149-161
[13] KING A, CHIU D C. Efficient Fault-Tolerant Preservation of Data Integrity During Dynamic RAID Data Migration: US Patent 6, 530,004[P]. 2003
[14] Chodorow K. MongoDB: the definitive guide [M], 2nd Edition. USA: O'Reilly Media, 2013
[15] CHANG F, DEAN J, GHEMAWAT S, et al. Bigtable: A Distributed Storage System for Structured Data [J]. ACM Transactions on Computer Systems (TOCS), 2008, 26(2): 4
[16] GEORGE L. HBase: the Definitive Guide [M]. USA: O'Reilly Media, 2011
[17] ANDERSON JC, LEHNARDT J, SLATER N. CouchDB: the Definitive Guide [M]. USA: O'Reilly Media, 2010
[18] ANDERSON E, HALL J, HARTLINE J, et al. An Experimental Study of Data Migration Algorithms [M]. British: Springer, 2001
[19] BAGUI S, NGUYEN L T. Database Sharding: To Provide Fault Tolerance and Scalability of Big Data on the Cloud [J]. International Journal of Cloud Application Computing, 2015, 5(2):36-52. http://dx.doi.org/10.4018/IJCAC.2015040103
[20] LIU Y, WANG Y, JIN Y. Research on the improvement of MongoDB Auto-Sharding in Cloud Environment[C]// 2012 7th International Conference on Computer Science & Education (ICCSE). USA: IEEE, 2012: 851-854

作者简介

李甜甜，东北大学计算机科学与理论专业博士生；主要研究方向为大数据计算、高能效计算；发表论文10篇。

王智，东北大学软件学院软件工程专业硕士；主要研究方向为大数据计算、数据密集型计算。

宋杰，东北大学副教授；主要研究方向为大数据存储与管理、迭代计算、高能效计算；主持国家级项目3项，省部级项目4项；发表论文30余篇。

DOI:10.3969/j.issn.1009-6868.2016.02.008
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20160302.1631.004.html

基于概念网的媒体大数据分析和结构化描述方法

Topic Network-Based Media Big Data Analysis and Structural Description

张宝鹏/ZHANG Baopeng[1]
彭进业/PENG Jinye[2]
范建平/FAN Jianping[2]
(1. 北京交通大学 计算机与信息技术学院,北京 100044;
2. 西北大学 信息科学与技术学院,西安 710069)
(1.School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China;
2. School of Information and Technology, Northwest University, Xi'an 710069, China)

中图分类号:TP393 文献标志码:A 文章编号:1009-6868 (2016) 02-0033-005

摘要: 提出基于概念网的媒体大数据结构化描述和分析的技术框架,该框架可以针对不同的数据获取来源,通过层次式多角度概念描述模型融合数据的视觉特征、实例和概念关联的语义,并提出面向单一媒体和多媒体文档的跨媒体概念提取及基于结构的语义对齐方法,从而有效支持媒体大数据的语义关联分析及多领域的智能应用。

关键词: 概念网;媒体大数据分析;概念抽取;结构化描述;可视化

Abstract: In this paper, we propose that a topic network-based enabling technology framework for big media analysis and structural description. And it proposes a hierarchical concept description model with multiple perspectives for different sources data to integrating semantic of visual, instance and concept correlation. And cross-media concept extraction method for single media and multimedia document and their structure-based semantic alignment method are also proposed, which can efficiently support the big media analysis and smart application in many domain.

Key words: topic network; big media analysis; concept extraction; structural description; visualization

随着互联网的普及和迅速发展,各类在线社交网络(如Facebook、Twitter、新浪微博、腾讯网等)的飞速发展,网络数据资源越来越多样化,并呈爆炸式增长。这种大数据的势态引发了多行业、多领域的时代性变革。大数据思想的重要在于[1]:人们可以在很大程度上从对于因果关系的追求中解脱出来,转而将注意力放在相关关系的发现和使用上。目前,在互联网中,大量文本、图像、音频、视频等媒体大数据迅速增长,其中蕴含了很多人类社会活动的基本规律,公共卫生、商业乃至思维模式因此酝酿着重大的机会和挑战。基于大数据的研究逐渐成为各国政府重点发展的国家战略,及时、准确地获取并理解这些数据及其关系不仅仅可以为政府在社会生活、金融服务、医疗卫生等方面发现和处理民生问题,辅助政府决策,同时也为互联网经济的发展提供有效的客户和经济规律的知识辅助,提供商业智能决策支持。

收稿时间:2016-02-18
网络出版时间:2016-03-02

尽管媒体大数据成长迅速,应用广泛,但其数据量大、种类繁杂、价值密度低以及时时刻刻不断变化的特点,使得存储、统计、分类以及调用都非常困难[2],其价值远没有得到充分的利用和开发。而人工智能领域的一些理论和比较实用的方法,已经开始用于大数据分析方面,推动两个领域技术和应用融合的加速,但依然只是初期。目前谷歌、百度等通用的搜索引擎提供了基于文本描述的多媒体的检索机制,但对于大数据背景下的多种媒体数据来说,还缺乏准确文本描述,需要不同的算法分析、理解其内容的语义,实现相应的文本描述,从而为搜索引擎所用。另外,媒体数据间的异构性特点,使得当前单一媒体的搜索引擎无法有效支持大数据条件下异构媒体间的数据语义关联检索。因此,从媒体大数据智能应用的角度来看,其表示、理解及检索是重要的环节,而根据异构媒体间语义关系实现媒体大数据的智能的模式发现是解决这些问题的关键点。

1 媒体大数据分析和描述的关键问题

根据媒体大数据深度分析的目

标，以及其支撑媒体搜索引擎、媒体消费和关联分析的需求，尽管当前异构媒体的关联和分析技术有一些相关研究，但有些关键问题还没有得到解决，包括：

（1）媒体数据标注的不确定性及歧义性

除了大数据的4个V（Volumn、Variety、Velocity、Value）之外，为充分利用大数据蕴含的知识信息，一个重要的问题是解决媒体数据标注的不确定性、歧义性，这种不确定的标签数据包括：

- 粗糙标注，例如图片中对象是在图片层次上给出的，而忽略了其区域性的语义；
- 抽象标注，指标签只从高层语义角度给出，缺乏具体语义关联；
- 无关标注，指标注和图像语义并无关联；
- 噪声标注，指错误的标注。

这些标签数据将误导数据驱动的机器学习方法，从而导致数据训练分类器在性能和准确率上的退化。目前很多项目开展了图像智能标注的工作，旨在提高标签的准确率，包括传统概率的方法[3-4]、场景限制下的综合方法[5]、深度学习方法[6]及面向大规模的方法[7]等，但面向媒体大数据的复杂结构，复杂的语义及智能化的需求使得当前技术还远远不能满足其需要。

（2）媒体大数据结构化描述及其机器学习的算法

媒体大数据包含大量的语义概念，而且语义概念之间有千丝万缕的关系；同时对于不同主域的应用环境，不同的语义关系需要不同的结构化描述。目前传统多媒体语义描述模型主要包括两种：词袋模型，其源于自然语言理解，适合于视觉的相似匹配，但与语义并没有直接的对应关系；基于特征-语义的分类模型，源于机器学习，其主要参考的是人类语义感知设计，提取难度较大，准确率不高。由于传统多媒体语义提取采用多类学习的方法，其中用两类分类器合成的方法，训练检测复杂度较高，训练难度大，而传统的多任务学习和结构化支持向量机（SVM）学习方法，无法真正发掘出概念间相似性结构的信息。两种方法必须要解决的问题就是面向媒体大数据的泛化能力。目前，基于深度学习的多媒体语义提取方法得到了空前的关注，如文本检索会议（TREC）的视频事件检测提出的基于卷积神经网的深度学习算法，微软的音、视频索引服务（MAVIS）的语音识别系统，Google的深度学习模型等，都获得了很好的效果。但它们主要对音频、视频或文本单一模态进行分析，没有充分利用多模态信息间的相互协同关系。

（3）媒体大数据的关联性分析

媒体大数据分析首先需要研究异构媒体的统一表示[8]，相似度计算及语义关联的分析方法。传统的异构媒体采用基于子空间的映射技术，包括典型关联分析（CCA）方法、概率潜语义分析（PLSA）方法等。在相似度计算方面，主要的度量方法是基于图模型的相似度度量方法和基于学习的相似度度量方法[9]，但目前两者主要都是依赖共生性假设，即如果两个多媒体文档包含同一个媒体对象，则它们具有相同语义，也可以说是基于概念和概念的相似性或简单的物理依赖。跨媒体数据中的内在语义关系和结构（概念相关性网络）并没有给予充分的考虑，并且概念间关系复杂，因此并不适用于媒体大数据的深度分析，而主流的机器学习方法可能无法直接解决其复杂、大规模学习问题。

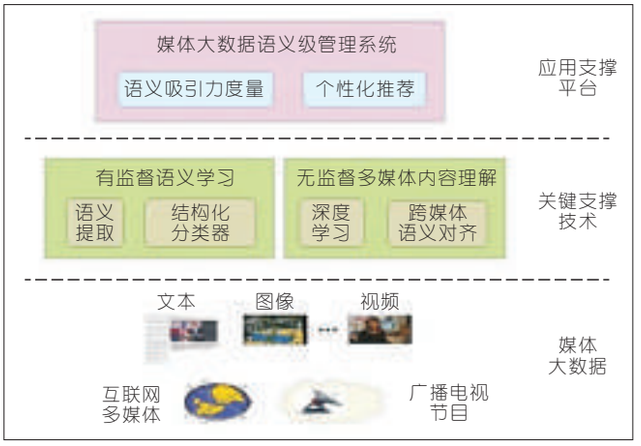
（4）媒体大数据的可视化与可视化分析

在媒体大数据的深度分析中，准确率和查全率是主要的分类器的评估标准，但由于学习分类器会过拟合，以及用于分类器训练和测试的样本是服从于同样的分布，因此评估标准会误导分类器的判定能力，也就是说不能显式地反映分类器和正确率和其辨识力。一种有效用于分类器评估的方法是可视化分类器的边界和类间的边缘，用户可以交互式地评估其正确率。因此，在机器学习过程中融合人的交互式操作，来改善分类器训练具有更高的应用价值。

2 媒体大数据关联分析的参考技术框架

针对目前在媒体大数据深度分析中所面临的问题，其未来发展的思路应该是基于内容语义的、全生命周期的支撑，因此我们提出了基于概念网的核心参考技术框架，如图1所示。针对媒体大数据处理的数据特点，我们需要考虑两种关键技术问题：有监督的媒体语义学习；无监督的多媒体内容理解。

目前媒体大数据的跨媒体概念的提取方法主要针对两种不同的媒体数据获取类型：一种是多媒体文档，主要是电视节目和网络媒体，包含图像、视频、音频和伴随文本描述等多种媒体形式。其关联关系隐含在多媒体文档中，重点解决的问题是多模态特征融合与跨模态关联分析的问题，而跨模态深度学习技术可以基于已有的图像、视频、音频及其对应的文本训练其语义概念检测模型，检测数据中的语义概念，并使用跨媒体语义对齐技术实现不同媒体语义概念的对齐。另外一种是单一视觉媒体，主要指监控录像和照片包含单一视频和图像，但没有伴随文本描述，需要进行多媒体数据中视觉语义概念的直接检测。其通过结合直接的标注数据和图像或视频的初级语义进行结构协同学习，提取语义概念并关联到对应的初级语义概念上，得到跨媒体语义。结构协同学习是基于概念相似性结构进行协同学习获得的分类模型的方法，其语义的统一于概念网络，有助于融合异构媒体的内容及关系特征，同时易于进行增量计算、测试修正及扩展。



◀图1
媒体大数据深度分析参考技术框架

该技术框架可以有效支持异构媒体大数据的可扩展应用，包括与当前搜索引擎的结合及面向不同应用领域的推荐系统等，如图2所示。

3 媒体大数据关联分析的关键技术

3.1 层次式多角度概念描述

多模态数据的语义提取并存储为语义库，需要一个能够描述所需语义信息，方便语义运算的语义模型作为数据语义存储和运算格式。由于相关的数据应用需要在高层语义、底层特征和实例样本等不同的层面处理海量数据及其语义，这要求语义描述模型要在统一的框架下存储所有这些信息。其难点在于：模型必须能够统一存储不同种类、不同层面差异巨大的媒体数据及其特征和语义。我们认为：应包括3层结构组成的描述模型，通过整合3个层次的关联（如图3所示），实现语义-实体-关系模型。其中位于语义层次的概念网应充分考虑大规模概念间的相关性，并提供能够对媒体大数据进行关联分析与结构化描述的新框架，从而用于指导训练大规模相关关联的分类器，并大幅度提高概念检测准确性。

3.2 基于多媒体文档的跨媒体概念提取

传统搜索引擎技术支持的图像-文本对应关系的获取具有很大的不确定性（如图4所示），而面向媒体大数据，语义对齐与关联分析可以利用视觉聚类、随机行走和概念语义网进行相关性重排以产生更准确的跨媒体语义对齐结果，并提取更准确的大规模跨媒体概念，同时利用视觉聚类可以进行跨媒体的语义消歧。这种跨媒体语义对齐方法可以为机器视觉研究提供大量的可靠标注的训练数据。

3.3 基于单一媒体的跨媒体概念提取

结构协同学习利用多个语义概念之间的相似性关系信息设计检测语义概念的分类器，通过充分利用这种相似性关系信息（该关系可以用结构表示），提升大类数媒体数据分类的性能和准确率。面向大媒体数据的大规模结构协同学习框架（如图5所示），首先将语义概念相似性网络进行分割以形成相似概念组，这一过程将最相似的语义概念分到同一组，而将差异较大的概念分到不同组，实现将语义概念中的相似结构表示为概念的分组情况。针对不同分类任务可选择不同分类算法和不同特征表示，有助于大量减少训练复杂度。同时，利用多层视觉树[10]来管理大量

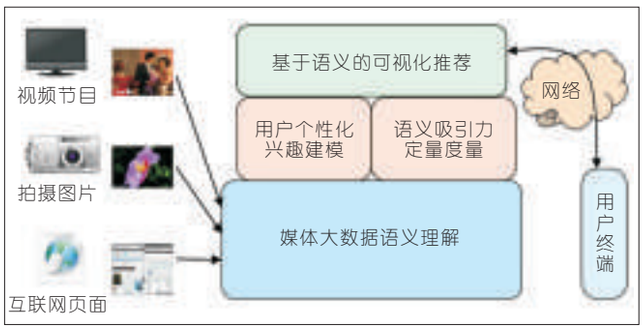
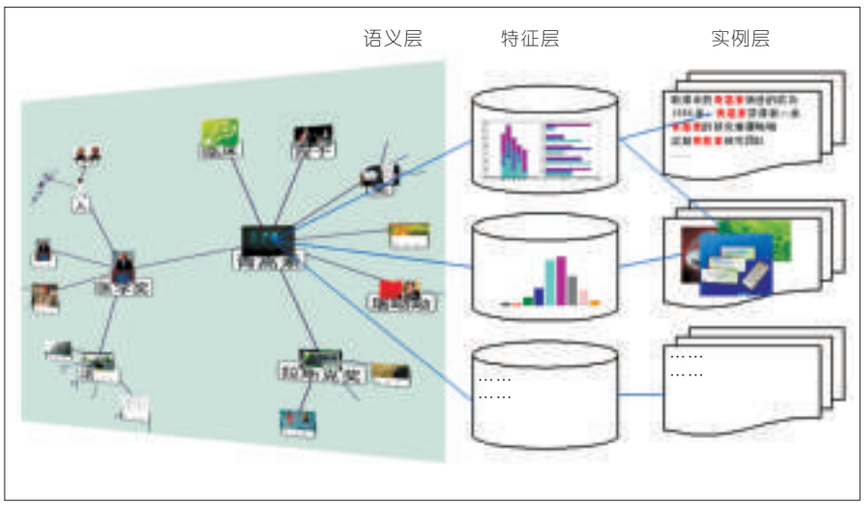
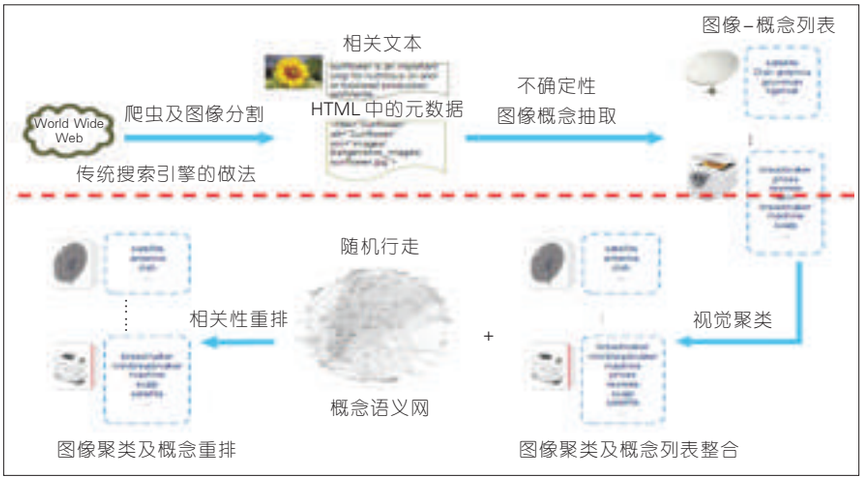


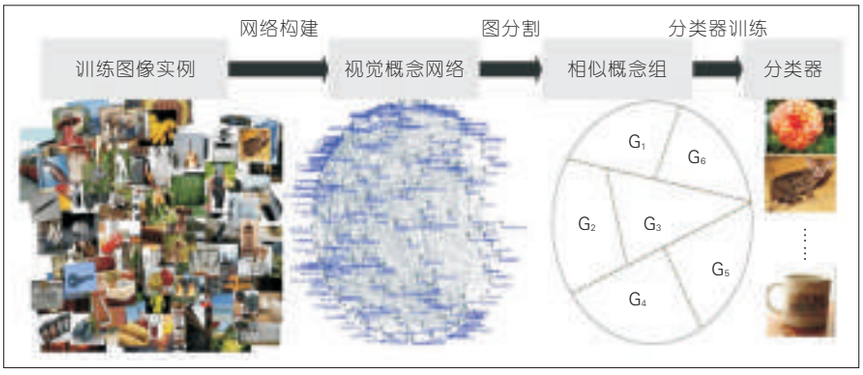
图2▶
基于语义分析的媒体大数据消费系统



▲图3 层次式多角度概念描述模型



▲图4 传统搜索引擎与跨媒体概念提取的技术思路对比



▲图5 大规模结构协同学习框架

分类器，实现快速提取大规模跨媒体概念。这其中一个重要的问题是：训练图像实例如何提取语义。目前，深度学习可以得到很好的特征提取及分类效果[11]，而更为有效的方法是将各种传统视觉特征作为先验知识模型加入到深度学习算法的训练当中。

3.4 跨媒体语义对齐

当前很多算法都是针对不同媒体的数据构建语义结构化模型。这些模型有的较好地关联到了高层语义，但因为缺乏相关的文本数据标注而无法关联到高层语义，只能通过深度学习算法获得大量抽象的语义概念及其关系。为了统一管理和挖掘媒体大数据，必须实现抽象的语义概念与具体的语义概念（语言）对齐。描述媒体的结构化语义信息的模型一般为图结构，我们需要研究语义对齐方法实现多个语义结构的对齐，提高语义信息的准确度。其难点在于：需要精确估计两个图的部分节点之间的相似度关系，但语义概念在不同媒体数据中的具体表现差异巨大，难以直接估算相似度。

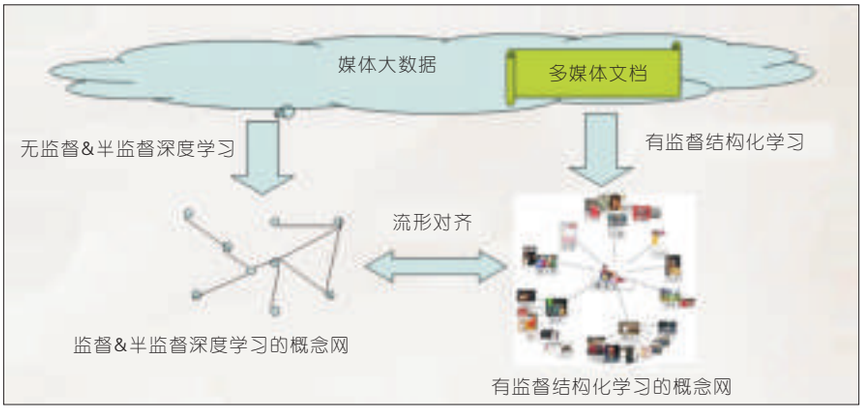
为了充分利用所有语义信息获得最高对齐精度，可以使用流形对齐方法，该方法对实现两个语义结构的对齐是个较好的选择。如图6所示：流形对齐算法综合计算两个语义空间的语义概念的相似度和语义概念的内在关联结构，从而实现两个语义空间的对齐，这比仅仅依据语义概念之间的各种相似性的方法具有更高性能。

为简化描述，下面我们把抽象的语义概念称为未标记实体，具体的语义概念称为语义实体。在使用流形对齐算法过程中，我们需要计算部分未标记实体和语义实体之间的相似度。我们提出了两种相似度计算方法：结构协同分类获得的语义概念包含对齐的图像视频数据，这些数据上也包括深度学习算法提取的未标记实体，通过统计未标记实体在某个语义实体对应的图像、视频数据中出现的概率，即可计算出未标记实体和语义实体的相似度；用结构协同学习获得的语义概念检测模型检测所有图像和视频关键帧，可以获得描述其语义的一个高维矢量，一对视觉实例间的语义相似度可以定义为其语义矢量之间的近似程度，未标记实体和语义实体的语义相似度则可基于两者对应的图像和视觉结构间的相似度进行计算。为了既可以体现跨媒体数据对齐的信息又利用结构协同学习的结果，有效的方法是将以上两种相似度加权组合获得未标记实体和语义实体之间的融合相似度，融合相似度可以用作流形对齐的节点间对应信息，从而实现大规模媒体数据的知识的融合和一致性处理。

3.5 基于概念网的媒体大数据关联性分析及其可视化

如果把语义概念之间的相似性用一个加权图表示，语义概念之间的相似性结构信息将形成一个语义概念相似性网络。这个网络的结构对应于语义概念之间的相似性结构，因此可以用于结构化学习指导分类器结构设计。构造语义概念相似性网络首先需要度量语义概念之间的视觉相似度，而语义概念之间的视觉相似度基于样本之间的相似度计算，样本之间的相似度要基于底层视觉特征计算。为了消除概念之间的相似性非常小却仍然有连接的现象，我们采用自底向上层次式聚类算法裁剪全连接的语义网络。

