

文章编号:1001-9081(2009)09-2562-06

云计算及其关键技术

陈 全, 邓倩妮

(上海交通大学 计算机科学与工程系, 上海 200240)

(squir3@gmail.com)

摘 要:云计算是一种新兴的计算模型,它是在网格计算的基础上发展而来的。介绍了云计算的发展历史和应用场景,比较了现有的云计算的定义并给出了新的定义,以谷歌的云计算技术为例,总结了云计算的关键技术:数据存储技术(Google File System)、数据管理技术(BigTable)、编程模型和任务调度模型(Map-Reduce)等,分析了云计算和网格计算以及传统超级计算的区别,并指出了云计算的广阔发展前景。

关键词:云计算;数据存储;数据管理;编程模型

中图分类号: TP338.8 **文献标志码:** A

Cloud computing and its key techniques

CHEN Quan, DENG Qian-ni

(Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: Cloud computing is a new computing model; it is developed based on grid computing. The authors introduced the development history of cloud computing and its application situation; compared existing definitions of cloud computing and gave a new definition; took google's cloud computing techniques as an example, summed up key techniques, such as data storage technology (Google File System), data management technology (BigTable), as well as programming model and task scheduling model (Map-Reduce), used in cloud computing; and analyzed the differences among cloud computing, grid computing and traditional super-computing, and fingered out the broad development prospects of cloud computing.

Key words: cloud computing; data storage; data management; programming model

0 引言

云计算(Cloud Computing)是一种新近提出的计算模式。是分布式计算(Distributed Computing)、并行计算(Parallel Computing)和网格计算(Grid Computing)的发展。

目前,亚马逊、微软、谷歌、IBM、英特尔等公司纷纷提出了“云计算”。例如亚马逊的AWS(Amazon Web Services)^[1]、IBM和谷歌联合进行的“蓝云”计划等。这对云计算的商业价值给予了巨大的肯定。同时学术界也纷纷对云计算进行深层次的研究。例如谷歌同华盛顿大学以及清华大学合作,启动云计算学术合作计划(Academic Cloud Computing Initiative),推动云计算的普及,加紧对云计算的研究。卡内基梅隆大学等对数据密集型的超级计算(Data Intensive Super Computing, DISC)^[2]进行研究,本质上也是对云计算相关技术开展研究。

IDC的调查显示,未来五年云计算服务将急速增长,预期2012年市场规模可达420亿美元。目前企业导入云计算已逐渐普及,并且有逐年成长趋势。估计在2012年,企业投入在云计算服务的支出将占整体IT成本的25%,甚至在2013年提高至IT总支出的三分之一。

由此可见,在各大公司以及学术界的共同推动下,云计算技术将会持续发展。

云计算有着广泛的应用前景。如表1所示。云计算在天文学^[3]、医学等各个领域有着广泛的应用前景。

趋势科技和瑞星等安全厂商纷纷提出了“安全云”计划。

如今,每天有2万多种新的病毒和木马产生,传统的通过更新用户病毒库的防毒模式,受到了严峻的挑战,该模式使用户端的病毒库过于庞大。趋势科技和瑞星的“安全云”将病毒资料库放在“云”端,与客户端通过网络相连,当“云”在网络上发现不安全链接时,可以直接形成判断,阻止其进入用户机器,从根本上保护机器的安全。

表1 云计算的应用领域

领域	应用场景
科研	地震监测
	海洋信息监控
	天文信息计算处理
医学	DNA 信息分析
	海量病例存储分析
	医疗影像处理
网络安全	病毒库存储
	垃圾邮件屏蔽
图形和图像处理	动画素材存储分析
	高仿真动画制作
	海量图片检索
互联网	E-mail 服务
	在线实时翻译
	网络检索服务

据趋势科技大中华区执行总裁张伟钦介绍,趋势科技已投入了大量资金,在全球数个地方建设了新型数据中心。同时,趋势科技还花费了1000多万美元,租借了34000多台服务器,构建了一个服务遍及全球的“安全云”。目前趋势科技

收稿日期:2009-03-14;修回日期:2009-05-09。

作者简介:陈全(1986-),男,安徽滁州人,硕士研究生,主要研究方向:云计算、数据密集型可扩展计算; 邓倩妮(1973-),女,广西柳州人,副教授,博士,主要研究方向:对等计算、网格计算、复杂网络拓扑。

