

分类号\_\_\_\_\_

密级\_\_\_\_\_

UDC\_\_\_\_\_

编号\_\_\_\_\_

華東交通大學

硕士学位论文

基于连续双向拍卖的云计算资源分配研究

学位申请人：	刘鹏
学科专业：	计算机应用技术
指导教师：	刘觉夫 教授
副指导教师：	

答辩日期：

华东交通大学 2012 届硕士学位论文

基于连续双向拍卖的云计算资源分配研究

信息工程学院

刘鹏

## 独创性声明

本人郑重声明：所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表和撰写的研究成果，也不包含为获得华东交通大学或其他教育机构的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

本人签名\_\_\_\_\_日期\_\_\_\_\_

## 关于论文使用授权的说明

本人完全了解华东交通大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅。学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

保密的论文在解密后遵守此规定，本论文无保密内容。

本人签名\_\_\_\_\_导师签名\_\_\_\_\_日期\_\_\_\_\_

## 基于连续双向拍卖的云计算资源分配研究

### 摘要

云计算是一种新兴商业计算模型，商业化的特性使其持续关注用户日益多样化的需求，目标旨在按需灵活合理分配资源，提升数据中心资源利用率。成熟虚拟化技术的运用使得资源分配与调度呈现与以往并行分布式计算不同的特性，因而对云计算中资源分配与调度的研究可以推动云计算进一步发展。

本文利用经济学中市场机制具有调动个体积极性，激励资源共享，实现合理资源配置的特点，将连续双向拍卖应用于商业化特性的云计算环境中，设计了基于连续双向拍卖的云计算资源分配策略算法。该算法考虑不同云用户对资源具有多样化的需求，分别为云用户、资源提供者设置公平透明的定价策略：云用户代理在每个时间单位内基于剩余可用资源数与平均剩余时间独立决定竞价值，资源提供者根据自身工作负载提出要价请求。云用户与资源提供者分别提交竞价和要价给拍卖商，拍卖商管理当前的竞价列表与要价列表，匹配最终交易。文中同时考虑少数对云任务截止时间紧迫程度特别高的特殊需求用户，允许这类用户依据实际情况选择预定适合的资源提供者并协商付费价格。最后采用服务等级协议（SLA, Service Level Agreement）正式确立云用户与云服务提供者之间的服务合约，指定云用户实现不同 QoS 方面的服务水平，探讨 QoS 请求的实现情况。

在介绍分析云计算仿真工具 CloudSim 的体系结构、技术实现与具体使用方法后，扩展重编译 CloudSim 平台，对本文策略算法进行仿真实验与性能分析，通过和其他相关调度算法进行比较，表明本文提出的 S-CDA 算法在一定程度上较好地适应用户对服务质量的不同需求，适合开放复杂的云计算环境。

**关键词：**云计算，资源分配，连续双向拍卖，服务等级协议，CloudSim

# RESEARCH OF CLOUD RESOURCE ALLOCATION BASED ON CONTINUOUS DOUBLE AUCTION

## ABSTRACT

Cloud computing is a new business computing model, the commercialization property of cloud computing makes it focus on the users' increasingly diversified demands. It aims at flexibly and reasonably allocating resources according to the needs, improving the data center resources utilization. The use of mature virtualization makes resource allocation and scheduling different from the previous parallel distributed computing. Therefore, the study on cloud resources allocation and scheduling can promote the development of cloud computing.

This paper adopts the market economics mechanism which mobilizes individual enthusiasm, inspires resources sharing, and realizes the reasonable allocation of resources. Applying continuous double auction in cloud computing with the commercial properties, it designs a resource allocation algorithm based on continuous double auction in cloud computing. The algorithm considers different cloud users with diversified demands, sets fair and transparent pricing strategies for the resources providers and the user respectively: cloud user agent decides the bid independently based on the available remaining resources and the average remaining time in each unit of time, while resource providers proposes the ask price according to their own workload. Cloud users and resources providers submit the bid and ask price respectively to auctioneer, while the auctioneer manages current list of bid and ask price and matches final deal. In this paper, a few cloud users with high urgent deadline tasks are also discussed, we assume that these cloud users are allowed to choose and reserve suitable resource providers with the consultation price based on their actual situation. It finally adopts the SLA to define the service contract between users and the cloud service providers, specifies services level for users to achieve different QoS, and explores the realization of the QoS requests.

