

云计算：体系架构与关键技术

罗军舟，金嘉晖，宋爱波，东方

(东南大学 计算机科学与工程学院，江苏 南京 211189)

摘 要：系统地分析和总结云计算的研究现状，划分云计算体系架构为核心服务、服务管理、用户访问接口等 3 个层次。围绕低成本、高可靠、高可用、规模可伸缩等研究目标，深入全面地介绍了云计算的关键技术及最新研究进展。在云计算基础设施方面，介绍了云计算数据中心设计与管理及资源虚拟化技术；在大规模数据处理方面，分析了海量数据处理平台及其资源管理与调度技术；在云计算服务保障方面，讨论了服务质量保证和安全与隐私保护技术。针对新型的云计算应用和云计算存在的局限性，又探讨并展望了今后的研究方向。最后，介绍了东南大学云计算平台以及云计算研究与应用方面的相关成果。

关键词：云计算；虚拟化；数据中心；海量数据处理；服务质量；安全与隐私

中图分类号：TP393

文献标识码：A

文章编号：1000-436X(2011)07-0003-19

Cloud computing: architecture and key technologies

LUO Jun-zhou, JIN Jia-hui, SONG Ai-bo, DONG Fang

(School of Computer Science and Engineering, Southeast University, Nanjing 211189, China)

Abstract: Based on the analysis and summary of the previous work, the cloud computing framework can be divided into three layers: core cloud services, cloud service management and user access interface. To realize low-cost, reliable, available and scalable cloud services, the up-to-date key technologies and research progresses of the three layers within the cloud framework are reviewed intensively and extensively. Regarding cloud infrastructure, the strategies for data center design and management are introduced together with the virtualization technology. In respect of large-scale data processing, several platforms and their resource management and task scheduling mechanisms are investigated extensively. With regard to cloud service quality, both QoS guarantee and security/privacy protection are discussed in depth. The prospects for future research are further explored based on the new application models and limitations of cloud computing. Finally, the cloud computing platform and research work of Southeast University are introduced.

Key words: cloud computing; virtualization; data center; large-scale data processing; quality of service; security and privacy

收稿日期：2011-05-20；修回日期：2011-06-30

基金项目：国家自然科学基金资助项目(61070161, 61070158, 61003257, 60773103, 90912002)；国家重点基础研究发展计划(“973”计划)基金资助项目(2010CB328104)；国家科技支撑计划课题基金资助项目(2010BAI88B03)；教育部博士点基金课题基金资助项目(200802860031)；江苏省自然科学基金资助项目(BK2008030)；国家科技重大专项课题基金资助项目(2009ZX03004-004-04)；江苏省“网络与信息安全”重点实验室基金资助项目(BM2003201)；“计算机网络与信息集成”教育部重点实验室项目(93K-9)

Foundation Items: The National Natural Science Foundation of China(61070161, 61070158, 61003257, 60773103, 90912002); The National Key Basic Research Program of China (973 Program) (2010CB328104); China National Key Technology R&D Program (2010BAI88B03); China Specialized Research Fund for the Doctoral Program of Higher Education (200802860031); The Natural Science Foundation of Jiangsu Province (BK2008030); China National S&T Major Project (2009ZX03004-004-04); Jiangsu Provincial Key Laboratory of Network and Information Security (BM2003201); The Key Laboratory of Computer Network and Information Integration of Ministry of Education of China (93K-9)

1 引言

近年来, 社交网络、电子商务、数字城市、在线视频等新一代大规模互联网应用发展迅猛。这些新兴的应用具有数据存储量大、业务增长速度快等特点。据统计至 2010 年, 社交网站 Facebook 已存储了 15PB 的数据, 并且每天新增 60TB 数据^[1]; 电子商务网站淘宝的 B2C 业务在 2010 年增长了 4 倍, 其数据中心存储了 14PB 数据, 并且每天需要处理 500TB 数据^[2]。与此同时, 传统企业的软硬件维护成本高昂: 在企业的 IT 投入中, 仅有 20% 的投入用于软硬件更新与商业价值的提升, 而 80% 则投入用于系统维护。根据 2006 年 IDC 对 200 家企业的统计, 部分企业的信息技术人力成本达到 1 320 美元/每人/每台服务器, 而部署一个新的应用系统需要花费 5.4 周^[3]。

为了解决上述问题, 2006 年 Google、Amazon 等公司提出了“云计算”的构想。根据美国国家标准与技术研究院 (NIST) 的定义^[4], 云计算是一种利用互联网实现随时随地、按需、便捷地访问共享资源池 (如计算设施、存储设备、应用程序等) 的计算模式。计算机资源服务化是云计算重要的表现形式, 它为用户屏蔽了数据中心管理、大规模数据处理、应用程序部署等问题。通过云计算, 用户可以根据其业务负载快速申请或释放资源, 并以按需支付的方式对所使用的资源付费, 在提高服务质量的同时降低运维成本。

作为信息产业的一大创新, 云计算模式一经提出便得到工业界、学术界的广泛关注。其中 Amazon 等公司的云计算平台提供可快速部署的虚拟服务器, 实现了基础设施的按需分配^[5]。MapReduce^[6] 等新型并行编程框架简化了海量数据处理模型。Google 公司的 App Engine^[7] 云计算开发平台为应用服务提供商开发和部署云计算服务提供接口。Salesforce 公司的客户关系管理 (CRM, customer relationship management) 服务^[8] 等云计算服务将桌面应用程序迁移到互联网, 实现应用程序的泛在访问。同时, 各国学者对云计算也展开了大量研究工作。早在 2007 年, 斯坦福大学等多所美国高校便开始和 Google、IBM 合作, 研究云计算关键技术。近年来, 随着云计算研究的深入, 众多国际会议 (如 SIGCOMM、OSDI、SIGMOD、CCS 等) 上陆续发表了云计算相关研究成果。此外, 以 Eucalyptus^[9]

为代表的开源云计算平台的出现, 加速了云计算服务的研究和普及。

不仅如此, 各国政府纷纷将云计算列为国家战略, 投入了相当大的财力和物力用于云计算的部署。其中, 美国政府利用云计算技术建立联邦政府网站, 以降低政府信息化运行成本。英国政府建立国家级云计算平台 (G-Cloud), 超过 2/3 的英国企业开始使用云计算服务。在我国, 北京、上海、深圳、杭州、无锡等城市开展了云计算服务创新发展试点示范工作; 电信、石油石化、交通运输等行业也启动了相应的云计算发展计划, 以促进产业信息化。

然而, 云计算本质上并非一个全新的概念。早在 1961 年, 计算机先驱 John McCarthy 就预言: “未来的计算资源能像公共设施 (如水、电) 一样被使用。”^[10] 为了实现这个目标, 在之后的几十年里, 学术界和工业界陆续提出了集群计算、效用计算、网格计算、服务计算等技术, 而云计算正是从这些技术发展而来。

在这些传统技术中, 集群计算将大量独立的计算机通过高速局域网相连, 从而提供高性能计算能力。效用计算为用户提供按需租用计算机资源的途径。网格计算整合大量异构计算机的闲置资源 (如计算资源和磁盘存储等), 组成虚拟组织, 以解决大规模计算问题。服务计算作为连接信息技术和商业服务的桥梁, 研究如何用信息技术对商业服务建模、操作和管理。

对云计算而言, 其借鉴了传统分布式计算的思想。通常情况下, 云计算采用计算机集群构成数据中心, 并以服务的形式交付给用户, 使得用户可以像使用水、电一样按需购买云计算资源。从这个角度看, 云计算与网格计算的目标非常相似^[10]。但是云计算和网格计算等传统的分布式计算也有着较明显的区别: 首先云计算是弹性的, 即云计算能根据工作负载大小动态分配资源, 而部署于云计算平台上的应用需要适应资源的变化, 并能根据变化做出响应; 其次, 相对于强调异构资源共享的网格计算, 云计算更强调大规模资源池的分享, 通过分享提高资源复用率, 并利用规模经济降低运行成本; 最后, 云计算需要考虑经济成本, 因此硬件设备、软件平台的设计不再一味追求高性能, 而要综合考虑成本、可用性、可靠性等因素。

基于上述比较并结合云计算的应用背景, 云计算的特点可归纳如下^[4]。

1) 弹性服务。服务的规模可快速伸缩，以自动适应业务负载的动态变化。用户使用的资源同业务的需求相一致，避免了因为服务器性能过载或冗余而导致的服务质量下降或资源浪费。

2) 资源池化。资源以共享资源池的方式统一管理。利用虚拟化技术，将资源分享给不同用户，资源的放置、管理与分配策略对用户透明。

3) 按需服务。以服务的形式为用户提供应用程序、数据存储、基础设施等资源，并可以根据用户需求，自动分配资源，而不需要系统管理员干预。

4) 服务可计费。监控用户的资源使用量，并根据资源的使用情况对服务计费。

5) 泛在接入。用户可以利用各种终端设备（如 PC 电脑、笔记本电脑、智能手机等）随时随地通过互联网访问云计算服务。

正是因为云计算具有上述 5 个特性，使得用户只需连上互联网就可以源源不断地使用计算机资源，实现了“互联网即计算机”的构想。

综上所述，云计算是分布式计算、互联网技术、大规模资源管理等技术的融合与发展（如图 1 所示），其研究和应用是一个系统工程，涵盖了数据中心管理、资源虚拟化、海量数据处理、计算机安全等重要问题。本文通过归纳云计算特点与体系架构，总结和分析云计算各层服务的关键技术及系统实例，针对当前云计算存在的问题，提出未来研究的方向。

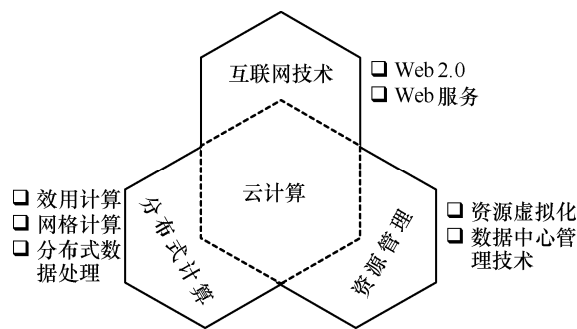


图 1 云计算与相关技术的联系

2 云计算体系架构

云计算可以按需提供弹性资源，它的表现形式是一系列服务的集合。结合当前云计算的应用与研究，其体系架构可分为核心服务、服务管理、用户访问接口 3 层，如图 2 所示。核心服务层将硬件基础设施、软件运行环境、应用程序抽象成服务，这

些服务具有可靠性强、可用性高、规模可伸缩等特点，满足多样化的应用需求。服务管理层为核心服务提供支持，进一步确保核心服务的可靠性、可用性与安全性。用户访问接口层实现端到云的访问。