已将公司中低端的部分产品线放到“云安全”计划中,而高端的大部分产品线,仍在准备过程中。

谷歌提供的 Gmail、Google Earth、Google Analytics 等服务都基于其云计算服务器运行^[4]。谷歌基于云计算提供的翻译服务具有现今最好的性能^[5]。对互联网和美国人生活的一项研究显示,大约 70% 的在线用户使用以上“云计算”服务。

1 云计算的定义

目前,云计算没有统一的定义,当前云计算的定义主要包括如下几种。

1) 维基百科给云计算下的定义。云计算将 IT 相关的能以服务的方式提供给用户,允许用户在不了解提供服务的技术、没有相关知识以及设备操作能力的情况下,通过 Internet 获取需要的服务^[6]。

2) 中国云计算网将云计算定义为:云计算是分布式计算(Distributed Computing)、并行计算(Parallel Computing)和网格计算(Grid Computing)的发展,或者说是这些科学概念的商业实现^[7]。

3) 文献[8]在综合多个云计算的定义之后,给“云”下了如下定义。云是一个包含大量可用虚拟资源(例如硬件、开发平台以及 I/O 服务)的资源池。这些虚拟资源可以根据不同的负载动态地重新配置,以达到更优化的资源利用率。这种资源池通常由基础设施提供商按照服务协议(Service Level Agreement, SLA)采用按时付费(Pay-Per-Use, PPU)的模式开发管理。

对比上述定义,可以发现,文献[6]和文献[8]中给出的定义的相似之处在于它们都提到了云计算提供服务。文献[6]中的定义主要从云计算用户的角度给出定义,即用户在不了解具体实现的情况下通过 Internet 获取服务。文献[7]从云计算的发展脉络的角度给出定义,主要说明了云计算的历史。文献[8]给出的定义则是从云计算提供者的角度来给出定义,定义了云计算的后台实现方式,以及管理模式。但是,这些定义都仅从一个方面定义云,并不完善。结合上述定义,可以总结出云计算的一些本质特征,即分布式的计算和存储特性,高扩展性,用户友好性,良好的管理性,按时付费等。云计算技术具有以下特点。

1) 云计算系统提供的是服务。服务的实现机制对用户透明,用户无需了解云计算的具体机制,就可以获得需要的服务。

2) 用冗余方式提供可靠性。云计算系统由大量商用计算机组成集群向用户提供数据处理服务。随着计算机数量的增加,系统出现错误的概率大大增加。在没有专用的硬件可靠性部件的支持下,采用软件的方式,即数据冗余和分布式存储来保证数据的可靠性。

3) 高可用性。通过集成海量存储和高性能的计算能力,云能提供较高的服务质量。云计算系统可以自动检测失效节点,并将失效节点排除,不影响系统的正常运行。

4) 高层次的编程模型。云计算系统提供高层次的编程模型。用户通过简单学习,就可以编写自己的云计算程序,在“云”系统上执行,满足自己的需求。现在云计算系统主要采用 Map-Reduce 模型。

5) 经济性。组建一个采用大量的商业机组成的集群相

对于同样性能的超级计算机花费的资金要少很多。

6) 服务多样性。用户可以支付不同的费用,以获得不同级别的服务等。

为了更好地定义云计算,必须从云计算的服务使用者,服务提供者,组织方式,内部实现机制等方面同时定义。

根据现有定义的不足,本文定义云计算为:“云计算是由网格计算发展而来的,前台采用按时付费的方式通过 Internet 向用户提供服务。云系统后台由大量的集群使用虚拟机的方式,通过高速互联网络互连,组成大型的虚拟资源池。这些虚拟资源可自主管理和配置。用数据冗余的方式保证虚拟资源的高可用性。并具有分布式存储和计算、高扩展性、高可用性、用户友好性等特征。”

2 云计算的关键技术

云计算是一种新型的超级计算方式,以数据为中心,是一种数据密集型的超级计算。在数据存储、数据管理、编程模式等多方面具有自身独特的技术。同时涉及了众多其他技术,如表 2 所示。本章主要介绍云计算特有的技术,包括数据存储技术,数据管理技术,编程模式等。其他相关技术请阅读给出的参考文献。

表 2 云计算涉及的关键技术

技术类型	具体技术	参考文献
设备架设	数据中心节能	文献[4, 9]
	节点互联技术	文献[10]
改善服务技术	可用性技术	文献[11-12]
	容错性技术	文献[12-13]
资源管理技术	数据存储技术	文献[14-16]
	数据管理技术	文献[11]
任务管理技术	数据切分技术	文献[17]
	任务调度技术	文献[12, 17]
	编程模型	文献[12, 15-17]
其他相关技术	负载均衡技术	文献[18]
	并行计算技术	文献[19-20]
	虚拟机技术	文献[13, 21-22]
	系统监控技术	文献[23]