该方法可以有效表示主域的数据相关性。例如，用于描述新闻热点



▲图6 深度学习与结构化学习相结合的对齐方法

间的相关性的新闻概念网，如图7所示。这种概念网提供了一个对大规模媒体概念进行关联分析和结构化描述的新框架结构，同时也便于面向不同消费系统进行扩展应用。

4 结束语

当前多领域、跨领域的网络媒体数据呈大规模增长的态势，而异构媒体的智能关联、知识表示是合理利用数据并为行业提供智能化服务的核心研究问题，因此，突破媒体大数据的基于内容的结构化描述、关联与深度分析，形成媒体内容语义的全生命周期的技术框架，以支持个性化搜索与智能推荐、跨终端的多媒体内容呈现等关键技术的发展，对建立面向用户的智能服务平台，推进知识获取及推广，改善用户体验具有非常重要的意义。

参考文献
[1] 维克托.迈尔.舍恩伯格, 肯尼思.库克耶. 大数据时代: 生活、工作与思维的大变革[M]. 盛扬燕, 周涛, 译. 杭州: 浙江人民出版社, 2013
[2] ZHU W, CUI P, WANG Z. Multimedia Big Data Computing [J]. IEEE Multimedia, 2015, 22(3): 96–105. DOI: 10.1109/MMUL.2015.66
[3] FENG S L, MANMATHA R, LAVRENKO V. Multiple Bernoulli Relevance Models for Image and Video Annotation[C]//Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. USA: IEEE, 2004(2): II–1002–II–1009. DOI: 10.1109/CVPR.2004.1315274
[4] BARNARD K, DUYGULU P, FORSYTH D, et al. Matching Words and Pictures [J]. J Mach Learn Res, 2013(3): 1107–1135
[5] LI J L, SOCHER R, LI F F. Towards Total Scene Understanding: Classification, Annotation and Segmentation in an Automatic Framework[C]//Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. USA: IEEE, 2009: 2036–2043
[6] FARABET C, COUPRIE C, NAJMAN L, et al. Learning Hierarchical Features for Scene Labeling[C]//IEEE Transactions on Pattern Analysis and Machine Intelligence. USA: IEEE, 2012, 35(8): 1915–1929
[7] WESTON J, BENGIO S, USUNIER N. Large Scale Image Annotation: Learning to Rank with Joint Word–Image Embeddings [J]. Machine Learning, 2010, 81 (1):21–35
[8] ZHU S C. Statistical Modeling and Conceptualization of Visual Patterns [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(6): 691–712. DOI: 10.1109/TPAMI.2003.1201820
[9] 唐杰, 陈文光, 面向大社交数据的深度分析与挖掘[J]. 科学通报, 2015, 60(5): 509–519
[10] ZHOU N, FAN J. Jointly Learning Visually Correlated Dictionaries for Large–scale Visual Recognition Applications [J]. IEEE Transaction. on Pattern Analysis and MachineIntelligence, 2014, 36(4):715–730
[11] DEAN J, CORRADO G S, MONGA R, et al. Large Scale Distributed Deep Networks[C]// Proceedings of the 26th Annual Conference on Neural Information Processing Systems. Canada, 2012: 1223–1231

◀图7
基于概念网的媒体大数据的关联性表示

作者简介

张宝鹏，北京交通大学计算机与信息技术学院讲师；主要研究方向为多媒体理解及检索、大数据管理及挖掘等；已主持及参与完成国家级、省部级项目20余项；已发表论文20余篇。

彭进业，西北大学信息科学与技术学院教授、博士生导师，教育部新世纪优秀人才支持计划获得者，陕西省图象图形学会常务理事；主要研究方向为信号处理与信息安全；发表论文60余篇。

范建平，西北大学信息科学与技术学院特聘教授、博士生导师；主要研究方向为统计机器学习、大规模视觉识别、社交图像/视频分析、大规模图像/视频检索等；已发表论文160余篇。

DOI：10.3969/j.issn.1009-6868.2016.02.009
网络出版地址：http://www.cnki.net/kcms/detail/34.1228.TN.20160222.1019.004.html

BC-BSP：一个基于BSP的高可扩展并行迭代图处理系统

BC-BSP: A BSP-Based High Scalable Parallel Iterative Graph Processing System

刘恩孚/LIU Enfu
冷芳玲/LENG Fangling
鲍玉斌/BAO Yubin
（东北大学 计算机科学与工程学院，辽宁 沈阳 110819）
(School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China)

中图分类号：TP393 文献标志码：A 文章编号：1009-6868 (2016) 02-0038-006

摘要： 提出了一个基于整体同步并行计算（BSP）模型的、具有磁盘暂存功能的大规模图处理系统——BC-BSP。该系统通过提供应用程序接口（API）实现系统配置和有关策略的可扩展性，通过优化的图数据磁盘存储实现了数据处理规模的高可扩展性以及高性能的容错方案，并且可以处理普通数据集的聚类和分类等需要迭代计算的数据挖掘算法。通过实验验证了该系统的可扩展性，其在真实数据集上性能优于Giraph1.0.0，在模拟数据集上稍逊于Giraph的内存版。

关键词： BSP；大规模图处理；迭代计算；磁盘缓存

Abstract: We describe a bulk synchronous parallel (BSP)-based parallel iterative processing system for graph data with disk caching assist. This system is called BC-BSP. The system can achieve the scalability of system configuration and policy by providing APIs, high scalability of the data scale processed, and high performance of fault-tolerant scheme by disk storage optimization to graph data. It can also execute some data mining algorithms with iterative processing, such as clustering and classification on non-graph data sets. The experimental results show that the scalability and performance of the proposed system are better than that of Giraph1.0.0 on the real data set, but it is lightly poorer than the memory version of Giraph.

Keywords: BSP; large-scale graph processing; iterative computing; disk cache

图是计算机科学中最常用的一类抽象数据结构，更具有一般性的表示能力。现实世界中的许多应用场景都可以很自然地使用图结构表示。例如，交通运输网络、社交网络中的资源对象之间的关系以及生物信息网络等。在大数据时代，需要分析的图规模越来越大。以互联网和社交网络为例，随着互联网的深入使用和Web 2.0技术的推动，网页数量增长迅猛，据中国互联网络信息中心（CNNIC）统计：截止2014年12月中国网页规模达到1 899亿个，年增长率26.6%；而基于互联网的社交网络更是如此，如全球最大的社交网络Facebook，2014年7月已有约22亿用户，其中月活跃用户数13亿人。在中国，如QQ空间、微博、开心网等，发展也异常迅猛。因此，实际应用中图的顶点可达10亿，而边就会更多，对应的数据文件会更大。对如此大规模图数据的存储和分析处理的时间和空间开销远远超出了传统集中式图数据处理的承受能力。因此，对大规模图的有效处理成为了一个新的挑战。

MapReduce计算模型可以实现对大规模（图）数据的处理，并且具有很好的容错性和可扩展性。但是由于图数据分析（如网页的PageRank[1]计算、最短路径计算、聚类分析）都需要多次迭代才能完成。每次迭代需要一个或多个开销较大的MapReduce作业完成。为解决迭代计算的时间性能问题，谷歌公司开发了基于整体同步并行计算（BSP）模型的Pregel[2]系统，之后Apache的两个开源项目Hama和Giraph也开展了基于BSP的迭代计算系统的开发。它们都是在内存中做数据处理，因此能够处理的图的规模有限。文中，我们设计开发了基于BSP模型的、能够处理大规模

收稿时间：2016-01-08
网络出版时间：2016-02-22
基金项目：国家自然科学基金重点项目（61433008）；国家自然科学基金（61173028）；教育部-中国移动科研基金（MCM20122051）

（图）数据的并行迭代计算系统——BC-BSP。该系统主要特色在于：（1）实现了具有磁盘辅助的基于BSP的大规模图数据并行迭代处理系统，该系统在内存受限的情况下具有很好的数据处理能力，即在可用的节点规模和内存配置的情况下，可以处理的数据规模较大；（2）系统多方面考虑负载均衡，在充分考虑数据本地化的前提下考虑了各个节点的负载均衡问题，并且结点的负载均衡优先于数据本地化。我们做了大量的实验，比较了基于BSP的大规模图处理系统的性能和扩展性。

1 BSP模型和相关工作

BSP是一种“块”同步模型[3]，即通过消息传递机制，实现块内异步并行，块间显式同步。一个基于BSP的计算系统是由具有处理机和存储器的多个自治的计算服务器组成的集群，并且这个集群采用主/从结构。主节点用于协调整个集群，包括接收用户的作业提交、作业调度、故障监控等功能，从节点（也称为工作节点）用于存储和处理数据。

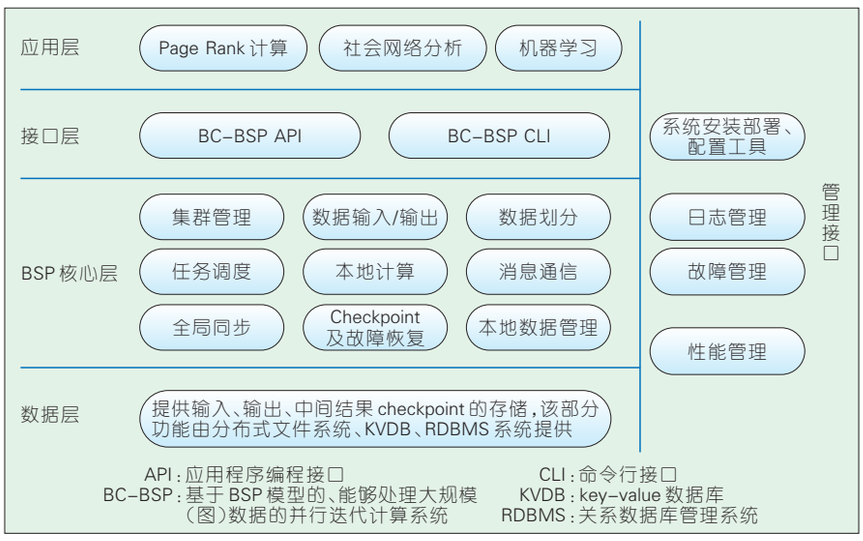
谷歌公司开发的基于BSP模型的分布式图计算框架Pregel主要是为了处理大规模图数据，如网页的PageRank计算、最短路径等。Pregel假设处理的数据都在内存中，因此在一定的节点规模下，它能够处理的数据规模是有限制的。基于Pregel的思想，许多基于BSP的大规模图处理系统被开发出来。例如，Apache推出了基于Java的开源项目Hama[4]，它是一个纯粹的基于BSP的用于大规模科学计算（如矩阵计算、图和网络算法）的计算框架，同样它的早期版本没有考虑磁盘辅助的问题，而是假设所有数据全部位于内存中，最新的版本也在添加磁盘辅助功能，但是很不完善；而Apache的另一个开源项目Giraph，是建立在Hadoop基础之上的Pregel的开源实现[5]，可以认为它是MapReduce模型和BSP模型的结合体，即它利用MapReduce作业的Map任务实现了基于BSP模型的迭代计算，而不需要Reduce任务，整个图处理过程只需要启动一次MapRedcue作业，但是一旦出现故障，整个作业需要重新启动；GraphLab是卡内基梅隆大学提出的面向大规模数据挖掘和图计算的分布式内存计算框架[6]。更多的基于BSP模型的类Pregel的大规模数据分布式并行处理系统和框架请见文献[7]。

2 BC-BSP概述

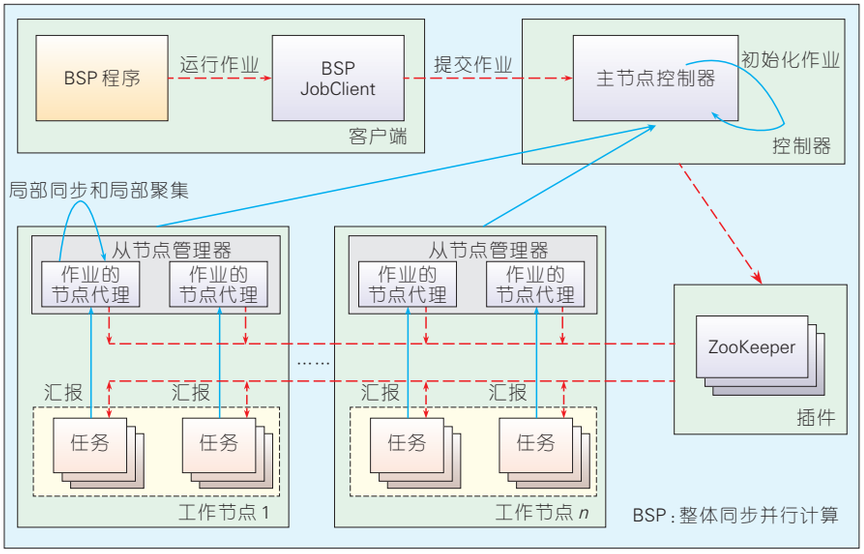
图1给出了BC-BSP系统的整体结构，主要包括BSP核心层、管理接口层和接口层。BC-BSP实现了对Hadoop分布式文件系统（HDFS）、HBase、MySQL等底层存储系统的支持，包括数据的输入和输出。BC-BSP系统内部核心层主要包括客户端作业提交和数据划分，主节点端的作业调度和集群监控，从节点端的本地计算处理、全局同步、消息通信和容错控制；接口层主要包括应用编程接口（API）和命令行接口（CLI）；管理接口层主要包括集群管理、系统自动化安装部署、日志管理、性能管理和故障管理等工具。

从系统实现的角度，BC-BSP系统是一个主从式结构，主要分为客户端、主控节点、工作节点、任务模块、全局同步模块。图2给出了BC-BSP的运行控制机制以及系统中客户端、主控节点、工作节点、任务模块、全局同步模块之间的协作关系。

在BC-BSP系统中，客户端主要根据用户指定的输入路径进行数据分片，调整分区数目，检查作业运行的可行性，向主控节点申请作业并将作业打包提交给BSP主控节点，当作业开始运行后，负责及时反馈作业运行状态；主控节点端管理集群工作节点的注册、心跳信息和状态信息收集等，并作为容错控制的控制中心，提供各种状态查询接口，并以作业为单位，负责作业的初始化、调度和同步控制等；工作节点端主要负责工作节点本地的任务管理和局部同步控制以及局部聚集计算等；任务模块端是任务运行的实体，主要负责执行用户的业务处理逻辑和数据输入输出处理等；全局同步负责同一作业的所有任务在各个超步之间的全局同步工作，超步路障同步由主节点端、工作节点端及任务模块端共同完成，在同步过程中，可以完成聚集计算，系统中的同步主要通过第三方组件Zookeeper实现；消息通信主要在每一个超步的本地计算执行过程中，负责异步地发送和接收消息，并将接收的



▲图1 BC-BSP系统的功能组件



▲图2 BC-BSP的运行控制机制

消息暂存到本地的接收消息队列中，当内存空间不足时，支持磁盘辅助存储，这里主要是通过远程过程调用协议（RPC）机制实现消息传递；容错控制模块负责容错备份、故障检测和故障恢复等功能，以写检查点机制作为主要的容错方案，支持手动备份和自动周期备份功能；管理工具主要通过Web界面或命令行的方式为用户提供可视化的系统管理和监控功能；接口模块主要为用户提供本地计算、消息发送/接收等的应用编程接口，以及为用户提供启动和关闭系统服务、作业提交等命令行接口。

3 BC-BSP提供的API

系统给用户提供了与作业建立相关的API，用于编写针对图处理或科学计算的处理程序。另外，系统还提供了用于系统功能扩展的接口。下面我们简单介绍这些接口。

（1）消息管理接口负责消息的发送/接收功能，在每一个超步的本地计算执行过程中，并行地发送和接收消息，并将接收的消息缓存到本地的接收消息队列中，在发送消息队列达到一定规模的时候，执行Combine操作，然后再将消息发送给目的节点。

（2）分区数据管理接口负责在进行图数据处理之前将待处理的图数据按照一定的原则划分给各个任务。本系统实现了基于Hash的划分方法和基于Hash的均衡划分方法。

（3）图顶点上下文接口负责在任务处理的一个超步中，处理每个图顶点时获取正在处理的图顶点的相关属性信息和方法。

（4）消息合并接口在图处理过程中，通常以顶点为中心进行处理，该接口为了减少在网络上传送的消息数量，在发送端对发给同一个顶点的消息进行合并。

（5）聚集计算接口许多图处理/机器学习算法中需要聚集计算，实现该接口可进行超步间的聚集值计算。

（6）数据输入输出接口包括输入接口和输出接口，用于实现将数据从指定数据存储系统中读入和写出。

4 BC-BSP系统的实现

本节介绍BC-BSP系统在实现上的一些主要策略和细节，主要包括图数据的表示、主节点控制器、从节点管理器、本地计算与消息通信、图数据划分以及故障恢复等的实现。

4.1 主节点控制器

主控节点是整个BC-BSP集群的控制中心，负责管理所有的工作节点，监控整个集群的工作状态，接收各工作节点的心跳信息并加以处理，完成整个作业的全局同步控制，并提供统一的信息查询接口和作业提交接口。当集群启动后，主控节点接收各工作节点的注册信息，形成统一的集群资源信息，在运行过程中通过心跳信息不断更新集群资源信息，例如，可用任务槽数量。当客户端请求提交作业时，将其放入作业等待队列，作业调度器按照优先级加先入先出队列（FIFO）的策略调度作业；而完成一个作业的具体任务的调度则是按照负载均衡和数据本地化的原则。因为本系统中一个作业的所有任务需要同时运行，所以系统中的任务调度是采用由BSP主节点控制器根据上述原则将任务依次不断下推给各个节点。

4.2 从节点管理器

工作节点是硬件上的计算单元，系统启动后，BC-BSP集群的各个节点上启动一个从节点管理器（WM）进程，负责完成具体的任务启动和消息通信。每个工作节点启动后，都首先向主控节点注册，使自己成为BC-BSP集群中的一员；之后，工作节点定期向主控节点发送心跳信息，汇报自己的状态；当有新任务下达时，工作节点根据新任务的指令，到HDFS上读取作业信息并下载到本地文件系统；然后创建任务控制对象和对应的执行进程，接着运行任务。WM为在本节点上运行的每个作业建立一个WorkerAgent对象，用于收集该作业在本节点上的各个任务的心跳信息、工作状态信息等。这样全局同步采用两级同步方式，即一个工作节点上的属于同一个作业的各个任务在本节点上实现局部同步，然后再以节点为单位向Zookeeper注册实现全局同步。工作节点以作业为单位维护在本节点上运行的隶属于同一个作业的所有任务，进行统一管理，完成

各种局部操作，例如本地聚集计算。

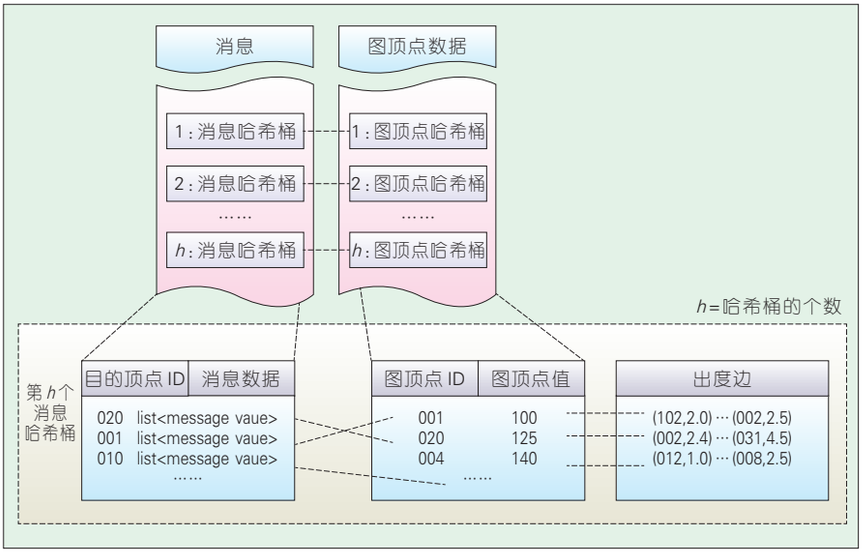
4.3 磁盘辅助的本地计算和消息通信

任务模块是逻辑上的计算处理单元，称为一个任务。BSP主节点控制器中的任务调度器根据负载均衡和数据本地化原则将任务分配到具体的工作节点上，由WM创建该任务模块进程。任务模块启动后，首先完成数据加载，将需要处理的数据分片从存储介质上按照指定的输入格式读入本地，并进行数据划分。计算过程中会定期地向WM的WorkerAgent对象发送心跳信息，报告任务的状态等信息。

在Pregel系统以及基于它思想的各种实现中，都假设集群的处理节点足够，使得待处理的数据等够完全存放在内存中。但是实际情况却不是这样的：一方面对于一个给定的待处理数据集，用户很难确定需要几个工作节点才能使得各个任务处理的数据能够存放在内存中；另一方面，当集群规模有限时，也希望能够处理相对较大规模的数据。对于系统中发送（或接收）的消息也是如此。鉴于以上原因，本系统中使用了磁盘临时存储数据和消息（也称之为磁盘暂存），以便能够处理较大规模的数据。

对于消息数据，将消息数据的内存占用比例按照用户指定的静态划分参数确定，系统运行时处理各种类型的消息时内存的使用单独分配处理，每种类型的消息内存占用都具有一个独立的阈值控制。

对于任务处理的数据而言，在迭代计算过程中常驻磁盘。对于出边表不变的计算情况，即不增加也不删除边的情形，将顶点的出边表与顶点的其他在计算中变化的部分，例如顶点的值或标签等信息，分开存放，但是同样使用记录的ID的Hash映射进行划分，如图3所示。将图数据分开处理的好处在于：每次迭代结束只需将本次迭代过程中变化的数据写回本地磁盘文件即可，不变的静态部分不需要写回磁盘，同时也为容错控制提供了方便。



▲图3 磁盘暂存的Hash索引示意

4.4 图的顶点类

一个图是由顶点集合和边集合构成，因此有顶点类和边类。本系统中使用邻接表的方式组织图数据。这样一个顶点类中除了顶点本身的属性之外，还有与之相连的出边信息，同时提供了对顶点和边进行操作的方法（见图4）。