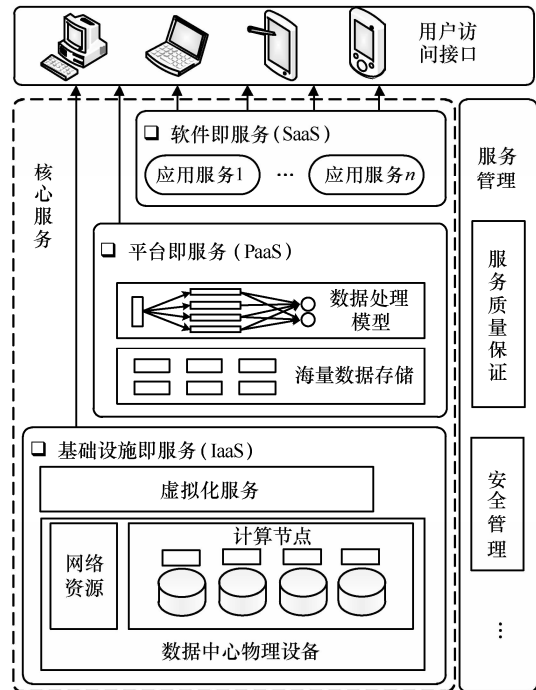


图 2 云计算体系架构

2.1 核心服务层

云计算核心服务通常可以分为 3 个子层：基础设施即服务层 (IaaS, infrastructure as a service)、平台即服务层 (PaaS, platform as a service)、软件即服务层 (SaaS, software as a service)。表 1 对 3 层服务的特点进行了比较。

IaaS 提供硬件基础设施部署服务，为用户按需提供实体或虚拟的计算、存储和网络等资源。在使用 IaaS 层服务的过程中，用户需要向 IaaS 层服务提供商提供基础设施的配置信息，运行于基础设施的程序代码以及相关的用户数据。由于数据中心是 IaaS 层的基础，因此数据中心的管理和优化问题近年来成为研究热点。另外，为了优化硬件资源的分配，IaaS 层引入了虚拟化技术。借助于 Xen、KVM、VMware 等虚拟化工具，可以提供可靠性高、可定制性强、规模可扩展的 IaaS 层服务。

PaaS 是云计算应用程序运行环境，提供应用程序部署与管理服务。通过 PaaS 层的软件工具和开发语言，应用程序开发者只需上传程序代码和数据即可使

表 1 IaaS、PaaS、SaaS 的比较

服务内容	服务对象	使用方式	关键技术	系统实例
IaaS 提供基础设施部署服务	需要硬件资源的用户	使用者上传数据、程序代码、环境配置	数据中心管理技术、虚拟化技术等	Amazon EC2、Eucalyptus 等
PaaS 提供应用程序部署与管理服务	程序开发者	使用者上传数据、程序代码	海量数据处理技术、资源管理与调度技术等	Google App Engine、Microsoft Azure、Hadoop 等
SaaS 提供基于互联网的应用程序服务	企业和需要软件应用的 用户	使用者上传数据	Web 服务技术、互联网应用开发技术等	Google Apps、Salesforce CRM 等

用服务，而不必关注底层的网络、存储、操作系统的管理问题。由于目前互联网应用平台（如 Facebook、Google、淘宝等）的数据量日趋庞大，PaaS 层应当充分考虑对海量数据的存储与处理能力，并利用有效的资源管理与调度策略提高处理效率。

SaaS 是基于云计算基础平台所开发的应用程序。企业可以通过租用 SaaS 层服务解决企业信息化问题，如企业通过 GMail 建立属于该企业的电子邮件服务。该服务托管于 Google 的数据中心，企业不必考虑服务器的管理、维护问题。对于普通用户来讲，SaaS 层服务将桌面应用程序迁移到互联网，可实现应用程序的泛在访问。

2.2 服务管理层

服务管理层对核心服务层的可用性、可靠性和安全性提供保障。服务管理包括服务质量（QoS, quality of service）保证和安全管理等。

云计算需要提供高可靠、高可用、低成本的个性化服务。然而云计算平台规模庞大且结构复杂，很难完全满足用户的 QoS 需求。为此，云计算服务提供商需要和用户进行协商，并制定服务水平协议（SLA, service level agreement），使得双方对服务质量的需求达成一致。当服务提供商提供的服务未能达到 SLA 的要求时，用户将得到补偿。

此外，数据的安全性一直是用户较为关心的问题。云计算数据中心采用的资源集中式管理方式使得云计算平台存在单点失效问题。保存在数据中心的 关键数据会因为突发事件（如地震、断电）、病毒入侵、黑客攻击而丢失或泄露。根据云计算服务特点，研究云计算环境下的安全与隐私保护技术（如数据隔离、隐私保护、访问控制等）是保证云计算得以广泛应用的关键。

除了 QoS 保证、安全管理外，服务管理层还包括计费管理、资源监控等管理内容，这些管理措施对云计算的稳定运行同样起到重要作用。

2.3 用户访问接口层

用户访问接口实现了云计算服务的泛在访问，

通常包括命令行、Web 服务、Web 门户等形式。命令行和 Web 服务的访问模式既可为终端设备提供应用程序开发接口，又便于多种服务的组合。Web 门户是访问接口的另一种模式。通过 Web 门户，云计算将用户的桌面应用迁移到互联网，从而使用户随时随地通过浏览器就可以访问数据和程序，提高工作效率。虽然用户通过访问接口使用便利的云计算服务，但是由于不同云计算服务商提供接口标准不同，导致用户数据不能在不同服务商之间迁移。为此，在 Intel、Sun 和 Cisco 等公司的倡导下，云计算互操作论坛（CCIF, cloud computing interoperability forum）宣告成立，并致力于开发统一的云计算接口（UCI, unified cloud interface），以实现“全球环境下，不同企业之间可利用云计算服务无缝协同工作”的目标。

3 云计算关键技术

云计算的目标是以低成本的方式提供高可靠、高可用、规模可伸缩的个性化服务。为了达到这个目标，需要数据中心管理、虚拟化、海量数据处理、资源管理与调度、QoS 保证、安全与隐私保护等若干关键技术加以支持。本节详细介绍核心服务层与服务管理层涉及的关键技术和典型应用，并从 IaaS、PaaS、SaaS 3 个方面依次对核心服务层进行分析。

3.1 IaaS

IaaS 层是云计算的基础。通过建立大规模数据中心，IaaS 层为上层云计算服务提供海量硬件资源。同时，在虚拟化技术的支持下，IaaS 层可以实现硬件资源的按需配置，并提供个性化的基础设施服务。基于以上两点，IaaS 层主要研究 2 个问题：①如何建设低成本、高效能的数据中心；②如何拓展虚拟化技术，实现弹性、可靠的基础设施服务。

3.1.1 数据中心相关技术

数据中心是云计算的核心，其资源规模与可靠性对上层的云计算服务有着重要影响。Google、Facebook 等公司十分重视数据中心的建设。在 2009

年，Facebook 的数据中心拥有 30 000 个计算节点，截止 2010 年，计算节点数量更是达到 60 000 个^[11]；Google 公司平均每季度投入约 6 亿美元用于数据中心建设，其中仅 2010 年第 4 季度便投入了 25 亿美元^[12]。

与传统的企业数据中心不同，云计算数据中心具有以下特点^[13]。

1) 自治性。相较传统的数据中心需要人工维护，云计算数据中心的大规模性要求系统在发生异常时能自动重新配置，并从异常中恢复，而不影响服务的正常使用。

2) 规模经济。通过对大规模集群的统一化标准化管理，使单位设备的管理成本大幅降低。

3) 规模可扩展。考虑到建设成本及设备更新换代，云计算数据中心往往采用大规模高性价比的设备组成硬件资源，并提供扩展规模的空间。

基于以上特点，云计算数据中心的相关研究工作主要集中在以下 2 个方面：研究新型的数据中心网络拓扑，以低成本、高带宽、高可靠的方式连接大规模计算节点；研究有效的绿色节能技术，以提高效能比，减少环境污染。

① 数据中心网络设计

目前，大型的云计算数据中心由上万个计算节点构成，而且节点数量呈上升趋势。计算节点的大规模性对数据中心网络的容错能力和可扩展性带来挑战。

然而，面对以上挑战，传统的树型结构网络拓扑^[14]（如图 3 所示）存在以下缺陷。首先，可靠性低，若汇聚层或核心层的网络设备发生异常，网络性能会大幅下降。其次，可扩展性差，因为核心层网络设备的端口有限，难以支持大规模网络。再次，网络带宽有限，在汇聚层，汇聚交换机连接边缘层的网络带宽远大于其连接核心层的网络带宽（带宽比例为 80:1，甚至 240:1）^[14]，所以对于连接在不同汇聚交换机的计算节点来说，它们的网络通信容易受到阻塞。

为了弥补传统拓扑结构的缺陷，研究者提出了 VL2^[14]、PortLand^[15]、DCCell^[16]、BCube^[17]等新型的网络拓扑结构。这些拓扑在传统的树型结构中加入了类似于 mesh 的构造，使得节点之间连通性与容错能力更高，易于负载均衡。同时，这些新型的拓扑结构利用小型交换机便可构建，使得网络建设成本降低，节点更容易扩展。

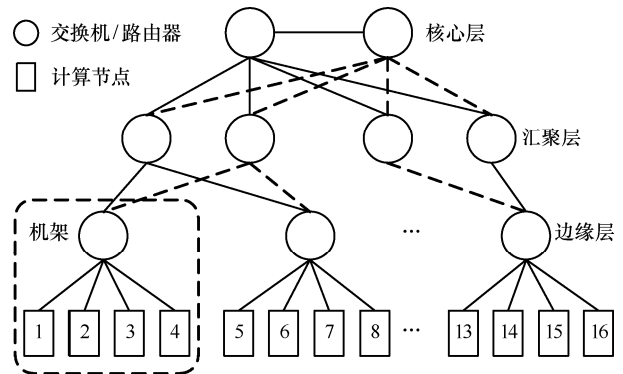


图 3 传统的树型网络拓扑

以 PortLand 为例来说明网络拓扑结构，如图 4 所示。PortLand 借鉴了 Fat-Tree 拓扑的思想，可以由 $5k^2/4$ 个 k 口交换机连接 $k^3/4$ 个计算节点。PortLand 由边缘层、汇聚层、核心层构成。其中边缘层和汇聚层可分解为若干 Pod，每一个 Pod 含 k 台交换机，分属边缘层和汇聚层（每层 $k/2$ 台交换机）。Pod 内部以完全二分图的结构相连。边缘层交换机连接计算节点，每个 Pod 可连接 $k^2/4$ 个计算节点。汇聚层交换机连接核心层交换机，每个 Pod 连接 $k^2/4$ 台核心层交换机。基于 PortLand，可以保证任意两点之间有多条通路，计算节点在任何时刻两两之间可无阻塞通信，从而满足云计算数据中心高可靠性、高带宽的需求。同时，PortLand 可以利用小型交换机连接大规模计算节点，既带来良好的可扩展性，又降低了数据中心的建设成本。

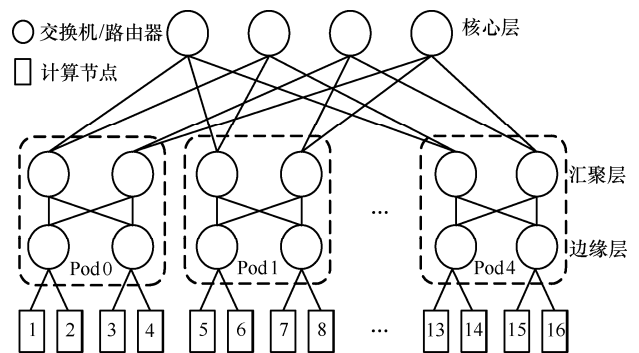


图 4 PortLand 网络拓扑

② 数据中心节能技术

云计算数据中心规模庞大，为了保证设备正常工作，需要消耗大量的电能。据估计，一个拥有 50 000 个计算节点的数据中心每年耗电量超过 1 亿千瓦时，电费达到 930 万美元^[13]。因此需要研究有效的绿色节能技术，以解决能耗开销问题。实施绿色节能技术，

