ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.5, May 2009, pp.1337−1348
doi: 10.3724/SP.J.1001.2009.03493
© by *Institute of Software, the Chinese Academy of Sciences*. All rights reserved.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

云计算:系统实例与研究现状*

陈 康[1,2+], 郑纬民[1,2]

[1](清华大学 清华信息科学与技术国家实验室(筹),北京 100084)

[2](清华大学 计算机科学与技术系,北京 100084)

Cloud Computing: System Instances and Current Research

CHEN Kang[1,2+], ZHENG Wei-Min[1,2]

[1](Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China)

[2](Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

+ Corresponding author: E-mail: ck99@mails.tsinghua.edu.cn

Chen K, Zheng WM. Cloud computing: System instances and current research. *Journal of Software*, 2009, 20(5):1337−1348. http://www.jos.org.cn/1000-9825/3493.htm

Abstract: This paper surveys the current technologies adopted in cloud computing as well as the systems in enterprises. Cloud computing can be viewed from two different aspects. One is about the cloud infrastructure which is the building block for the up layer cloud application. The other is of course the cloud application. This paper focuses on the cloud infrastructure including the systems and current research. Some attractive cloud applications are also discussed. Cloud computing infrastructure has three distinct characteristics. First, the infrastructure is built on top of large scale clusters which contain a large number of cheap PC servers. Second, the applications are co-designed with the fundamental infrastructure that the computing resources can be maximally utilized. Third, the reliability of the whole system is achieved by software building on top of redundant hardware instead of mere hardware. All these technologies are for the two important goals for distributed system: high scalability and high availability. Scalability means that the cloud infrastructure can be expanded to very large scale even to thousands of nodes. Availability means that the services are available even when quite a number of nodes fail. From this paper, readers will capture the current status of cloud computing as well as its future trends.

Key words: cloud computing; distributed infrastructure; distributed paradigm

摘 要: 针对云计算这样一个范畴综述了当前云计算所采用的技术,剖析其背后的技术含义以及当前云计算参与企业所采用的云计算实现方案.云计算包含两个方面的含义:一方面是底层构建的云计算平台基础设施,是用来构造上层应用程序的基础;另外一方面是构建在这个基础平台之上的云计算应用程序.主要是针对云计算的基础架构的研究与实现状况给出综述,对于云计算的应用也有所涉及.云计算有 3 个最基本的特征:第 1 个是基础设施架构在大规模的廉价服务器集群之上;第二是应用程序与底层服务协作开发,最大限度地利用资源;第 3 个是通过多个廉价服

* Supported by the National Natural Science Foundation of China under Grant No.90718040 (国家自然科学基金); the National Basic Research Program of China under Grant No.2007CB310900 (国家重点基础研究发展计划(973)); the National High-Tech Research and Development Plan of China under Grant No.2008AA01Z112 (国家高技术研究发展计划(863))

Received 2008-06-13; Accepted 2008-10-09

© 中国科学院软件研究所 http://www.jos.org.cn

务器之间的冗余,通过软件获得高可用性.云计算达到了两个分布式计算的重要目标:可扩展性和高可用性.可扩展性表达了云计算能够无缝地扩展到大规模的集群之上,甚至包含数千个节点同时处理.高可用性代表了云计算能够容忍节点的错误,甚至有很大一部分节点发生失效也不会影响程序的正确运行.通过此文可以了解云计算的当前发展状况以及未来的研究趋势.

关键词: 云计算;分布式基础架构;分布系统范例

中图法分类号: TP393 **文献标识码:** A

IBM 公司于 2007 年底宣布了云计算计划[1],云计算的概念出现在大众面前.为了更好地了解云计算这个名字后面的本质含义,本文拟通过具体分析工业界推出的几个被广泛接受的云计算实现,以及学术界针对当前大规模数据处理上所作的努力,为读者剖析云计算背后所采用的具体技术.首先看一下在 IBM 的技术白皮书“Cloud Computing”[2]中的云计算定义:

“云计算一词用来同时描述一个系统平台或者一种类型的应用程序.一个云计算的平台按需进行动态地部署(provision)、配置(configuration)、重新配置(reconfigure)以及取消服务(deprovision)等.在云计算平台中的服务器可以是物理的服务器或者虚拟的服务器.高级的计算云通常包含一些其他的计算资源,例如存储区域网络(SANs),网络设备,防火墙以及其他安全设备等.云计算在描述应用方面,它描述了一种可以通过互联网 Internet 进行访问的可扩展的应用程序.“云应用”使用大规模的数据中心以及功能强劲的服务器来运行网络应用程序与网络服务.任何一个用户可以通过合适的互联网接入设备以及一个标准的浏览器就能够访问一个云计算应用程序.”

上述定义给出了云计算两个方面的含义:一方面描述了基础设施,用来构造应用程序,其地位相当于 PC 机上的操作系统;另一方面描述了建立在这种基础设施之上的云计算应用.在与网格计算的比较上,网格程序是将一个大任务分解成很多小任务并行运行在不同的集群以及服务器上,注重科学计算应用程序的运行.而云计算是一个具有更广泛含义的计算平台,能够支持非网格的应用,例如支持网络服务程序中的前台网络服务器、应用服务器、数据库服务器三层应用程序架构模式,以及支持当前 Web 2.0 模式的网络应用程序.云计算是能够提供动态资源池、虚拟化和高可用性的下一代计算平台.现有的云计算实现使用的技术体现了以下 3 个方面的特征:

1) 硬件基础设施架构在大规模的廉价服务器集群之上.与传统的性能强劲但价格昂贵的大型机不同,云计算的基础架构大量使用了廉价的服务器集群,特别是 x86 架构的服务器.节点之间的互联网络一般也使用普遍的千兆以太网.

2) 应用程序与底层服务协作开发,最大限度地利用资源.传统的应用程序建立在完善的基础结构,如操作系统之上,利用底层提供的服务来构造应用.而云计算为了更好地利用资源,采用了底层结构与上层应用共同设计的方法来完善应用程序的构建.

3) 通过多个廉价服务器之间的冗余,使用软件获得高可用性.由于使用了廉价的服务器集群,节点的失效将不可避免,并且会有节点同时失效的问题.为此,在软件设计上需要考虑节点之间的容错问题,使用冗余的节点获得高可用性.