2.1 数据存储技术

为保证高可用、高可靠和经济性,云计算采用分布式存储的方式来存储数据,采用冗余存储的方式来保证存储数据的可靠性,即为同一份数据存储多个副本。

另外,云计算系统需要同时满足大量用户的需求,并行地为大量用户提供服务。因此,云计算的数据存储技术必须具有高吞吐率和高传输率的特点。

云计算的数据存储技术主要有谷歌的非开源的 GFS (Google File System)^[14]和 Hadoop 开发团队开发的 GFS 的开源实现 HDFS (Hadoop Distributed File System)^[15-16]。大部分 IT 厂商,包括雅虎、英特尔的“云”计划采用的都是 HDFS 的数据存储技术。

云计算的数据存储技术未来的发展将集中在超大规模的数据存储、数据加密和安全性保证以及继续提高 I/O 速率等方面。

以 GFS 为例。GFS 是一个管理大型分布式数据密集型计算的可扩展的分布式文件系统。它使用廉价的商用硬件搭建系统并向大量用户提供容错的高性能的服务。

GFS 和普通的分布式文件系统有以下区别,如表 3 所示。

表 3 GFS 与传统分布式文件系统的区别

文件系统	组件失败管理	文件大小	数据写方式	数据流和控制流
GFS	不作为异常处理	少量大文件	在文件末尾附加数据	数据流和控制流分开
传统分布式文件系统	作为异常处理	大量小文件	修改现存数据	数据流和控制流结合

GFS 系统由一个 Master 和大量块服务器构成。Master 存放文件系统的所有元数据,包括名字空间、存取控制、文件分块信息、文件块的位置信息等。GFS 中的文件切分为 64 MB 的块进行存储。

在 GFS 文件系统中,采用冗余存储的方式来保证数据的可靠性。每份数据在系统中保存 3 个以上的备份。为了保证数据的一致性,对于数据的所有修改需要在所有的备份上进行,并用版本号的方式来确保所有备份处于一致的状态。

客户端不通过 Master 读取数据,避免了大量读操作使 Master 成为系统瓶颈。客户端从 Master 获取目标数据块的位置信息后,直接和块服务器交互进行读操作。

GFS 的写操作将写操作控制信号和数据流分开,如图 1^[17]所示。

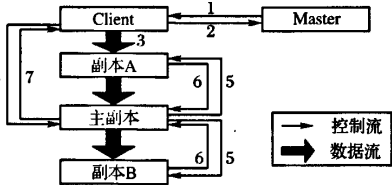


图 1 写控制信号和写数据流

即客户端在获取 Master 的写授权后,将数据传输给所有的数据副本,在所有的数据副本都收到修改的数据后,客户端才发出写请求控制信号。在所有的数据副本更新完数据后,由主副本向客户端发出写操作完成控制信号。具体请见文献[17]。

当然,云计算的数据存储技术并不仅仅只是 GFS,其他 IT 厂商,包括微软、Hadoop 开发团队也在开发相应的数据管理工具。本质上是一种分布式的数据存储技术,以及与之相关的虚拟化技术,对上层屏蔽具体的物理存储器的位置、信息等。快速的数据定位、数据安全性、数据可靠性以及底层设备内存存储数据量的均衡等方面都需要继续研究完善。

2.2 数据管理技术

云计算系统对大数据集进行处理、分析向用户提供高效的服务。因此,数据管理技术必须能够高效地管理大数据集。其次,如何在规模巨大的数据中找到特定的数据,也是云计算数据管理技术所必须解决的问题。

云计算的特点是对海量的数据存储、读取后进行大量的分析,数据的读操作频率远大于数据的更新频率,云中的数据管理是一种读优化的数据管理。因此,云系统的数据管理往往采用数据库领域中列存储的数据管理模式。将表按列划分后存储。

云计算的数据管理技术中最著名的是谷歌在文献[11]提出的 BigTable 数据管理技术。

由于采用列存储的方式管理数据,如何提高数据的更新速率以及进一步提高随机读速率是未来的数据管理技术必须解决的问题。

以 BigTable 为例。BigTable 数据管理方式设计者——Google 给出了如下定义^[11]:“BigTable 是一种为了管理结构化数据而设计的分布式存储系统,这些数据可以扩展到非常大的规模,例如在数千台商用服务器上的达到 PB (Petabytes) 规模的数据。”