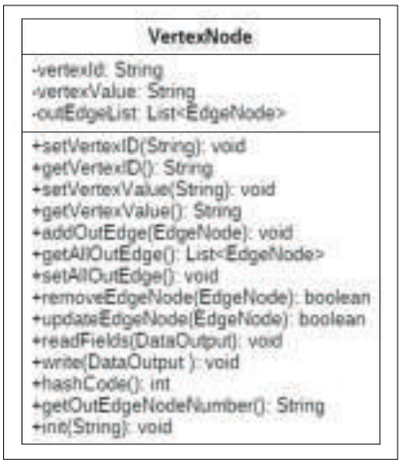
4.5 数据划分

数据划分是BSP计算与MapReduce计算不同的地方。前者需要在迭代计算中能够定位消息发送的目的地在哪里。因此，数据划分是将各个任务与之绑定的数据分片的数据从数据源读入，然后利用一定的数据划分原则，例如Hash划分，将图数据分配给某个任务，以便形成超步迭代计算时的数据分区。

一个作业的各个数据分区大小是否均匀直接影响系统的负载均衡，但是Hash函数很难保证各个分区大小的均衡。为此，我们采用了多Hash桶合并的划分方法，以实现数据的近似均衡划分。合并的原则可以是各个桶中的对象数据尽可能均衡，还可以考虑数据的本地性。本系统目前是按照各个桶中数据对象近似均衡为主兼顾本地性的原则进行合并。

4.6 容错机制

容错是本分布式处理系统必须考虑的问题。BC-BSP系统中考虑两类故障：一类是任务故障，例如任务进程宕掉；另一类是工作节点故障，例如一个Worker出现网络断开故障或者磁盘读写故障。系统中各个任务通过心跳机制向所在Worker的WM汇报自己的工作状态，而各个工作节点也是通过心跳机制定期向BSP主节点控制器汇报工作状态。



▲图4 图顶点类结构

本模块包括写检查点、故障检测和故障诊断以及故障恢复等功能。写检查点是定期或者人工控制方式将某个时刻的作业运行快照保存到分布式文件系统，如HDFS；故障检测与故障诊断是完成故障信息的收集与故障类型的判断，不同阶段的不同类型的故障，采用不同的恢复机制。BC-BSP系统实现了基本的基于检查点的故障恢复策略和面向磁盘驻留的多级容错处理策略。

所谓的面向磁盘驻留的多级容错处理策略，是利用了本系统的磁盘辅助机制的一些措施，即将图数据分成不变的常驻磁盘的静态部分（例如图顶点的出边表）和每次迭代计算几乎都会变化的需要写回磁盘的动态部分。因此在进行系统快照备份时，实现增量备份，即对图数据的静态部分只需要备份一次即可，而每次迭代计算时只需增量地备份动态变化部分。当然每次备份时需要备份本次收到的所有消息。

5 BC-BSP系统应用示例

本节讨论使用本系统进行图数据的PageRank计算和多维数值型数据集的k-means聚类分析的示例。在k-means示例中，可以论证BC-BSP系统也可以有效地处理非图数据的数据挖掘算法。

5.1 PageRank

使用BC-BSP系统实现PageRank计算中，首先将一个顶点的PageRank值按照一定的规则（如各个出边顶点平分），通过发送消息的方式发送给出边顶点，同时获得来自入边顶点的消息；之后按照PageRank算法的PageRank值计算公式，将一个顶点的消息值（即PageRank贡献值）累加，计算当前顶点新的PageRank值。因此用户可以提供combine方法实现消息发送前的合并，再基于顶点的新PageRank值重复上面的计算过程，直到满足收敛条件结束计算，并按预先的用户配置输出计算结果。

5.2 多维数值型数据集的k-means聚类

使用BC-BSP系统对多维数值型数据集进行k-means聚类，不需要进行顶点间的消息传递，但是需要利用聚集器计算新的聚类中心，可以通过各个聚簇的所有数据点的累计和与累计数据点计数两种聚集器实现。因此，用户可以实现BC-BSP系统提供的staffStartup接口，完成整个聚类作业开始之前的聚类中心初始化工作，例如读取预先设定好的存储在分布式文件中的初始聚类中心，利用系统提供的聚集器接口实现聚簇内数据点累计和与累计计数计算新的聚类中心，这样就需要每个任务计算自己任务内的局部累计和与累计计数，然后在BSP主节点控制器计算各个类的总累计和以及总类内数据点数，在新的超步开始时计算聚集中心。

当k-means聚类的k值较小（例如几十个）时，这种利用聚集器的方法是可行的。然而，实验中我们发现：当k值上百或更大时，就会出现异常。这是因为需要向Zookeeper写的内容太多。因为系统框架中聚集器的实现利用了Zookeeper，所以在实现k-means聚类时，使用了分布式文件暂存各个任务的局部聚集结果。在执行超步计算前读取这些临时文件，计算新的聚类中心，可以解决k值较大时引起的异常问题。

6 BC-BSP系统的实验

选择同样基于BSP模型的Hama[4]和Giraph[5]作为参照比较系统，并且使用它们的API实现了PageRank算法。实验软硬件配置是：30个工作节点，一个作为控制节点，29个用作存储和计算的工作节点，Java虚拟机（JVM）的内存设置为2 GB。每个节点的配置如下：Intel Core i3-2100双核中央处理器（CPU）、8 GB双倍速率同步动态随机存储器（DDR）3内存、500 G/7200 RPM磁盘，安装了Red Hat Centos 6.0操作系统、JDK1.6.0-30、Hadoop-0.20.2和Zookeeper-3.3.2。统计了运行PageRank 10次迭代的运行时间开销。

测试数据采用不同规模的真实数据和人工合成数据；人工合成数据集由数据生成器生成。实验中我们选择了定点规模不同的5个真实数据集[8]，它们的统计信息见表1。

6.1 真实数据集测试结果

利用表1中描述的5个真实数据集，在Giraph1.0.0的内存版（Giraph 1.0.0_MEM）和磁盘版（Giraph 1.0.0_HDD）、Hama 0.6.4和BC-BSP 2.0系统上分别运行了PageRank算法，得到了图5所示的结果。

由图5展示的结果可得出：BC-BSP2.0的性能优于另外3个对比系统，总体上比Giraph1.0.0的内存版的性能好。

6.2 虚拟数据集测试结果

通过测试虚拟数据集进行系统可扩展性的对比，我们可知：数据从1 000万顶点至11 000万顶点，主要用于测试系统的可扩展性和计算性能，平均出度规模为11.5。

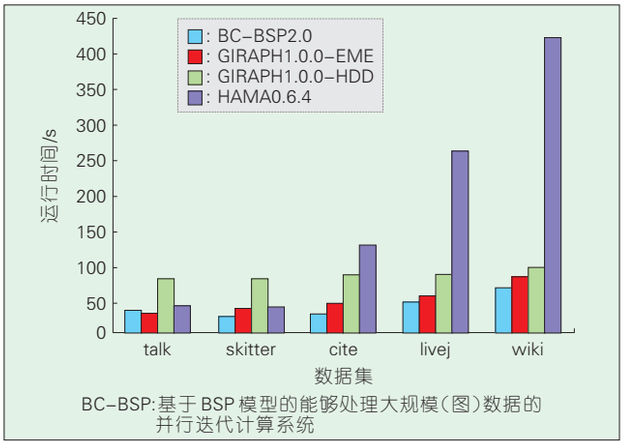
由图6展示的结果可得出：图数据的顶点从1 000万到11 000万，BC-BSP 2.0在数据吞吐量以及在相同数据集的处理效率上都要优于HAMA-0.6.4，并优于GIRAPH-1.0.0_HDD，效率略低于GIRAPH-1.0.0_MEM，但可扩展性更好。

7 结束语

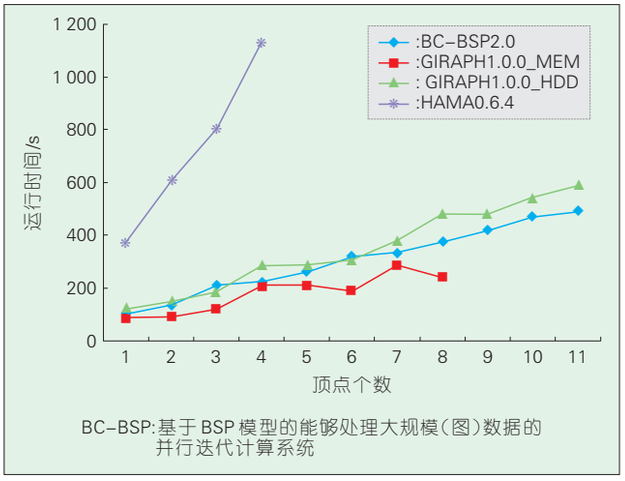
文章描述了在Java语言环境下基于BSP模型实现的用于大规模图数据迭代处理的系统BC-BSP。该系统在Pregel思想的基础上，实现了它的基本功能，同时增加了若干优化策略，包括增加了均衡的数据划分策略，使得每个任务处理的节点数量尽可能相近，图数据处理和消息通信过

▼表1 测试用真实数据集信息

名称	顶点数目	边数目	平均出度	磁盘文件规模/ MB
1: skitter	1、696、414	12、791、712	7.54	99.29
2: talk	2、393、819	7、415、229	3.098	67.42
3: cite-pr	6、009、555	22、528、503	3.7488	215
4: livej-pr	4、847、571	73、841、344	15.233	553.56
5: wiki-pr	5、716、808	135、877、199	23.768	1 047.1



◀图5 真实数据集PageRank测试结果



◀图6 模拟数据集PageRank测试结果

程中的磁盘暂存使得在计算节点及其内存资源有限的情况下可以处理较大的数据，具有更高的可扩展性。

尽管在系统开发过程中已经做了大量的优化工作，但是系统还有可优化的地方。例如，关于图数据结构的优化与改进：(1)目前不论是图顶点对象还是边对象都采用字符串方式存储，可以改成支持泛型的实现；(2)系统利用写检查点机制实现了故障恢复，但是对于故障类型的捕获和诊断还有待进一步加强；(3)在系统实现中发现Java环境对内存的开销巨大，因此对数据结构的设计以及使用需要仔细地斟酌。

致谢

本研究得到东北大学于戈教授和谷峪副教授的帮助，以及中国移动(苏州)研发中心钱岭博士的支持，谨致谢意！

本系统开发工作是由东北大学计算机软件所王志刚博士研究生以及许多已经毕业的研究生共同完成，对他们谨致谢意！

参考文献

[1] SERGEY B, LARRY P. The Anatomy of a Large-Scale Hypertextual Web Search Engine [J]. Computer Networks and ISDN Systems, 1998, 30(98): 1-7
[2] GUERON M, LLIA R, MARGULIS G. Pregel: A System for Large-Scale Graph Processing [J]. American Journal of Emergency Medicine, 2009, 18(18):135-146
[3] VALIANT L G. Bulk-Synchrony: A Bridging Model for Parallel Computation [J]. Communications of the ACM, 1990, 33(8): 103-111
[4] Welcome to Hama Project [EB/OL].[2011-07-13] . http://incubator.apache.org/hama/
[5] AVERY C, CHRISTAN K. Giraph: Large-Scalegraph Processing Infrastructure on Hadoop [EB/OL]. [2011-06-29]. Hadoop Summit 2011, https://github.com/aching/Giraph
[6] LOW Y, BICKSON D, GONZALEZ J, GUESTRIN C, et al. Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud [J]. Proceedings of the VLDB Endowment, 2012, 5(8): 716-727
[7] MAMOU H. An Experimental Comparison of Pregel-Like Graph Processing Systems [C]// Proceedings of Vldb Endowment. USA: ACM 2014: 7(12):1047-1058
[8] Using the Stanford Large Network Dataset Collection [EB/OL], https://snap.stanford.edu/data/index.html

作者简介

刘恩孚，东北大学计算机科学与工程学院在读硕士研究生；主要研究方向为数据挖掘、图数据管理。

冷芳玲，东北大学计算机科学与工程学院讲师，中国计算机学会高级成员；主要研究方向为数据仓库和联机分析处理等。

鲍玉斌，东北大学计算机科学与工程学院教授，中国计算机学会高级成员；主要研究方向为联机分析处理、云计算、图数据管理等。

DOI:10.3969/j.issn.1009-6868.2016.02.010
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20160219.1622.002.html

大数据安全必须面对的攻击假设矩阵

Matrix of Attack Hypothesis Faced in Big Data Security

潘柱廷/PAN Zhuting
(启明星辰公司,北京 100193)
(Venustech Group Inc., Beijing 100193, China)

中图分类号:TP393 文献标志码:A 文章编号:1009-6868 (2016) 02-0044-005

摘要: 认为大数据安全研究需要从大数据攻击研究出发。大数据攻击不仅仅需要考虑针对大数据系统的攻击,更要综合考虑针对系统、过程、数据、语义等多层次的攻击,还要综合看待攻击面和背后的攻击目标。为了更好地理解大数据攻击,提出了意识信息物理系统(MCPs)这样的多层次复杂系统的认识模型,并根据MCPs的多层次,建立起[攻击面x攻击目标]的攻击假设矩阵。对于攻击假设矩阵中每个格子的研究,可以帮助人们构建更有效的保障体系。

关键词: 大数据安全;攻击假设矩阵;攻击面;攻击目标;MCPs

Abstract: Big data attacks are the foundation of big data security. For data attacks, we need to consider attacks for the systems with large data, and for the system, process data and semantic level of attack, and also need a comprehensive look at the target of the attack surface. In order to better understand big data attacks, a new model of the multi-level complex system—Mentality-Cyber-Physical system/space (MCPs) is proposed. Based on this multi-level of MCPs, a attack hypothesis matrix by [attack surface, attack target] is built up. Research on every grid of the matrix will lead to more effective assurance solutions.

Key words: big data security; matrix of attack hypothesis; attack surface; attack target; MCPs

1 大数据安全的范畴

大数据作为一个新的技术模式和学科分支,已经开始对网络信息安全产生深刻的影响,这种影响既要循着安全本身的固有规律,也会带着数据自身以前不被重视的新特性。

1.1 安全的本质性结构

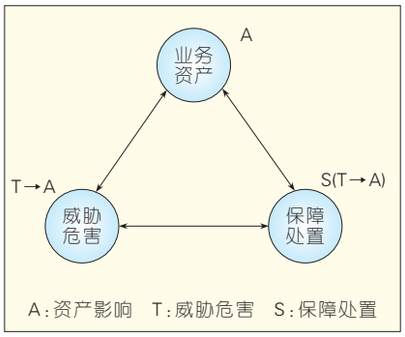
在IT领域的各个分支中,网络信息安全区别于其他分支的根本不同,就是安全永远是一个三要素互相交织、博弈的课题。这3个要素为:业务和资产、威胁和危害、保障和处置,如图1所示。

安全的独特性在于:有难以控制、难以意料的"威胁和危害"一方,自然就有了特有的"保障和处置"这一方,两者和业务资产一起形成了一个三方博弈关系。

所有的安全问题,都要就这3方面分别阐述清楚才谈得到思考的完备性,而大数据安全这个话题也不例外。

1.2 大数据安全的方向

大数据安全的如下3个方向,是大数据方法和技术作用于安全三要素所演绎出来的方向。



▲图1 安全的3个要素

(1)大数据作用于业务和资产,即大数据的主流应用。这必然会面临新的针对大数据的攻击和威胁,进而对大数据的保护要对抗这种针对大数据的攻击。

(2)大数据作用于威胁和危害,即大数据攻击和副作用。如果是主动和故意的举措,那就是大数据攻击;如果是被动的,就是大数据产生的副作用,比如大数据技术对于公民隐私保护的破坏。

(3)大数据作用于保障和处置,即安全大数据应用。就是在对抗各类安全威胁的时候,运用大数据技术进行分析和检测,特别是无特征检测、异常检测、态势分析等方面。

文章论述的重点是大数据安全的第1个方向。研究对大数据的保护必须先研究针对大数据的攻击,如果没有真正研究、设计、实现并测度大数据的攻击,那么之前所设计的所

收稿时间:2016-01-10
网络出版时间:2016-02-19

谓大数据防护就都是臆想，只有真实的攻击才能够验证保护和防护的有效性。

2 数据本质和特质

研究针对大数据的攻击，我们必须搞清楚针对大数据的攻击的对象——大数据对象。

2.1 大数据的7V特性

在描述大数据问题时，我们常说其有7个V的特性[1]，具体如下：

1V（Volume），即海量的数据规模。这体现了大数据问题在数据量上的海量。

2V（Velocity），即快速数据流转和动态数据体系。这代表了时间轴上的大数据，除了对于分析快速及时的要求之外，还体现海量数据可能来自于时间轴的长度延展（存储）和颗粒度的细化（频度）；时间的相关性也是数据间相关性的一大类，比如视频和音频数据就是“顺序时间”的典型结构。

3V（Vast），即数据来自广大无边的空间。每个数据都来自于一个空间的位置，可能是物理空间（现实世界），也可能是网络空间，空间的相关性也是数据间相关性的一大类，也是一大类典型结构。

4V（Variety），即多样的数据类型。大数据，比所谓的“量大”更重要的一个特性就是“高维”。特别是当数据样本的数量难以满足对于高维问题求解的基本要求时，大数据更倾向于回避精确解的求解，而满足于有价值的近似解。这种不追求精确解的特性，让大数据及其系统具有了一定的鲁棒性基础，增加了攻击难度。

5V（Veracity），即数据的真实和准确性更难判断。数据有好坏问题，而这个好坏问题在大数据中会更加极端地被放大，更泛地表达这个话题就是数据的“质”，即数据质量[2]的相关问题。

6V（Value），即大数据的低价值密度。对于大数据的攻击，背后必然要针对其价值进行。

7V（Visualization），即大数据可视化的重要性。大数据的价值需要展现，如果能够破坏和斩断价值链，也是重要的攻击成果。

在这7个V中：第1个V，表达的是大数据外在表现的“大”量；第2～4个V是从时间、空间和多样性这3个方面说明大数据的“大”；第5～7个V阐述的是大数据的价值流转，即从数据本身的客观质量，到有立场的价值认识和价值挖掘，最后到价值的展示和利用。

2.2 攻击大数据的常规理解

在传统的网络信息安全领域中（这里指融合大数据特有特征的思考之前），对于攻防的认知主要集中于系统方面：漏洞是系统的漏洞，越权是对于系统访问控制的突破，拒绝服务攻击是对网络系统的拥塞，伪装是对于系统访问者身份的假冒等；安全方法也主要都围绕系统的防护而展开。当然，这个系统是包括了节点式的系统（如主机操作系统）、结构化的网络系统。

在探讨攻击大数据的时候，我们首先想到的就是如何攻击大数据系统，而由于大数据目前的主要应用模式就是分析和决策支持，其系统的对外暴露面非常少，因此至今还没有关于重要的大数据系统遭遇渗透性攻击的报道。能够见诸报道的大数据系统出现的问题和故障，常常是由于电力故障等物理性故障导致的可用性事故，而这些所谓的问题并没有体现出大数据的独特性。

对于大数据系统的、具有针对性的攻击假设，需要针对大数据系统的分布式特色发起攻击。对于大数据的特色攻击还没有太多的研究，可能有两个原因：第一，大数据系统还在快速地演化和发展；第二，攻击研究者要搭建一个接近真实的大数据系统，其成本比较高，技术门槛也较高。但是，由于大数据系统的高价值聚集，这样的攻击早晚会到来。

2.3 MCPs结构

网络空间已经成为了大家非常熟悉的一个词，它不仅仅指网络相关的IT系统，更被人们理解为一个空间，在这个空间中主要体现了Cyber实体及其“活动”。

这里所说的活动指Cyber过程，主要体现为操作和流。数据实体对应的是数据流，应用系统对应业务流和服务关系，节点系统对应了计算操作和存储承载，网络系统对应了网络流和连接关系，而物理实体则是对前述Cyber实体的承载。Cyber实体就如同生物体的解剖关系，而Cyber过程如同生物体的生理关系。当前流行概念中的云计算、移动互联网等等都是Cyber自身形态的多样化、高能化和效益化。

信息物理系统（CPS）强调了Cyber与物理空间的关系：可将Cyber与物理空间的关系简化为控制与感知的关系。CPS类似的模型将物理世界和网络空间关联起来了，其关联的根本媒介其实是数据。当前流行概念中的物联网、工业控制、智能生活等等都是将Cyber空间与物理世界更加紧密地关联起来。

网络空间安全领域被分为两大领域：一个是从技术上说的网络安全，比如加解密、攻防渗透、系统加固等；另一个是从系统的内容上说的信息安全，比如舆情态势感知、社交网络策动攻击等。这两方面现在是单独研究和治理的，交集不大。

现在，随着大数据的方法和技术日益得到重视，数据也越来越受到人们的重视。大数据又是一个应用驱动、价值驱动的领域。当数据与数据的语义总是密切关联在一起的时候，我们就发现人的意识空间和Cyber空间的关系变得密切起来。多人的共同意识空间就是群体社交意识。

数据将人的意识空间（包括群体

意识）、Cyber 空间、物理世界 3 方面链接在一起，形成了一个整体意识信息物理系统（MCPs），如图 2 所示。

当我们有了 MCPs 这样的整体认识，在考虑安全问题（特别是大数据安全问题）的时候，就要考虑 MCPs 模式下的攻击。

3 MCPs 的攻击假设矩阵

3.1 攻击面和攻击目标

攻击面是指攻击者的着手之处和着手模式；攻击目标是指攻击者希望被攻击体系中的某个部分或环节出现重大偏差。我们将攻击面和攻击目标分开来定义，是因为两者并非总是同一的。

3.2 MCPs 的 3x3 攻击假设矩阵

在系统攻击中，攻击面和攻击目标可能不同。这种攻击面与攻击目标的错位，可能出现在 MCPs 的 3 个方面，由意识空间、网络空间、物理空间（现实世界）的交叉攻击假设，形成如图 3 所示的 3x3 攻击假设矩阵。

3.3 MCPs 的 14x14 攻击假设矩阵

要对 MCPs 攻击假设矩阵进行更具体的研究，就需要将 MCPs 分解成更细致的环节。我们可以将 MCPs 简单分解为 14 个方面，其编码如下：

- Mm：动机
- Mv：价值
- Ms：语义
- Cd：数据和数据流
- Cm：元数据和纯数据
- Ca：应用和业务流
- Cc：计算节点
- Cs：存储节点
- Cn：网络和网络流
- Cp：Cyber 物理实体
- Pc：控制器
- Ps：传感器
- PS：空间关系
- PT：时间关系