After introduction and analysis CloudSim's system architecture, the technical implementation and the concrete use method, the paper expands and recompiles CloudSim platform to implement the proposed algorithm, makes a simulation experiment and performance analysis. Comparing with other related scheduling algorithms, it is proved that the S-CDA presented by this paper adapts to different users' QoS at a certain extent and is suitable for open complex cloud computing environment.

**Key Words:** cloud computing, resource allocation, CDA,SLA,CloudSim

# 目录

第一章 绪论.....	1
1.1 研究背景及意义 .....	1
1.2 国内外研究现状 .....	2
1.3 主要研究内容与创新 .....	4
1.4 论文组织结构 .....	4
第二章 云计算技术.....	6
2.1 定义与特征 .....	6
2.2 相关概念比较 .....	7
2.3 体系结构 .....	11
2.3.1 一般体系结构.....	11
2.3.2 面向市场的体系结构.....	12
2.4 云计算应用 .....	13
2.4.1 Microsoft 云计算 .....	13
2.4.2 IBM 云计算 .....	15
2.4.3 Google 云计算 .....	16
2.4.4 国内典型云计算应用.....	19
2.5 云计算标准化 .....	21
第三章 资源分配相关研究.....	23
3.1 网格资源分配与调度研究 .....	23
3.1.1 网格资源特征与调度目标.....	23
3.1.2 网格资源管理模型.....	24
3.2 云计算资源分配研究 .....	24
3.2.1 云资源特征与分配目标.....	25
3.2.2 面向市场的拍卖机制.....	26
第四章 基于 CDA 的云计算资源分配.....	27
4.1 云计算资源分配模型 .....	27
4.2 CDA 拍卖模型.....	28
4.2.1 云用户代理策略.....	28
4.2.2 资源代理策略.....	30
4.2.3 拍卖商策略.....	30
4.3 SLA .....	30
4.4 算法伪代码 .....	31

第五章 仿真实验与结果分析.....	33
5.1 CloudSim 介绍.....	33
5.1.1 CloudSim 体系结构 .....	33
5.1.2 CloudSim 技术实现 .....	35
5.2 CloudSim 使用方法.....	38
5.2.1 环境配置.....	38
5.2.2 程序运行.....	38
5.2.3 仿真基本步骤.....	39
5.2.4 CloudSim 扩展 .....	40
5.2.5 平台重编译.....	41
5.3 实验仿真与结果分析 .....	41
第六章 总结与展望.....	46
6.1 全文总结 .....	46
6.2 展望 .....	46
参考文献.....	47
个人简历 在读期间发表的学术论文.....	50
致谢.....	51

## 第一章 绪论

### 1.1 研究背景及意义

云计算思想起源于 20 世纪 60 年代,当时提出把计算能力当作一种像水和电一样的公用事业的理念,最近这个梦想逐渐成为商业实现。而 80 年代的网格计算,90 年代的公用计算以及虚拟化技术,SOA 和 SaaS 的应用使得云计算成为一种新兴的资源使用与交付模式。云计算被看作是继个人计算机,互联网之后的第三次 IT 变革,它必将带来生活、生产方式和商业模式根本性的改变,是当今社会的关注热点。