BigTable 对数据读操作进行优化,采用列存储的方式,提高数据读取效率。BigTable 管理的数据的存储结构为: < row: string, column: string, time: int64 > -> string。BigTable 的基本元素是:行,列,记录版和时间戳。其中,记录板是一段行的集合体。如图 2 所示。

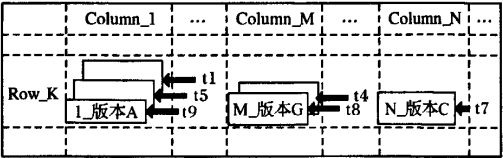


图 2 BigTable 的逻辑结构

BigTable 中的数据项按照行关键字的字典序排列,每行动态地划分到记录板中。每个节点管理大约 100 个记录板。时间戳是一个 64 位的整数,表示数据的不同版本。列族是若干列的集合, BigTable 中的存取权限控制在列族的粒度进行。

BigTable 在执行时需要三个主要的组件:链接到每个客户端的库,一个主服务器,多个记录板服务器。主服务器用于分配记录板到记录板服务器以及负载均衡,垃圾回收等。记录板服务器用于直接管理一组记录板,处理读写请求等。

为保证数据结构的高可扩展性, BigTable 采用三级的层次化的方式来存储位置信息,如图 3 所示^[11]。

其中第一级的 Chubby file 中包含 Root Tablet 的位置, Root Tablet 有且仅有一个,包含所有 METADATA tablets 的位置信息,每个 METADATA tablets 包含许多 User Table 的位置信息。

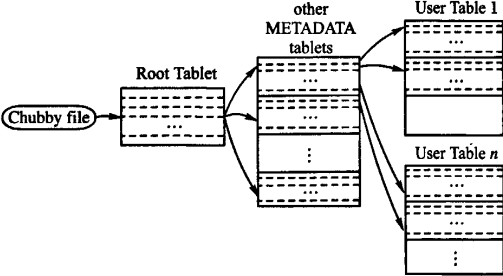


图 3 BigTable 中存储记录板位置信息的结构

当客户端读取数据时,首先从 Chubby file 中获取 Root Tablet 的位置,并从中读取相应 METADATA tablet 的位置信息。接着从该 METADATA tablet 中读取包含目标数据位置信息的 User Table 的位置,然后从该 User Table 中读取目标数据的位置信息项。据此信息到服务器中特定位置读取数据。具体见文献[11]。

这种数据管理技术虽然已经投入使用,但是仍然具有部分缺点。例如,对类似数据库中的 Join 操作效率太低,表内数据如何切分存储,数据类型限定为 string 类型过于简单等。而微软的 DryadLINQ 系统^[20]则将操作的对象封装为 .NET 类,这样有利于对数据进行各种操作,同时对 Join 进行了优化,得到了比 BigTable + MapReduce 更快的 Join 速率和更易用的数据操作方式。

2.3 编程模型

为了使用户能更轻松地享受云计算带来的服务,让用户能利用该编程模型编写简单的程序来实现特定的目的,云计算上的编程模型必须十分简单。必须保证后台复杂的并行执行和任务调度向用户和编程人员透明。

云计算大部分采用 Map-Reduce^[12] 的编程模式。现在大部分 IT 厂商提出的“云”计划中采用的编程模型,都是基于 Map-Reduce 的思想开发的编程工具。

Map-Reduce 不仅仅是一种编程模型,同时也是一种高效的调度模型。Map-Reduce 这种编程模型并不仅适用于云计算,在多核和多处理器、cell processor 以及异构机群上同样有良好的性能^[24-27]。

该编程模式仅适用于编写任务内部松耦合、能够高度并行化的程序。如何改进该编程模式,使程序员能够轻松地编写紧耦合的程序,运行时能高效地调度和执行任务,是 Map-Reduce 编程模型未来的发展方向。

Map-Reduce 是一种处理和产生大规模数据集的编程模型,程序员在 Map 函数中指定对各分块数据的处理过程,在 Reduce 函数中指定如何对分块数据处理的中间结果进行归约。用户只需要指定 map 和 reduce 函数来编写分布式的并行程序。当在集群上运行 Map-Reduce 程序时,程序员不需要关心如何将输入的数据分块、分配和调度,同时系统还将处理集群内节点失败以及节点间通信的管理等。图 4^[15] 给出了一个 Map-Reduce 程序的具体执行过程。