不仅可以降低数据中心的运行开销,而且能减少二氧化碳的排放,有助于环境保护。

当前,数据中心能耗问题得到工业界和学术界广泛关注。Google 的分析表明,云计算数据中心的能源开销主要来自计算机设备、不间断电源、供电单元、冷却装置、新风系统、增湿设备及附属设施(如照明、电动门等)^[18]。如图 5 所示,IT 设备和冷却装置的能耗比重较大。因此,需要首先针对 IT 设备能耗和制冷系统进行研究,以优化数据中心的能耗总量或在性能与能耗之间寻求最佳的折衷。针对 IT 设备能耗优化问题,Nathuji 等人提出一种面向数据中心虚拟化的自适应能耗管理系统 Virtual Power,该系统通过集成虚拟化平台自身具备的能耗管理策略,以虚拟机为单位为数据中心提供一种在线能耗管理能力^[19]。Pallipadi 等人根据 CPU 利用率,控制和调整 CPU 频率以达到优化 IT 设备能耗的目的^[20]。Rao 等人研究多电力市场环境中,如何在保证服务质量前提下优化数据中心能耗总量的问题^[21]。针对制冷系统能耗优化问题,Samadiani 等人综合考虑空间大小、机架和风扇的摆放以及空气的流动方向等因素,提出一种多层次的数据中心冷却设备设计思路,并对空气流和热交换进行建模和仿真,为数据中心布局提供理论支持^[22]。此外,数据中心建成以后,可采用动态制冷策略降低能耗。例如对于处于休眠的服务器,可适当关闭一些制冷设施或改变冷气流的走向,以节约成本^[23]。

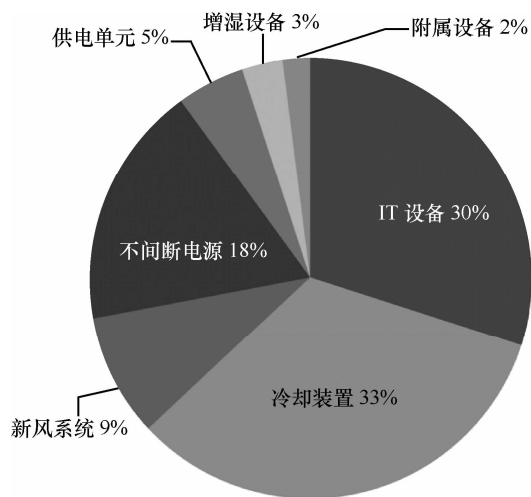


图 5 数据中心的能耗分布

3.1.2 虚拟化技术

数据中心为云计算提供了大规模资源。为了实

现基础设施服务的按需分配,需要研究虚拟化技术。虚拟化是 IaaS 层的重要组成部分,也是云计算的最重要特点。虚拟化技术可以提供以下特点。

1) 资源分享。通过虚拟机封装用户各自的运行环境,有效实现多用户分享数据中心资源。

2) 资源定制。用户利用虚拟化技术,配置私有的服务器,指定所需的 CPU 数目、内存容量、磁盘空间,实现资源的按需分配。

3) 细粒度资源管理。将物理服务器拆分成若干虚拟机,可以提高服务器的资源利用率,减少浪费,而且有助于服务器的负载均衡和节能。

基于以上特点,虚拟化技术成为实现云计算资源池化和按需服务的基础。为了进一步满足云计算弹性服务和数据中心自治性的需求,需要研究虚拟机快速部署和在线迁移技术。

1) 虚拟机快速部署技术

传统的虚拟机部署分为 4 个阶段:创建虚拟机;安装操作系统与应用程序;配置主机属性(如网络、主机名等);启动虚拟机。该方法部署时间较长,达不到云计算弹性服务的要求。尽管可以通过修改虚拟机配置(如增减 CPU 数目、磁盘空间、内存容量)改变单台虚拟机性能,但是更多情况下云计算需要快速扩张虚拟机集群的规模。

为了简化虚拟机的部署过程,虚拟机模板技术^[24]被应用于大多数云计算平台。虚拟机模板预装了操作系统与应用软件,并对虚拟设备进行了预配置,可以有效减少虚拟机的部署时间。然而虚拟机模板技术仍不能满足快速部署的需求:一方面,将模板转换成虚拟机需要复制模板文件,当模板文件较大时,复制的时间开销不可忽视;另一方面,因为应用程序没有加载到内存,所以通过虚拟机模板转换的虚拟机需要在启动或加载内存镜像后,方可提供服务。为此,有学者提出了基于 fork 思想的虚拟机部署方式^[25,26]。该方式受操作系统的 fork 原语启发,可以利用父虚拟机迅速克隆出大量子虚拟机。与进程级的 fork 相似,基于虚拟机级的 fork,子虚拟机可以继承父虚拟机的内存状态信息,并在创建后即时可用。当部署大规模虚拟机时,子虚拟机可以并行创建,并维护其独立的内存空间,而不依赖于父虚拟机。为了减少文件的复制开销,虚拟机 fork 采用了“写时复制”(COW, copy-on-write)技术:子虚拟机在执行“写操作”时,将更新后的文件写入本机磁盘;在执行“读操作”时,通过判

断该文件是否已被更新，确定本机磁盘或父虚拟机的磁盘读取文件。在虚拟机 fork 技术的相关研究工作中，Potemkin 项目实现了虚拟机 fork 技术^[25]，并可在 1s 内完成虚拟机的部署或删除，但要求父虚拟机和子虚拟机在相同的物理机上。Lagar-Cavilla 等人研究了分布式环境下的并行虚拟机 fork 技术^[26]，该技术可以在 1s 内完成 32 台虚拟机的部署。虚拟机 fork 是一种即时（on-demand）部署技术，虽然提高了部署效率，但通过该技术部署的子虚拟机不能持久化保存。

2) 虚拟机在线迁移技术

虚拟机在线迁移是指虚拟机在运行状态下从一台物理机移动到另一台物理机。虚拟机在线迁移技术对云计算平台有效管理具有重要意义。

① 提高系统可靠性。一方面，当物理机需要维护时，可以将运行于该物理机的虚拟机转移到其他物理机。另一方面，可利用在线迁移技术完成虚拟机运行时备份，当主虚拟机发生异常时，可将服务无缝切换至备份虚拟机。

② 有利于负载均衡。当物理机负载过重时，可以通过虚拟机迁移达到负载均衡，优化数据中心性能。

③ 有利于设计节能方案。通过集中零散的虚拟机，可使部分物理机完全空闲，以便关闭这些物理机（或使物理机休眠），达到节能目的。

此外，虚拟机的在线迁移对用户透明，云计算平台可以在不影响服务质量的情况下优化和管理数据中心。在线迁移技术^[27]于 2005 年由 Clark 等人提出，通过迭代的预复制（pre-copy）策略同步迁移前后的虚拟机的状态。传统的虚拟机迁移是在 LAN 中进行的，为了在数据中心之间完成虚拟机在线迁移，Hirofuchi 等人介绍了一种在 WAN 环境下的迁移方法^[28]。这种方法在保证虚拟机数据一致性的前提下，尽可能少地牺牲虚拟机 I/O 性能，加快迁移速度。利用虚拟机在线迁移技术，Remus 系统设计了虚拟机在线备份方法^[29]。当原始虚拟机发生错误时，系统可以立即切换到备份虚拟机，而不会影响到关键任务的执行，提高了系统可靠性。

3.1.3 典型的 IaaS 层平台

本节介绍了 3 种典型的 IaaS 平台，包括 Amazon EC2、Eucalyptus 和东南大学云计算平台。

Amazon 弹性计算云（EC2，elastic computing

cloud）为公众提供基于 Xen 虚拟机的基础设施服务。Amazon EC2 的虚拟机分为标准型、高内存型、高性能型等多种类型，每一种类型的价格各不相同。用户可以根据自身应用的特点与虚拟机价格，定制虚拟机的硬件配置和操作系统。Amazon EC2 的计费系统根据用户的使用情况（一般为使用时间）对用户收费。在弹性服务方面，Amazon EC2 可以根据用户自定义的弹性规则，扩张或收缩虚拟机集群规模。目前，Amazon EC2 已拥有 Ericsson、Active.com、Autodesk 等大量用户。

Eucalyptus 是加州大学圣巴巴拉分校开发的开源 IaaS 平台。区别于 Amazon EC2 等商业 IaaS 平台，Eucalyptus 的设计目标是成为研究和发展的云计算的基础平台。为了实现这个目标，Eucalyptus 的设计强调开源化、模块化，以便研究者对各功能模块升级、改造和更换。目前，Eucalyptus 已实现了和 Amazon EC2 相兼容的 API，并部署于全球各地的研究机构。

东南大学云计算平台面向计算密集型和数据密集型应用，由 3 500 颗 CPU 内核和 500TB 高速存储设备构成，提供 37 万亿次浮点计算能力。其基础设施服务不仅支持 Xen、KVM、VMware 等虚拟化技术，而且支持物理计算节点的快速部署，可根据科研人员的应用需求，为其按需配置物理的或虚拟的私有计算集群，并自动安装操作系统、应用软件。由于部分高性能计算应用对网络延时敏感，其数据中心利用 40Gbit/s QDR InfiniBand 作为数据传输网络，提供高带宽低延时的网络服务。目前，东南大学云计算平台承担了 AMS-02（见第 5 节）数据分析处理、电磁仿真、分子动力学模拟等科学计算任务。

3.2 PaaS

PaaS 层作为 3 层核心服务的中间层，既为上层应用提供简单、可靠的分布式编程框架，又需要基于底层的资源信息调度作业、管理数据，屏蔽底层系统的复杂性。随着数据密集型应用的普及和数据规模的日益庞大，PaaS 层需要具备存储与处理海量数据的能力。本节先介绍 PaaS 层的海量数据存储与处理技术，然后讨论基于这些技术的资源管理与调度策略。

3.2.1 海量数据存储与处理技术

1) 海量数据存储技术

云计算环境中的海量数据存储既要考虑存储

系统的 I/O 性能，又要保证文件系统的可靠性与可用性。

Ghemawat 等人为 Google 设计了 GFS (google file system) [30]。根据 Google 应用的特点，GFS 对其应用环境做了 6 点假设：①系统架设在容易失效的硬件平台上；②需要存储大量 GB 级甚至 TB 级的大文件；③文件读操作以大规模的流式读和小规模的随机读构成；④文件具有一次写多次读的特点；⑤系统需要有效处理并发的追加写操作；⑥高持续 I/O 带宽比低传输延迟重要。