云计算(Cloud Computing)是分布式处理(Distributed Computing)、并行处理(Parallel Computing)和网格计算(Grid Computing)的商业实现。同时它也是效用计算(Utility Computing)、虚拟化(Virtualization)、基础设施即服务(Infrastructure as a Service)、平台即服务(Platform as a Service)、软件即服务(Software as a Service)等概念的综合演进与跃升。云计算由专门软件实现对动态可扩展的集中资源进行管理,用户按照实际需求动态申请获取存储空间、计算力和信息服务并按量付费。这一程省略了繁琐细节,使用户专心自身业务,有效地降低成本,提高效率和技术创新。网格计算虚拟化不同分布的地理位置资源为一有机整体,从而最终实现数据、存储、计算和软件等资源的共享与协同工作。云计算不仅包括网格计算的内涵,同时也拥有一些新特点:1、一般来说,网格计算对分布式异构的资源进行聚合,支持构成的虚拟组织;而云计算资源相对集中,提供的底层资源主要是大型数据中心。2、网格计算通过对资源的整合支持挑战性应用,如科学应用研究等;云计算对广泛企业计算和 Web 应用等具有很好普适性。3、网格计算借助中间件完成异构集成,提供给用户统一环境;云计算的异构使用提供不同服务机制或镜像加以解决。4、网格计算采用执行任务的形式,阶段内完成产生的数据;云计算动态回收重用处理能力。5、相比被限制的商业应用,网格计算更多的针对科学研究;云计算立足商业化,实现不同用户多样化需求。6、云计算采取虚拟化技术,提供虚拟化资源供用户使用,与网格计算的资源分配不同。

成熟的虚拟化技术赋予云计算新特点,有效精简化资源分配与调度的过程。云计算商业特性要求不断满足用户日益增长的多样化需求,其中云计算资源分配与调度显得尤为关键。云计算目标旨在实现资源优化使用,更好完成用户提交的任务,提供更为优质服务;对资源按需灵活分配,进一步提升数据中心资源利用率。本文针对此,探讨云计算环境下资源的分配与调度,设计实现了基于连续双向拍卖机制的云计算资源分配策略算法。文中引入连续双向拍卖机制,积极探索云计算资源分配与调度问题。



## 1.2 国内外研究现状

美国是云计算概念的发祥地，在众多大型 IT 公司的极力推动下，有着成熟的产品与技术，而且市场应用程度高。美国总统奥巴马任命政府首席信息官，将云计算提上美国政府的 IT 议事日程。联邦政府同时启动了新网站 [info.apps.gov](http://info.apps.gov)，向大众普及云计算的理念，期望借助虚拟化压缩政府开支。美国国家标准技术研究院（NIST）提出了较全面的云计算概念及特征，评估云计算和制定云计算标准。美国航空航天局制定了“星云”计划，通过整合系列开源组件，形成无缝服务平台，虚拟化，可扩展技术提供存储、网络和高性能计算。美国国防信息系统局开发一系列的云计算方案，包括：全球信息栅格内容服务（GCDS）：服务分散用户群，提供稳定安全的数据信息共享的全球性计算平台；软件协同开发与使用系统（[forge.mil](http://forge.mil)）：提供开发者分享软件、系统组件及网络服务；快速访问计算环境（Rapid Access Computing Environment, RACE），以降低成本、整合应用程序、缩短交付时间和简化用户体验的云计算基础架构。美国能源部投入巨资 3200 万美元实施“麦哲伦”（Magellan）项目，以期全世界科学家共同使用一套云计算基础设施。这种系统不仅方便科学研究，还将很大程度上降低不少实验室与大学对计算基础设施的成本投入。

欧洲国家及欧盟也迅速跟进，欧盟的第七框架计划（FP7）资金支持若干个云计算项目，云计算是 2011 年的工作计划中重要研究主题<sup>[1]</sup>：（1）基础设施虚拟化及跨平台执行；（2）云计算环境的互操作；（3）无缝支持云计算软件和服务；（4）网络环境集成架构及技术；（5）软件栈开源执行等。欧盟投资 1700 万欧元，欧洲科研机构与 IBM 联合，旨在建立“基于服务的网络经济模式”，开展了 RESERVOIR 云计算计划。计划通过企业服务、分布式虚拟管理和虚拟化技术，实现跨平台的 IT 应用服务，对基于服务的在线经济，资源与服务透明式配置管理予以支持。英国云计算基础设施 G-Cloud 作为数字英国战略的一部分改进政府效率。