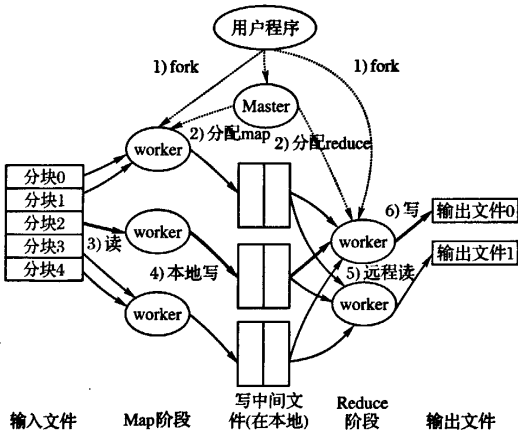


图4 Map-Reduce 程序的具体执行过程

从图4可以看出,执行一个 Map-Reduce 程序需要五个步骤:输入文件、将文件分配给多个 worker 并行地执行、写中间文件(本地写)、多个 Reduce workers 同时运行、输出最终结果。本地写中间文件在减少了对网络带宽的压力同时减少了写中间文件的时间耗费。执行 Reduce 时,根据从 Master 获得的中间文件位置信息,Reduce 使用远程过程调用,从中间文件所在节点读取所需的数据。

Map-Reduce 模型具有很强的容错性,当 worker 节点出现错误时,只需要将该 worker 节点屏蔽在系统外等待修复,并将该 worker 上执行的程序迁移到其他 worker 上重新执行,同时将该迁移信息通过 Master 发送给需要该节点处理结果的节点。Map-Reduce 使用检查点的方式来处理 Master 出错失败的问题,当 Master 出现错误时,可以根据最近的一个检查点重新选择一个节点作为 Master 并由此检查点位置继续运行。具体请见文献[12]。

Map-Reduce 仅为编程模式的一种,微软提出的 DryadLINQ^[17] 是另外一种并行编程模式。但它局限于 .NET 的 LINQ 系统同时并不开源,限制了它的发展前景。

Map-Reduce 作为一种较为流行的云计算编程模型,在云计算系统中应用广阔。但是基于它的开发工具 Hadoop 并不完善。特别是其调度算法过于简单,判断需要进行推测执行的任务的算法造成过多任务需要推测执行,降低了整个系统的性能。改进 Map-Reduce 的开发工具,包括任务调度器、底层数据存储系统、输入数据切分、监控“云”系统等方面是将来一段时间的主要发展方向。另外,将 Map-Reduce 的思想运用在云计算以外的其他方面也是一个流行的研究方向。

3 云计算和其他超级计算的区别

3.1 云计算与网格计算的区别

Ian Foster 将网格定义为:支持在动态变化的分布式虚拟组织(Virtual Organizations)间共享资源,协同解决问题的系统^[28]。所谓虚拟组织就是一些个人、组织或资源的动态组合。

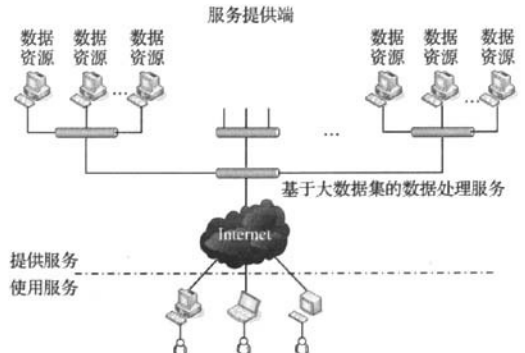


图5 “云”系统的结构

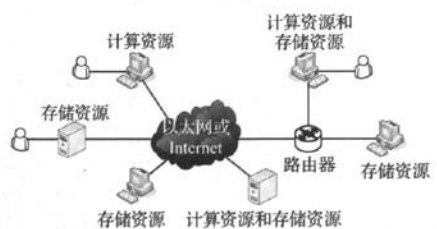


图6 网络的结构

图5和图6分别为云及网格的结构示意图。图5显示,云计算是一种生产者—消费者模型,云计算系统采用以太网等快速网络将若干集群连接在一起,用户通过因特网获取云计算系统提供的各种数据处理服务。图6显示,网格系统是一种资源共享模型,资源提供者亦可以成为资源消费者,网格侧重研究的是如何将分散的资源组合成动态虚拟组织。主要区别如表4所示。