将 MSPs 的这 14 个方面组成一个矩阵，矩阵不同的行代表不同的攻击面，矩阵不同的列代表不同的攻击目标。如表 1 所示。

表 1 中，蓝色区域就是从传统的系统攻击视角看到的攻击假设，攻击面可能是网络系统、存储节点、计算主机、应用系统，而最终最受影响的攻击目标也在这其中。

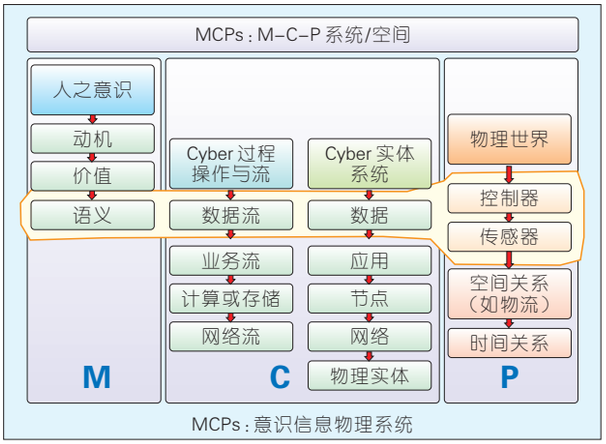


图 2 数据贯穿 MCPs 模型示意

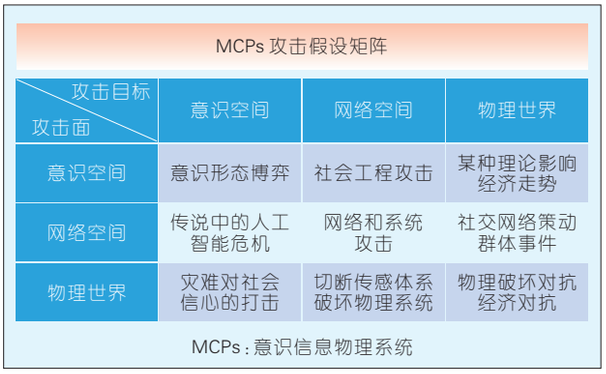


图 3 3x3 攻击假设矩阵及其示例

▼表 1 MCPs 攻击假设矩阵（14x14）

	Mm	Mv	Ms	Cd	Cm	Ca	Cc	Cs	Cn	Cp	Pc	Ps	PS	PT
Mm						4.6								
Mv														
Ms														
Cd			4.5											
Cm											4.7			
Ca														
Cc							4.1							
Cs			4.5	4.4										
4.5														
Cn					4.7		4.2		4.3		4.7			
Cp					4.8									
Pc														
Ps														
PS								4.8						
PT														

Ca：应用和业务流　Cm：元数据和纯数据　Cs：存储节点　Mv：价值　PS：空间关系
Cc：计算节点　Cn：网络和网络流　Mm：动机　Pc：控制器　PT：时间关系
Cd：数据和数据流　Cp：Cyber 物理实体　Ms：语义　Ps：传感器

注：表格中的数字是文章的章节编号，对应了这个矩阵节点的解释和示例。

数据Cd和元数据Cm，将MCPs三大空间连接起来。表1中的红色部分表示数据作为攻击面和攻击目标会横纵贯穿整个攻击假设矩阵，而且数据会成为MCPs3个空间的桥梁，产生交叉攻击的可能性。

表1反映了大数据和数据视角引入后，给我们带来的更加全面统合的攻击假设视界。

4 MCPs攻击假设矩阵的归类分析

MCPs的14x14攻击假设矩阵中的每一个格子，都是一种攻击模式，甚至是一个攻击链的索引。归类后的每个格子，都具有一定的攻击模式共性；格子之间则应当有攻击模式的差异化特点。

做出这样的分类研究，可以让我们把攻击研究得更细致，比如可以将计算节点（Cs）进一步细分为PC节点、移动节点、工控节点等。这样还可提醒我们注意那些原先忽视的空白部分，是否有攻击可能存在。只有对于攻击的全面和细致的研究，才能让我们对于防御和对抗的问题上有更多的把握。

4.1 [Cc，Cc]攻击

[Cc，Cc]攻击是最常被关注到的攻击模式，比如，对于操作系统漏洞的挖掘和利用，进而对于系统进行破坏和渗透，其攻击面和受影响目标都是系统。

4.2 [Cn，Cc]攻击

与节点攻击不同，[Cn，Cc]对网络的攻击是对结构的攻击。另外，一般把对于网络设备的攻击归类为对于网络的攻击。

分布式拒绝服务攻击（DDoS）是一个典型例子，其通过对于网络结构性的攻击，并通过占领海量节点而构成了一个攻击网络结构，将流量导入给一个目标系统使其瘫痪。这是典型的攻击网络最终危害节点系统。

网络劫持窃听也是一个典型例子，攻击点在网络的路上。通过窃听下来的明文或者密文进行分析，达到渗透相关系统的目的。

从Cn到Cc的影响传递很直接，因为计算节点都自然连接在网络中，所以对网络的攻击会很快传递给计算节点。

4.3 [Cn，Cn]攻击

内容分发网络（CDN）是当前一个非常重要的网络服务。如果能够利用CDN服务构建一个CDN指向的环，当向这个环投入足够多的流量时，环就会利用CDN机制在网络中形成一种自激振荡式的流量洪流，可能导致网络风暴的发生[3]。这是典型的攻击网络而危害网络，是一种结构性破坏。

4.4 [Cs，Cd]攻击

[Cs，Cd]攻击存储设备，甚至渗透并控制存储设备，自然会对于存储设备上存储的数据产生直接的危害。

4.5 [Cs，Ms]:=[Cs，Cd][Cd，Ms]攻击

如果[Cs，Ms]:=[Cs，Cd][Cd，Ms]攻击存储并对存储进行破坏，或者对于存储的攻击和篡改被较快发现，那么这种影响就难于进一步传递到其他攻击假设矩阵格子。

如果对于存储的攻击充分考虑了存储的数据结构，在篡改中保持其基本的数据结构，不让这样的篡改被轻易发现；同时，篡改的数据又能够借助应用系统的分析对于分析结果进行有效影响，那么就能够将这样的攻击传递到语义层，进而影响人的意识空间，影响人的决策。

而如果要在大数据存储环境下达到[Cs，Ms]，就要顺应大数据存储的系统模式和其存储数据的数据结构，做到篡改不易被发现；还要了解大数据存储的数据将如何被分析和应用，让篡改的数据能够污染到大数据分析的结果。

大数据相关的攻击假设，能够让我们反思如何对抗这种攻击。如果将存储的系统模式和数据结构进行一定的随机化（仿效操作系统中的地址随机化思想），那么大量篡改数据就很容易被发现；如果将大数据分析的容错能力（容忍不良质量数据）提高，那么就迫使要污染大数据分析结果必须篡改更多的数据。让“篡改不易被发现”与“大量篡改数据才能产生语义污染”形成矛盾，进而将攻击的效果阻隔在Cyber空间中，不让其有效影响人的意识空间。

4.6 [Mm，Ca]攻击

2016年初的一个突发案例[4]：一则谣言，经过微信朋友圈的扩散，震动了大半个互联网金融圈。

2016年1月10日下午，回顾2015年微信数据的“我和微信的故事”在朋友圈突然被刷屏，正当大家玩得非常欢快时，一个哑弹突然向社群中抛来。当晚，有用户在自己的朋友圈中称：该链接“千万不要进，（黑客）马上把支付宝的钱转出去，已经有人被盗”，还称加载该链接时“很慢，已经在盗取资料。”朋友圈截图被疯转，引发用户集体不安。很多人吓得把支付宝的银行卡都解除绑定，支付宝里的余额全部打回银行卡，还一一提醒朋友“如果我这个号向你借钱，千万别理。”

在1月11日的一个报告中，张小龙说起10日晚的事称：“我和微信的故事”的链接没想到被分享出去，这样带来了3个问题。第1个问题：访问太高，基本挂掉了；第2个问题，有人造谣说，打开链接支付宝的钱被偷了，这个时候，链接也确实因访问量太高打不开了；第3个问题，百万级用户开始解绑银行卡了，结果服务器也快挂了，银行卡也解绑不了了。

这是一个典型案例：一个谣言（在人的群体意识空间），影响了人们的操作行动，进而让一个应用系统崩溃（网络空间中）。

对于这类有意的攻击和无意的危害，有些防范措施可能在意识空间，有些防范措施就要在网络空间，甚至需要二者结合。比如，针对这类[Mm，Ca]风暴，就可以考虑建立态势感知监控和相关性研判，当然这就要将舆情监控和系统风暴监控进行相关性联动分析。这在以前是没有的，从这个事件让我们意识到这种联动分析的必要性。

4.7 [Cn，Pc]:=[Cn，Cm][Cm，Pc]攻击

光大证券乌龙指事件[5]给我们展示了一种可能性。

2013年8月16日11点05分上证指数出现大幅拉升，大盘一分钟内涨超5%，最高涨幅5.62%，指数最高报2 198.85点，盘中逼近2 200点。11点44分上交所称系统运行正常，下午2点，光大证券公告称策略投资部门自营业务在使用其独立的套利系统时出现问题。有媒体将此次事件称为“光大证券乌龙指事件”。

一个系统网络的故障，可能导致应用系统和大量数据的错误，这些可能是数据Cd或者元数据Cm。如果一些金融衍生品应用系统是通过对数据监测和分析自动进行买卖操作的，就可能因为被监测数据的错误导致错误的买卖决策（控制现实世界的控制器行动）；而如果错误的买卖决策又继续导致被监测数据的错误效果放大，可能就在市场中产生连锁效应，甚至有引发或诱发证券市场的瞬间大波动甚至股灾。

这种危害的可能性，对于社会的危害是极为严峻的。

4.8 [Cp，Cm]攻击和[PS，Cs]攻击

在密码破译和密钥分析领域，有一种方法：通过对密码芯片外部的热量分布进行跟踪分析，从而达到破解和猜测密钥的目的。这是典型的[Cp，Cm]攻击，用对系统物理实体的分析来攻击到数据层。

对于系统的运行状态进行分析，我们也可以通过对系统的能量消耗进行分析。这是典型的[PS，Cs]攻击，用物理世界的物理测度PS来分析系统Cs。

上述两个分析（攻击）都需要对物理世界测度并产生相当大量的数据，才能完成对于Cyber内部的分析。换句话说，这个分析过程需要大数据技术和分析方法的支持。

5 结束语

MCPs攻击假设矩阵还有很多空白之处需要填补和研判。可以想象：当我们把各个格子的攻击都能够假想并模拟出来，那么对于有效的安全保障和问题防范就会产生不可估量的支撑。

大数据安全绝对不能停留在系统层面，一定要在MCPs的统合视角下研究整个攻击假设矩阵。特别是跨MCP三大空间的攻击，将是非常值得研究的，很多“黑天鹅”式的攻击必然由此而产生。

参考文献

[1] 潘柱廷. 安全大数据的7个V——大数据基础问题与信息安全的交叉探究[J].中国信息安全, 2013(9):74-77
[2] MCGILVRAY D. 数据质量工程实践[M]. 刁兴春, 曹建军, 张健美, 译. 北京: 电子工业出版社, 2010
[3] CHEN J J, JIANG J, ZHENG X F, et al. Forwarding-Loop Attacks in Content Delivery Networks [EB/OL]. [2010-12-10]. http://netsec.ccert.edu.cn/duanhx/files/2010/12/cdn_loop-final-camera-ready.pdf
[4] 微信之父张小龙：“微信盗号谣言”引发蝴蝶效应[EB/OL]. [2016-01-11].http://news.ifeng.com/a/20160111/47022386_0.shtml

作者简介

潘柱廷，教授级高工，启明星辰公司首席战略官，中国计算机学会CCF第十一届常务理事，CCF大数据专家委员会专家，CCF计算机安全专业委员会常务委员等；长期从事网络信息安全技术的研究、开发，以及公司的战略研究策划、技术管理等工作。

综合信息

全球公共云市场规模2016年将达2 040亿美元

市场研究公司Gartner发布报告称：全球公共云服务市场规模2016年有望达到2 040亿美元，较2015年的1 750亿美元增长16.5%。

据预计，公共云服务市场将继续呈现出高速发展态势，并一直持续至2017年。Gartner公司指出：虽然公共云服务呈现出稳定发展的态势，但是2016年发展速度最快的却是IaaS。2016年，IaaS有望增长38.4%；到2016年年底，IaaS市场规模有望达到224亿美元。

此外，SaaS也有望实现年增长20.3%，达到377亿美元。云管理和安全服务的增长率有望达到24.7%。PaaS也有望表现出非常强劲的发展势头，达到21.1%的增长率。

市场研究公司ZK Research的分析师宙斯·科拉瓦拉则预测：云服务呈现出的这种强劲发展势头有望在未来5～7年内仍然保持下去。“我想我们尚处于云服务时代的起步阶段。”他表示，“我们可能会继续看到越来越多的公司逐渐向云端转移。很多公司因为担心安全问题而对云服务避而远之。但是，随着时间的推移，人们的这种担心最终会淡化乃至消失，越来越多的企业将会增强对云服务的信心，而这必将会继续促进云服务市场的发展。（转载自《中国信息产业网》）

DOI:10.3969/j.issn.1009-6868.2016.02.011
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20160229.1613.004.html

应用驱动的大数据挖掘
Application-Driven Big Data Mining

中图分类号:TP393 文献标志码:A 文章编号:1009-6868 (2016) 02-0049-004

摘要: 认为大数据挖掘的核心和本质是应用、数据、算法和平台4个要素的紧密结合。从大数据的特点出发,结合大数据挖掘的案例,提出大数据挖掘中的平台架构、数据获取和预处理、算法的选择和集成都是应用驱动的。强调大数据挖掘的目标来自实际应用的真实需求,只有结合具体应用数据和适合应用的算法,利用高效处理平台的支撑,并将挖掘到的模式或知识应用在实践中,才能体现大数据挖掘的真正价值。

关键词: 大数据;数据挖掘;应用驱动;FIU-Miner;高端制造业

Abstract: The core of big data analysis is the combination of applications, data, algorithms and platforms. Big data mining platforms, algorithms, and big data itself are driven by applications. Big data mining tasks come from real applications. With specific application data and appropriate algorithms, using efficient processing platform, digging into the patterns or knowledge in practice, big data mining platform can show its true value.

Key words: big data; data mining; application-driven; FIU-Miner; advanced manufacturing

李涛/LI Tao[1,2]
刘峥/LIU Zheng[1]
周绮凤/ZHOU Qifeng[3]
(1. 南京邮电大学 计算机学院,南京 210023,中国;
2. 佛罗里达国际大学 计算机学院,迈阿密 33199,美国;
3. 厦门大学 自动化系,厦门 361005,中国)
(1. School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China;
2. School of Computing and Information Sciences, Florida International University, Miami 33199, USA;
3. Department of Automation, Xiamen University ,Xiamen 361005, China)

- 应用驱动的大数据挖掘能够有效处理大数据的复杂特征,真正体现大数据挖掘的价值
- 大数据的获取与预处理是应用驱动大数据挖掘的前提
- 应用驱动的大数据获取和预处理能够有效地连接企业业务需求和数据挖掘平台

1 大数据时代的发展

数字化变革推动信息技术(IT)和通信技术(CT)的飞速发展,人类社会所产出的信息总量呈爆发式增长。一方面,各行各业在日常运作中借助IT产生和存储了海量的运营数据,如商业运营、金融证券、健康医疗、科学研究等,分布在世界各地的10 000多家沃尔玛超市1 h需要处理百万条以上顾客的消费记录,数据量高达2.5 PB[1],欧洲的大型电子对撞机每天产生的记录有500 EB[2];另一方面,CT使得全世界数十亿用户通过互联网链接在一起。目前全球移动互联网的流量每月约4.2 EB,思科预计:2019年全球移动互联网的流量会增长到每年292 EB[3]。

这些海量数据被称为大数据。维基百科对大数据的定义是:"大数据是由于规模、复杂性、实时性而导致的无法在一定时间内用常规软件工具对其进行获取、存贮、搜索、分享、分析、可视化的数据集合"[4]。知名技术咨询公司Gartner对大数据的定义是:"大数据是需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力的海量、高增长率和多样化的信息资产"[5]。

大数据技术的发展使得收集、处理、管理、分析在各行各业产生的海量数据成为可能:企业利用大数据技术理解客户的属性和行为,可以提供给客户更好的个性化服务,并可以利用大数据技术改善和优化商业流程,提高企业的运营效率;政府通过大数据技术来更智能的管理城市,包括公共交通、医疗服务、可持续性发展[7]等;超市可以向用户推销所需的商品;车险公司可以知道客户的驾驶水平;甚至2012年的美国总统大选,奥巴马的竞选团队也是依赖卓越的大数据分析取得胜利。大数据已经融入各行各业,大数据时代已经来临。

2 大数据的特点与理解

2.1 大数据的特点

目前业界普遍用4V的特点来衡

收稿时间:2016-02-03
网络出版时间:2016-02-29

量大数据所带来的挑战[7]，从数据本身的表现形式上描述了大数据与以往部分抽样的“小数据”的主要区别。

大量（Volume）：大数据的体量巨大，从TB级别跃升到PB级别；

多样（Variety）：大数据面对数据类型种类繁多，例如地理位置等结构化数据，事件日志等非结构化数据，还包括图片、视频等多媒体数据等；

高速（Velocity）：大数据产生和累计的速度快，要求处理速度快，做到实时分析，和传统的离线方式的数据挖掘技术有着本质的不同；

价值（Value）：大数据所蕴含的价值密度低，但有效价值高，合理利用低密度价值的数据并对其进行正确、准确的分析，将会带来巨大的商业和社会价值。

从现有的一些大数据挖掘应用案例出发[8]，大数据挖掘的流程可以总结为：

（1）准确定义大数据挖掘问题的目标；

（2）获取大数据，并对收集到的大数据进行数据清洗等预处理；

（3）选择合适的大数据挖掘平台架构和算法；

（4）进行大数据挖掘；

（5）理解所发现的模式或应用所产生的知识。

可以看到：只有应用才能体现大数据的价值。在大数据挖掘的流程和案例中，可以充分体现出实际应用中大数据所具有的以下一些新的4V的特点：

变化性（Variable）：不同的应用场景、不同的研究目标下，大数据的机构和意义均会发生变化，在大数据的实际应用和研究中需要考虑具体的上下文，从而体现大数据的价值。

真实性（Veracity）：大数据应用的基础是真实、可靠的大数据，它们是保证分析结果准确、挖掘知识有效的前提，只有真实而准确的大数据才能获取真正有意义的结果。

波动性（Volatility）：大数据本身往往含有噪音，加上有时分析流程的不规范，导致不同的算法、不同的分析流程、不同的衡量标准下，会得到不同的分析结果。

可视化（Visualization）：数据可视化可以在大数据应用中直观地阐述分析的结果以及数据的意义，帮助用户更好地理解、应用大数据。

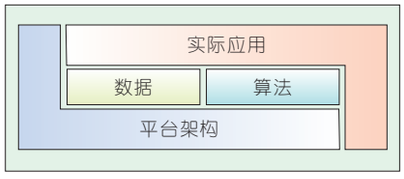
2.2 应用驱动的大数据架构

从上述大数据本身的表现形式上的4V特点出发，结合实际应用中大数据所具有的新4V特点，我们认为大数据的核心和本质是应用、算法、数据和平台4个要素的有机结合，如图1所示。大数据的基础是平台架构，数据和算法是大数据的核心，而实际应用是大数据的关键。上文所述的大数据挖掘的流程中，大数据挖掘的目标必须是来自实际应用的真实需求，只有结合具体应用数据和适合应用的算法，利用高效处理平台的有效支撑，并将挖掘到的模式或知识应用在实践中，才能提供量化、合理、可行、有价值的信息。这个应用、算法、数据和平台相结合的思想体现了大数据的本质和核心，可见大数据挖掘是应用驱动的，应用驱动的大数据挖掘能够有效处理大数据的复杂特征，体现大数据挖掘的价值。

3 应用驱动的大数据挖掘

3.1 应用驱动的大数据平台

一个高效的大数据平台可以有力地支撑海量数据的集成和数据挖掘算法，以及可视化的步骤执行，并可以利用规范的数据分析流程来保证结果的稳定性。传统的数据挖掘工具，如Weka、统计产品与服务解决方案（SPSS）等提供了友好的用户界面，但并不适合对海量数据进行挖掘分析。另外，最终用户很难对这些商业工具添加应用所需的合适算法。流行的数据挖掘算法库，如Mahout，提供了大量的数据挖掘算法，但需要数据挖掘专家来进行任务配置和算法集成，才能解决具体应用中的数据挖掘任务。最近出现的大数据挖掘产品，如Radoop等对于非基于Hadoop的算法支持有限，在多用户、多任务环境下的资源分配上也存在不足。



▲图1 大数据框架

应用驱动的大数据平台应该满足如下关键需求：

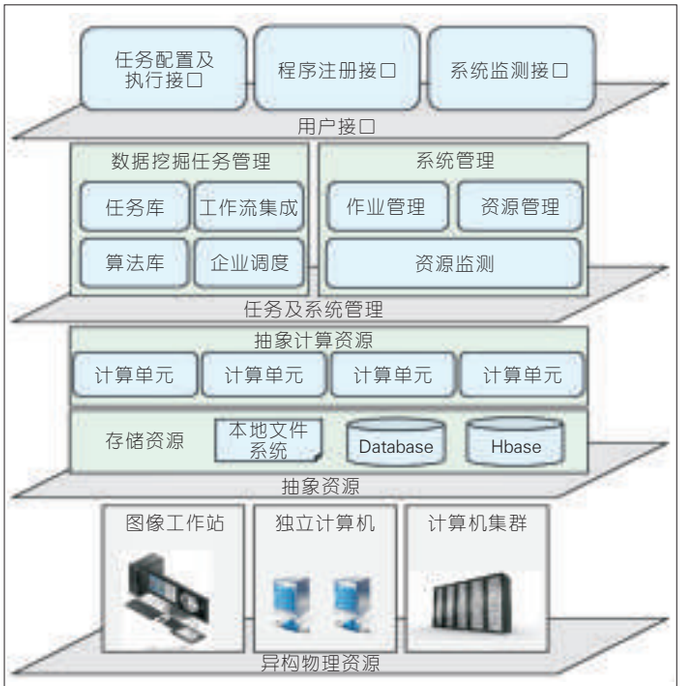
（1）人性化、友好的用户界面，快速任务配置；

（2）灵活的多语言，多算法集成；

（3）高效的分布式异构环境下的资源管理。

我们以一个快速、集成和用户友好的分布式数据挖掘系统（FIU-Miner）[9]为例介绍应用驱动的大数据平台如何满足这些需求。FIU-Miner友好的用户界面可以可视化地直接将现有算法配置成工作流，甚至无需编写任何代码，其他与挖掘任务无关的底层细节都由FIU-Miner进行管理。FIU-Miner不仅支持直接导入外部算法库来扩充分析工具集合，还会根据所导入算法的语言和运行环境自动分配对应任务到合适的计算节点。FIU-Miner可以支持各种异构的计算环境，包括PC、服务器、图形处理器（GPU）工作站等，同时根据算法实现、负载平衡、数据位置等因素来优化计算资源的利用率。