通过上面的技术手段,云计算达到了两个分布式计算的重要目标:可扩展性和高可用性.可扩展性表达了云计算能够无缝地扩展到大规模的集群之上,甚至包含数千个节点同时处理.高可用性代表了云计算能够容忍节点的错误,甚至有很大一部分节点发生失效也不会影响程序的正确运行.学术界其实很早就展开了针对云计算方便用户使用方面的研究,很多学术上的研究成果远早于云计算概念提出来之前就已得出.我国的计算机研究人员远在“云计算”这个名词提出之前就已有透明计算[3,4]的构思.透明计算体现了云计算的特征,即资源池动态的构建、虚拟化、用户透明等.清华大学张尧学教授(中国工程院院士)早在 1998 年就开始从事透明计算系统和理论的研究.工业界很多公司则分别提出了自己针对云计算的理解,用不同的技术来实现上述云计算的目标.本文主要研究工业界 3 个具体的云计算实例,具体包括 Google 的云计算平台以及云计算的网络应用程序、IBM

© 中国科学院软件研究所 http://www.jos.org.cn

公司的“蓝云”平台产品以及 Amazon 公司的弹性计算云.此外,对云计算的学术研究现状也进行了调研与分析.

1 清华大学透明计算平台

张尧学教授领导的研究小组从 1998 年开始就从事透明计算系统和理论的研究,到 2004 年前后正式提出,并不断完善了透明计算的概念和相关理论[3,4].随着硬件、软件以及网络技术的发展,计算模式从大型机的方式逐渐过渡到微型个人计算机的方式,并且近年来过渡到普适计算上.但是用户仍然很难获得异构类型的操作系统以及应用程序,在轻量级的设备上很难获得完善的服务.而在透明计算中,用户无须感知计算具体所在位置以及操作系统、中间件、应用等技术细节,只需要根据自己的需求,通过连通在网络之上的各种设备选取相应的服务.

图 1 显示了透明计算平台的 3 个重要组成部分.用户的显示界面是前端的轻权设备,包括各种个人计算机、笔记本、PDA、智能手机等,被统称为透明客户端.透明客户端可以是没有安装任何软件的裸机,也可以是装有部分核心软件平台的轻巧性终端.中间的透明网络则整合了各种有线和无线网络传输设施,主要用来在各种透明客户端与后台服务器之间完成数据的传递,而用户无须意识到网络的存在.与云计算基础服务设施构想一致,透明服务器不排斥任何一种可能的服务提供方式,既可通过当前流行的 PC 服务器集群方式来构建透明服务器集群,也可使用大型服务器等.

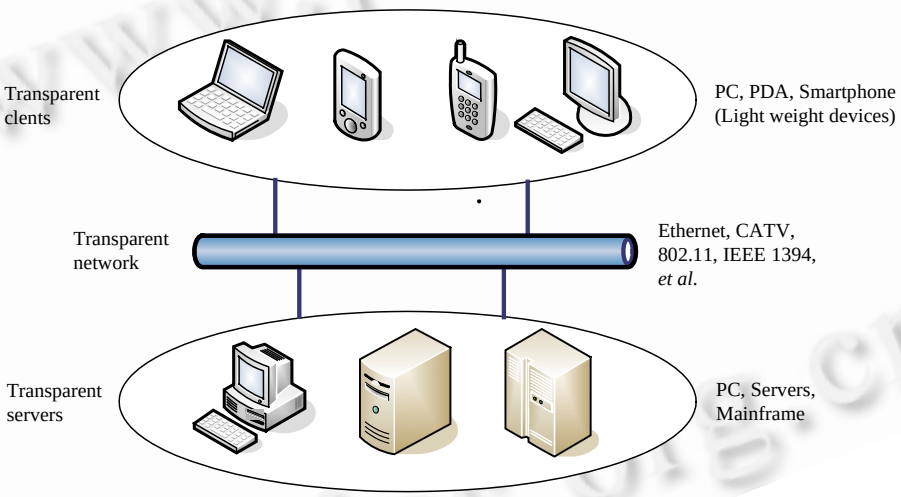

Fig.1 Architecture of transparent computing system

图 1 透明计算系统的组成结构

当前透明计算平台已经达到了平台异构的目的,能够支持 Linux 以及 Windows 操作系统的运行.用户具有很大的灵活性,能够自主选择自己所需要的操作系统运行在透明客户端上.透明服务器使用了流行的PC服务器集群的方式,预先存储了各种不同的操作平台,包括操作系统的运行环境、应用程序以及相应的数据.每个客户端从透明服务器上获取并建立整个运行环境以满足用户对于不同操作环境的需求.由于用户之间的数据相互隔离,因此服务器集群可以选取用户相对独立的方式进行存储,使得整个系统能够扩展到很大的规模.在服务器集群之上进行相应的冗余出错处理,很好地保护了每个用户的透明计算数据安全性.

2 Google的云计算平台

Google 公司有一套专属的云计算平台[5],这个平台先是为 Google 最重要的搜索应用提供服务[6],现在已经扩展到其他应用程序.Google 的云计算基础架构模式包括 4 个相互独立又紧密结合在一起的系统:Google File System[7]分布式文件系统,针对 Google 应用程序的特点提出的 MapReduce 编程模式[8],分布式的锁机制

© 中国科学院软件研究所 http://www.jos.org.cn

Chubby[9]以及 Google 开发的模型简化的大规模分布式数据库 BigTable[10].

• Google File System 文件系统(GFS)

除了性能,可伸缩性、可靠性以及可用性以外,GFS 设计还受到 Google 应用负载和技术环境的影响[7].体现在 4 个方面:1) 充分考虑到大量节点的失效问题,需要通过软件将容错以及自动恢复功能集成在系统中;2) 构造特殊的文件系统参数,文件通常大小以 G 字节计,并包含大量小文件;3) 充分考虑应用的特性,增加文件追加操作,优化顺序读写速度;4) 文件系统的某些具体操作不再透明,需要应用程序的协助完成.

图 2 给出了 Google File System 的系统架构.如图 2 所示,一个 GFS 集群包含一个主服务器和多个块服务器,被多个客户端访问.大文件被分割成固定尺寸的块,块服务器把块作为 Linux 文件保存在本地硬盘上,并根据指定的块句柄和字节范围来读写块数据.为了保证可靠性,每个块被缺省保存 3 个备份.主服务器管理文件系统所有的元数据,包括名字空间、访问控制、文件到块的映射、块物理位置等相关信息.通过服务器端和客户端的联合设计,GFS 对应用支持达到性能与可用性最优.GFS 是为 Google 应用程序本身而设计的,在内部部署了许多 GFS 集群.有的集群拥有超过 1 000 个存储节点,超过 300T 的硬盘空间,被不同机器上的数百个客户端连续不断地频繁访问着.