日本提出了数字日本创新计划以期建立全国范围的云计算基础设施，开拓信息通信技术市场，提升日本经济。发布的《云计算与日本竞争力研究》报告，将计划从基础设施、制度改善和创新三方面推动并开启云计算服务的全球市场。云计算也是韩国未来和绿色发展战略的一部分。为进一步推进发展云计算，韩国政府成立了联合计划组，实施了数字化广播云计算环境的云计算研发项目。此外，韩国还成立了云服务协会对政策、云技术、云服务、云安全和云试验平台进行研究。印度政府宣布在多年信息技术领域成功发展经验基础上，印度将打造全球首个向市民提供使用云计算技术的电子政府服务系统。该系统能使得政府在节支增效的同时，实现电子政务覆盖 90% 的市民服务，包括轻松快速地处理网上预订火车票等。

中国也积极开展云计算。“太湖云谷”是国内第一批云计算建设城市无锡联手 IBM 打造的集开发、商务和政务一体的云平台。北京实施了“祥云计划”，依托人才，科技优势，产学研结合，计划在 2012 年前完成产业链布局。上海发布“云海计划”，以亚太

云计算中心为定位。深圳也在积极打造“华南云计算中心”，近期深圳云计算产业协会联合英特尔、IBM、金蝶等国内外相关企业创建的专业性技术与应用研发实验室云计算国际联合实验室正式揭牌。南昌鄱湖云计算规划投资2亿元，将与戴尔、甲骨文、深圳华为合作共建全球第一个运用开源技术建设的公有云计算。作为十二五规划的重点项目，工信部与发改委于去年10月18日联合印发《关于做好云计算服务创新发展试点示范工作的通知》，确定在北京、上海等五个城市开展云计算服务创新发展试点示范工作。为推动我国在全球卫星导航产业实现赶超，科技产业界去年8月面向全球发布了将云计算应用于全球卫星导航的“位置云”概念。“位置云”是基于IT技术、3S技术以及网络与通信技术的综合体系，其中包含基础设施、开发平台与服务、产品解决方案等，吸纳与位置相关所有资讯，为各领域提供基于位置需求的解决方案。这一体系一定会对促进全球卫星导航产业应用有着重大深远意义<sup>[2]</sup>。

云计算的研究在学术界主要集中以下方面<sup>[3]</sup>：1、体系结构划分。分类云系统，介绍云的层次栈，加强对云计算系统的分类、联系与内部关系理解。2、云计算关键技术：（1）虚拟机 云计算中最关键的一种技术就是虚拟化技术，目前普遍使用的三种虚拟机技术是Xen、KVM和Vmware Infrastructure。（2）云安全 云计算的广泛使用使得云安全管理成为亟待解决的重要问题。很多研究人员做了大量理论研究以及工程实践，Siani Pearson等提出了云计算的服务设计中保证隐私不泄露的六大设计原则。（3）能耗管理 如何有效整合云计算基础设施资源、降低成本、节省能源也是关注的热点，由此产生了绿色云计算的概念。（4）云监测 海量数据处理与虚拟机监控能够体现云计算强大数据处理能力，Jerome Boulon等设计了建立在Hadoop上的数据收集系统，用于监测分析的大规模分布式系统Chukwa。（5）数据管理 云计算资源具有弹性特性，拥有无限的存储、计算能力。拥有大量硬件资源的云服务商为用户提供大量虚拟设备。（6）资源调度 云系统通过提高并行度来提升性能，依据用户的请求动态分配计算资源。3、编程模型：（1）All-Pairs模型 适用于数据密集型科学应用。（2）GridBatch模型 解决大规模数据批处理问题，让用户控制数据的分发和计算力分配。4、云平台研究：（1）Cumulus（数据中心科学云） 主要服务于科学计算应用程序提供虚拟机、虚拟应用程序和虚拟计算的平台。（2）CARMEN（e-Science云计算） 项目供神经科学家共享、整合以及分析数据。（3）Tplatform 由PC集群层、云计算平台基础设施层与数据处理应用层组成。（5）Yahoo云平台 构建一系列可扩展、高度可用数据存储和处理服务，并部署在云模型中。（6）开源云计算系统平台 IT厂商不断推出各自平台，而研究者利用较低的成本的开源云计算系统模拟出近似商业云的环境。5、云计算应用研究：（1）生物学应用 当今生物信息科学要求海量数据管理和大规模计算能力，MapReduce等并行计算的方法可以满足要求。（2）数据库应用 用户把越来越多的软件，如数据库应用，部署在云平台。（3）语义分析研究 针对动态用户进行大规模数据集有效查询与推理。（4）地理信息应用。（5）商业应用 企业中半结构化与非结构化数据大量增长，开发研究有效商业应用很有必要。6、云安全 借鉴云计算的思想并将其应用于信息网络安全。云计