如图2所示的FIU-Miner的系统架构，包括用户界面层、任务和系统管理层、抽象计算资源层和异构物理资源层。抽象计算资源层屏蔽了不同物理环境给大数据挖掘带来的资源调度的复杂度，提高了分布式计算的效率；任务及系统管理层方便了不同数据挖掘算法的集成，多种分析任务的配置管理；友好的用户接口为基于FIU-Miner构建不同的大数据挖掘



FIU-Miner：分布式环境下数据挖掘的一个快速、集成和用户友好的系统

◀图2 FIU-Miner的系统架构

应用提供了极大的便捷，帮助数据分析人员方便有效地开展各项复杂的数据挖掘任务。

3.2 应用驱动的大数据获取与预处理

大数据的获取与预处理是应用驱动大数据挖掘的前提。以企业大数据挖掘为例，一个企业中所面临的大数据的任务多种多样，当确定大数据挖掘任务的目标时，企业对挖掘的对象和所能发现的知识往往缺乏理解，而大企业的业务流程复杂，具体业务逻辑和数据之间的对应关系十分琐碎，运营数据往往来自不同的数据源，具有不同的类型和格式，所以大数据通常无法预先规划和准备好，数据的获取是一个难题。在具体应用的大数据挖掘任务中，需要在数据的导入、整合上有很大的灵活性，只有通过业务人员和数据挖掘工程师的配合，不断尝试，才能有效地将企业的业务需求与数据挖掘的功能联系起来。在大数据获取过程中还需要根据应用需求注意数据聚合过程中的隐私保护，避免泄露用户的敏感信息。

由于大数据的多样性，所获取和整合的大数据通常还不能直接应用于数据挖掘算法，需要对数据进行预处理，结合具体应用处理数据的结构信息，抽象数据的语义信息等，并需要对所获得的大数据中的各种属性进行选择，剔除与应用无关的属性，或者引入额外的抽象测度等。大数据的质量是知识发现结果有效的保证，所以需要对数据中的噪音进行过滤，对缺失值进行处理。

3.3 应用驱动的大数据挖掘算法

数据挖掘领域中的很多算法都是从实际应用的具体需求衍生和发展出来的。从顾客交易数据分析到隐私保护数据挖掘，从文本数据挖掘到多媒体数据挖掘，从Web挖掘到社交网络挖掘，这些不同子领域的算法都是由应用推动的。数据挖掘是个交叉学科，融合了统计分析、数据库、信息检索、机器学习、模式识别、人工智能等领域的研究成果。大数据挖掘要以具体应用为驱动，根据应用数据特性，挖掘任务需求，选择、集成相应的数据挖掘和机器学习算法，并根据需要进一步进行研究，在实际问题中得到应用和验证。如基于关联规则和时间序列分析的分类算法就是关联规则发现和时间序列模式识别的有机结合；半监督学习和半监督聚类也是分类和聚类的融合结果。在处理高维、稀疏的数据时，数据的分布不明显，需要注意算法的可靠性。在处理复杂关系网络的数据时，需要根据应用的数据特征来研究能够处理异构信息网络的图挖掘算法。

4 应用驱动大数据挖掘的应用

4.1 高端制造业大数据挖掘挑战

高端制造业是指制造业中新出现的具有高技术含量、高附加值、强竞争力的产业，包括电子半导体生产、精密仪器制造、生物制药等。这些制造领域往往涉及严密的工程设计，复杂的装配生产线，大量的控制加工设备与工艺参数，精确的过程控制和材料的严格规范。随着信息技术在高端制造业中的普及，高端制造业中积累了大量的生成设计、机器设备、原材料、环境条件、生成流程等生产要素相关的历史数据，其中蕴含了对生产和管理有帮助的高价值信息。通过大数据挖掘，企业可以把隐藏在这些海量数据中有用的、深层次的信息挖掘出来，用来指导流程控制、生产调度、优化决策等方面，从而能够在实际应用中改进产品品质，提升产品性能和生产效率，最终达到提高企业行业竞争力的目的。

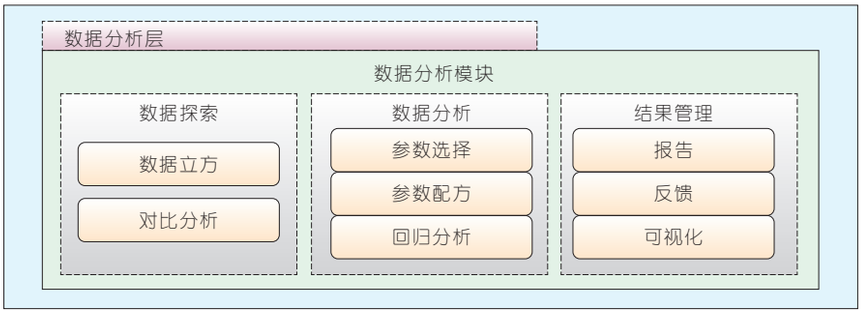
高端制造业中的数据挖掘面临很多挑战[10]，比如：如何有效分析大规模数据，如何保证对数据分析效率和分析结果的准确性等。在实际应用中，依靠传统信息系统从海量数据中进行查询和报警或单纯利用专家经验来分析和发现潜在有价值的信息已经变得不太现实。因此，企业需

要利用数据分析技术、工具或平台，智能地从大量复杂的生产原始数据中发现新的模式和知识作为改善生产过程的决策依据，系统性地提高生产效率。

4.2 等离子显示器制造中基于FIU-Miner 的大数据解决方案

四川虹欧显示器件有限公司就是利用大数据挖掘来提高等离子屏的生产良率。我们可以通过下面这个案例来阐述应用驱动的大数据挖掘。等离子显示器制造中大数据挖掘的难点是：自动化的生产方式中自动采集的数据急剧增长，需要强大的数据分析能力来支撑；大量的生成过程控制参数对高维数据分析的效率和结果的准确性提出了更高要求。这个过程本身就是对数据进行探索、分析和理解的一个循序渐进的迭代过程。因此，一个实用的系统应该提供一个集成的、高效率的分析平台来支持这个过程。

在平台方面，基于FIU-Miner，结合实际挖掘任务的具体需求和难点，我们在架构上增加了数据分析层，如图3所示。其中数据探索系统主要提供对数据的宏观理解和快速预览，以及敏感参数验证。利用联机分析处理（OLAP）技术帮助分析人员快速掌握挖掘任务相关数据的特性，指导后续的数据预处理，如属性选择和测度建立等。数据分析系统集成了根据实际大数据挖掘任务的需要所选择数据挖掘算法，包括参数选择、参数配置和回归分析。数据分析人员通过操作界面调用算法，聚焦具体的分析任务，并且算法对数据分析人员透明。结果管理系统基于业务分析结果产生分析报告，这些分析报告可以直接给决策者提供决策依据，同时报告系统也为领域专家提供收集反馈的接口。领域专家知识的引入对优化模型、改进算法具有很大的指导意义。



▲图3 数据分析层

5 结束语

大数据一词经常被用以描述和指代信息爆炸时代产生的海量信息，研究大数据的意义在于发现和理解信息内容及信息与信息之间的联系。文章从大数据本身的表现形式的4V特点出发，结合大数据挖掘的案例中体现的新4V特点，提出应用驱动的大数据挖掘思想，指出大数据的本质是应用、算法、数据和平台四个要素的有机结合。应用驱动的平台、应用驱动的数据获取和预处理、应用驱动的算法是大数据挖掘成功实施的关键。应用驱动的大数据挖掘在高端制造业的成功实施案例，验证了本文所提思想的正确性和可行性。未来，随着大数据挖掘技术的不断深入，应用驱动的大数据挖掘将会体现更大的价值和广泛的应用前景。

致谢

感谢南京邮电大学曾春秋、郑理老师在本篇文章的撰写过程中提出很多有意义的见解，并在相关工作中给予了很多帮助和贡献。

参考文献

[1] Data, Data Everywhere [EB/OL]. [2010-02-25]. http://www.economist.com/node/15557443
[2] HRUMFIEL G. High-EnergyPhysics: Down the Petabyte Highway [J]. Naure, 2011, 469 (19): 282-283
[3] BAMETT J T, SUMITS A, JAIN S, et al, Global Mobile Data Traffic Forecast, 2014-2019 [EB/OL].[2015-02-18]. http://www.ciscoknowledgenetwork.com/files/496_02-24-15_VNI_Mobile_Forecast_Prezo_for_CKN.pdf
[4] Big Data [EB/OL]. [2013-02-22]. https://en.wikipedia.org/wiki/Big_data
[5] GARTER. What Is Big Data [EB/OL]. [2014-10-20]. http://www.gartner.com/it-glossary/big-data
[6] 周绮凤, 李涛.大数据与计算可持续性[J]. 南京邮电大学学报, 2015(5): 20-31
[7] 严霄凤, 张德馨. 大数据研究[J]. 计算机技术与发展, 2013, 23(4): 168-172
[8] 李涛. 数据挖掘的应用与实践——大数据时代的案例分析[M]. 厦门：厦门大学出版社, 2015
[9] ZENG C, JIANG Y. ZHENG L, et al. Fiu-Miner: A Fast, Integrated, and User-Friendly System for Data Mining in Distributed Environment[C]// Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD' 13). USA: ACM, 2013: 1506-1509
[10] 李涛, 曾春秋, 周武柏等. 大数据时代的数据挖掘——从应用的角度看大数据挖掘[J]. 大数据, 2015, 1(4) :11-17

作者简介

李 涛，2004年7月获美国罗彻斯特大学计算机科学博士学位；现任美国佛罗里达国际大学计算机学院教授、博导，同时担任南京邮电大学计算机学院、软件学院院长，南京邮电大学大数据研究院院长；2006年获得美国国家自然科学基金委颁发的杰出青年教授奖，2009年获得佛罗里达国际大学最高学术研究奖，2010年获得IBM大规模数据分析创新奖；发表文章250余篇。

刘峥，南京邮电大学计算机学院讲师；主要研究方向为图数据挖掘与查询、网络数据挖掘等；已在国际知名会议发表多篇关于数据挖掘方面的论文。

周绮凤，厦门大学自动化系教授；研究方向为机器学习、数据挖掘及其在可持续发展等领域的应用。

DOI:10.3969/j.issn.1009-6868.2016.02.012
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20160225.1441.002.html

大数据安全与隐私保护态势
Big Data Security and Privacy Protection

中图分类号:TP393 文献标志码:A 文章编号:1009-6868 (2016) 02-0053-004

摘要: 指出安全与隐私防护是大数据面临的两个重要的问题。认为大数据在引入新的安全问题和挑战的同时,也为信息安全领域带来了新的发展契机,即基于大数据的信息安全相关技术可以反过来用于大数据的安全和隐私保护。目前,基于大数据的数据真实性分析被广泛认为是最为有效的方法。认为信息安全企业未来的发展前景为:以底层大数据服务为基础,各个企业之间组成相互依赖、相互支撑的信息安全服务体系,通过构建安全大数据,逐步形成大数据安全生态环境。

关键词: 大数据;安全;隐私;认证

Abstract: Security and privacy protection are two important issues with big data. On the one hand, big data creates new security problems and challenges. On the other hand, it creates new opportunities for the development of information security. Big-data-based information security technologies can be used for security and privacy protection. Big-data-based data authenticity analysis is widely considered to be the most effective method. Development prospects for information security are: the data underlying service is the foundation, and enterprises between each other can form the system in which they have mutual dependence, mutual support of information security service. By building up the security big data system, a good environment for information security industry is formed.

Key words: big data; security; privacy; authentication

范渊/FAN Yuan
(杭州安恒信息技术有限公司,浙江 杭州 310051)
(DBAPPSecurity, Hangzhou 310051, China)

- 基于大数据的数据真实性分析被广泛认为是最为有效的方法
- 基于大数据的数据真实性分析技术能够提高垃圾信息的鉴别能力
- 只有通过技术手段与相关政策法规等相结合,才能更好地解决大数据安全与隐私保护问题

1 大数据研究现状

目前,社会信息化和网络化的发展导致数据爆炸式增长。据统计,平均每秒有200万用户在使用谷歌搜索,Facebook用户每天共享的东西超过40亿,Twitter每天处理的推特数量超过3.4亿。同时,科学计算、医疗卫生、金融、零售业等各行业也有大量数据在不断产生。2012年全球信息总量已经达到2.7 ZB,而到2015年这一数值预计会达到8 ZB。这一现象引发了人们的广泛关注。

在学术界,图灵奖获得者Jim Gray提出了科学研究的第4范式,即以大数据为基础的数据密集型科学研究;2008年《Nature》推出了大数据专刊对其展开探讨;2011年《Science》也推出类似的数据处理专刊。IT产业界行动更为积极,持续关注数据再利用,挖掘大数据的潜在价值。目前,大数据已成为继云计算之后信息技术领域的另一个信息产业增长点。据Gartner预测:2016年全球在大数据方面的总花费将达到2 320亿美元。Gartner将大数据技术列入对众多公司和组织机构具有战略意义的十大技术与趋势之一。

不仅如此,作为国家和社会的主要管理者,各国政府也是大数据技术推广的主要推动者。2009年3月美国政府上线了data.gov网站,向公众开放政府所拥有的公共数据。随后,英国、澳大利亚等政府也开始了大数据开放的进程。截至目前,全世界已经正式有35个国家和地区构建了自己的数据开放门户网站[1]。美国政府联合6个部门宣布了2亿美元的"大数据研究与发展计划"。在中国,2012年中国通信学会、中国计算机学会等重要学术组织先后成立了大数据专家委员会,为中国大数据应用和发展提供学术咨询。

目前,大数据的发展仍然面临着许多问题,安全与隐私问题是人们公认的关键问题之一。当前,人们在互联网上的一言一行都掌握在互联网商家手中,包括购物习惯、好友联络情况、阅读习惯、检索习惯等。多项

收稿时间:2016-02-23
网络出版时间:2016-02-25

实际案例说明：即使无害的数据被大量收集后，也会暴露个人隐私。

事实上，大数据安全含义更为广泛，人们面临的威胁并不仅限于个人隐私泄漏。与其他信息一样，大数据在存储、处理、传输等过程中面临诸多安全风险，具有数据安全与隐私保护需求。而实现大数据安全与隐私保护，较以往其他安全问题（如云计算中的数据安全等）更为棘手。这是因为在云计算中，虽然服务提供商控制了数据的存储与运行环境，但是用户仍然有些办法保护自己的数据，例如通过密码学的技术手段实现数据安全存储与安全计算，或者通过可信计算方式实现运行环境安全等。而在大数据的背景下，Facebook 等商家既是数据的生产者，又是数据的存储、管理者和使用者。单纯通过技术手段限制商家对用户信息的使用，实现用户隐私保护是极其困难的事。

当前很多组织都认识到大数据的安全问题，并积极行动起来关注大数据安全问题。2012 年云安全联盟（CSA）组建了大数据工作组，旨在寻找针对数据中心安全和隐私问题的解决方案。文章在梳理大数据研究现状的基础上，重点分析了当前大数据所带来的安全挑战，详细阐述了当前大数据安全与隐私保护的关键技术。需要指出的是：大数据在引入新的安全问题和挑战的同时，也为信息安全领域带来了新的发展契机，即基于大数据的信息安全技术可以反过来用于大数据的安全和隐私保护[1]。

2 大数据安全的挑战

科学技术是一把双刃剑。大数据所引发的安全问题与其带来的价值同样引人注目。而近年爆发的“棱镜门”事件更加剧了人们对大数据安全的担忧。与传统的信息安全问题相比，大数据安全面临的挑战性问题主要体现在以下几个方面。

（1）大数据中的用户隐私保护

大量事实表明：大数据未被妥善处理会对用户的隐私造成极大的侵害。根据需要保护的内容不同，隐私保护又可以进一步细分为位置隐私保护、标识符匿名保护、连接关系匿名保护等。人们面临的威胁并不仅限于个人隐私泄漏，还在于基于大数据对人们状态和行为的预测。一个典型的例子是某零售商通过历史记录分析，比家长更早知道其女儿已经怀孕的事实，并向其邮寄相关广告信息。而社交网络分析研究也表明，可以通过其中的群组特性发现用户的属性。例如通过分析用户的 Twitter 信息，可以发现用户的政治倾向、消费习惯以及喜好的球队等。当前企业常常认为经过匿名处理后，信息不包含用户的标识符，就可以公开发布了。但事实上，仅通过匿名保护并不能很好地达到隐私保护目标。例如，AOL 公司曾公布了匿名处理后的 3 个月内部分搜索历史，供人们分析使用。虽然个人相关的标识信息被精心处理过，但其中的某些记录项还是可以被准确地定位到具体的个人。纽约时报随即公布了其识别出的 1 位用户。编号为 4、417、749 的用户是 1 位 62 岁的寡居妇人，家里养了 3 条狗，患有某种疾病，等等。另一个相似的例子是，著名的 DVD 租赁商 Netflix 曾公布了约 50 万用户的租赁信息，悬赏 100 万美元征集算法，以期提高电影推荐系统的准确度。但是当上述信息与其他数据源结合时，部分用户还是被识别出来了。研究者发现，Netflix 中的用户有很大概率对非 top 100、top 500、top 1 000 的影片进行过评分，而根据对非 top 影片的评分结果进行去匿名化攻击的效果更好。

目前用户数据的收集、存储、管理与使用等均缺乏规范，更缺乏监管，主要依靠企业的自律。用户无法确定自己隐私信息的用途。而在商业化场景中，用户应有权决定自己的信息如何被利用，实现用户可控的隐私保护。包括：数据采集时的隐私保护，如数据精度处理；数据共享、发布时的隐私保护，如数据的匿名处理、人工加扰等；数据分析时的隐私保护；数据生命周期的隐私保护；隐私数据可信销毁等。

（2）大数据的可信性

关于大数据的一个普遍的观点是：数据自己可以说明一切，数据自身就是事实。但实际情况是：如果不仔细甄别，数据也会欺骗，就像人们有时会被自己的双眼欺骗一样。

大数据可信性的威胁之一是：伪造或刻意制造的数据，而错误的数据往往会导致错误的结论。若数据应用场景明确，就可能有人刻意制造数据、营造某种“假象”，诱导分析者得出对其有利的结论。

由于虚假信息往往隐藏于大量信息中，使得人们无法鉴别真伪，从而做出错误判断。例如，一些点评网站上的虚假评论，混杂在真实评论中使得用户无法分辨，可能误导用户去选择某些劣质商品或服务。由于当前网络社区中虚假信息的产生和传播变得越来越容易，其所产生的影响不可低估。用信息安全技术手段鉴别所有来源的真实性是不可能的。

大数据可信性的威胁之二是：数据在传播中的逐步失真。原因之一是人工干预的数据采集过程可能引入误差，由于失误导致数据失真与偏差，最终影响数据分析结果的准确性。此外，数据失真还有数据的版本变更的因素。在传播过程中，现实情况发生了变化，早期采集的数据已经不能反映真实情况[2]。例如，餐馆电话号码已经变更，但早期的信息已经被其他搜索引擎或应用收录，所以用户可能看到矛盾的信息而影响其判断。因此，大数据的使用者应该有能力基于数据来源的真实性、数据传播途径、数据加工处理过程等，了解各项数据可信度，防止分析得出无意义或者错误的结果。

密码学中的数字签名、消息鉴别码等技术可以用于验证数据的完整

性，但应用于大数据的真实性时面临很大困难，主要根源在于数据粒度的差异。例如，数据的发源方可以对整个信息签名，但是当信息分解成若干组成部分时，该签名无法验证每个部分的完整性。而数据的发源方无法事先预知哪些部分被利用，如何被利用，难以事先为其生成验证对象。

（3）大数据访问控制的实现

访问控制是实现数据受控共享的有效手段。由于大数据可能被用于多种不同场景，其访问控制需求十分突出。大数据访问控制的特点与难点在于：

难以预设角色，实现角色划分。由于大数据应用范围广泛，它通常要为来自不同组织或部门、不同身份与目的的用户所访问，实施访问控制是基本需求。然而，在大数据的场景下，有大量的用户需要实施权限管理，且用户具体的权限要求未知。面对未知的大量数据和用户，预先设置角色十分困难。

难以预知每个角色的实际权限。由于大数据场景中包含海量数据，安全管理员可能缺乏足够的专业知识，无法准确地为用户指定其所可以访问的数据范围。而且从效率角度讲，定义用户所有授权规则也不是理想的方式。以医疗领域应用为例，医生为了完成其工作可能需要访问大量信息，但对于数据能否访问应该由医生来决定，不应该需要管理员对每个医生做特别的配置。但同时又应该能够提供对医生访问行为的检测与控制，限制医生对病患数据的过度访问。此外，不同类型的大数据中可能存在多样化的访问控制需求。例如，在Web 2.0个人用户数据中，存在基于历史记录的访问控制；在地理地图数据中，存在基于尺度以及数据精度的访问控制需求；在流数据处理中，存在数据时间区间的访问控制需求等。如何能够统一地描述与表达访问控制需求也是一个极具挑战性的问题[3]。

由于大数据分析技术的出现，企业可以超越以往的“保护－检测－响应－恢复（PDRR）”模式，更主动地发现潜在的安全威胁。例如，IBM推出了名为“IBM大数据安全智能”的新型安全工具，可以利用大数据来侦测来自企业内外部的安全威胁，包括扫描电子邮件和社交网络，标示出明显心存不满的员工，提醒企业注意，预防其泄露企业机密。“棱镜”计划也可以被理解为应用大数据方法进行安全分析的成功故事。通过收集各个国家各种类型的数据，利用安全威胁数据和安全分析形成系统方法发现潜在危险局势，在攻击发生之前识别威胁。

3 基于认证分析的大数据分析技术

相比于传统技术方案，基于大数据的威胁发现技术具有以下优点。

（1）分析内容的范围更大。传统的威胁分析主要针对的内容为各类安全事件。一个企业的信息资产则包括数据资产、软件资产、实物资产、人员资产、服务资产和其他为业务提供支持的无形资产。由于传统威胁检测技术的局限性，其并不能覆盖这6类信息资产，因此所能发现的威胁也是有限的。通过在威胁检测方面引入大数据分析技术，可以更全面地发现针对这些信息资产的攻击。例如通过分析企业员工的即时通信数据、Email数据等可以及时发现人员资产是否面临其他企业“挖墙脚”的攻击威胁。再比如，通过对企业的客户部订单数据的分析，也能够发现一些异常的操作行为，进而判断是否危害公司利益。可以看出：分析内容范围的扩大使得基于大数据的威胁检测更加全面。

（2）分析内容的时间跨度更长。现有的许多威胁分析技术都是内存关联性的，也就是说实时收集数据，采用分析技术发现攻击。分析窗口通常受限于内存大小，无法应对持续性和潜伏性攻击。引入大数据分析技术后，威胁分析窗口可以横跨若干年的数据，因此威胁发现能力更强，可以有效应对高级持续性威胁（APT）类攻击。