图 6 展示了 GFS 的执行流程。在 GFS 中，一个大文件被划分成若干固定大小（如 64MB）的数据块，并分布在计算节点的本地硬盘，为了保证数据可靠性，每一个数据块都保存有多个副本，所有文件和数据块副本的元数据由元数据管理节点管理。GFS 的优势在于：①由于文件的分块粒度大，GFS 可以存取 PB 级的超大文件；②通过文件的分布式存储，GFS 可并行读取文件，提供高 I/O 吞吐率；③鉴于上述假设 4，GFS 可以简化数据块副本间的数据同步问题；④文件块副本策略保证了文件可靠性。

Bigtable 是基于 GFS 开发的分布式存储系统[31]，它将提高系统的适用性、可扩展性、可用性和存储

性能作为设计目标。Bigtable 的功能与分布式数据库类似，用以存储结构化或半结构化数据，为 Google 应用（如搜索引擎、Google Earth 等）提供数据存储与查询服务。在数据管理方面，Bigtable 将一整张数据表拆分成许多存储于 GFS 的子表，并由分布式锁服务 Chubby 负责数据一致性管理。在数据模型方面，Bigtable 以行名、列名、时间戳建立索引，表中的数据项由无结构的字节数组表示。这种灵活的数据模型保证 Bigtable 适用于多种不同应用环境。图 7 展示了如何在 Bigtable 中存储网页，其中 $t_1 \sim t_5$ 为时间戳。

由于 Bigtable 需要管理节点集中管理元数据，所以存在性能瓶颈和单点失效问题。为此，DeCandia 等人设计了基于 P2P 结构的 Dynamo 存储系统[32]，并应用于 Amazon 的数据存储平台。借助于 P2P 技术的特点，Dynamo 允许使用者根据工作负载动态调整集群规模。另外，在可用性方面，Dynamo 采用零跳分布式散列表结构降低操作响应时间；在可靠性方面，Dynamo 利用文件副本机制应对节点失效。由于保证副本强一致性会影响系统性能，所以，为了承受每天数千万的并发读写请求，Dynamo 中设计了最终一致性模型，弱化副本一致性，保证提高性能。

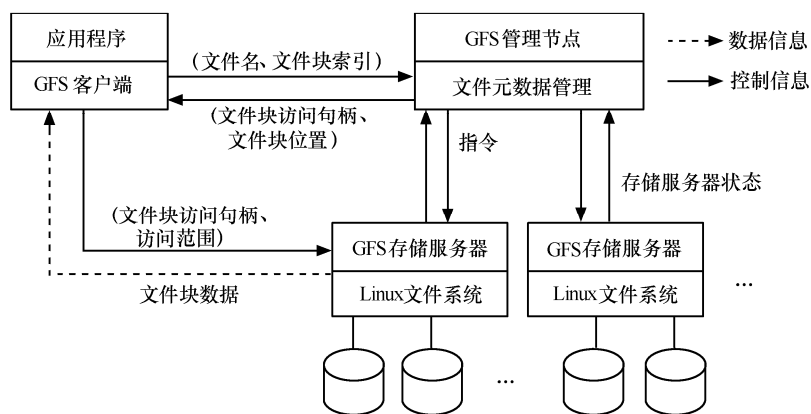


图 6 GFS 执行流程

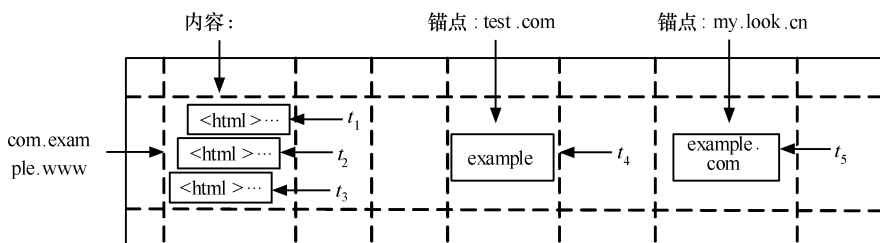


图 7 Bigtable 的存储方式

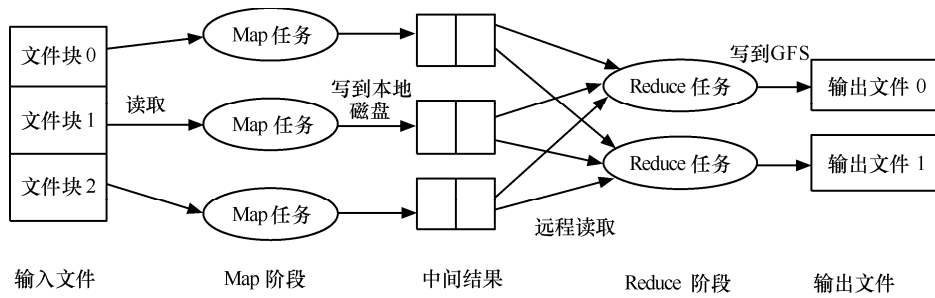


图 8 MapReduce 的执行过程

2) 数据处理技术与编程模型

PaaS 平台不仅要实现海量数据的存储,而且要提供面向海量数据的分析处理功能。由于 PaaS 平台部署于大规模硬件资源上,所以海量数据的分析处理需要抽象处理过程,并要求其编程模型支持规模扩展,屏蔽底层细节并且简单有效。

MapReduce 是 Google 提出的并程序编程模型,运行于 GFS 之上。如图 8 所示,一个 MapReduce 作业由大量 Map 和 Reduce 任务组成,根据两类任务的特点,可以把数据处理过程划分成 Map 和 Reduce 2 个阶段:在 Map 阶段,Map 任务读取输入文件块,并行分析处理,处理后的中间结果保存在 Map 任务执行节点;在 Reduce 阶段,Reduce 任务读取并合并多个 Map 任务的中间结果。MapReduce 可以简化大规模数据处理的难度:首先,MapReduce 中的数据同步发生在 Reduce 读取 Map 中间结果的阶段,这个过程由编程框架自动控制,从而简化数据同步问题;其次,由于 MapReduce 会监测任务执行状态,重新执行异常状态任务,所以程序员不需考虑任务失败问题;再次,Map 任务和 Reduce 任务都可以并发执行,通过增加计算节点数量便可加快处理速度;最后,在处理大规模数据时,Map/Reduce 任务的数目远多于计算节点的数目^[6],有助于计算节点负载均衡。

虽然 MapReduce 具有诸多优点,但仍具有局限性:①MapReduce 灵活性低,很多问题难以抽象成 Map 和 Reduce 操作;②MapReduce 在实现迭代算法时效率较低;③MapReduce 在执行多数据集的交运算时效率不高。为此, Sawzall 语言^[33]和 Pig 语言^[34]封装了 MapReduce,可以自动完成数据查询操作到 MapReduce 的映射;Ekanayake 等人设计了 Twister 平台^[35],使 MapReduce 有效支持迭代操作;Yang 等人设计了 Map-Reduce-Merge 框架^[36],通过加入 Merge 阶段实现多数据集的交操作。在此基

础上, Wang 等人将 Map-Reduce-Merge 框架应用于构建 OLAP 数据立方体^[37];文献[38, 39]将 MapReduce 应用到并行求解大规模组合优化问题(如并行遗传算法)。

由于许多问题难以抽象成 MapReduce 模型,为了使并行编程框架灵活普适, Isard 等人设计 Dryad 框架^[40]。Dryad 采用了基于有向无环图(DAG, directed acyclic graph)的并行模型(如图 9 所示)。在 Dryad 中,每一个数据处理作业都由 DAG 表示,图中的每一个节点表示需要执行的子任务,节点之间的边表示 2 个子任务之间的通信。Dryad 可以直观地表示出作业内的数据流。基于 DAG 优化技术, Dryad 可以更加简单高效地处理复杂流程。同 MapReduce 相似, Dryad 为程序开发者屏蔽了底层的复杂性,并可在计算节点规模扩展时提高处理性能。在此基础上, Yu 等人设计了 DryadLINQ 数据查询语言^[41],该语言和 .NET 平台无缝结合,并利用 Dryad 模型对 Azure^[42]平台上的数据进行查询处理。

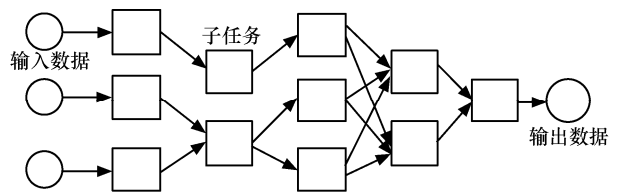


图 9 Dryad 的任务模型

3.2.2 资源管理与调度技术

海量数据处理平台的大规模性给资源管理与调度带来挑战。研究有效的资源管理与调度技术可以提高 MapReduce、Dryad 等 PaaS 层海量数据处理平台的性能。

1) 副本管理技术

副本机制是 PaaS 层保证数据可靠性的基础,有效的副本策略不但可以降低数据丢失的风险,而

且能优化作业完成时间。目前，Hadoop（见 3.2.3 节）采用了机架敏感的副本放置策略。该策略默认文件系统部署于传统网络拓扑的数据中心。以放置 3 个文件副本为例，由于同一机架的计算节点间网络带宽高，所以机架敏感的副本放置策略将 2 个文件副本置于同一机架，另一个置于不同机架。这样的策略既考虑了计算节点和机架失效的情况，也减少了因为数据一致性维护带来的网络传输开销。除此之外，文件副本放置还与应用有关，Eltabakh 等人提出了一种灵活的数据放置策略 CoHadoop^[43]，用户可以根据应用需求自定义文件块的存放位置，使需要协同处理的数据分布在相同的节点上，从而在一定程度上减少了节点之间的数据传输开销。但是，目前 PaaS 层的副本调度大多局限于单数据中心，从容灾备份和负载均衡角度，需要考虑面向多数据中心的副本管理策略。郑湃等人提出了三阶段数据布局策略^[44]，分别针对跨数据中心数据传输、数据依赖关系和全局负载均衡 3 个目标对数据布局方案进行求解和优化。虽然该研究对多数据中心间的数据管理起到优化作用，但是未深入讨论副本管理策略。因此，需在多数据中心环境下研究副本放置、副本选择及一致性维护和更新机制。

2) 任务调度算法

PaaS 层的海量数据处理以数据密集型作业为主，其执行性能受到 I/O 带宽的影响。但是，网络带宽是计算集群（计算集群既包括数据中心中物理计算节点集群，也包括虚拟机构建的集群）中的紧缺的资源：①云计算数据中心考虑成本因素，很少采用高带宽的网络设备；②IaaS 层部署的虚拟机集群共享有限的网络带宽；③海量数据的读写操作占用了大量带宽资源。因此 PaaS 层海量数据处理平台的任务调度需要考虑网络带宽因素。

为了减少任务执行过程中的网络传输开销，可以将任务调度到输入数据所在的计算节点，因此，需要研究面向数据本地性（data-locality）的任务调度算法。Hadoop 以“尽力而为”的策略保证数据本地性。虽然该算法易于实现，但是没有做到全局优化，在实际环境中不能保证较高的数据本地性。为了达到全局优化，Fischer 等人为 MapReduce 任务调度建立数学模型^[45]，并提出了 HTA（Hadoop task assignment）问题。该问题为一个变形的二部图匹配（如图 10 所示），目标是将任务分配到计算节点，并使各计算节点负载均衡，其中 s_i 、 t_j 分别表示计