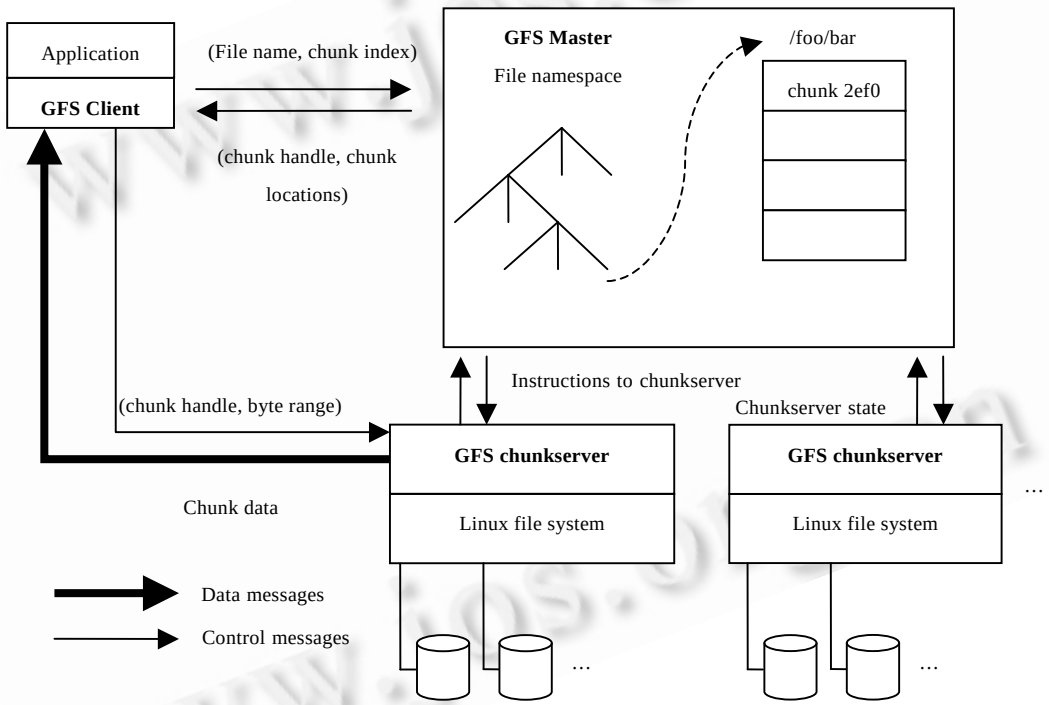

Fig.2 Google File System architecture

图 2 Google File System 的系统架构

• MapReduce 分布式编程环境

Google 构造 MapReduce 编程规范[8,11,12]来简化分布式系统的编程.应用程序编写人员只需将精力放在应用程序本身,而关于集群的处理问题,包括可靠性和可扩展性,则交由平台来处理.MapReduce 通过“Map(映射)”和“Reduce(化简)”这样两个简单的概念来构成运算基本单元,用户只需提供自己的 Map 函数以及 Reduce 函数即可并行处理海量数据.为了进一步理解 MapReduce 的编程方式,下面给出一个基于 MapReduce 编程方式的程序伪代码.程序功能是统计文本中所有单词出现的次数.

在图 3 所示的 map 函数中,用户的程序将文本中所有出现的单词都按照出现计数 1(以 Key-Value 对的形式)发射到 MapReduce 给出的一个中间临时空间中.通过 MapReduce 中间处理过程,将所有相同的单词产生的中间

© 中国科学院软件研究所 http://www.jos.org.cn

结果分配到同样一个 Reduce 函数中.而每一个 Reduce 函数则只需把计数累加在一起即可获得最后结果.

图 4 给出了 MapReduce 执行过程,分为 Map 阶段以及 Reduce 两个阶段,都使用了集群中的所有节点.在两个阶段之间还有一个中间的分类阶段,即将中间结果包含相同的 key 的中间结果交给同一个 Reduce 函数去执行.

```
map(String input_key, String input_value):
  // input_key: document name
  // input_value: document contents
  for each word w in input_value:
    EmitIntermediate(w,"1");

reduce(String output_key, Interator intermediate_values):
  // output_key: a word
  // output_values: a list of counts
  int result = 0;
  for each v in intermediate_values:
    result+=ParseInt(v);
  Emit(AsString(result));
```

Fig.3 WordCount program using MapReduce framework

图 3 基于 MapReduce 框架的单词统计程序举例

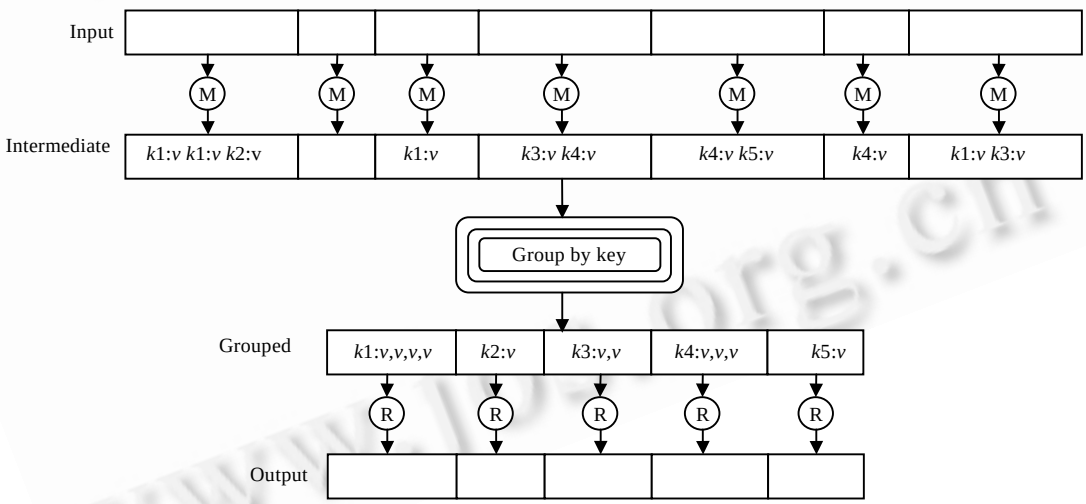

Fig.4 Execution steps of MapReduce processing programs (M stands for the exectuion of Map while R stands for the execution of Reduce)

图 4 MapReduce 处理程序的执行过程(M 代表 Map 函数的执行,R 代表 Reduce 函数的执行)

- 分布式的大规模数据库管理系统 BigTable

由于一部分 Google 应用程序需要处理大量的格式化以及半格式化数据, Google 构建了弱一致性要求的大规模数据库系统 BigTable[10].BigTable 的应用包括 Search History,Maps,Orkut,RSS 阅读器等.