（3）攻击威胁的预测性。传统的安全防护技术或工具大多是在攻击发生后对攻击行为进行分析和归类，并做出响应。基于大数据的威胁分析，可进行超前的预判，它能够寻找潜在的安全威胁，对未发生的攻击行为进行预防。

（4）对未知威胁的检测。传统的威胁分析通常是由经验丰富的专业人员根据企业需求和实际情况展开，然而这种威胁分析的结果很大程度上依赖于个人经验。同时，分析所发现的威胁也是已知的。大数据分析的特点是侧重于普通的关联分析，而不侧重因果分析，因此通过采用恰当的分析模型，可发现未知威胁。

虽然基于大数据的威胁发现技术具有上述的优点，但是该技术目前也存在一些问题和挑战，主要集中在分析结果的准确程度上。一方面，大数据的收集很难做到全面，而数据又是分析的基础，它的片面性往往会导致分析出的结果的偏差。为了分析企业信息资产面临的威胁，不但要全面收集企业内部的数据，还要对一些企业外的数据进行收集，这些在某种程度上是一个大问题。另一方面，大数据分析能力的不足影响威胁分析的准确性。例如，纽约投资银行每秒会有5 000次网络事件，每天会从中捕捉25 TB数据。如果没有足够的分析能力，要从如此庞大的数据中准确地发现极少数预示潜在攻击的事件，进而分析出威胁是几乎不可能完成的任务。

身份认证是信息系统或网络中确认操作者身份的过程。传统的认证技术主要通过用户所知的秘密，例如口令，或者持有的凭证，例如数字证书，来鉴别用户。这些技术面临着两个问题：（1）攻击者总是能够找到

方法来骗取用户所知的秘密，或窃取用户持有的凭证，从而通过认证机制的认证。例如攻击者利用钓鱼网站窃取用户口令，或者通过社会工程学方式接近用户，直接骗取用户所知秘密或持有的凭证。(2)传统认证技术中认证方式越安全往往意味着用户负担越重。例如，为了加强认证安全而采用的多因素认证。用户往往需要同时记忆复杂的口令，还要随身携带硬件USB Key，一旦忘记口令或者忘记携带USB Key，就无法完成身份认证。为了减轻用户负担，一些生物认证方式出现，利用用户具有的生物特征，例如指纹等，来确认其身份。然而，这些认证技术要求设备必须具有生物特征识别功能，例如指纹识别。因此很大程度上限制了这些认证技术的广泛应用。

认证技术中引入大数据分析则能够有效地解决这两个问题。基于大数据的认证技术指的是收集用户行为和设备行为数据，并对这些数据进行分析，获得用户行为和设备行为的特征，进而通过鉴别操作者行为及其设备行为来确定其身份。这与传统认证技术利用用户所知秘密，所持有凭证，或具有的生物特征来确认其身份有很大不同。这种新的认证技术具有如下优点。

(1)攻击者很难模拟用户行为特征来通过认证，因此更加安全。利用大数据技术所能收集的用户行为和设备行为数据是多样的，可以包括用户使用系统的时间，经常采用的设备，设备所处物理位置，甚至是用户的操作习惯数据。通过这些数据的分析能够为用户勾画一个行为特征的轮廓。攻击者很难在方方面面都模仿到用户行为，因此其与真正用户的行为特征轮廓必然存在一个较大偏差，无法通过认证。

(2)减轻了用户负担。用户行为和设备行为特征数据的采集、存储等都由认证系统完成。相比于传统认证技术，极大地减轻了用户负担。

(3)可以更好地支持各系统认证机制的统一。基于大数据的认证技术可以让用户在整个网络空间采用相同的行为特征进行身份认证，避免不同系统采用不同认证方式，且用户所知秘密或所持有凭证也各不相同而带来的种种不便。

虽然基于大数据的认证技术具有上述优点，但同时也存在一些问题和挑战亟待解决。

(1)初始阶段的认证问题。基于大数据的认证技术是建立在大量用户行为和设备行为数据分析的基础上，而初始阶段不具备大量数据。因此，无法分析出用户行为特征，或者分析的结果不够准确。

(2)用户隐私问题。基于大数据的认证技术为了能够获得用户的行为习惯，必然要长期持续地收集大量的用户数据。那么如何在收集和分析这些数据的同时，确保用户隐私也是亟待解决的问题。它是影响这种新的认证技术是否能够推广的主要因素。

目前，基于大数据的数据真实性分析被广泛认为是最为有效的方法。许多企业已经开始了这方面的研究工作，例如Yahoo和Thinkmail等利用大数据分析技术来过滤垃圾邮件；Yelp等社交点评网络用大数据分析来识别虚假评论；新浪微博等社交媒体利用大数据分析来鉴别各类垃圾信息等。基于大数据的数据真实性分析技术能够提高垃圾信息的鉴别能力。一方面，引入大数据分析可以获得更高的识别准确率，例如，对于点评网站的虚假评论，可以通过收集评论者的大量位置信息、评论内容、评论时间等进行分析，鉴别其评论的可靠性，如果某评论者为某品牌多个同类产品都发表了恶意评论，则其评论的真实性就值得怀疑；另一方面，在进行大数据分析时，通过机器学习技术，可以发现更多具有新特征的垃圾信息。然而该技术仍然面临一些困难，主要是虚假信息的定义，分析模型的构建等。

4 结束语

前面列举了部分当前基于大数据的信息安全技术，未来必将涌现出更多、更丰富的安全应用和安全服务。由于此类技术以大数据分析为基础，因此如何收集、存储和管理大数据就是相关企业或组织所面临的核心问题。除了极少数企业有能力做到之外，对于绝大多数信息安全企业来说，更为现实的方式是通过某种方式获得大数据服务，结合自己的技术特色领域，对外提供安全服务。一种未来的发展前景是：以底层大数据服务为基础，各个企业之间组成相互依赖、相互支撑的信息安全服务体系，总体上形成信息安全产业界的良好生态环境。大数据带来了新的安全问题，但它自身也是解决问题的重要手段。文章从大数据的隐私保护、信任、访问控制等角度出发，梳理了当前大数据安全与隐私保护相关关键技术。但总体上来说，当前全球针对大数据安全与隐私保护的相关研究还不充分，只有通过技术手段与相关政策法规等相结合，才能更好地解决大数据安全与隐私保护问题。

参考文献

[1] 冯登国, 张敏, 李昊. 大数据安全与隐私保护[J]. 计算机学报, 2014, 36(01): 246–258
[2] 谢邦昌, 姜叶飞. 大数据时代 隐私如何保护[J]. 中国统计, 2013(06): 24–28
[3] 应钦. 大数据安全与隐私保护技术探究[J]. 硅谷, 2014(10): 15–19. doi:10.3969/j.issn.1671–7597.2014.10.044

作者简介

范渊，毕业于美国加州州立大学，现任杭州安恒信息技术有限公司董事长兼CEO；长期从事在线应用安全、数据库安全和审计、Compliance（如SOX、PCI、ISO17799/27001）等方面的研究；并作为项目负责人已承担国家级科技计划项目8项，省、市级科技计划项目16项；先后入选国家“千人计划”特聘专家，国家科技部“科技创新创业人才”，第3届世界浙商大会创业创新奖等；申请专利45项，授权发明专利8项，并在重要学术期刊发表多篇重要论文。

DOI:10.3969/j.issn.1009-6868.2016.02.013
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20160224.1559.002.html

编者按:网络空间安全作为一项新的全球治理议程,已经成为世界关注的焦点、各国政府的战略目标之一,但人们对网络空间安全的研究,还缺乏全面系统的理论指导,针对该问题,我刊特转载自《科学网》一篇由北京邮电大学杨义先、钮心忻教授编写的《安全通论——攻防篇之“盲对抗”》(原文网址:http://blog.sciencenet.cn/blog-453322-947304.html)。在该文章中,给出了黑客攻击能力和红客防御能力的可达理论极限,并巧妙地构造了一个随机变量$Z=(X+Y)\bmod 2$,将一次真正成功的攻防问题,等价地转换成了攻击信道$(X;Z)$,同时恰到好处地应用了看似并不相关的仙农编码定理。

安全通论——攻防篇之“盲对抗”

The General Theory of Security: Blind Confrontation in Offensive and Defensive

中图分类号:TN929.5 文献标志码:A 文章编号:1009-6868 (2016) 02-0057-004

摘要:精确地给出了黑客攻击能力和红客防御能力的可达理论极限。对黑客来说,如果他想真正成功地把红客打败k次,一定有某种技巧,使他能够在k/C次进攻中,以任意接近1的概率达到目的;如果黑客经过n次攻击,获得了S次真正成功,那么一定有$S\leqslant nC$。对红客来说,如果他想真正成功地把黑客挡住R次,一定有某种技巧,使得他能够在R/C次防御中,以任意接近1的概率达到目的;如果红客经过n次防卫,获得了R次真正成功,一定有$R\leqslant ND$。这里C和D分别是攻击信道和防御信道的信道容量。如果$C<D$,则黑客输;如果$C>D$,则红客输;如果$C=D$,则红黑实力相当。

关键词:黑客;红客;进攻;防御;信道

Abstract: In this paper, the limit theory of hacker attack ability and honker defense ability is given. If a hacker wants to beat a honker K times, there must be some skills, so that he can achieve the purpose with probability arbitrarily close to 1 in the k/C times' offensive. If a hacker achieves S successes after n attacks, there must be $S\leqslant nC$. If a honker wants to defend against a hacker R times, there must be some skills, so he can achieve the purpose with probability arbitrarily close to 1 in the k/C times' defensive; If a honker achieve n times' real success after n times' defensive, there must be $R\leqslant ND$. C and D respectively represent the channel capacity of offensive channel and defensive channel. If $C<D$, then the hacker loses. If $C>D$, then the honker loses. If $C=D$, they have considerable strength.

Key words: hacker; honker; offensive; defensive; channel

杨义先/YANG Yixian
钮心忻/NIU Xinxin
(北京邮电大学 信息安全中心,北京 100876)
(Information Security Center, Beijing University of Post and Telecommunication, Beijing 100876, China)

- 攻防是安全的核心,特别是在有红黑双方对抗的场景下,攻防几乎就等于安全
- 在攻防系统中,只有攻方和守方这两个直接利益相关方,但绝没有利益无关的第三方
- 无裁判攻防可分为盲攻防和非盲攻防

1 盲对抗场境

攻防是安全的核心,特别是在有红黑双方对抗的场景下(比如,战场、公安、网络安全等),攻防几乎就等于安全。所以,在安全通论的建立过程中,我们将花费更多的篇幅来研究攻防问题。但是,长期以来,人们并未对攻防场景进行过清晰的整理,再加上攻防一词经常被滥用,从而导致攻防几乎成了一个只能意会不能言传的名词,当然就更无法对攻防进行系统的理论研究了。

因此,为了开始我们的研究,必须首先理清攻防场景。更准确地说,下面我们只考虑无裁判的攻防,因为像日常看到的诸如拳击比赛等有裁判攻防的体育项目,并不是真正的攻防。在攻防系统中,只有攻方和守方这两个直接利益相关方(虽然有时涉及的人员会超过两个),但绝没有利益无关的第三方。所以,对攻防结果

收稿时间:2016-01-22
网络出版时间:2016-02-24

来说，吹哨的裁判员其实是干扰，是噪音，而且还是主观的噪音，因此必须要去除。

无裁判攻防又可以进一步分为两大类：盲攻防、非盲攻防。所谓盲攻防，指每次攻防后，双方都只知道自己的损益情况，而对另一方却一无所知，比如，大国博弈、网络攻防、实际战场、间谍战、泼妇互骂等都是盲攻防的例子；非盲攻防，指每次攻防后，双方都知道本次攻防的结果，而且还一致认同这个结果，比如，石头剪刀布游戏、下棋、炒股等都是非盲攻防的例子。一般来说，盲对抗更血腥和残酷，而非盲对抗的娱乐味更浓。在文章中，我们只考虑盲攻防[1]。

为了更形象地说明，下面我们仍然借用拳击的术语来介绍盲攻防系统。当然，这时裁判已经被赶走，代替裁判的是无所不知的上帝。

攻方（黑客）是个神仙拳击手，永远不知累，他可用随机变量 X 来表示。黑客每次出击后，都会对自己的本次出击给出一个真心盲评价（比如，自认为本次出击成功或失败，当自认为本次出击成功时，记为 $X=1$；当自认为出击失败时，记为 $X=0$），但是，他绝不将这个真心盲评价系统告诉任何人。此处，之所以假定攻方（黑客）的盲自评要对外保密，是因为我们可以因此认定他的盲自评是真心的，不会也没有必要弄虚作假。

守方（红客）也是个神仙拳击手，他也永远不知累，可用随机变量 Y 来表示。红客每次守卫后，也都会对自己的这次守卫给出一个真心盲评价（比如，自认为本次守卫是成功或失败，当自认为守卫成功时，记为 $Y=1$；当自认为守卫失败时，记为 $Y=0$）。这个评价也仍然绝不告诉任何人。同样，之所以要假定红客的盲自评要对外保密，是因为我们可以因此认定他的自评是真心的，不会也没有必要弄虚作假。

裁判员虽然被赶走了，但是我们却把上帝请来了。不过，上帝只是远远地呆在凌霄宝殿看热闹，他知道攻守双方心里的真实想法，因此也知道双方对每次攻防的真心盲自评，于是他可将攻守双方过去 N 次对抗的盲自评结果记录下来：$(X, Y)=(X_1, Y_1)、(X_2, Y_2)、\cdots、(X_N, Y_N)$。

由于当 N 趋于无穷大时，频率趋于概率 P_r，所以只要攻守双方足够长时间对抗之后，上帝便可以得到随机变量 X、Y 的概率分布和 (X, Y) 的联合概率分布：

P_r（攻方盲自评为成功）= $P_r(X=1)=p$

P_r（攻方盲自评为失败）= $P_r(X=0)=1-p$，$0<p<1$

P_r（守方盲自评为成功）= $P_r(Y=1)=q$

P_r（守方盲自评为失败）= $P_r(Y=0)=1-q$，$0<q<1$

P_r（攻方盲自评为成功，守方盲自评为成功）= $P_r(X=1, Y=1)=a$，$0<a<1$

P_r（攻方盲自评为成功，守方盲自评为失败）= $P_r(X=1, Y=0)=b$，$0<b<1$

P_r（攻方盲自评为失败，守方盲自评为成功）= $P_r(X=0, Y=1)=c$，$0<c<1$

P_r（攻方盲自评为失败，守方盲自评为失败）= $P_r(X=0, Y=0)=d$，$0<d<1$

这里，a、b、c、d、p、q 之间还满足式(1) ~ (3)：

$$a+b+c+d=1 \tag{1}$$

$$p=P_r(X=1)=P_r(X=1,Y=0)+P_r(X=1,Y=1)=a+b \tag{2}$$

$$q=P_r(Y=1)=P_r(X=1,Y=1)+P_r(X=0,Y=1)=a+c \tag{3}$$

所以，6 个变量 a、b、c、d、p、q 中，其实只有 3 个是独立的。

足够长的时间之后，上帝看够了，便叫停攻守双方。让他们分别对擂台进行有利于自己的秘密调整，当然某方（或双方）也可以放弃本次调整的机会，如果他（他们）认为当前擂台对自己更有利的话。这里，所谓的秘密调整，指双方都不知道对方做了些什么调整。比如，针对网络空间安全对抗，也许红客安装了一个防火墙，也许黑客植入了一种新的恶意代码等；针对阵地战的情况，也许攻方调来了一去增援部队，也许守方又埋了一批地雷等。

总之，攻守双方调整完成后，双方又在新的擂台上，再开始下一轮的对抗。

不过，我们不研究攻守双方的下一轮对抗，只考虑当前轮，即由上面的 X、Y、(X, Y) 等随机变量所组成的系统。

至此，盲攻防场景的精确描述就完成了。可见，网络战、间谍战、泼妇互骂等对抗性很惨烈的攻防，都是典型的盲对抗。

2 黑客攻击能力极限

根据第 1 节中的随机变量 X 和 Y，上帝再新造一个随机变量 $Z=(X+Y) \bmod 2$。由于任何两个随机变量都可以组成一个通信信道，所以我们把 X 作为输入，Z 作为输出，上帝便可构造出一个通信信道 F，我们称之为攻击信道。

由于攻方（黑客）的目的是要打败守方（红客），所以黑客是否真正成功，不能由自己的盲评价来定（虽然这个盲评价是真心的），而应该是由红客的真心盲评价说了算，所以则有式(4)：

$$\begin{aligned}&\{\text{攻方的某次攻击真正成功}\}\\&=\{\text{攻方本次盲自评为成功}\cap\text{守方本次盲自评为失败}\}\cup\{\text{攻方本次盲自评为失败}\cap\text{守方本次盲自评为失败}\}\\&=\{X=1,Y=0\}\cup\{X=0,Y=0\}\\&=\{X=1, Z=1\}\cup\{X=0,Z=0\}\\&=\{\text{1 bit 信息被成功地从通信系统 F 的发端}(X)\text{传输到了收端}(Z)\}。\end{aligned} \tag{4}$$

另一方面，如果有 1 bit 信息被成功地从发端 (X) 传到了收端 (Z)，那么要么是“$X=0$，$Z=0$”，要么是“$X=1$，$Z=1$”。由于 $Y=(X+Z) \bmod 2$，所以由“$X=0$，$Z=0$”推知“$X=0$，$Y=0$”；由“$X=1$，$Z=1$”推知“$X=1$，$Y=0$”。而“$X=0$，$Y=0$”意味着攻防本次盲自评为失败∩守方本次盲自评为失败；“$X=1$，$Y=0$”意味着

攻方本次盲自评为成功∩守方本次盲自评为失败；综合起来就意味着攻方获得某次攻击的真正成功。

简而言之，我们可以知道：如果黑客的某次攻击真正成功，那么攻击信道F就成功地传输1 bit到收端；如果有1 bit被成功地从攻击信道F的发端，传送到了收端，那么黑客X就获得了一次真正成功攻击。

引理1：黑客获得一次真正成功的攻击，其实就对等于攻击信道F成功地传输了1 bit。

根据仙农信息论的著名《信道编码定理》[2]：如果信道F的容量为C，那么对于任意传输率$k/n\leq C$，都可以在译码错误概率任意小的情况下，通过某个n bit长的码字，成功地把k bit传输到收信端；如果信道F能够用n长码字，把S bit无误差地传输到收端，那么，一定有$S\leq nC$。

定理1（黑客攻击能力极限定理）：设由随机变量$(X;Z)$组成的攻击信道F的信道容量为C。那么：如果黑客想真正成功地把红客打败k次，一定有某种技巧（对应于仙农编码），使得他能够在k/C次攻击中，以任意接近1的概率达到目的；如果黑客经过n次攻击，获得了S次真正成功的攻击，一定有$S\leq nC$。

由定理1可知：只要求出攻击信道F的信道容量C，那么黑客的攻击能力极限就确定了。

下面我们需要计算出F的信道容量C。

首先，由于随机变量$Z=(X+Y) \bmod 2$，所以可以由X和Y的概率分布，得到Z的概率分布如下：

$$P_r(Z=0)$$
$$= P_r(X=Y)$$
$$= P_r(\text{攻守双方的盲自评结果一致})$$
$$= P_r(X=0,\ Y=0)+ P_r(X=1,\ Y=1)$$
$$= a+d$$
$$P_r(Z=1)$$
$$= P_r(X\neq Y)$$
$$= P_r(\text{攻守双方的盲自评结果相反})$$
$$= P_r(X=0,\ Y=1)+ P_r(X=1,\ Y=0)$$
$$= b+c$$
$$= 1-(a+d)$$

考虑通信系统F，它由随机变量X和Z构成的，即它以X为输入，Z为输出，它的2X2阶转移概率矩阵为$A=[A(x,z)]=[P_r(z \mid x)]$，这里$x$，$z$=0或1。

$$A(0,0)$$
$$= P_r(Z=0 \mid X=0)$$
$$= [P_r(Z=0,\ X=0)]/\ P_r(X=0)$$
$$= [P_r(Y=0,\ X=0)]/(1-p)$$
$$= d/(1-p)$$
$$A(0,1)$$
$$= P_r(Z=1 \mid X=0)$$
$$= [P_r(Z=1,\ X=0)]/\ P_r(X=0)$$
$$= [P_r(Y=1,\ X=0)]/(1-p)$$
$$= c/(1-p)$$
$$= 1-d/(1-p)$$
$$A(1,0)$$
$$= P_r(Z=0 \mid X=1)$$
$$= [P_r(Z=0,\ X=1)]/\ P_r(X=1)$$
$$= [P_r(Y=1,\ X=1)]/p$$
$$=a/p$$
$$A(1,1)$$
$$= P_r(Z=1 \mid X=1)$$
$$= [P_r(Z=1,\ X=1)]/\ P_r(X=1)$$
$$= [P_r(Y=0,\ X=1)]/p$$
$$=b/p$$
$$=(p-a)/p$$

由于随机变量(X,Z)的联合概率分布为：

$$P_r(X=0,\ Z=0)=P_r(X=0,Y=0)= d$$
$$P_r(X=0,\ Z=1)=P_r(X=0,Y=1)= c$$
$$P_r(X=1,\ Z=0)=P_r(X=1,Y=1)= a$$
$$P_r(X=1,\ Z=1)=P_r(X=1,Y=0)= b$$

所以，随机变量X与Z之间的互信息为：

$$I(X,\ Z)$$
$$=\sum_x\sum_z p(x,\ z)\log(p(x,\ z)/[p(x)p(z)])$$
$$=d\log[d/[(1-p)(a+d)]] +$$
$$c\log[c/[(1-p)(b+c)]] +$$
$$a\log[a/[p(a+d)]] +b\log[b/[p(b+c)]] \quad (5)$$

由于此处有：$a+b+c+d=1$，$p=a+b$，$q=a+c$，$0<a,b,c,d,p,q<1$，所以式(5)可以进一步转化为只与变量a和p有关的式(6)（注意：此时q已不再是变量，而是确定值了）：

$$I(X,\ Z)$$
$$=[1+a-(p+q)]\log[[1+a-$$
$$(p+q)]/[(1-p)(1+2a-p-q)]] +$$
$$(q-a)\log[(q-a/[(1-p)(p+q-2a)])] +$$
$$a\log[a/[p(1+2a-p-q)]] +$$
$$(p-a)\log[(p-a)/[p(p+q-2a)]] \quad (6)$$