算节点和任务，实边表示 s_i 有 t_j 的输入数据，虚边表示 s_i 没有 t_j 的输入数据， w_l 和 w_r 分别表示调度开销。该研究利用 3-SAT 问题证明了 HTA 问题是 NP 完全的，并设计了 MaxCover-BalAssign 算法解决该问题。虽然 MaxCover-BalAssign 算法的理论上限接近最优解，但是时间复杂度过高，难以应用在大规模环境中。为此，Jin 等人设计了 BAR 调度算法^[46]，基于“先均匀分配再均衡负载”的思想，BAR 算法在快速求解大规模 HTA 问题同时，得到优于 MaxCover-BalAssign 算法的调度结果。

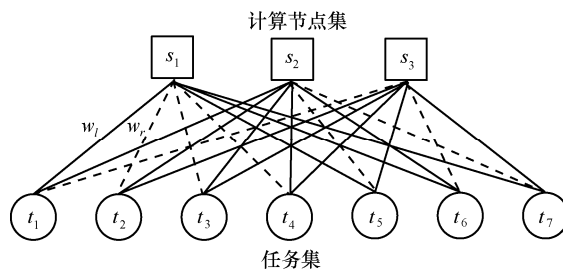


图 10 HTA 问题模型

除了保证数据本地性，PaaS 层的作业调度器还需要考虑作业之间的公平调度。PaaS 层的工作负载中既包括子任务少、执行时间短、对响应时间敏感的即时作业（如数据查询作业），也包括子任务多、执行时间长的长期作业（如数据分析作业）。研究公平调度算法可以及时为即时作业分配资源，使其快速响应。因为数据本地性和作业公平性不能同时满足，所以 Zaharia 等人在 Max-Min 公平调度算法的基础上设计了延迟调度（delay scheduling）算法^[47]。该算法通过推迟调度一部分作业并使这些作业等待合适的计算节点，以达到较高的数据本地性。但是在等待开销较大的情况下，延迟策略会影响作业完成时间。为了折衷数据本地性和作业公平性，Isard 等设计了基于最小代价流的调度模型^[48]，并应用于 Microsoft 的 Azure 平台。如图 11 所示， w_j^i 表示作业 i 的子任务 j ， w_j^i 和 C, R, X, U 的边分别表示任务被调度到计算节点、机架、数据中心和不被调度，每条边带有权值，并需要根据集群状态实时更新权值。当系统状态发生改变时（如有计算节点空闲、有新任务加入），调度器对调度图求解最小代价流，并做出调度决策。虽然该方法可以得到全局优化的调度结果，但是求解最小代价流会带来计算开销，当图的规模很大时，计算开销将严重影响系统性能。

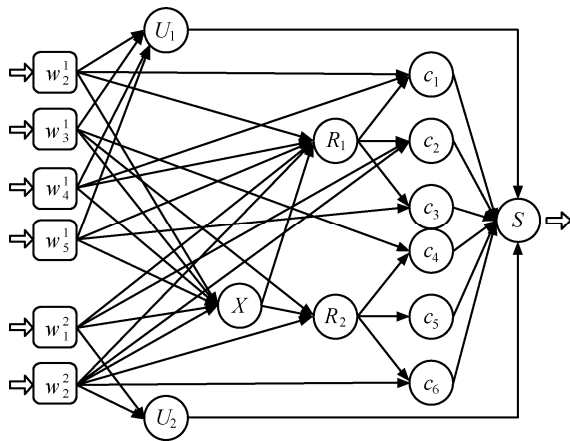


图 11 Quincy 的调度模型

3) 任务容错机制

为了使 PaaS 平台可以在任务发生异常时自动从异常状态恢复，需要研究任务容错机制。MapReduce 的容错机制在检测到异常任务时，会启动该任务的备份任务。备份任务和原任务同时进行，当其中一个任务顺利完成时，调度器立即结束另一个任务。Hadoop 的任务调度器实现了备份任务调度策略。但是现有的 Hadoop 调度器检测异常任务的算法存在较大缺陷：如果一个任务的进度落后于同类型任务进度的 20%，Hadoop 则把该任务当做异常任务，然而，当集群异构时，任务之间的执行进度差异较大，因而在异构集群中很容易产生大量的备份任务。为此，Zaharia 等人研究了异构环境下异常任务的发现机制，并设计了 LATE (longest approximate time to end) 调度器^[49]。通过估算 Map 任务的完成时间，LATE 为估计完成时间最晚的任务产生备份。虽然 LATE 可以有效避免产生过多的备份任务，但是该方法假设 Map 任务处理速度是稳定的，所以在 Map 任务执行速度变化的情况下（如先快后慢），LATE 便不能达到理想的性能。

3.2.3 典型的 PaaS 平台

本节介绍了 3 种典型的 PaaS 平台：Google App Engine、Hadoop 和 Microsoft Azure。这些平台都基于海量数据处理技术搭建，且各具代表性。图 12 比较了上述 PaaS 平台所采用的关键技术。

Google App Engine 是基于 Google 数据中心的开发、托管 Web 应用程序的平台。通过该平台，程序开发者可以构建规模可扩展的 Web 应用程序，而不用考虑底硬件基础设施的管理。App Engine 由 GFS 管理数据、MapReduce 处理数据，并用 Sawzall 为编程语言提供接口。

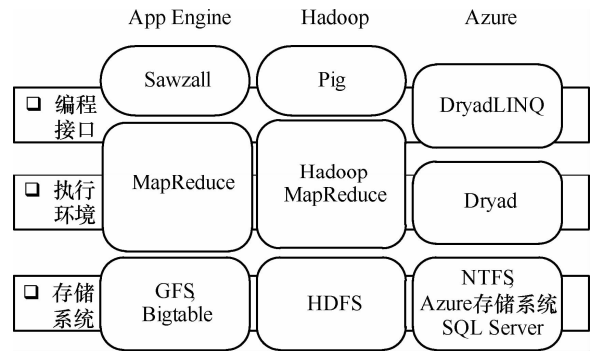


图 12 典型 PaaS 平台的比较

Hadoop 是开源的分布式处理平台，其 HDFS、Hadoop MapReduce 和 Pig 模块实现了 GFS、MapReduce 和 Sawzall 等数据处理技术。与 Google 的分布式处理平台相似，Hadoop 在可扩展性、可靠性、可用性方面做了优化，使其适用于大规模的云环境。目前，Hadoop 由 Apache 基金会维护，Yahoo!、Facebook、淘宝等公司利用 Hadoop 构建数据处理平台，以满足海量数据分析处理需求。

Microsoft Azure 以 Dryad 作为数据处理引擎，允许用户在 Microsoft 的数据中心上构建、管理、扩展应用程序。目前，Azure 支持按需付费，并免费提供 750h 的计算时长和 1GB 数据库空间，其服务范围已经遍布 41 个国家和地区。

3.3 SaaS

SaaS 层面向的是云计算终端用户，提供基于互联网的软件应用服务。随着 Web 服务、HTML5、Ajax、Mashup 等技术的成熟与标准化，SaaS 应用近年来发展迅速。

典型的 SaaS 应用包括 Google Apps、Salesforce CRM 等。

Google Apps 包括 Google Docs、GMail 等一系列 SaaS 应用。Google 将传统的桌面应用程序（如文字处理软件、电子邮件服务等）迁移到互联网，并托管这些应用程序。用户通过 Web 浏览器便可随时随地访问 Google Apps，而不需要下载、安装或维护任何硬件或软件。Google Apps 为每个应用提供了编程接口，使各应用之间可以随意组合。Google Apps 的用户既可以是个人用户也可以是服务提供商。比如企业可向 Google 申请域名为 @example.com 的邮件服务，满足企业内部收发电子邮件的需求。在此期间，企业只需对资源使用量付费，而不必考虑购置、维护邮件服务器、邮件管

理系统的开销。

Salesforce CRM 部署于 Force.com 云计算平台，为企业提供客户关系管理服务，包括销售云、服务云、数据云等部分。通过租用 CRM 的服务，企业可以拥有完整的企业管理系统，用以管理内部员工、生产销售、客户业务等。利用 CRM 预定义的服务组件，企业可以根据自身业务的特点定制工作流程。基于数据隔离模型，CRM 可以隔离不同企业的数

3.4 服务管理层

为了使云计算核心服务高效、安全地运行，需要服务管理技术加以支持。服务管理技术包括 QoS 保证机制、安全与隐私保护技术、资源监控技术、服务计费模型等。其中，QoS 保证机制和安全与隐私保护技术是保证云计算可靠性、可用性、安全性的基础。为此，本节着重介绍 QoS 保证机制和安全与隐私保护技术的研究现状。

3.4.1 QoS 保证机制

云计算不仅要为用户提供满足应用功能需求的资源和服务，同时还需要提供优质的 QoS（如可用性、可靠性、可扩展、性能等），以保证应用顺利高效地执行。这是云计算得以被广泛采纳的基础。图 13 给出了云计算中 QoS 保证机制^[50]，首先，用户从自身应用的业务逻辑层面提出相应的 QoS 需求；为了能够在使用相应服务的过程中始终满足用户的需求，云计算服务提供商需要对 QoS 水平进行匹配并且与用户协商制定服务水平协议；最后，根据 SLA 内容进行资源分配以达到 QoS 保证的目的。针对以上 3 个步骤，本节依次介绍 IaaS、PaaS 和 SaaS 中的 QoS 保证机制。

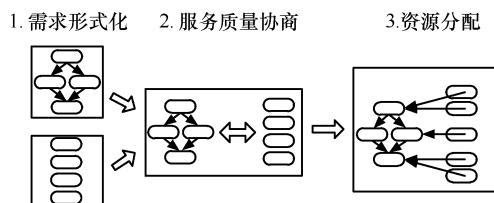


图 13 QoS 保证机制

1) IaaS 层的 QoS 保证机制

IaaS 层可看作是一个资源池，其中包括可定制

的计算、网络、存储等资源，并根据用户需求按需提供相应的服务能力。文献[51]指出，IaaS 层所关心 QoS 参数主要可分为两类：一类是云计算服务提供商所提供的系统最小服务质量，如服务器可用性

及网络性能等；另一类服务提供商承诺的服务响应时间。为了能够在服务运行过程中有效保证其性能，IaaS 层用户需要针对 QoS 参数同云计算服务提供商签订相应的 SLA。根据应用类型不同可分为两类：确定性 SLA (deterministic SLA) 以及可能性 SLA (probabilistic SLA)。其中确定性 SLA 主要针对关键性核心服务，这类服务通常需要十分严格的性能保证（如银行核心业务等），因此需要 100% 确保其相应的 QoS 需求。对于可能性 SLA，其通常采用可用性百分比表示（如保证硬件每月 99.95% 的时间正常运行），这类服务通常并不需要十分严格的 QoS 保证，主要适用于中小型商业模式及企业级应用。在签订完 SLA 后，若服务提供商未按照 SLA 进行 QoS 保障时，则对服务提供商启动惩罚机制（如赔款），以补偿对用户造成的损失。

在实际系统方面，近年来出现了若干通过 SLA 技术实现 IaaS 层 QoS 保证机制的商用云计算系统或平台，主要包括 Amazon EC2、GoGrid^{注1}、Rackspace^{注2}等，其 QoS 参数如表 2 所示。