图 5 给出了在 BigTable 模型中给出的数据模型.数据模型包括行列以及相应的时间戳,所有的数据都存放在表格单元中.BigTable 的内容按照行来划分,将多个行组成一个小表,保存到某一个服务器节点中.这个小表就被称为 Tablet.图 6 是整个 BigTable 的存储服务体系结构.

© 中国科学院软件研究所 http://www.jos.org.cn

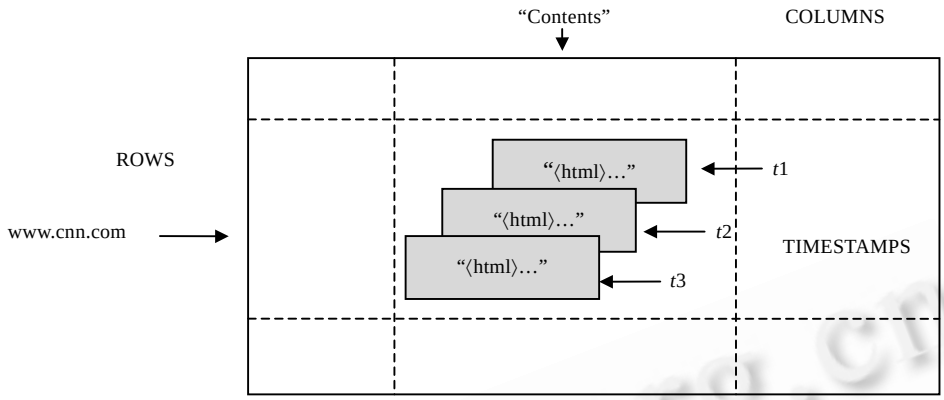

Fig.5 Data model of Google BigTable

图 5 Google BigTable 的数据模型

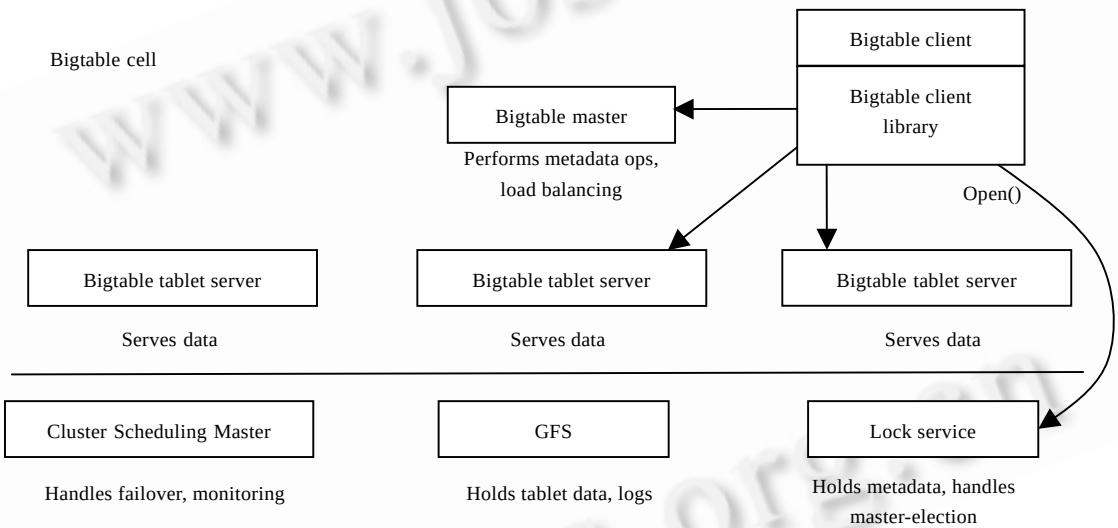

Fig.6 Organization of BigTable System

图 6 BigTable 系统的组织结构

与前述的系统类似,BigTable 也是客户端和服务器端的联合设计,使得性能能够最大程度地符合应用的需求.BigTable 系统依赖于集群系统的底层结构,一个是分布式的集群任务调度器,一个是前述的 Google 文件系统,还有一个分布式的锁服务 Chubby.Chubby 是一个非常鲁棒的粗粒度锁,BigTable 使用 Chubby 来保存根数据表格的指针,即用户可以首先从 Chubby 锁服务器中获得根表的位置,进而对数据进行访问.BigTable 使用一台服务器作为主服务器,用来保存和操作元数据.主服务器除了管理元数据之外,还负责对 tablet 服务器(即一般意义上的数据服务器)进行远程管理与负载调配.客户端通过编程接口与主服务器进行元数据通信,与 tablet 服务器进行数据通信.

以上是 Google 内部云计算基础平台的 3 个主要部分.Google 还构建其他云计算组件,包括一个领域描述语言以及分布式锁服务机制等.文献[5]描述了 Google 内部构建集群的方法,即使用了大量的 x86 服务器集群来构建整个计算的硬件基础设施.Sawzall[13]是一种建立在 MapReduce 基础上的领域语言,专门用于大规模的信息处理.Chubby[9]是前述的一个高可用、分布式数据锁服务.当有机器失效时,Chubby 使用 Paxos 算法[14]来保证备份

© 中国科学院软件研究所 http://www.jos.org.cn

的一致性.Chubby 的小型分布式文件系统的每一个单元都可以用来提供锁服务.

3 IBM“蓝云”计算平台

IBM 的“蓝云”计算平台是一套软、硬件平台,将 Internet 上使用的技术扩展到企业平台上,使得数据中心使用类似于互联网的计算环境.“蓝云”大量使用了 IBM 先进的大规模计算技术,结合了 IBM 自身的软、硬件系统以及服务技术,支持开放标准与开放源代码软件.“蓝云”基于 IBM Almaden 研究中心的云基础架构,采用了 Xen[15,16]和 PowerVM[17]虚拟化软件,Linux 操作系统映像以及 Hadoop[18]软件(Google File System 以及 MapReduce 的开源实现).IBM 已经正式推出了基于 x86 芯片服务器系统的“蓝云”产品.