利用此$I(X,\ Z)$就可知：以X为输入，Z为输出的信道F的信道容量C就等于$\text{Max}[I(X,\ Z)]$（这里最大值是针对X为所有可能的二元离散随机变量来计算的）。更简单地说：容量C等于$\text{Max}_{0<a,p<1}[I(X,Z)]$（这里的最大值是对仅仅两个变量$a$和$p$在条件$0<a$，$p<1$下之取的），所以该信道容量的计算就很简单了。

3 红客守卫能力极限

设随机变量X、Y、Z和(X,Y)等都与前面相同。

根据随机变量Y（红客）和Z，上帝再组成另一个通信信道G，称为防御信道，即把Y作为输入，把Z作为输出。

由于守方（红客）的目的是要挡住攻方（黑客）的进攻，所以红客是否真正成功，不能由自己的盲评价来定，而应该是由黑客的真心盲评价说了算，所以就应该有如式(7)中的等式成立：

$$\{\text{守方的某次防卫真正成功}\}$$
$$=\{\text{守方本次盲自评为成功}\cap\text{攻方本次盲自评为失败}\}\cup\{\text{守方本次盲自评为失败}\cap\text{攻方本次盲自评为失败}\}$$
$$=\{Y=1,\ X=0\}\cup\{Y=0,\ X=0\}$$
$$=\{Y=1,\ Z=1\}\cup\{Y=0,\ Z=0\}$$
$$=\{1\text{ bit信息被成功地从防御信道G的发端}(Y)\text{传输到了收端}(Z)\}。 \quad (7)$$

与攻击信道的情况类似，反过来，式(7)也就意味着：如果在防御信道G中，1 bit信息被成功地从发端(Y)传到了收端(Z)，那么红客就获得了一次真正成功的防卫。

引理2：红客获得一次真正成功

的守卫，其实就是防御信道G成功地传输了1 bit。

定理2（红客守卫能力极限定理）：设由随机变量$(Y;Z)$组成的防御信道G的信道容量为D。那么则有：如果红客想真正成功地把黑客挡住R次，那么一定有某种技巧（对应于仙农编码），使得他能够在R/C次防御中，以任意接近1的概率达到目的；如果红客经过N次守卫，获得了R次真正成功的守卫，那么，一定有$R\leqslant ND$。

考虑通信系统G，它由随机变量Y和Z构成的，即它以Y为输入，Z为输出，它的2X2阶转移概率矩阵为$B=[B(y,z)]=[P_t(z \mid y)]$，这里$y,z=0$或1。

$$
\begin{aligned}
&B(0,0)\\
&=P_t(Z=0 \mid Y=0)\\
&=[P_t(Z=0,Y=0)]/P_t(Y=0)\\
&=[P_t(X=0,Y=0)]/(1-q)\\
&=d/(1-q)\\
&B(0,1)\\
&=P_t(Z=1 \mid Y=0)\\
&=[P_t(Z=1,Y=0)]/P_t(Y=0)\\
&=[P_t(X=1,Y=0)]/(1-q)\\
&=b/(1-q)\\
&B(1,0)\\
&=P_t(Z=0 \mid Y=1)\\
&=[P_t(Z=0,Y=1)]/P_t(Y=1)\\
&=[P_t(X=1,Y=1)]/q\\
&=a/q\\
&B(1,1)\\
&=P_t(Z=1 \mid Y=1)\\
&=[P_t(Z=1,Y=1)]/P_t(Y=1)\\
&=[P_t(X=0,Y=1)]/q\\
&=c/q
\end{aligned}
$$

由于随机变量(Y,Z)的联合概率分布为：

$$P_t(Y=0,Z=0)=P_t(X=0,Y=0)=d$$
$$P_t(Y=0,Z=1)=P_t(X=1,Y=0)=b$$
$$P_t(Y=1,Z=0)=P_t(X=1,Y=1)=a$$
$$P_t(Y=1,Z=1)=P_t(X=0,Y=1)=c$$

所以，随机变量Y与Z之间的互信息为：

$$
\begin{aligned}
&I(Y,Z)\\
&=\sum_Y\sum_Z p(y,z)\log(p(y,z)/[p(y)p(z)])\\
&=d\log[d/[(1-q)(a+d)]]+\\
&b\log[b/[(1-q)(b+c)]]+\\
&a\log[a/[q(a+d)]]+c\log[c/[q(b+c)]] \qquad (8)
\end{aligned}
$$

由于此处有：$a+b+c+d=1$，$p=a+b$，$q=a+c$，$0<a,b,c,d,p,q<1$，所以式(8)可以进一步转化为只与变量a和q有关的式(9)（注意：此时p不再是变量，而是确定值了）：

$$
\begin{aligned}
&I(Y,Z)\\
&=(1+a-p-q)\log[(1+a-p-q)/\\
&[(1-q)(1+2a-p-q)]]+(p-a)\log[(p-a)/\\
&[(1-q)(p+q-2a)]]+a\log[a/\\
&[q(1+2a-p-q)]]+(q-a)\log[(q-a)/\\
&[q(p+q-2a)]] \qquad (9)
\end{aligned}
$$

利用此$I(Y,Z)$可知：以Y为输入，Z为输出的防御信道G的信道容量D就等于$\mathrm{Max}[I(Y,Z)]$（这里最大值是针对Y为所有可能的二元离散随机变量来计算的）或者更简单地说，容量D等于$\mathrm{Max}_{0<a,\ q<1}[I(Y,Z)]$（这里的最大值是对仅仅两个变量$a$和$q$在条件$0<a,q<1$下之取的），所以该信道容量的计算就很简单。

4 攻守双方的实力比较

由于信道容量是在传信率k/n保持不变的情况下，系统所能够传输的最大信息比特数，而每成功传输1 bit，就相当于攻方的一次攻击真正成功（或守方的一次防守真正成功），所以从宏观的角度来看，我们可以推导出定理3。

定理3（攻守实力定理）：设C和D分别表示攻击信道F和防御信道G的信道容量，如果$C<D$，那么整体上黑客处于弱势；如果$C>D$，那么整体上红客处于弱势；如果$C=D$，那么红黑双方实力相当。

我们需要注意到：攻击信道的容量C，其实是q的函数，所以可以记之为$C(q)$；同理，防御信道的容量D是p的函数，可以记之为$D(p)$。由此，在盲对抗中，红黑双方可以通过对自己预期的调整，即改变相应的概率分q和p，从而改变$C(q)$和$D(p)$的大小，并最终提升自己在盲对抗中的胜算情况。换句话说，我们证明了一个早已熟知的社会事实，即定理4。

定理4（知足常乐定理）：在盲对抗中，黑客（或红客）有两种思路来提高自己的业绩，或称为“幸福指数”。增强自身的相对打击（或抵抗）力，即增加b和d（或c和a）；降低自己的贪欲，即增加p（或q）。但是，需要注意你可能无法改变外界，即调整b和d（或c和a），但却可以改变自身，即调整p（或q）。由此可见：知足常乐是盲对抗中的一个真理。

5 结束语

我们的诀窍有两点：巧妙地构造了一个随机变量$Z=(X+Y)\mathrm{mod}2$，并将一次真正成功的攻防问题，等价地转换成了攻击信道$(X;Z)$（或者防守信道$(Y;Z)$）的1 bit成功传输问题；恰到好处地应用了看似风马牛不相关的仙农编码定理。以上两点，任缺一项，就不会找到让“黑客悟空”永远也跳不出去的“如来手掌”。

参考文献

[1] THOMAS M C, THOMAS J A.信息论基础[M]. 阮吉寿, 张华, 译. 北京：机械工业出版社出版, 2007

[2] SHU L, DANIEL J C. 差错控制编码[M]. 晏坚, 何元智, 潘亚汉, 等译. 北京：机械工业出版社出版, 2007

作者简介

杨义先，灾备技术国家工程实验室主任，北京邮电大学教授、博士生导师，信息安全中心主任，首批长江学者特聘教授，首届国家杰出青年基金获得者，中国密码学会副理事长；目前研究方向为网络空间安全、现代密码学和纠错编码等；获得包括国家发明奖和省部级科技进步奖等在内的各类科技奖励20余项，授权发明专利4项，主持和参与多项国家“863”、国家自然科学基金、省部级等科研项目；发表高水平论文500余篇，出版专著及教材20多部。

钮心忻，北京邮电大学计算机学院教授、博士生导师；长期从事网络与信息安全、信号与信息处理等方面的研究工作。

DOI:10.3969/j.issn.1009-6868.2016.02.014
网络出版地址:http://www.cnki.net/kcms/detail/34.1228.TN.20160301.1646.004.html

M-ICT时代融合业务技术发展趋势

Development Trend of Integration Business Technology in the M-ICT Era

陆平/LU Ping
董振江/DONG Zhenjiang
杨勇/YANG Yong
(中兴通讯股份有限公司,深圳 518052)
(ZTE Corporation, Shenzhen 518052, China)

中图分类号:TN393 文献标志码:A 文章编号:1009-6868 (2016) 01-0061-006

摘要: 提出了M-ICT时代融合业务技术的发展趋势:(1)虚拟数据中心已经成为下一代IT基础设施的通用解决方案,其通用功能架构包括软件定义计算、软件定义存储、软件定义网络和软件定义安全等核心子系统,技术方案尚在不断完善之中;(2)容器已大规模应用于互联网,传统的电信域急需应对这种挑战,容器技术也使得平台即服务PaaS产品拓宽了发展空间;(3)基于NFV架构的云化是大势所趋,以用户体验为驱动,基于融合CDN和智能数据分析,提供智能视频服务是竞争力提升的关键;(4)增强现实技术的多媒体视频应用必将极大地改进用户体验,人工智能技术在近几年将会产生更多的应用形态,进一步重塑和重建各行各业。

关键词: 云数据中心;软件定义存储;软件定义安全;容器;平台即服务;互联网电视;人工智能

Abstract: In this paper, we discuss the development trend of integration business technology in the M-ICT era: (1) Virtual data center has become the next-generation IT solution to common infrastructure, which includes software-defined computation, software-defined storage, software-defined network and software-defined security and is becoming more and more mature; (2) containers have been widely applied in Internet applications, and it is very necessary to deal with this challenge in the traditional telecommunication domain, and the container technology also makes Platform as a Service (PaaS) vigorous again; (3) network function virtualization (NFV) is the trend for network development, and intelligent data analysis based on convergent content delivery network (CDN) is the key to enhance the competitiveness of video products as well as driven by user experience; (4) augmented reality-based multimedia video application will greatly improve reality of user experience, while application of artificial intelligence technology can produce more forms of applications , further remodeling and reconstruction of all walks of life.

Keywords: virtual datacenter; software-defined storage; software-defined security; container; PaaS; Internet TV; artificial intelligence

1 云数据中心

传统的互联网数据中心(IDC)与云计算技术的结合已经成为业界关于未来基础网络的共识[1-2]。西班牙电信提出的统一的信息技术网络基础设施架构(UNICA)架构[3],该架构基于云基础设施,深度融合,对外提供标准的虚拟数据中心(VDC)服务,并通过VDC的分级建设,提供就近接入和全网服务。从VDC建设的实施效果上看,我们需要重点考虑4个特性。

(1)可靠高效:海量数据能够催生高性能计算、海量存储及资源动态调度,以及高效便捷的应用生命的周期管理;

(2)跨域弹性:与客户业务发展同步需要深度解耦,软硬件定制,云间数据共享、自动部署、灵活迁移、按需扩展,并支持数据中心快速扩容;

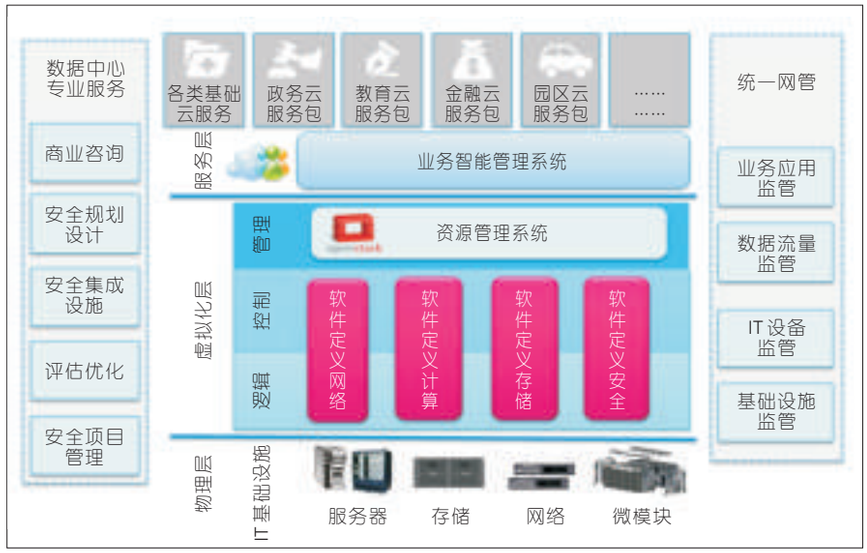
(3)安全可信:从基础设施、运行环境到数据的端到端的安全性,智能地服务感知;

(4)开放共赢:在水平方向上,支持异构的基础设施;在垂直方向上,提供不同类型的服务模式,同时还能对上层应用,提供多领域业务的支持能力。

基于以上要求,我们结合中兴通讯在VDC建设上的丰富经验,提出了云数据中心VDC的相关功能架构,如图1所示。

其中,软件定义网路、软件定义计算、软件定义存储和软件定义安全4个子系统[4-6]是整个软件定义数据中心的核心,通过资源管理系统进行统一的调度管理。软件定义网络(SDN)和虚拟化技术基于内核的虚拟机(KVM)/XEN/VMWare等已经可以较好地支持软件定义网络和软件定义计算等系统的实现,而由于存储的多样性需求和安全的复杂性,软件定义存储和软件定义安全也成为了

收稿时间:2016-02-05
网络出版时间:2016-03-01
基金项目:国家科技重大专项(2013ZX03002004)



▲图1 云数据中心功能架构

目前云数据中心的关键问题。

1.1 软件定义存储

2012年，VMware在其VMworld大会上首次提出软件定义数据中心（SDDC）的概念。经过了几年的发展，软件定义存储的概念和产品形态逐渐明朗，相关的产品开始呈现并被慢慢接受。软件定义存储的核心是提供自助的服务接口，用于分配和管理虚拟存储空间。目前，软件定义存储处于应用的初级阶段，预计到2020年在企业市场的份额将超过50%。

软件定义存储分为控制平面（如图2所示）和数据平面，在整个软件定义存储架构中，控制平面的难度、价值也最大。各厂商也采取了不同的软件定义存储构建方式。我们认为一个理想的控制平面应该具备如下特征：

（1）支持Openstack Cinder、Manila接口，构建开放生态链；

（2）北向提供开放应用程序编程接口（API），完善生态链；

（3）业务驱动存储服务，智能管控存储资源；

（4）基于策略精细化调配存储资源，业务支撑更灵活；

（5）跨异构资源抽象池化，软硬件更新解耦，硬件维护、数据迁移更便捷。

软件定义存储不是一蹴而就的，需要一个过程。随着软件定义存储的数据服务的完善，通过不断努力，增强互操作性、策略驱动异构存储的能力，使其部分的型号或者部分的模块逐渐上升到控制平面。在目前的传统存储阵列市场，我们已经看到这样的趋势。

1.2 软件定义安全

安全问题是数据中心最复杂、最关键的问题之一。在云数据中心的场景下，由于虚拟化技术的引入，物理资源由多租户共享，传统的硬件防火墙、入侵防御系统等安全设备已经无法满足应用的要求，安全产品的虚拟化就成为解决这一问题的必需。在安全设备虚拟化的实现过程中，我们要关注以下关键特征点：

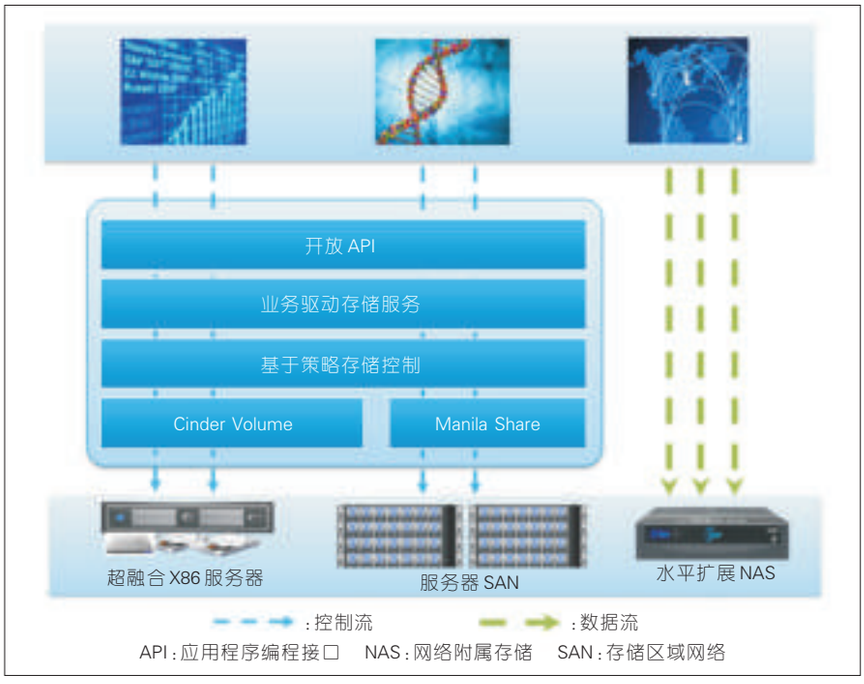
（1）安全控制层引入安全控制器，可基于全局视图，增强云数据中心的安全管理和集中调度能力；

（2）安全控制器开放接口，增强数据中心安全类应用定制化能力，有助于安全应用的生态链建设；

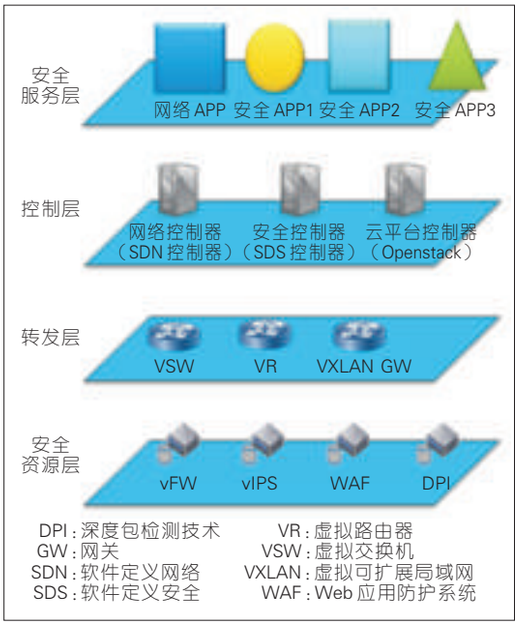
（3）租户可按需订制安全策略，通过控制层下发，实现安全防护的智能化、自动化。

综合中兴通讯在VDC安全产品方面的实践经验，我们提出了云数据中心软件定义安全的基本架构，如图3所示。

其中，安全控制器调用SDN控制



▲图2 软件定义存储的控制平面



▲图3 软件定义安全的基本架构

器接口来控制数据包转发行为（前转、引流、丢弃等等），同时获得全网设备的日志和拓扑信息，用于安全分析；安全控制器与安全资源池对接实现安全策略下发，及告警、日志信息的上报；基于大数据分析和知识库，安全控制器和安全应用实现数据中心数据流量的智能分析和管控，可大大强化数据中心的安全级别。

安全设备虚拟化的还带来了明显的优势：它大大降低了成本，提高了资源利用率与敏捷度，同时通过水平方向的动态扩容，满足大规模、高性能高并发性要求。

2 容器技术及PaaS产品的发展

以Docker[7]为代表的容器技术在2015年大行其道，不仅成为新技术的热点话题，而且在互联网公司中大规模应用，实现了其价值。容器技术的出现，也使得平台即服务（PaaS）云产品的发展再次出现重大的机遇。

2.1 容器技术

Docker是一个基于Linux容器（LXC）技术构建的容器引擎，源代码托管在GitHub上，基于Go语言并遵从Apache 2.0开源协议。Docker让开发者将他们的应用和依赖文件打包并发布成一个可移植的容器，实现应用的快速部署和迁移。

以Docker为代表的容器技术正在引领云计算的下一次变革，Docker改变了云端应用交付和部署的方式，它所提供的轻量级、面向应用的虚拟化运行环境为微服务架构的实现提供了理想的载体，并带动了容器服务的兴起和高速发展。

同虚拟机相比，Docker的优势主要有：镜像占用空间小，启动速度快，资源利用率较高，应用性能也接近物理机。与此同时，由于Docker技术本身的实现机制以及生态环境的发展尚不太完善，在应用中也暴露出了一些问题，如Docker系统在内核、运行环境上的隔离性，Docker网络的性能及可移植性问题，缺乏成熟的集群管理方案等。

Docker的生态环境已经呈现出欣欣向荣的景象，包括Docker社区本身以及一些大的企业，如谷歌和亚马逊等，都在参与Docker相关的生态建设工作。

在标准方面，除Docker外，还有Rocket和Warden等多种容器技术，多个标准组织都在大力推动容器技术的标准化。其中，Linux基金会于2015年6月成立开放容器组织（OCI），旨在围绕容器格式和运行时制定一个开放的工业化标准，该组织一成立便得到了包括谷歌、微软、亚马逊等一系列云计算厂商的支持。2015年7月，谷歌与Linux基金会以及众多行业合作伙伴为一起推动基于容器的云计算的发展，共同建立一个云原生计算基金会（CNCF），它成立的目的是构建“云原生”计算并推动其逐步落地，在微服务、容器和应用动态调度、容器编排工具等方面形成标准。

总体来说，容器技术的生态环境已经建立，尽管在容器管理、编排调度、网络和存储等方面尚不成熟，但是这并不妨碍Docker在互联网中的应用，其轻量级的特性已经在互联网应用中发挥出了较大的作用，对传统电信服务部署方案的演进也影响极大，众多电信服务商提出基于Docker将电信应用解耦形成微服务，以实现高效的电信应用服务管理。

2.2 PaaS平台

PaaS作为云计算的一个模式[8-9]，一直被业界看好。业界PaaS平台实现方案主要包括Cloud Foundry和Openshift，业界许多厂家给予开源的Cloud Foundry实现了自己的PaaS产品。Cloud Foundry是一个开放的框架，支持多种语言、运行时环境、云平台及应用服务，使开发人员能够在几秒钟内进行应用程序的部署和扩展，无需担心任何基础架构的问题，可以部署在多种私有云和共有云上。

Cloud Foundry作为开源PaaS平台，许多公司根据市场发展方向开辟出自己的商业模式，不同的商业模式使Cloud Foundry应用到以下不同的场景中：面向企业级私有云市场，面向企业级公有云市场，面向中小企业/独立软件开发商（ISV）/个人开发者PaaS公有云。