2) PaaS/SaaS 层的 QoS 保证机制

在云计算环境中，PaaS 层主要负责提供云计算应用程序（服务）的运行环境及资源管理。SaaS 提供以服务为形式的应用程序。与 IaaS 层的 QoS 保证机制相似，PaaS 层和 SaaS 层的 QoS 保证也需要经历 3 个阶段，典型的 QoS 参数如表 3 所示。PaaS 层和 SaaS 层 QoS 保证的难点在第 3 阶段（资源分配阶段）。由于在云计算环境中，应用服务提供商同底层硬件服务提供商之间可以是松耦合的，所以 PaaS 层和 SaaS 层在第 3 阶段需要综合考虑 IaaS 层的费用、IaaS 层承诺的 QoS、PaaS/SaaS 层服务对用户承诺的 QoS 等。

为此，本节介绍 PaaS 层和 SaaS 层的资源分配策略。为了便于讨论，本节中 PaaS 层和 SaaS 层统称为应用服务层。弹性服务是云计算的特性之一，为了保证服务的可用性，应用服务层需要根据业务

注 1 <http://www.gogrid.com>

注 2 <http://www.rackspace.com>

表 2 IaaS 层的 QoS 参数定义

QoS 参数(IaaS)	描述	云计算服务
服务器可用性	云计算提供的服务器（如虚拟机）、存储系统正常运行的保障	GoGrid、Rackspace、Amazon EC2
网络性能保障	数据包丢失率、网络时延、网络抖动的保障	GoGrid
负载均衡器可用性	处理时延、吞吐率、访问并发率的保障	GoGrid、Rackspace
异常通知保障	在发生基础设施异常（如虚拟机需迁移）时的通知时间	Rackspace
支持响应时间	当服务发生异常时，云服务提供商提供排错支持服务的响应时间	GoGrid
惩罚机制保障	不能按照 SLA 合约进行 QoS 保障时的惩罚机制	GoGrid、Rackspace、Amazon EC2

表 3 PaaS/SaaS 层的 QoS 参数定义

QoS 参数(PaaS/SaaS)	描述	云计算服务
服务请求差错率	单位时间内服务请求发生异常的概率	Google App Engine
网络连接可用性	网络连接畅通且不被中断的可用性保障	Microsoft Azure
服务稳定性	某个用户正常使用服务且服务不失效的稳定性保障	Microsoft Azure、Google Apps、Salesforce CRM
惩罚机制保障	不能按照 SLA 合约进行 QoS 保障时的惩罚机制	Microsoft Azure、Google Apps、Salesforce CRM、Google App Engine

负载动态申请或释放 IaaS 层的资源。Calheiros 等基于排队论设计了负载预测模型，通过比较硬件设施工作负载、用户请求负载及 QoS 目标，调整虚拟机的数量^[52]。由于同类 IaaS 层服务可能由多个服务提供商提供，应用服务提供商需要根据 QoS 协定选择合适的 IaaS 层服务。为此，Xiao 等人设计了基于信誉的 QoS 部署机制^[53]，该机制综合考虑 IaaS 层服务层提供商的信誉、应用服务同用户的 SLA 以及 QoS 的部署开销，选择合适的 IaaS 层服务。除此之外，由于 Amazon EC2 的 Spot Instance 服务可以以竞价方式提供廉价的虚拟机，Andrzejak 等人为应用服务层设计了的竞价模型，使其在满足用户 QoS 需求的前提下降低硬件设施开销^[54]。

3.4.2 安全与隐私保护

虽然通过 QoS 保证机制可以提高云计算的可靠性和可用性，但是目前实现高安全性的云计算环境仍面临诸多挑战。一方面，云平台上的应用程序（或服务）同底层硬件环境间是松耦合的，没有固定不变的安全边界，大大增加了数据安全性与隐私保护的难度。另一方面，云计算环境中的数据量十分巨大（通常都是 TB 甚至 PB 级），传统安全机制在可扩展性及性能方面难以有效满足需求。随着云计算的安全问题日益突出，近年来研究者针对云计算的模型和应用，讨论了云计算安全隐患，研究了云计算环境下的数据安全与隐私保护技术。本节结合云计算核心服务的层次模型，介绍云计算环境下的数据安全与隐私保护技

术的研究现状。

1) IaaS 层的安全

虚拟化是云计算 IaaS 层普遍采用的技术。该技术不仅可以实现资源可定制，而且能有效隔离用户的资源。Santhanam 等人讨论了分布式环境下基于虚拟机技术实现的“沙盒”模型，以隔离用户执行环境^[55]。然而虚拟化平台并不是完美的，仍然存在安全漏洞。基于 Amazon EC2 上的实验，Ristenpart 等人发现 Xen 虚拟化平台存在被旁路攻击的危险^[56]。他们在云计算中心放置若干台虚拟机，当检测到有一台虚拟机和目标虚拟机放置在同一台主机上时，便可通过操纵自己放置的虚拟机对目标虚拟机进行旁路攻击，得到目标虚拟机的更多信息。为了避免基于 Cache 缓存的旁路攻击，Raj 等人提出了 Cache 层次敏感的内核分配方法和基于页染色的 Cache 划分 2 种资源管理方法，以实现性能与安全隔离^[57]。

2) PaaS 层的安全

PaaS 层的海量数据存储和处理需要防止隐私泄露问题。Roy 等人提出了一种基于 MapReduce 平台的隐私保护系统 Airavat^[58]，集成强访问控制和区分隐私，为处理关键数据提供安全和隐私保护。在加密数据的文本搜索方面，传统的方法需要对关键词进行完全匹配，但是云计算数据量非常大，在用户频繁访问的情况下，精确匹配返回的结果会非常少，使得系统的可用性大幅降低，Li 等人提出了

基于模糊关键词的搜索方法^[59]，在精确匹配失败后，还将采取与关键词近似语义的关键词集的匹配，达到在隐私保护的前提下为用户检索更多匹配文件的效果。

3) SaaS 层的安全

SaaS 层提供了基于互联网的应用程序服务，并会保存敏感数据（如企业商业信息）。因为云服务器由许多用户共享，且云服务器和用户不在同一个信任域里，所以需要敏感数据建立访问控制机制。由于传统的加密控制方式需要花费很大的计算开销，而且密钥发布和细粒度的访问控制都不适合大规模的数据管理，Yu 等人讨论了基于文件属性的访问控制策略^[60]，在不泄露数据内容的前提下将与访问控制相关的复杂计算工作交给不可信的云服务器完成，从而达到访问控制的目的。

从以上研究可以看出，云计算面临的核心安全问题是用户不再对数据和环境拥有完全的控制权。为了解决该问题，云计算的部署模式被分为公有云、私有云和混合云。

公有云是以按需付费方式向公众提供的云计算服务（如 Amazon EC2、Salesforce CRM 等）。虽然公有云提供了便利的服务方式，但是由于用户数据保存在服务提供商，存在用户隐私泄露、数据安全得不到保证等问题。

私有云是一个企业或组织内部构建的云计算系统。部署私有云需要企业新建私有的数据中心或改造原有数据中心。由于服务提供商和用户同属于一个信任域，所以数据隐私可以得到保护。受其数据中心规模的限制，私有云在服务弹性方面与公有云相比较差。

混合云结合了公有云和私有云的特点：用户的关键数据存放在私有云，以保护数据隐私；当私有云工作负载过重时，可临时购买公有云资源，以保证服务质量。部署混合云需要公有云和私有云具有统一的接口标准，以保证服务无缝迁移。

此外，工业界对云计算的安全问题非常重视，并为云计算服务和平台开发了若干安全机制^[61]。其中 Sun 公司发布开源的云计算安全工具可为 Amazon EC2 提供安全保护。微软公司发布基于云计算平台 Azure 的安全方案，以解决虚拟化及底层硬件环境中的安全性问题。另外，Yahoo! 为 Hadoop 集成了 Kerberos 验证，Kerberos 验证

有助于数据隔离，使对敏感数据的访问与操作更为安全。

4 云计算的机遇与挑战

云计算的研究领域广泛，并且与实际生产应用紧密结合。纵观已有的研究成果，还可从以下 2 个角度对云计算做深入研究：①拓展云计算的外沿，将云计算与相关应用领域相结合，本节以移动互联网和科学计算为例，分析新的云计算应用模式及尚需解决的问题；②挖掘云计算的内涵，讨论云计算模型的局限性，本节以端到云的海量数据传输和大规模程序调试诊断为例，阐释云计算面临的挑战。

1) 云计算和移动互联网的结合

云计算和移动互联网的联系紧密，移动互联网的发展丰富了云计算的外沿。由于移动设备的在硬件配置和接入方式上具有特殊性，所以有许多问题值得研究。首先，移动设备的资源是有限的。访问基于 Web 门户的云计算服务往往需要在浏览器端解释执行脚本程序（如 JavaScript、Ajax 等），因此会消耗移动设备的计算资源和能源。虽然为移动设备定制客户端可以减少移动设备的资源消耗，但是移动设备运行平台种类多，更新快，导致定制客户端的成本相对较高。因此需要为云计算设计交互性强、计算量小、普适性强的访问接口。其次是网络接入问题。对于许多 SaaS 层服务来说，用户对响应时间敏感。但是，移动网络的时延比固定网络高，而且容易丢失连接，导致 SaaS 层服务可用性降低。因此，需要针对移动终端的网络特性对 SaaS 层服务进行优化。

2) 云计算与科学计算的结合

科学计算领域希望以经济的方式求解科学问题，云计算可以为科学计算提供低成本的计算能力和存储能力。但是，在云计算平台上进行科学计算面临着效率低的问题^[62]。虽然一些服务提供商推出了面向科学计算的 IaaS 层服务，但是其性能和传统的高性能计算机相比仍有差距。研究面向科学计算的云计算平台首先要从 IaaS 层入手。IaaS 层的 I/O 性能成为影响执行时间的重要因素：①网络时延问题，MPI 并行程序对网络时延比较敏感，传统高性能计算集群采用 InfiniBand 网络降低传输时延，但是目前虚拟机对 InfiniBand 的支持不够，不能满足低时延需求；②I/O 带宽问题，

虚拟机之间需要竞争磁盘和网络 I/O 带宽，对于数据密集型科学计算应用，I/O 带宽的减少会延长执行时间。其次要在 PaaS 层研究面向科学计算的编程模型。虽然 Moretti 等提出了面向数据密集型科学计算的 All-Pairs 编程模型^[63]，但是该模型的原型系统只运行于小规模集群，并不能保证其可扩展性。最后，对于复杂的科学工作流，要研究如何根据执行状态与任务需求动态申请和释放云计算资源，优化执行成本。

3) 端到云的海量数据传输

云计算将海量数据在数据中心进行集中存放，对数据密集型计算应用提供强有力的支持。目前许多数据密集型计算应用需要在端到云之间进行大数据量的传输，如 AMS-02 实验每年将产生约 170TB 的数据量，需要将这些数据传输到云数据中心存储和处理，并将处理后的数据分发到各地研究中心进行下一步的分析。若每年完成 170TB 的数据传输，至少需要 40Mbit/s 的网络带宽，但是这样高的带宽需求很难在当前的互联网中得到满足。另外，按照 Amazon 云存储服务的定价，若每年传输上述数据量，则需花费约数万美元，其中并不包括支付给互联网服务提供商的费用。由此可见，端到云的海量数据传输将耗费大量的时间和经济开销。由于网络性价比的增长速度远远落后于云计算技术的发展速度，目前传输主要通过邮寄方式将存储数据的磁盘直接放入云数据中心，这种方法仍然需要相当的经济开销，并且运输过程容易导致磁盘损坏。为了支持更加高效快捷的端到云的海量数据传输，需要从基础设施层入手研究下一代网络体系结构，改变网络的组织方式和运行模式，提高网络吞吐量。