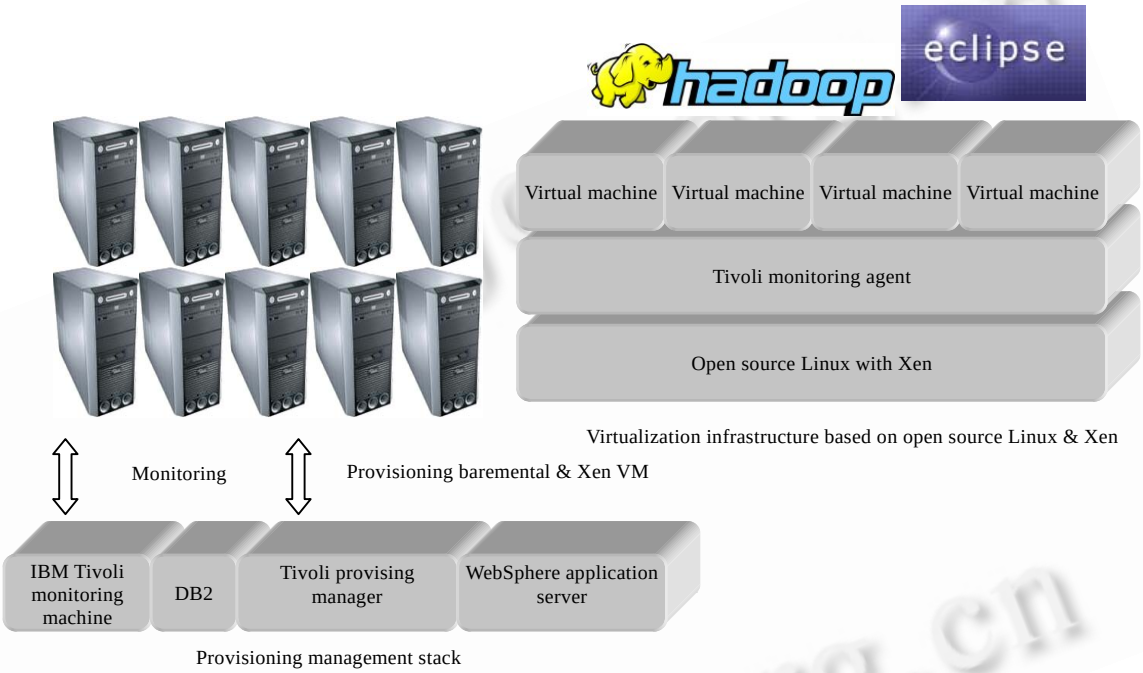

Fig.7 Architecture of IBM “Blue Cloud”

图 7 IBM“蓝云”产品架构

由图可知,“蓝云”计算平台由一个数据中心、IBM Tivoli 部署管理软件(Tivoli provisioning manager)、IBM Tivoli 监控软件(IBM Tivoli monitoring)、IBM WebSphere 应用服务器、IBM DB2 数据库以及一些开源信息处理软件和开源虚拟化软件共同组成.“蓝云”的硬件平台环境与一般的 x86 服务器集群类似,使用刀片的方式增加了计算密度.“蓝云”软件平台的特点主要体现在虚拟机以及对于大规模数据处理软件 Apache Hadoop 的使用上.Hadoop 是开源版本的 Google File System 软件和 MapReduce 编程规范.

- “蓝云”计算平台中的虚拟化技术

“蓝云”软件的一个重要特点是虚拟化技术的使用[19].虚拟化的方式在“蓝云”中有两个级别,一个是在硬件级别上实现虚拟化,另一个是通过开源软件实现虚拟化.硬件级别的虚拟化可以使用 IBM p 系列的服务器,获得硬件的逻辑分区 LPAR(logic partition).逻辑分区的 CPU 资源能够通过 IBM Enterprise Workload Manager 来管理.通过这样的方式加上在实际使用过程中的资源分配策略,能够使相应的资源合理地分配到各个逻辑分区.p 系列系统的逻辑分区最小粒度是 1/10 颗中央处理器(CPU).Xen 则是软件级别上的虚拟化,能够在 Linux 基础上运行另外一个操作系统.

虚拟机是一类特殊的软件,能够完全模拟硬件的执行,运行不经修改的完整的操作系统,保留了一整套运行

© 中国科学院软件研究所 http://www.jos.org.cn

环境语义.通过虚拟机的方式,在云计算平台上获得如下一些优点:1) 云计算的管理平台能够动态地将计算平台定位到所需要的物理节点上[20,21],而无须停止运行在虚拟机平台上的应用程序,进程迁移方法更加灵活;2) 降低集群电能消耗,将多个负载不是很重的虚拟机计算节点合并到同一个物理节点上,从而能够关闭空闲的物理节点,达到节约电能的目的;3) 通过虚拟机在不同物理节点上的动态迁移,迁移了整体的虚拟运行环境,能够获得与应用无关的负载平衡性能;4) 在部署上也更加灵活,即可以将虚拟机直接部署到物理计算平台上,而虚拟机本身就包括了相应的操作系统以及相应的应用软件,直接将大量的虚拟机映像复制到对应的物理节点即可.

• “蓝云”计算平台中的存储体系结构

“蓝云”计算平台中的存储体系结构对于云计算来说也是非常重要的,无论是操作系统、服务程序还是用户的应用程序的数据都保存在存储体系中.“蓝云”存储体系结构包含类似于 Google File System 的集群文件系统以及基于块设备方式的存储区域网络 SAN.

在设计云计算平台的存储体系结构时,不仅仅是需要考虑存储容量的问题.实际上,随着硬盘容量的不断扩充以及硬盘价格的不断下降,可以通过组合多个磁盘获得很大的磁盘容量.相对于磁盘的容量,在云计算平台的存储中,磁盘数据的读写速度是一个更重要的问题,因此需要对多个磁盘进行同时读写.这种方式要求将数据分配到多个节点的多个磁盘当中.为达到这一目的,存储技术有两个选择,一个是使用类似于Google File System的集群文件系统,另一个是基于块设备的存储区域网络 SAN 系统.

在蓝云计算平台上,SAN 系统与分布式文件系统(例如 Google File System)并不是相互对立的系统,SAN 提供的是块设备接口,需要在此基础上构建文件系统,才能被上层应用程序所使用.而 Google File System 正好是一个分布式的文件系统,能够建立在 SAN 之上.两者都能提供可靠性、可扩展性,至于如何使用还需要由建立在云计算平台上的应用程序来决定,这也体现了计算平台与上层应用相互协作的关系.