然而，从目前发发展现状来看，公有PaaS云服务运营并不成功。一方面在于竞争的激烈；另一方面在于，基于PaaS的开发模式要求开发者的开发习惯以及应用的架构做出较大的改变，包括应用解耦、服务交互以及开发环境等，包括12因子APP的特性要求等。在面向大企业的私有PaaS建设上，基于Cloudfoundry的PaaS平台在解决企业应用的快速开发、部署、迁移，以及devops的实现，发挥出了一定的价值。

Docker与PaaS的密切融合，将大

力推动PaaS的发展与落地，会成为未来PaaS发展的主流形式。

3 互联网电视及内容分发网络

近几年来，随着移动互联网的快速发展，视频网站纷纷向移动视频延伸发展互联网电视（OTT），各种移动终端的视频客户端发展得如火如荼，逐步覆盖PC、手机/PAD、TV。更有甚者，互联网视频厂家在发展智能电视，电信运营商在大力发展交互式网络电视（IPTV）。随着4G等移动宽带网络的发展，基于IPTV的TV视频逐步向手机/PAD等移动终端发展。国际上的电信运营商，有在专网运营IPTV的，有在公网运营OTT的，总的趋势是IPTV、OTT逐步和数字影像广播（DVB）电视运营商进行商业合纵联合[10]，技术和产品也在逐步融合。

基于融合内容分发网络（CDN）构建智能内容服务管道是电信运营商提升电信视频服务能力的必由之路。融合CDN是在传统CDN基础上实现了多业务的融合承载以及多种终端统一接入技术，提供机顶盒（STB）、PC、移动设备等多种终端上各种业务的内容分发服务，提供面向固定网络和宽带移动网络的全网服务能力，从而可以满足海量用户的个性化需求。融合CDN除了具备基础的内容分发和加速能力，还需要具备有效优化数据网络流量分布，降低数据网络流量瓶颈的能力，并根据业务和用户要求提供不同质量的服务。融合CDN不再是一张孤立的分发网络，需要和其他网元互动，参与业务行为，感知内容变化，响应用户需求，进一步演进为智能分发、业务融合、多终端接入的智能管道。

电信运营商是智能管道的主导方，因而移动CDN的建设是关键。

宽带移动通信技术的快速发展，让CDN延伸到无线网络的需求日益迫切。据统计：2014年首次出现了用户使用移动终端连接互联网时长超过通过电脑访问互联网时长，随着互联网用户的使用习惯逐渐向移动端迁移，良好的用户体验更成为直接影响用户选择。针对移动CDN的建设，我们需要重点考虑以下3个方面：

（1）对于4G等移动网络，可以将CDN下沉至最靠近用户的基站，同时存储热点内容。移动CDN具备高密度数据处理能力和存储容量，对移动终端访问互联网流量进行优化、缓存和加速，保证热点内容分发和高等级用户的服务质量，提升用户的体验。图4给出了移动CDN部署位置建议。

（2）采用码流自适应技术，来优化因为无线信号的不稳定性而导致的视频卡顿、响应慢的问题。采用码流自适应技术，移动CDN可动态检测用户空口资源的变化情况，及时调整发送内容的码率，从而充分保障用户观看视频的流畅性，提升用户体验。

（3）采用移动CDN用户识别会话保存技术来解决移动终端用户在基站之间切换时的业务连续性。这样可以保证用户在文件下载、视频观看时，即使发生了接入基站的切换变化也无须重新下载文件或者中断视频，从根本上屏蔽了移动网络位置改变带来的影响。

总体而言，由于网络架构和运营模式上存在的重大差异，使得目前运营商在视频服务面临互联网厂商的巨大冲击，建议从以下几方面来考虑提升服务竞争力：

（1）构建云化的视频网络是系统架构优化的重点，而网络功能虚拟化（NFV）[11]是必然的选择；

（2）基于融合CDN构建智能的视频管道是充分发挥CDN价值和作用的关键；

（3）构建视频服务能力开放平台，促进视频服务生态链建设，促进多媒体应用创新；

（4）坚持以用户体验为驱动，通过大数据和智能数据分析技术提升用户体验，满足用户个性化要求。

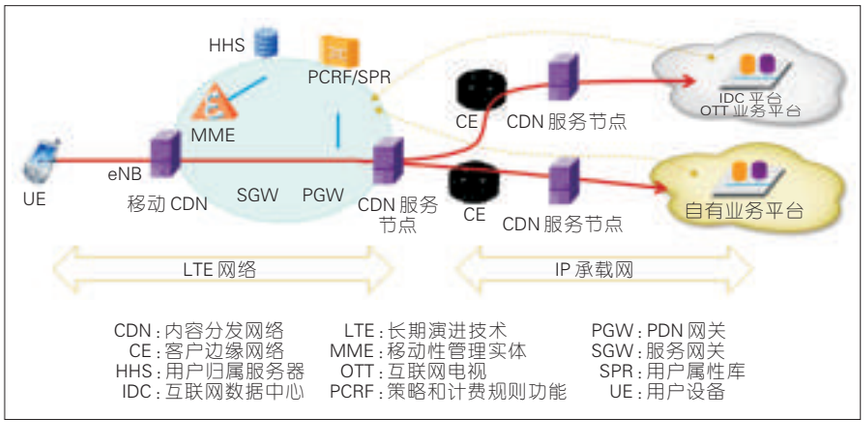
OTT应用的生态环境，仅仅依赖现有的、上下游的设备商参与是无法有效建设起来的，需要业务提供商（SP）一起参与，并需要OTT的服务接口开放标准，这是当前OTT方案急需完善的问题。

4 增强现实与人工智能

4.1 增强现实

2015年微软、苹果、谷歌在增强现实（AR）[12]领域进行大规模的投资和整合，引起了业界的关注。2016年成为AR元年已经成为众多机构对技术的趋势的预测之一，这也是媒体继2010年之后，又一次热捧AR元年的概念。与上一次的基于智能移动手机有所不同，这次的焦点集中在AR Glass技术进展带来的各种行业应用的可能性。

增强现实技术包含了多模传感



▲图4 移动CDN部署位置

技术、智能感知技术、三维注册跟踪技术、实时视频融合渲染等新技术与新手段。其本质是利用各种传感器技术对周围的环境和目标进行智能感知，在识别和理解后，对海量信息进行实时检索，并基于实时采集的视频流将数字世界的多媒体内容（图片、语音、视频、3D模型）进行融合显示，为用户提供超越现实的感官体验，如图5所示。其典型特征表现为虚实融合、三维注册和实时交互。由于获取信息更加便捷，呈现方式更加自然，互动方式更加丰富，为下一代的业务体验和模式带来了更多的想象力。

（1）虚实融合。虚实融合是将数字世界的信息实时叠加到实时采集的视频流上，对现有的场景进行增强渲染。其挑战在于达到以假乱真的沉浸感，核心技术是对各种格式（2D图片、3D模型、视频、音频）的增强信息进行实时渲染。

（2）三维注册。三维注册技术是增强现实技术中最关键、最难有效解决的技术之一。注册跟踪技术可以分为基于发射和接收装置的追踪器、基于手机的惯性传感单元的手机位姿计算、基于计算机视觉的注册跟踪技术、多传感器的融合方法，关键挑战在于能够实现实时、稳定、鲁棒的效果。注册跟踪的算法计算量通常很大，这对于移动终端的计算力和电池都提出了挑战。将算法固化到芯片中是一个降低功耗，解放CPU资源的一种选择。

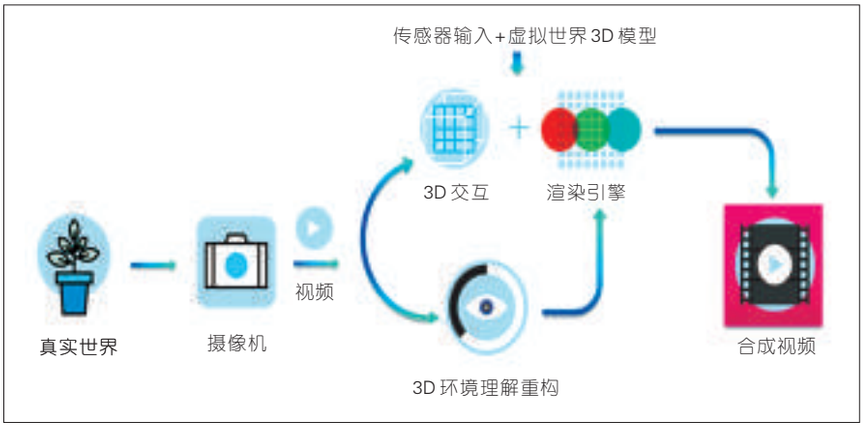
（3）实时交互。增强现实系统中的交互技术是指在真实和虚拟的融合场景中与虚拟目标之间的实时的调整，使得达到虚实融合的目的的技术。在这类场景中，交互的自然性和实时性将成为影响系统体验的关键因素。常用的人机交互技术有手势识别技术、动作识别技术、眼球跟踪技术等。

增强现实技术正在快速地发展中，一些技术演进趋势已经凸显：

（1）增强现实的入口是以视觉为主的、基于多传感器融合的智能感知技术。基于海量数据的大规模视觉检索将成为未来的发展方向，通过海量结构化和非结构的采集和存储技术、大规模的分布式计算技术，更高精度的智能媒体分析技术，可以将复杂的计算上移到云端，实现AR无处不在的用户体验。

（2）将基于终端的注册跟踪算法固化到芯片，在传感器层面进一步地进行融合设计，可以提升感知精度，使得AR成为未来的终端设计的必选属性。

（3）以magic leap为代表的光场重建技术，通过实时重建目标的全息数字信息，直接以视网膜的精度投影到人的眼睛，解决了快速配准、体验舒适的问题，使得AR显示技术走上了一个新的台阶。



▲图5 增强现实系统实现

（4）5G技术演进与发展为实时分享丰富的多媒体互动内容扫清了带宽的限制。

4.2 人工智能

2015年是人工智能从技术、产品到公众认知都有重大突破的一年。几乎所有的科技调查公司，如Gartner，宣布的战略预测都包含人工智能（AI），同时Google、Facebook、微软等都进行人工智能工具和算法的开源。因此，2016年人工智能技术将产生更多的应用形态，改变现有生活方式，提升用户体验[13]。

人工智能的本质是用机器去实现所有目前必须借助人类智慧才能实现的任务，主要分为三大部分任务：运动控制、感知智能和认知智能，实现图6所示的功能。

近年来，基于认知智能的应用层出不穷。从应用场景的角度来看，主要有智能问答、智能客服、个人助理、智能机器人等。在这些应用场景中，涉及的技术包括：语音识别、语音合成、自然语言理解、对话管理和自然语言生成，如图7所示。其中，语音识别和语音合成技术相对已经成熟并得到商用，而自然语言理解、对话管理和自然语言生成是当前最关键的三大核心技术，特征分别为：

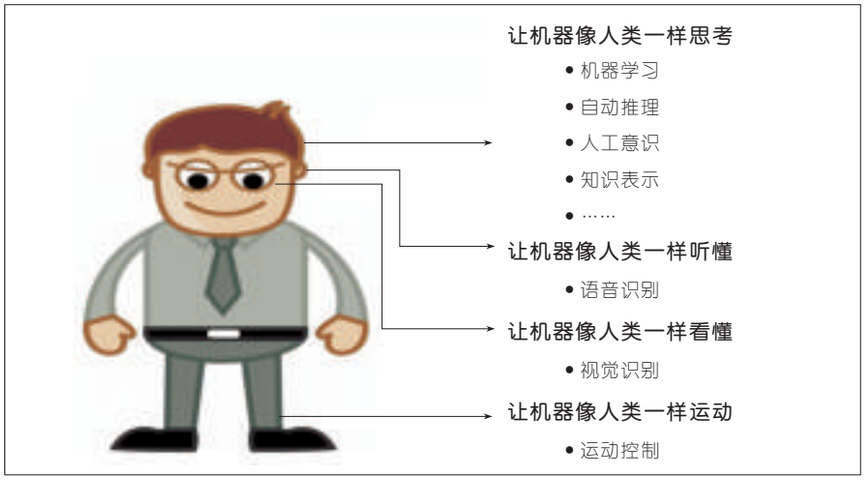
（1）自然语言理解，包括自然语言处理、智能纠错、和情感计算；

（2）对话管理，包括省略恢复、指代消解和问题追问；

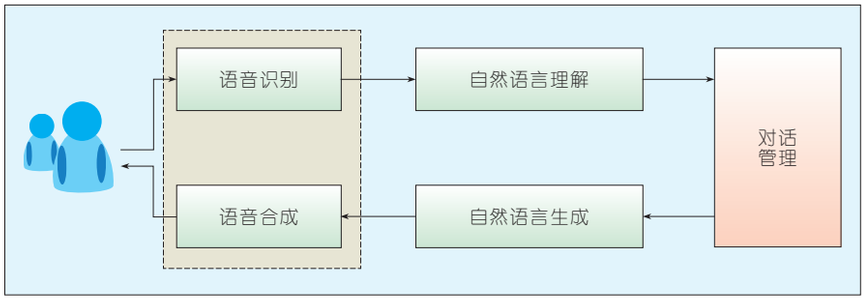
（3）自然语言生成，包括内容检索、相似度的计算以及答案的获取与生成。

随着人工智能技术与认知智能技术的不断成熟，各行业市场的巨大需求量，与互联网+、物联网、智能终端的资源整合，未来5年将产生如下的一些应用形态，这将重塑和重建各行各业，如电信、办公、家庭、医疗、金融等。

（1）基于自然语言处理（NLP）的智能客服替换呼叫中心人工坐席渐



▲图6 人工智能的实现目标



▲图7 认知智能的关键技术

成趋势；

（2）开拓向实体店营业厅，部分替代大堂客服；

（3）智慧家庭人机交互，替代各类遥控操作；

（4）服务机器人，养老助残，逐渐成为家庭必需品。

5 结束语

万物互联的新时代，电信网络和互联网都面临着极大的挑战，构建统一的云计算基础设施，提供以计算、存储、网络和安全为主要功能特征的服务能力，是服务提供商提升自身竞争力的一大趋势，尽管软件定义存储和软件定义安全的实现方案还在不断完善之中。此外，容器技术作为热点技术之一，尽管存在一些问题，但是容器技术在轻量级的应用部署、迁移和运维管理方面所表现出来的巨大优势，已经成为一个较好的IT新技术实践。对于电信网络来说，NFV是一个概念架构，其完善和落地还需要一个过程。以用户体验为驱动，基于融合CDN和智能数据分析，结合增强现实技术的多媒体视频应用必将极大地改进用户体验。人工智能技术在近几年将会产生更多的应用形态，进一步重塑和重建各行各业。

参考文献

[1] 石屹嵘, 龚德志. 基于云模式的新一代IDC系统框架研究[J]. 电信科学. 2010(06):18–24
[2] 魏进武，张云勇，陈清金. 云计算推动IDC向VDC转型的研究[J]，电信科学，2010(11):34–38
[3] Telecompaper. Telefonica Unveils UNICA Virtualization Infrastructure[EB/OL].[2013–10–23]. http://www.telecompaper.com/news/telefonica–unveils–unica–virtualization–infrastructure––998081
[4] 徐宁宁. 云平台中软件定义存储资源[J]. 通讯世界, 2015(11):117–118
[5] SHIN S, PORRAS P, YEGNESWARAN V, et al. Fresco: Modular Composable Security Services for Software–Defined Networks[C]// Internet Society NDSS 2013. USA: ACM, 2013: 223–228
[6] 刘文懋，裘晓峰，陈鹏程，等. 面向SDN环境的软件定义安全架构[J]. 计算机科学与探索, 2015(01): 63–70
[7] 张建，谢天钧. 基于Docker的平台即服务架构研究[J]. 信息技术与信息化, 2014(10): 131–134
[8] GARCIA G, ANDRES, ALFONSON D, et al. Overview of Current Commercial PaaS Platforms [C]// Proceedings of the 6th International Conference on Software and Database Technologies. USA: ACM, 2011: 231–238
[9] STEFAN W, EDDY T, WOUTER J. Comparing PaaS Offerings in Light of SaaS Development [J]. Computing, 2014 (8): 669–724
[9] 施唯佳，蒋力，贾立鼎. OTT TV和IPTV的技术比较分析. 电信科学, 2014(5): 14–19
[10] Network Functions Virtualisation(NFV). Network Operator Perspectives on Industry Progress, ETSI [EB/OL]. [2015–02–10]. http://portal.etsi.or/NFV/NFV_White_Paper2.pdf
[11] 林倞，杨珂，王涌天，等. 移动增强现实系统的关键技术研究[J]. 中国图象图形学报, 2009, 19(03): 560–564
[12] 宋章军. 服务机器人的研究现状与发展趋势[J]. 集成技术, 2012(3): 1–9
[13] GARTER. The Top Ten Technology Trends for 2016 [EB/OL]. [2015–10–09]. http://www.gartner.com/newsroom/id/3143521

作者简介

陆平，中兴通讯云计算及IT研究院院长，北京邮电大学和南京邮电大学兼职教授；主要研究方向为云计算与大数据、新媒体、移动互联网等技术；主持国家、省部级基金项目10多项；获得省部级科技进步奖多项；发表学术论文15篇，出版专著2部。

董振江，中兴通讯战略与技术专家委员会业务专家组组长、云计算及IT研究院副院长，中国人工智能学会常务理事；主要研究方向为云计算与大数据、新媒体、移动互联网等技术；主持基金项目10余项；获得国家科技进步二等奖，电子学会科技进步一等奖，省科技进步奖等多项；发表学术论文10余篇，出版专著1部。

杨勇，中兴通讯云计算及IT研究院总工程师，中兴通讯技术专家委员会委员；主要研究方向为Web技术、多媒体处理技术、业务能力开放等，长期从事电信增值业务及移动互联网相关的研发工作；先后获得多项省部级科技进步奖以及电子学会科技进步一等奖1项；已发表学术论文18篇，拥有授权专利15项，出版专著1部。

《中兴通讯技术》杂志(双月刊)投稿须知

一、杂志定位

《中兴通讯技术》杂志为通信技术类学术期刊。通过介绍、探讨通信热点技术,以展现通信技术最新发展动态,并促进产学研合作,发掘和培养优秀人才,为振兴民族通信产业做贡献。

二、稿件基本要求

1. 投稿约定

(1)作者需登录《中兴通讯技术》投稿平台:www.zte.com.cn/paper,并上传稿件。第一次投稿需完成新用户注册。

(2)编辑部将按照审稿流程聘请专家审稿,并根据审稿意见,公平、公正地录用稿件。审稿过程需要1个月左右。

2. 内容和格式要求

(1)稿件须具有创新性、学术性、规范性和可读性。

(2)稿件需采用WORD文档格式。

(3)稿件篇幅一般不超过6 000字(包括文、图),内容包括:中、英文题名,作者姓名及汉语拼音,作者中、英文单位,中文摘要、关键词(3~8个),英文摘要、关键词,正文,参考文献,作者简介。

(4)中文题名一般不超过20个汉字,中、英文题名含义应一致。

(5)摘要尽量写成报道性摘要,包括研究的目的、方法、结果/结论,150~200字为宜。摘要应具有独立性和自明性。中英文摘要应一致。

(6)文稿中的量和单位应符合国家标准。外文字母的正斜体、大小写等须写清楚,上下角的字母、数据和符号的位置皆应明显区别。

(7)图、表力求少而精(以8幅为上限),应随文出现,切忌与文字重复。图、表应保持自明性,图中缩略词和英文均要在图中加中文解释。表应采用三线表,表中缩略词和英文均要在表内加中文解释。

(8)参考文献以20条左右为宜,不允许公开发表的资料不应列入。所有文献必须在正文中引用,文献序号按其在文中出现的先后次序编排。常用参考文献的书写格式为:

- 期刊[序号]作者. 题名[J]. 刊名, 出版年, 卷号(期号): 引文页码. 数字对象唯一标识符
- 书籍[序号]作者. 书名[M]. 出版地: 出版者, 出版年: 引文页码. 数字对象唯一标识符
- 论文集中析出文献[序号]作者. 题名[C]//论文集编者. 论文集名(会议名). 出版地:出版者, 出版年(开会年): 引文页码. 数字对象唯一标识符
- 学位论文[序号]作者. 题名[D].保存地点:保存单位,授予年. 数字对象唯一标识符
- 专利[序号]专利所有者. 专利题名:专利号[P]. 出版日期. 数字对象唯一标识符
- 国际、国家标准[序号] 标准名称: 标准编号[S]. 出版地: 出版者, 出版年. 数字对象唯一标识符

(9)作者超过3人时,可以感谢形式在文中提及。作者简介包括:姓名、工作单位、职务或职称、学历、毕业于何校、现从事的工作、专业特长、科研成果、已发表的论文数量等。

(10)提供正面、免冠、彩色标准照片一张,最好采用JPG格式(文件大小超过100 kB)。

(11)应标注出研究课题的资助基金或资助项目名称及编号。

(12)提供联系方式,如:通信地址、电话(含手机)、Email等。

3. 其他事项

(1)请勿一稿两投。凡在2个月(自来稿之日算起)以内未接到录用通知者,可致电编辑部询问。

(2)为了促进信息传播,加强学术交流,在论文发表后,本刊享有文章的转摘权(包括英文版、电子版、网络版)。作者获得的稿费包括转摘酬金。如作者不同意转摘,请在投稿时说明。

编辑部地址:安徽省合肥市金寨路329号国轩凯旋大厦1201室,邮政编码:230061
联系电话:0551-65533356,联系邮箱:magazine@zte.com.cn
本刊只接受在线投稿,欢迎访问本刊投稿平台:www.zte.com.cn/paper

中兴通讯技术
ZHONGXING TONGXUN JISHU

双月刊 1995年创刊 总第127期
2016年4月 第22卷第2期

主管：安徽省科学技术厅
主办：安徽省科学技术情报研究所
中兴通讯股份有限公司
编辑：《中兴通讯技术》编辑部

《中兴通讯技术》编辑部
地址：合肥市金寨路329号凯旋大厦12楼
邮编：230061
网址：www.zte.com.cn/magazine
投稿平台：www.zte.com.cn/paper
电子信箱：magazine@zte.com.cn
电话：(0551)65533356
传真：(0551)65850139

出版、发行：中兴通讯技术杂志社
发行范围：全球发行
印刷：合肥添彩包装有限公司
出版日期：2016年4月10日
刊号：ISSN 1009-6868
CN 34-1228/TN
广告经营许可证：皖合工商广字0058
定价：每册20.00元，全年120.00元