4) 大规模应用的部署与调试

云计算采用虚拟化技术在物理设备和具体应用之间加入了一层抽象，这要求原有基于底层物理系统的应用必须根据虚拟化做相应的调整才能部署到云计算环境中，从而降低了系统的透明性和应用对底层系统的可控性。另外，云计算利用虚拟技术能够根据应用需求的变化弹性地调整系统规模，降低运行成本。因此，对于分布式应用，开发者必须考虑如何根据负载情况动态分配和回收资源。但该过程很容易产生错误，如资源泄漏、死锁等。上述情况给大规模应用在云计算环境中的部署带来了巨大挑战。为解决这一问题，需要研究适应云计

算环境的调试与诊断开发工具以及新的应用开发模型。

5 东南大学云计算平台

近年来，东南大学在云计算领域进行了许多有效的尝试，也获得了较为丰硕的研究与应用成果。东南大学云计算平台的一个典型应用是 AMS-02 (alpha magnetic spectrometer 02) 海量数据处理。

AMS-02 实验是由诺贝尔物理学奖获得者丁肇中教授领导的由美、俄、德、法、中等 15 个国家和地区共 800 多名科学家参加的大型国际合作项目，其目的是为寻找由反物质所组成的宇宙和暗物质的来源以及测量宇宙线的来源。AMS-02 探测器于 2011 年 5 月 16 日搭乘美国奋进号航天飞机升空至国际空间站，将在国际空间站上运行 10~18 年，其间大量的原始数据将传送到分别设立在瑞士 CERN 和中国东南大学的地面数据处理中心 SOC (scientific operation center)，由地面数据处理中心对其进行传输、存储、处理、计算和分析。

东南大学在针对 AMS-02 海量数据处理的前期理论工作主要集中于网格环境下的自适应任务调度^[64]、分布式资源发现^[65,66]以及副本管理^[67]等方面。然而随着 AMS-02 探测器的升空和运行，AMS-02 实验对实际处理平台提出了更高的要求：首先，AMS-02 实验相关的数据文件规模急剧增加，需要更大规模、更加高效的数据处理平台的支持；其次，该平台需要提供数据访问服务，以满足世界各地的科学家分析海量科学数据的需求。

为了满足 AMS-02 海量数据处理应用的需求，东南大学构建了相应的云计算平台。如图 14 所示，该平台提供了 IaaS、PaaS 和 SaaS 层的服务，IaaS 层的基础设施由 3 500 颗 CPU 内核和容量为 500TB 的磁盘阵列构成，提供虚拟机和物理机的按需分配（见 3.1.3 节）。在 PaaS 层，数据分析处理平台和应用开发环境为大规模数据分析处理应用提供编程接口。在 SaaS 层，以服务的形式部署云计算应用程序，便于用户访问与使用。

对于 AMS-02 海量数据处理应用，东南大学云计算平台提供了如下支持。

首先，云计算平台可根据 AMS-02 实验的需求，为其分配独占的计算集群，并自动配置运行环境

(如操作系统、科学计算函数库等)。利用资源隔离技术,既保证 AMS-02 应用不会受到其他应用的影响,又为 AMS-02 海量数据处理应用中执行程序的更新和调试带来便利。

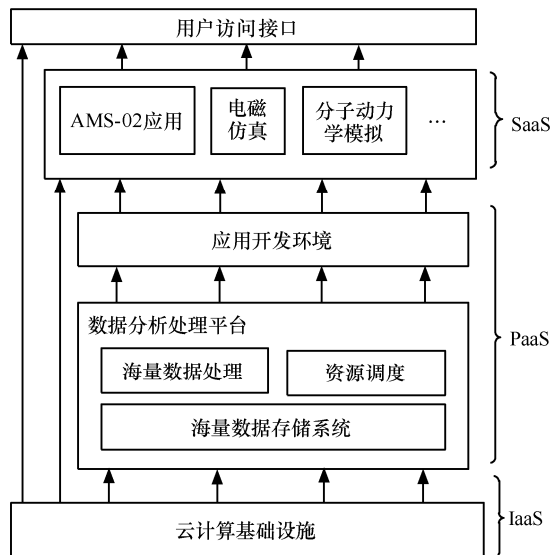


图 14 东南大学云计算平台的架构

其次,世界各国物理学家可通过访问部署于 SaaS 层的 AMS-02 应用服务,得到所需的原始科学数据和处理分析结果,以充分实现数据共享和协同工作。

最后,随着 AMS-02 实验的不断进行,待处理的数据量及数据处理的难度会大幅增加。此时相应的云计算应用开发环境将为 AMS-02 数据分析处理程序提供编程接口,在提供大规模计算和数据存储能力的同时,简化了海量数据处理的难度。

除了 AMS-02 实验之外,东南大学云计算平台针对不同学科院系的应用需求,还分别部署了电磁仿真、分子动力学模拟等科学计算应用。由于这些科学计算应用对计算平台的性能要求较高,因此为了优化云计算平台的运行性能,同时也进行了大量理论研究工作。

针对海量数据处理应用中数据副本的选择问题,在综合考虑副本开销及数据可用性因素的基础上提出一种基于 QoS 偏好感知的副本选择策略^[38],通过实现灵活可靠的副本管理机制以提高应用的数据访问效率。

针对大规模数据密集型任务的调度问题,提出了一个低开销的全局优化调度算法 BAR^[46]。BAR 能够根据集群的网络与工作负载动态调整

数据本地性,采用网络流思想,并结合负载均衡策略,以获得最小化的作业完成时间,为 AMS-02 等相关数据密集型任务的高效调度与执行提供了保证。

除了针对科学计算应用之外,东南大学在现有云计算平台的基础上,针对云计算环境中的若干共性问题也进行了相应的研究。

在 IaaS 层,部署了开源云计算系统 OpenQRM^[68]。基于该系统,研究虚拟机的放置、部署与迁移机制,完善其资源监控策略,使云计算平台可以快速感知资源工作负载的变化,从而提供弹性服务。此外,基于经济模型,探讨了云计算数据中心的资源管理、能耗及服务定价之间的关系。

在 PaaS 层,深入分析了 Hadoop 平台上的资源管理与调度机制,对多数据中心间的副本管理策略进行了研究。在此基础上,利用云计算在海量数据存储与处理方面的优势,将云计算应用于 OLAP 聚集计算和大规模组合优化问题的求解。

在 SaaS 层,针对移动社会网络中位置信任安全问题,开发设计了基于云计算的位置信任验证服务系统。该系统分为智能手机客户端和云计算平台端两部分。其中,在智能手机客户端,系统提取用户位置属性,并使用蓝牙无线传播技术进行位置信任凭证收集;在云计算平台上,基于凭证收集和验证算法,系统利用云计算弹性服务的特点来满足大规模用户的验证需求。

基于以上研究成果,东南大学今后还将在云计算数据中心的网络优化、云计算服务的安全隐私保护等方面开展深入研究,并进一步构建面向科学计算与应用的云计算平台。

6 结束语

云计算作为一种新兴的信息技术发展迅速。通过总结最近几年在该领域的应用与研究成果,将云计算体系架构划分为 3 个层次。综述了体系架构中主要关键技术的研究现状,包括数据中心设计与管理、虚拟化、海量数据存储与处理、资源管理与调度、服务质量保证和安全与隐私保护等。同时,介绍了东南大学的云计算平台及相关研究成果。总体来说,云计算的研究正处于发展阶段,从拓展云计算应用模式,解决内在的局限性等角度出发,围绕可用性、可靠性、规模弹性、成本能耗等因素,仍有大量关键问题需要深入研究。

致谢: 衷心感谢博士生周经亚、熊润群、田田、张竞慧、王巍、倪旭东、朱夏、王宇翔在本文写作中给予的帮助!

参考文献:

- [1] THUSOO A, SHAO Z, ANTHONY S, *et al.* Data warehousing and analytics infrastructure at facebook[A]. SIGMOD'10[C]. Indianapolis, Indiana, USA: ACM, 2010.1013-1020.
- [2] 淘宝数据开放策略[EB/OL]. <http://www.aliresearch.com/wp-content/uploads/2011/03/taobaoshuju.pdf>. 2011.
The open strategy of taobao's data[EB/OL] <http://www.ali-research.com/wp-content/uploads/2011/03/taobaoshuju.pdf>. 2011.
- [3] GILLEN A, BROUSSARD F W, PERRY R, *et al.* Optimizing infrastructure: the relationship between it labor costs and best practices for managing the windows desktop[EB/OL]. http://download.microsoft.com/download/a/4/4/a4474b0c-57d8-41a2-afe6-32037fa93ea6/ IDC_windesktop_IO_whitepaper.pdf 2007.
- [4] MELL P, GRANACE T. The NIST Definition of Cloud Computing[R]. National Institute of Standards and Technology, 2011.
- [5] Amazon EC2[EB/OL]. <http://aws.amazon.com/ec2> 2011.
- [6] DEAN J, GHEMAWAT S. MapReduce: a flexible data processing tool[J]. Commun ACM, 2010, 53(1): 72-77.
- [7] Google App Engine[EB/OL]. <http://code.google.com/appengine/> 2011.
- [8] Salesforce CRM[EB/OL]. <http://www.salesforce.com/>.
- [9] MILOJICIC D, WOLSKI R. Eucalyptus: delivering a private cloud[J]. Computer. 2011, 44(4): 102-104.
- [10] FOSTER I, YONG Z, RAICU I, *et al.* Cloud Computing and Grid Computing 360-Degree Compared[Z]. 2008.1-10.
- [11] Facebook server count: 60 000 or more[EB/OL]. <http://www.datacenterknowledge.com/archives/2010/06/28/facebook-server-count-60000-or-more/>. 2011.
- [12] Google investor relations[EB/OL].<http://investor.google.com/financial>. 2011.
- [13] GREENBERG A, HAMILTON J, MALTZ D A, *et al.* The cost of a cloud: research problems in data center networks[J]. SIGCOMM Comput Commun Rev, 2008, 39: 68-73.
- [14] GREENBERG A, HAMILTON J R, JAIN N, *et al.* VL2: a scalable and flexible data center network[A]. SIGCOMM'09[C]. Barcelona, Spain: ACM, 2009.51-62.
- [15] MYSORE R N, PAMBORIS A, FARRINGTON N, *et al.* PortLand: a scalable fault-tolerant layer 2 data center network fabric[A]. SIGCOMM'09[C]. Barcelona, Spain: ACM, 2009.39-50.
- [16] GUO C, WU H, TAN K, *et al.* Dcell: a scalable and fault-tolerant network structure for data centers[A]. SIGCOMM'08[C]. Seattle, WA, USA: ACM, 2008.75-86.
- [17] GUO C, LU G, LI D, *et al.* BCube: a high performance, server-centric network architecture for modular data centers[A]. SIGCOMM'09[C]. Barcelona, Spain: ACM, 2009.63-74.
- [18] HOELZLE U, BARROSO L A. The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines[M]. 1st ed. Morgan and Claypool Publishers, 2009.
- [19] NATHUJI R, SCHWAN K. VirtualPower: coordinated power management in virtualized enterprise systems[A]. SOSP '07[C]. New York, NY, USA: ACM, 2007.265-278.
- [20] PALLIPADI V, STARIKOVSKIY A. The ondemand governor: past, present and future[A]. Proceedings of Linux Symposium[C]. 2006. 223-238.
- [21] RAO L, LIU X, LE XIE, *et al.* Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment[A]. INFOCOM'10[C]. San Diego, California, USA: IEEE Press, 2010.1145-1153.
- [22] SAMADIANI E, JOSHI Y, MISTREE F. The Thermal Design of a Next Generation Data Center: a Conceptual Exposition[Z]. 2007. 93-102.
- [23] CHEN G, HE W, LIU J, *et al.* Energy-aware server provisioning and load dispatching for connection-intensive internet services[A]. NSDI'08[C]. Berkeley, CA, USA: USENIX Association, 2008. 337-350.
- [24] About virtual machine templates[EB/OL].<http://technet.microsoft.com/en-us/library/bb740838.aspx>, 2011.
- [25] VRABLE M, MA J, CHEN J, *et al.* Scalability, fidelity, and containment in the potemkin virtual honeyfarm[A]. SOSP'05[C]. Brighton, United Kingdom: ACM, 2005.148-162.
- [26] ANDR H, LAGAR-CAVILLA S, WHITNEY J A, *et al.* SnowFlock: virtual machine cloning as a first-class cloud primitive[J]. ACM Trans Comput Syst, 2011, 29(1): 1-45.
- [27] CLARK C, FRASER K, HAND S, *et al.* Live migration of virtual machines[A]. NSDI'05[C]. USENIX Association, 2005.273-286.
- [28] HIROFUCHI T, NAKADA H, OGAWA H, *et al.* A live storage migration mechanism over wan and its performance evaluation[A]. VTDC'09[C]. Barcelona, Spain: ACM, 2009.67-74.
- [29] CULLY B, LEFEBVRE G, MEYER D, *et al.* Remus: high availability via asynchronous virtual machine replication[A]. NSDI'08[C]. San Francisco, California: USENIX Association, 2008.161-174.
- [30] GHEMAWAT S, GOBIOFF H, LEUNG S. The Google file system[A]. SOSP'03[C]. Bolton Landing, NY, USA: ACM, 2003.29-43.
- [31] CHANG F, DEAN J, GHEMAWAT S, *et al.* Bigtable: a distributed storage system for structured data[J]. ACM Trans Comput Syst, 2008,