4 Amazon的弹性计算云

Amazon 是互联网上最大的在线零售商,每天负担着大量的网络交易,同时 Amazon 也为独立软件开发人员以及开发商提供云计算服务平台.Amazon 将他们的云计算平台称为弹性计算云(elastic compute cloud,简称 EC2)[22],是最早提供远程云计算平台服务的公司.Amazon 将自己的弹性计算云建立在公司内部的大规模集群计算的平台上,而用户可以通过弹性计算云的网络界面去操作在云计算平台上运行的各个实例(instance).用户使用实例的付费方式由用户的使用状况决定,即用户只需为自己所使用的计算平台实例付费,运行结束后计费也随之结束.这里所说的实例即是由用户控制的完整的虚拟机运行实例.通过这种方式,用户不必自己去建立云计算平台,节省了设备与维护费用.

Amazon 的弹性计算云由名为 Amazon 网络服务(Amazon Web services)的现有平台发展而来.2006 年 3 月,Amazon 发布了简单存储服务(simple storage service,简称 S3),用户使用 SOAP 协议存放和获取自己的数据对象.在 2007 年 7 月,Amazon 公司推出了简单队列服务(simple queue service,简称 SQS),这项服务能够使得托管虚拟主机之间发送的消息,支持分布式程序之间的数据传递,无须考虑消息丢失的问题.Amazon 又继续提供了 EBS(elastic block storage)服务,为用户提供块级别的存储接口.在提供这些基础设施的同时,Amazon 公司开发了弹性计算云 EC2 系统,开放给外部开发人员使用.图 8 给出了一个 EC2 系统的使用模式.

从图 8 中可以看出,弹性计算云用户使用客户端通过 SOAP over HTTPS 协议与 Amazon 弹性计算云内部的实例进行交互.这样,弹性计算云平台为用户或者开发人员提供了一个虚拟的集群环境,在用户具有充分灵活性的同时,也减轻了云计算平台拥有者(Amazon 公司)的管理负担.弹性计算云中的每一个实例代表一个运行中的虚拟机.用户对自己的虚拟机具有完整的访问权限,包括针对此虚拟机操作系统的管理员权限.虚拟机的收费也是根据虚拟机的能力进行费用计算的,实际上,用户租用的是虚拟的计算能力.

总而言之,Amazon 通过提供弹性计算云,满足了小规模软件开发人员对集群系统的需求,减小了维护负担.其收费方式相对简单明了:用户使用多少资源,只需为这一部分资源付费即可.

© 中国科学院软件研究所 http://www.jos.org.cn

为了弹性计算云的进一步发展,Amazon 规划了如何在云计算平台基础上帮助用户开发网络化的应用程序.除了网络零售业务以外,云计算也是 Amazon 公司的核心价值所在,必然会在弹性计算云的平台上添加更多的网络服务组件模块,为用户构建云计算应用提供方便.

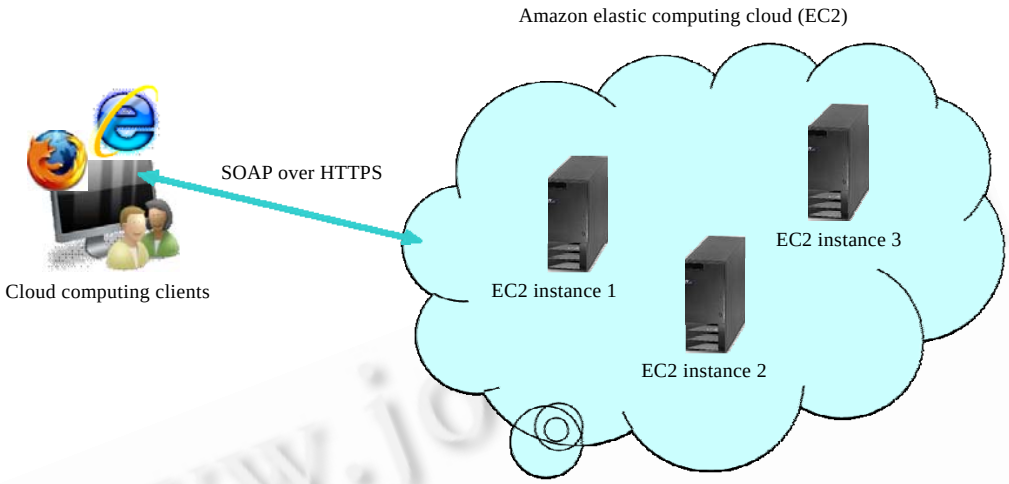

Fig.8 Usage model of Amazon elastic computing cloud

图 8 Amazon 弹性计算云使用模式

5 云计算的学术研究

本节我们将主要讨论与上述系统有关的文献资料以及分布式计算实际系统最新进展的论文.由于云计算技术在互联网应用上的重要性,有很多论文都出自各个大公司的研究院,这包括传统的软件和服务公司(如 Microsoft,惠普)以及新兴的网络服务公司(如 Yahoo,Google 等).关于云计算技术上的研究主要包括两个方面,一个是如何构建分布式平台的基础设施,另一个是如何帮助开发人员在云计算的分布式平台上进行编程.

在分布式平台的基础设施研究上,主要包括微软的 Dryad 框架.Amazon 公司的 Dynamo 框架,以及应用于 Ask.com 公司的 Neptune 框架.微软公司为了方便应用程序开发人员进行分布式程序的开发,提供了一个平台 Dryad[23],以支持有向无环图类型数据流的并行程序.Dryad 是一个一般化的框架,能够支持 MapReduce 类型的应用程序以及关系代数的一些操作.而 Dryad 的整体框架则根据程序的要求完成调度工作,自动完成任务在各个节点上的运行.Amazon 公司的研究人员研究了如何通过集群的技术快速存取大量的(键值,数据)对的问题(即 Key,Value 对),并建立 Dynamo[24]系统来维护这些信息.由于 Amazon 公司的特殊性,其公司内部的应用程序在很多情况下需要处理(键值,数据)对,并且需要扩展到大规模集群上.在对于读写控制方面,传统的读写处理方式是尽量简化读的操作,而将复杂性放在写操作上,Dynamo 系统则将复杂性放在读的方面,将整个系统设计成总是可以写入的,以提高网络用户购物的体验.Dynamo 主要使用结构化的 P2P 结构一致性哈希算法来对数据进行划分与存储,使用向量时钟的方式帮助完成数据读取,并采用哈希树与 Gossip 协议等一些手段对错误进行处理.应用于 Ask.com 的 Neptune[25]技术则针对大量数据进行归并.总体框架首先将数据分布到大规模集群网络上,每一个网络中的节点只需保存一部分数据即可.而后每一个节点在数据上做相应的操作,将操作输出的中间结果进行归并操作即可获得最终的结果.这种归并方式在网络数据处理的应用上非常广泛.