云计算和网格计算^[29]的一个重要区别在于资源调度模式。云计算采用集群来存储和管理数据资源,运行的任务以数据为中心。即调度计算任务到数据存储节点运行。而网格计算,则以计算为中心。计算资源和存储资源分布在因特网的各个角落,不强调任务所需的计算和存储资源同处一地。由于网络带宽的限制,网格计算中的数据传输时间占总运行时间的很大一部分。

网格将数据和计算资源虚拟化,而云计算则进一步将硬

件资源虚拟化,活用虚拟机技术,对失败任务重新执行,而不必重启任务。同时,网格内各节点采用统一的操作系统(大部分为 UNIX),而云计算放宽了条件,在各种操作系统的虚拟机上提供各种服务。和网络的复杂管理方式不同,云计算提供一种简单易用的管理环境。另外,网格和云在付费方式上有着显著的不同。网格按照固定的资费标准收费或者若干组织之间共享空闲资源。而云则采用用时付费以及服务等级协议的模式收费。其他区别不再赘述。

表4 网格计算和云计算的主要区别

区别点	网格计算	云计算
发起者	学术界	工业界
标准化	是(OGSA)	否
开源	是	部分开源
互联网络	因特网,高延时低带宽	高速网络,低延时高带宽
关注点	计算密集型	数据密集型
节点	分散的PC或服务器	集群
获取的对象	共享的资源	提供的服务
安全保证	公私钥技术, 账户技术	每个任务一个虚拟机, 保证隔离性
节点操作系统	相同的系统(UNIX)	多种操作系统上的虚拟机
虚拟化	虚拟数据和计算资源	虚拟软硬件平台
节点管理方式	分散式管理	集中式管理
易用性	难以管理,使用	用户友好
付费方式	/	用时付费
失败管理	失败的任务重启	虚拟机迁移到其他节点继续执行
对第三方插件的兼容性	难以兼容	易于兼容,通过提供不同的服务来兼容
自我管理方式	重新配置	重新配置,自我修复

3.2 云计算系统与传统超级计算机的区别

超级计算机拥有强大的处理能力,特别是计算能力。2008年11月17日,最新一期的Top500^[30]榜单发布。冠军“RoadRunner”是IBM为美国Los Alamos国家实验室建造的计算机系统。它的运算速度达到了1.026 PFLOPS。RoadRunner超级计算机包含12960个IBM PowerXcell 8i处理器以及6948个分布于刀片服务器上的AMD Opteron芯片。刀片服务器安装在288个IBM BladCenter机架架上。RoadRunner拥有80TB的内存,外存使用1.5PB容量的Panasas存储,外存通过10Gbps的以太网进行连接。耗资超过1亿美元。

从TOP500对超级计算机的排名方式可以看出,传统的超级计算机注重运算速度和任务的吞吐率。以运算速度为核心进行计算机的研究和开发。而云计算则以数据为中心,同时兼顾系统的运算速度。传统的超级计算机耗资巨大,远超云计算系统。例如,趋势科技花费1000多万美元租用34000多台服务器,构建自身的“安全云”系统。

4 结语

云计算具有广阔的发展前景,相关的各项关键技术也在迅速发展。

首先,当前的云计算系统的能耗过大,因此,减少能耗,提高能源的使用效率,建造高效的冷却系统是当前面临的一个主要问题。例如,谷歌的数据中心的能耗相当于一个小型城市的总能耗^[31]。因为,过大的能耗使得数据中心内发热量剧增,要保证云计算系统的正常运行,必须使用高效的冷却系统来保持数据中心在可接受的温度范围内。

其次,云计算对面向市场的资源管理方式的支持有限。可以加强相应的服务等级协议,使用户和服务提供者能更好的协商提供的服务质量。

另外,需要对云计算的接口进行标准化并且制定交互协议。这样可以支持不同云计算服务提供者之间进行交互,相互合作提供更加强大和更好的服务。

再者,需要开发出更易用的编程环境和编程工具,这样可以更加方便地创建云计算应用,拓展云计算的应用领域。

最后,虽然云计算还有很多问题需要解决,但是云计算必将得到更大的发展。

参考文献:

- [1] VARIA J. Cloud architectures - Amazon Web services [EB/OL]. [2009-03-01]. <http://acmbangalore.org/events/monthly-talk/may-2008--cloud-architectures---amazon-web-services.html>.
- [2] BRYANT R E. Data-intensive supercomputing: The case for DISC, CMU-CS-07-128 [R]. Pittsburgh, PA, USA: Carnegie Mellon University, Department of Computer Science, 2007.
- [3] SZALAY A S, KUNSZT P, THAKAR A, *et al.* Designing and mining multi-terabyte astronomy archives: The sloan digital sky survey [C]// Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2000: 451-462.
- [4] BARROSO L A, DEAN J, HÖLZLE U. Web search for a planet: The Google cluster architecture [J]. IEEE Micro, 2003, 23(2): 22-28.
- [5] GILES J. Google tops translation ranking [EB/OL]. (2006-11-06) [2009-03-06]. <http://www.nature.com/news/2006/061106/full/news061106-6.html>.
- [6] 维基百科. Cloud computing [EB/OL]. [2009-03-10]. http://en.wikipedia.org/wiki/Cloud_computing.
- [7] 中国云计算网. 什么是云计算? [EB/OL]. (2008-05-14) [2009-02-27]. <http://www.cloudcomputing-china.cn/Article/ShowArticle.asp?ArticleID=1>.
- [8] VAQUERO L M, RODERO-MERINO L, CACERES J, *et al.* A break in the clouds: Towards a cloud definition [J]. ACM SIGCOMM Computer Communication Review, 2009, 39(1): 50-55.
- [9] WEISS A. Computing in the clouds [J]. ACM Networker, 2007, 11(4): 16-25.
- [10] GRIFFITHS A, METHERALL G. Cluster interconnection networks [EB/OL]. (2000-09-01) [2009-03-03]. <http://www.gridbus.org/~raj/cse433/ClusterNets.pdf>.
- [11] CHANG F, DEAN J, GHEMAWAT S, *et al.* BigTable: A distributed storage system for structured data [J]. ACM Transactions on Computer Systems, 2008, 26(2): 1-26.
- [12] DEAN J, GHEMAWAT S. MapReduce: Simplified data processing on large clusters [C]// OSDI'04: Proceedings of the 6th Symposium on Operating System Design and Implementation. New York: ACM Press, 2004: 137-150.
- [13] NAGARAJAN A B, MUELLER F, ENGELMANN C, *et al.* Proactive fault tolerance for HPC with Xen virtualization [C]// Proceedings of the 21st Annual International Conference on Supercomputing. New York: ACM Press, 2007: 23-32.
- [14] GHEMAWAT S, GOBIOFF H, LEUNG P T. The Google file system [C]// Proceedings of the 19th ACM Symposium on Operating Systems Principles. New York: ACM Press, 2003: 29-43.
- [15] Apache Hadoop. Hadoop [EB/OL]. [2009-03-06]. <http://hadoop.apache.org/>.
- [16] Yahoo. Yahoo! Hadoop tutorial [EB/OL]. [2009-02-27]. <http://>

public.yahoo.com/gogate/hadoop-tutorial/start-tutorial.html.

[17] YU Y, ISARD M, FETTERLY D, *et al.* DryadLINQ: A system for general-purpose distributed data-parallel computing using a high-level language [C]// Proceedings of the 8th USENIX Symposium on Operating Systems Design and Implementation. New York: ACM Press, 2008: 1–14.

[18] PINAR A, HENDRICKSON B. Exploiting flexibly assignable work to improve load balance [C]// Proceedings of the 14th Annual ACM Symposium on Parallel Algorithms and Architectures. New York: ACM Press, 2002: 155–163.

[19] GRAMA A, KARYPIS G, KUMAR V, *et al.* Introduction to parallel computing [M]. 2nd ed. London: Pearson Education, 2003.

[20] YAN B Q, RHODES P J. Toward automatic parallelization of spatial computation for computing clusters [C]// Proceedings of the 17th International Symposium on High Performance Distributed Computing. New York: ACM Press, 2008: 45–54.

[21] KRSUL I, GANQULY A, ZHANG P J, *et al.* VMPlants: Providing and managing virtual machine execution environments for grid computing [C]// Proceedings of the 2004 ACM/IEEE Conference on Supercomputing. New York: ACM Press, 2004: 7–7.

[22] BARHAM P, DRAGOVIC B, FRASER K, *et al.* Xen and the art of virtualization [J]. ACM SIGOPS Operating Systems Review, 2003, 32(5): 164–177.

[23] SAAB C B, AIRE X B, FOLLIOT B. Phoenix: A self adaptable monitoring platform for cluster management [J]. Cluster Computing, 2002, 5(1): 75–85.

[24] RANGER C, RAGHURAMAN R, PENMETSA A. Evaluating MapReduce for multi-core and multiprocessor systems [C]// Proceedings of the 2007 IEEE 13th International Symposium on High Performance Computer Architecture. Washington, DC: IEEE Computer Society, 2007: 13–24.