- 26(2): 1-26.
- [32] DECANDIA G, HASTORUN D, JAMPANI M, *et al.* Dynamo: amazon's highly available key-value store[A]. SOSP'07[C]. Stevenson, Washington, USA: ACM, 2007.205-220.
- [33] PIKE R, DORWARD S, GRIESEMER R, *et al.* Interpreting the Data: Parallel Analysis with Sawzall[J]. Scientific Programming Journal, 2005, 13(4): 227-298.
- [34] OLSTON C, REED B, SRIVASTAVA U, *et al.* Pig latin: a not-so-foreign language for data processing[A]. SIGMOD'08[C]. New York, NY, USA: ACM, 2008.1099-1110.
- [35] EKANAYAKE J, LI H, ZHANG B, *et al.* Twister: a runtime for iterative MapReduce[A]. HPDC'10[C]. Chicago, Illinois: ACM, 2010.810-818.
- [36] YANG H, DASDAN A, HSIAO R, *et al.* Map-reduce-merge: simplified relational data processing on large clusters[A]. SIGMOD'07[C]. New York, NY, USA: ACM, 2007.1029-1040.
- [37] WANG Y, SONG A, LUO J. A MapReduceMerge-based Data Cube Construction Method[Z]. 2010.1-6.
- [38] XIONG R, LUO J, SONG A, *et al.* QoS preference-aware replica selection strategy using mapreduce-based PGA in data grids[A]. ICPP'11[C]. Taipei, Taiwan, China.
- [39] VERMA A, LLOR X, GOLDBERG D E, *et al.* Scaling genetic algorithms using mapreduce[A]. ISDA'09[C]. IEEE Computer Society, 2009.13-18.
- [40] ISARD M, BUDI M, YU Y, *et al.* Dryad: distributed data-parallel programs from sequential building blocks[A]. EuroSys'07[C]. Lisbon, Portugal: ACM, 2007.59-72.
- [41] YU Y, ISARD M, FETTERLY D, *et al.* DryadLINQ: a system for general-purpose distributed data-parallel computing using a high-level language[A]. OSDI'08[C]. San Diego, California: USENIX Association, 2008.1-14.
- [42] Microsoft Azure[EB/OL]. <http://www.microsoft.com/windowsazure/>. 2011.
- [43] ELTABAK M Y, TIAN Y, OZCAN F, *et al.* CoHadoop: flexible data placement and its exploitation in hadoop[A]. Proc VLDB Endowment[C]. 2011.
- [44] 郑湃, 崔立真, 王海洋等. 云计算环境下面向数据密集型应用的数据布局策略与方法[J]. 计算机学报. 2010(8): 1472-1480.
ZHENG P, CUI L Z, WANG H Y, *et al.* A data placement strategy for data-intensive applications in cloud[J]. Chinese Journal of Computers, 2010(8): 1472-1478.
- [45] FISCHER M J, SU X, YIN Y. Assigning tasks for efficiency in Hadoop: extended abstract[A]. SPAA'10[C]. New York, NY, USA: ACM, 2010.30-39.
- [46] JIN J, LUO J, SONG A, *et al.* BAR: an efficient data locality driven task scheduling algorithm for cloud computing[A]. CCGRID'11[C]. Newport Beach, CA, USA: IEEE Computer Society, 2011.295-304.
- [47] ZAHARIA M, BORTHAKUR D, SEN SARMA J, *et al.* Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling[A]. EuroSys'10[C]. New York, NY, USA: ACM, 2010. 265-278
- [48] ISARD M, PRABHAKARAN V, CURREY J, *et al.* Quincy: fair scheduling for distributed computing clusters[A]. SOSP '09[C]. New York, NY, USA: ACM, 2009.261-276.
- [49] ZAHARIA M, KONWINSKI A, JOSEPH A D, *et al.* Improving MapReduce performance in heterogeneous environments[A]. OSDI'08[C]. Berkeley, CA, USA: USENIX Association, 2008.29-42.
- [50] STANTCHEV V, SCHROEDER C. Negotiating and Enforcing QoS and SLAs in grid and cloud computing[A]. GPC '09[C]. Berlin, Heidelberg: Springer-Verlag, 2009.25-35.
- [51] BUYYA R, BROBERG J, GOSCINSKI A M. Cloud Computing Principles and Paradigms[M]. Wiley Publishing, 2011.
- [52] CALHEIROS R N, RANJANY R, BUYYA R. Virtual machine provisioning based on analytical performance and QoS in cloud computing environments[A]. ICPP'11[C]. Taipei, Taiwan, China.
- [53] XIAO Y, LIN C, JIANG Y, *et al.* Reputation-BASED QoS provisioning in cloud computing via dirichlet multinomial model[A]. ICC'10[C]. 2010.1-5.
- [54] ANDRZEJAK A, KONDO D, YI S. Decision model for cloud computing under SLA constraints[A]. MASCOTS'10[C]. 2010. 257-266.
- [55] SANTHANAM S, ELANGO P, ARPACI-DUSSEAU A, *et al.* Deploying virtual machines as sandboxes for the grid[A]. WORLDS'05[C]. Berkeley, CA, USA: USENIX Association, 2005. 7-12.
- [56] RISTENPART T, TROMER E, SHACHAM H, *et al.* Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds[A]. CCS'09[C]. Chicago, Illinois, USA: ACM, 2009. 199-212.
- [57] RAJ H, NATHUJI R, SINGH A, *et al.* Resource management for isolation enhanced cloud services[A]. CCSW'09[C]. New York, NY, USA: ACM, 2009.77-84.
- [58] ROY I, SETTY S T V, KILZER A, *et al.* Airavat: security and privacy for MapReduce[A]. NSDI'10[C]. Berkeley, CA, USA: USENIX Association, 2010.20.
- [59] LI J, WANG Q, WANG C, *et al.* Fuzzy keyword search over encrypted data in cloud computing[A]. INFOCOM'10[C]. Piscataway,

NJ, USA: IEEE Press, 2010.441-445.

- [60] YU S, WANG C, REN K, *et al.* Achieving secure, scalable, and fine-grained data access control in cloud computing[A]. INFO-COM'10[C]. Piscataway, NJ, USA: IEEE Press, 2010.534-542.
- [61] 冯登国, 张敏, 张妍等. 云计算安全研究[J]. 软件学报. 2011, 22(1): 71-83.
- FENG D G, ZHANG M, ZHANG Y, *et al.* Study on Cloud Computing Security, 2011, 22(1):71-83.
- [62] ARMBRUST M, FOX A, GRIFFITH R, *et al.* Above the Clouds: A Berkeley View of Cloud Computing[R]. EECS Department, University of California, Berkeley, 2009.
- [63] MORETTI C, BUI H, HOLLINGSWORTH K, *et al.* All-pairs: an abstraction for data-intensive computing on campus grids[J]. IEEE Transactions on Parallel and Distributed Systems, 2010, 21: 33-46.
- [64] DONG F, LUO J, SONG A, *et al.* Resource load based stochastic DAGs scheduling mechanism for grid environment[A]. HPC'10[C]. Washington, DC, USA: IEEE Computer Society, 2010.197-204.
- [65] ZHOU J, LUO J, SONG A. NETOP: a non-cooperative game based topology optimization model towards improving search performance[J]. Journal of Internet Technology, 12(3): 477-490
- [66] ZHOU J, LUO J, SONG A. Grid service discovery based on cross-service domain model[A]. NPC'08 [C]. Berlin, Heidelberg: Springer-Verlag, 2008.327-338.
- [67] TIAN T, LUO J, WU Z. A replica replacement algorithm based on value-cost prediction[J]. Lecture Notes in Computer Science, 2008: 365-373.
- [68] OpenQRM [EB/OL]. <http://www.openqrm.com>, 2011.

作者简介:



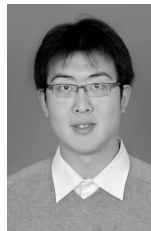
罗军舟 (1960-), 男, 浙江宁波人, 博士, 东南大学教授、博士生导师, 主要研究方向为网络与云计算、下一代网络体系结构、协议工程、网络安全和管理、服务计算。



金嘉晖 (1986-), 男, 浙江温州人, 东南大学博士生, 主要研究方向为云计算、海量数据处理。



宋爱波 (1970-), 男, 山东烟台人, 博士, 东南大学副教授, 主要研究方向为网络与云计算、海量数据处理、Petri 网理论与应用。



东方 (1982-), 男, 江苏南京人, 博士, 东南大学讲师, 主要研究方向为云计算、网络计算、海量数据处理。