在帮助开发人员在云计算的分布式平台上进行编程的研究方面,有很多研究机构开发了新的编程模式,对 MapReduce 编程模式进行扩展或者更新.Yahoo 公司扩展了 MapReduce 框架,在 MapReduce 步骤之后加入一个 Merge 的步骤,从而形成一个新的 MapReduceMerge[26]框架.使用这样的框架应用程序开发人员可以自己提供 Merge 函数,做两个数据集合的合并操作.Stanford 大学的研究人员将 MapReduce 的思想应用到多核处理器上[27],主要工作是在多核处理器的基础上构建了一套 MapReduce 的编程框架,并结合各种不同的应用程序在多

© 中国科学院软件研究所 http://www.jos.org.cn

核上的表现与现有的使用 pthread 编程方式进行比较.结果表明,在适合 MapReduce 表达的应用程序上,MapReduce 效率较高,在多核上的应用是有价值的.Wisconsin 大学的研究人员在 Cell 处理器上运行了基于 MapReduce 的应用程序[28].由于 Cell 处理器是异构多核的处理器,由 1 个中央处理器和 8 个协处理器构成,对此编程比较困难.他们将 MapReduce 的框架移植到 Cell 处理器的架构上.实验结果表明,Cell 处理器上的 MapReduce 程序有一定程度的性能提高.在不同于 MapReduce 编程方面,HP 的 Sinfonia[29]将注意力关注于分布式共享内存的使用.Sinfornia 提供了一个新的编程接口,一个对于内存的读写操作三元组(Compare,Read,Write).在这个三元组中,Compare 是比较列表,由应用程序提供一系列的值与相应的集群内存中的数值进行比较,类似地,Read 和 Write 是读出和写入的列表,表明一系列的读写操作.其语意是首先进行 Compare 列表的比较,如果所有的比较都能得到满足,则进行三元组中的读写操作.如果上述的任何一个部分操作失败,则整个操作回卷到操作之前的状态,保持系统一致.目前已经在这种模式上完成了分布式文件系统的构建以及分布式的垃圾收集系统等.同时,这样的一种系统也能够容忍大量节点的失败,完成了对于可用性的保证.

6 云计算系统的特征比较与未来的发展

本节主要讨论云计算的共同特征以及它们之间的差异点.从用户的角度来看,云计算系统将各种数据包括用户数据都通过网络保存到远端的云存储平台上,减小了用户对于数据管理的负担;同时,云计算系统也将处理数据的服务程序通过远程的大规模云计算处理平台进行,能够负担大量数据的处理工作.可以说,云计算是数据共享计算模式与服务共享计算模式的结合体,是下一代计算模式的发展方向.从平台技术构建来看,云计算具有 3 个基本特征,即系统建立在大规模的廉价服务器集群之上,通过基础设施与上层应用程序的协同构建以达到最大效率利用硬件资源的目的,以及通过软件的方法容忍多个节点的错误.通过云计算对这 3 个方面基本特征的体现,达到了分布式系统两个方面的目标,即系统的可扩展性和可靠性.

各个云计算平台也各自具有不同的特点.特别是在平台的使用上,透明计算平台为用户同时提供了用户实际接触的客户端节点以及无法接触的远程虚拟存储服务器,是一个半公开的环境.Google 的云计算平台环境是私有的环境,除了开放有限的应用程序接口,例如 GWT(Google Web toolkit),Google App Engine 以及 Google Map API 等以外,Google 并没有将云计算的内部基础设施共享给外部的用户使用.IBM 的“蓝云”计算平台则是可供销售的软、硬件集合,用户基于这些软、硬件产品构建自己的云计算应用.Amazon 的弹性计算云则是托管式的云计算平台,用户可以通过远端的操作界面直接操作使用,看不到实际的物理节点.表 1 从其他角度比较了各个云计算系统的不同之处.可以看出,虽然云计算系统在很多方面具有共性,但实际上各个系统之间还是有很大不同的,这也给云计算用户或者开发人员带来了不同的体验.

云计算未来主要有两个发展方向:一个是构建与应用程序紧密结合的大规模底层基础设施,使得应用能够扩展到很大的规模;另一个是通过构建新型的云计算应用程序,在网络上提供更加丰富的用户体验.第 1 个发展趋势能够从现有的云计算研究状况中体现出来,详见上一节的分析.而在云计算应用的构造上,很多新型的社会服务型网络,如 facebook 等,已经体现了这个发展趋势,而在研究上则开始注重如何通过云计算基础平台将多个业务融合起来.

7 结束语

本文讨论了当前云计算技术发展的前沿技术,通过具体的云计算实例进行详细分析与研究,使读者能够掌握云计算的内部含义.从总体上来说,云计算可以从大规模分布式的基础架构以及平台之上的云计算应用程序两个方面去理解,而本文的重点则放在云计算的基础架构建设方面,这是构建整个云计算平台的根本.总结本文的云计算系统实例以及相关的研究现状,相信随着云计算概念的提出以及相应系统构建实践经验的获得,在不久的将来,工业界和学术界对云计算的研究无疑会成为一个热点.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 1 Features comparison among cloud computing systems

表 1 云计算系统之间的特性比较

	Transparent computing platform from Tsinghua University	Google cloud computing infrastructure	IBM BlueCloud product	Amazon elastic computing cloud
Compatibility to traditional software	Based on the transparent management technology, completely compatible to the current software. Current system and software can run on top of transparent computing platform directly	The new network system is built from scratch; current software cannot run on the infrastructure. Not compatible	Virtualization provided, can run traditional software as well as new cloud computing interface for programming the new applications	Virtualization provided, can run traditional software
System openness	Developed with private technologies	Developed with private technologies	Developed with open source technologies	Combine the open source and private technologies together
Adoption of system virtualization technology	Provide the runtime environment directly on metal hardware, no overhead that might be brought by virtualization	No system virtualization technology adopted, only support new applications	Use open source virtualization software Xen with virtualization overheads	Use open source virtualization software Xen with virtualization overheads
Target users	For end users to use directly	End users can use directly, also open specific interfaces for developers for building new applications	For developers	For developers
Programming support	No programming interface	Specific network application programming interfaces are provided	Local distributed application programming interface	Network remote operation interface