[25] KRUIJF M D, SANKARALINGAM K. MapReduce for the cell B. E. architecture, CS-TR-2007-1625 [R]. Madison, MI, USA: University of Wisconsin-Madison, 2007.

[26] HE B S, FANG W B, LUO Q, *et al.* Mars: A MapReduce framework on graphics processors [C]// Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques. New York: ACM Press, 2008: 260–269.

[27] ZAHARIA M, KONWINSKI A, JOSEPH A D. Improving MapReduce performance in heterogeneous environments [C]// Proceedings of the 8th USENIX Symposium on Operating Systems Design and Implementation. New York: ACM Press, 2008: 29–42.

[28] FOSTER I, KESSELMAN C, TUECKE S. The anatomy of the grid enabling scalable virtual organizations [J]. International Journal of High Performance Computing Applications, 2001, 15(3): 200–222.

[29] BERMAN F, FOX G, HEY T. The grid: Past, present, and future [M]// Grid computing: Making the global infrastructure a reality. England: John Wiley and Sons, 2003: 9–50.

[30] Top 500 supercomputing sites. Top500 list-November 2008 (1–100) [EB/OL]. (2008–11–17) [2009–03–05]. <http://www.top500.org/list/2008/11/100>.

[31] BUYYA R, YEO C S, VENUGOPAL S, *et al.* Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility [J]. Future Generation Computer Systems, 2008, 25(6): 599–616.

(上接第 2547 页)

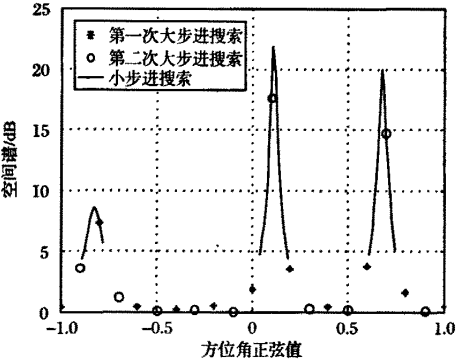


图 2 改进 MUSIC 算法的空间谱

两种算法性能的比较如表 1 所示。如果选择 5 为门限,邻域宽度 0.2 时,谱峰宽度 $\Delta = 0.09$,根据式(13)一次搜索出三个信号的概率是:

$$Z_1 = (0.09/0.2)^3 = 9.1\% \tag{21}$$

此时,加法运算量只有传统算法 40%,乘法运算量只有传统算法 40%。两次搜索出三个信号的概率是 $Z_2 = 72.9\%$,此时时加法运算量只有传统算法 50%,乘法运算量只有传统算法 50%。

表 1 算法性能比较

搜索方式	加法次数	乘法次数	概率/%
传统搜索	30 000	57 400	100.0
一次搜索	12 000	22 960	9.1
两次搜索	15 000	28 700	72.9

5 结语

通过大步进和小步进联合搜索,可以有效减少 MUSIC 法运算量,同时保持对信号 DOA 的估计精度。新算法对于解决 MUSIC 算法工程应用中面临的大运算量问题,具有重要意义。

参考文献:

[1] PUSKA H, SAARNISAARI H, IINATTI J. Serial search code acquisition using smart antennas with single correlator or matched filter [J]. IEEE Transactions on Communications, 2008, 56(2): 299–307.

[2] HIROTA A, ARAI H, NAKANO M. Direction-of-arrival estimation system for multipath propagation in code-division multiple-access systems with synthesized virtual planar array using pilot signals [J]. IEEE Transactions on Vehicular Technology, 2008, 57(4): 2153–2161.

[3] ZHANG T T, LU Y L, HUI H T. Compensation for the mutual coupling effect in uniform circular arrays for 2D DOA estimations employing the maximum likelihood technique [J]. IEEE Transactions on Aerospace and Electronic Systems, 2008, 44(3): 1215–1221.

[4] SCHMIDT R O. Multiple emitter location and signal parameter estimation [J]. IEEE Transactions on Antennas and Propagation, 1986, 34(3): 276–280.

[5] YE Z F, LIU C. On the resiliency of MUSIC direction finding against antenna sensor coupling [J]. IEEE Transactions on Antennas and Propagation, 2008, 56(2): 371–380.

[6] LI B, PENG C Y, BISWAS S. Association of DOA estimation from two ULAs [J]. IEEE Transactions on Instrumentation and Measurement, 2008, 57(6): 1094–1101.

[7] van TREES H L. Optimum array processing [M]. New York: John Wiley, 2002.