References:

[1] Sims K. IBM introduces ready-to-use cloud computing collaboration services get clients started with cloud computing. 2007. http://www-03.ibm.com/press/us/en/pressrelease/22613.wss

[2] Boss G, Malladi P, Quan D, Legregni L, Hall H. Cloud computing. IBM White Paper, 2007. http://download.boulder.ibm.com/ibmdl/pub/software/dw/wes/hipods/Cloud_computing_wp_final_8Oct.pdf

[3] Zhang YX, Zhou YZ. 4VP+: A novel meta OS approach for streaming programs in ubiquitous computing. In: Proc. of IEEE the 21st Int'l Conf. on Advanced Information Networking and Applications (AINA 2007). Los Alamitos: IEEE Computer Society, 2007. 394−403.

[4] Zhang YX, Zhou YZ. Transparent Computing: A new paradigm for pervasive computing. In: Ma JH, Jin H, Yang LT, Tsai JJP, eds. Proc. of the 3rd Int'l Conf. on Ubiquitous Intelligence and Computing (UIC 2006). Berlin, Heidelberg: Springer-Verlag, 2006. 1−11.

[5] Barroso LA, Dean J, Hölzle U. Web search for a planet: The Google cluster architecture. IEEE Micro, 2003,23(2):22−28.

[6] Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine. Computer Networks, 1998,30(1-7):107−117.

[7] Ghemawat S, Gobioff H, Leung ST. The Google file system. In: Proc. of the 19th ACM Symp. on Operating Systems Principles. New York: ACM Press, 2003. 29−43.

[8] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. In: Proc. of the 6th Symp. on Operating System Design and Implementation. Berkeley: USENIX Association, 2004. 137−150.

[9] Burrows M. The chubby lock service for loosely-coupled distributed systems. In: Proc. of the 7th USENIX Symp. on Operating Systems Design and Implementation. Berkeley: USENIX Association, 2006. 335−350.

[10] Chang F, Dean J, Ghemawat S, Hsieh WC, Wallach DA, Burrows M, Chandra T, Fikes A, Gruber RE. Bigtable: A distributed storage system for structured data. In: Proc. of the 7th USENIX Symp. on Operating Systems Design and Implementation. Berkeley: USENIX Association, 2006. 205−218.

[11] Dean J, Ghemawat S. Distributed programming with Mapreduce. In: Oram A, Wilson G, eds. Beautiful Code. Sebastopol: O'Reilly Media, Inc., 2007. 371−384.

[12] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. Communications of the ACM, 2005,51(1):

107−113.

[13] Pike R, Dorward S, Griesemer R, Quinlan S. Interpreting the data: Parallel analysis with Sawzall. Scientific Programming Journal, 2005,13(4):277−298.

[14] Lamport L. Paxos made simple. ACM SIGACT News, 2001,32(4):51−58.

[15] Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebaur R, Pratt I, Warfield A. Xen and the art of virtualization. In: Proc. of the 9th ACM Symp. on Operating Systems Principles. New York: Bolton Landing, 2003. 164−177.

[16] Citrix systems, citrix XenServer: Efficient virtual server software. XenSource Company. http://www.xensource.com/

[17] IBM. IBM virtualization. 2009. http://www.ibm.com/virtualization

[18] Apache. Apache hadoop. http://hadoop.apache.org/core/

[19] Smith JE, Nair R. Virtual Machines: Versatile Platforms for Systems and Processes. San Francisco: Morgan Kaufmann Publishers, 2005.

[20] Clark C, Fraser K, Hansen JG, Jul E, Pratt I, Warfield A. Live migration of virtual machines. In: Proc. of the 2nd Symp. on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2005. 273−286.

[21] Nelson M, Lim BH, Hutchins G. Fast transparent migration for virtual machines. In: Proc. of the USENIX 2005 Annual Technical Conf. Berkeley: USENIX Association, 2005. 391−394.

[22] Amazon. Amazon elastic compute cloud (Amazon EC2). 2009. http://aws.amazon.com/ec2/

[23] Isard M, Budiu M, Yu Y, Birrell A, Fetterly D. Dryad: Distributed data-parallel programs from sequential building blocks. In: Proc. of the 2nd European Conf. on Computer Systems (EuroSys)., 2007. 59−72.

[24] DeCandia G, Hastorun D, Jampani M, Kakulapati G, Lakshman A, Pilchin A, Sivasubramanian S, Vosshall P, Vogels W. Dynamo: Amazon's highly available key-value store. In: Proc. of the 21st ACM Symp. on Operating Systems Principles. New York: ACM Press, 2007. 205−220.

[25] Chu LK, Tang H, Yang T, Shen K. Optimizing data aggregation for cluster-based Internet services. In: Proc. of the ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming. New York: ACM Press, 2003. 119−130.

[26] Yang HC, Dasdan A, Hsiao RL, Parker DS. Map-Reduce-Merge: Simplified relational data processing on large clusters. In: Proc. of the 2007 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2007. 1029−1040.

[27] Ranger C, Raghuraman R, Penmetsa A, Bradski G, Kozyrakis C. Evaluating MapReduce for multi-core and multiprocessor systems. In: Proc. of the 13th Int'l Symp. on High-Performance Computer Architecture. Los Alamitos: IEEE Computer Society, 2007. 13−24.

[28] de Kruijf M, Sankaralingam K. MapReduce for the Cell B.E. architecture. Technical Report, CS-TR-2007-1625, University of Wisconsin Computer Sciences, 2007.

[29] Aguilera MK, Merchant A, Shah M, Veitch A, Karamanolis C. Sinfonia: A new paradigm for building scalable distributed systems. In: Proc. of the 21st ACM Symp. on Operating Systems Principles. New York: ACM Press, 2007. 159−174.

陈康(1976－),男,浙江台州人,博士,讲师,主要研究领域为分布与并行计算,操作系统,虚拟化.

郑纬民(1946－),男,教授,博士生导师,CCF高级会员,主要研究领域为分布与并行计算,存储系统.