算研究发展极其迅速，并将逐步深入，比如，云和云之间的互操作、跨云虚拟化，研究网格计算与云计算的结合以及云计算的可靠性研究。

### 1.3 主要研究内容与创新

论文主要探讨研究基于连续双向拍卖机制的云计算资源分配策略问题，主要工作如下：

(1) 阐述云计算定义、特征、体系结构，以及云计算相关概念比较。归纳介绍主流云计算应用，包括其中的关键技术与解决方案，总结云计算标准化现状发展状况。

(2) 分析网格计算、云计算资源分配相关研究。利用经济学中市场机制可以激励资源共享，实现合理资源配置的特点，考虑用户的 SLA (Service Level Agreement) 和少数特殊需求用户，探讨将连续双向拍卖机制应用于云计算资源分配，设计了基于连续双向拍卖的云计算资源分配算法 S-CDA。

(3) 介绍云计算仿真工具 CloudSim 的体系结构、技术实现与具体使用方法。分析并扩展重编译仿真平台，仿真评价本文提出的策略算法性能，通过和其他相关调度算法进行比较，表明本文提出的 S-CDA 算法一定程度上较好适应用户对服务质量的不同需求，适合开放复杂的云计算环境。

本文创新点有以下：

(1) 结合经济学中市场机制，将连续双向拍卖应用于具有商业特性的云计算环境中，构建基于连续双向的云计算资源分配模型，实现高效、公平与灵活的资源分配与调度。

(2) 鉴于不同用户对不同资源有着多样化的需求，分别为云用户，资源提供者设置公平透明的定价策略，揭示实际需求变化。同时考虑少数对云任务截止时间紧迫程度特别高的特殊需求用户。采用服务等级协议 (SLA, Service Level Agreement) 确定云服务提供者与云用户间的服务合约，指定云用户实现不同 QoS 方面的服务水平。

(3) 扩展重编译 CloudSim 仿真平台，评价本文策略算法性能。

### 1.4 论文组织结构

第一章主要介绍云计算的时代背景与研究意义，讨论当前国内外云计算发展研究现状，包括国内外的发展与学术界对云计算的研究热点，阐述本文主要研究内容，简述本文组织结构，研究内容与创新。

第二章对云计算技术进行阐述。对云计算的基本概念、特征、代表性体系结构予以归纳，就涉及云计算的相关概念进行详细比较，包括分布式、并行计算，效用计算，网格计算、框计算与物联网，介绍主流的云计算应用，云计算标准化现状。

第三章介绍资源分配相关研究。介绍网格计算的资源分配与调度研究，概述云计算

资源分配相关已有研究。概括了面向市场的资源分配模型，引出拍卖机制。

第四章主要阐述基于连续双向拍卖的云计算资源分配算法模型，给出算法相应的伪代码描述。

第五章主要介绍了云计算仿真工具 CloudSim 与具体使用方法，扩展重编译 CloudSim 平台对本文设计的 S-CDA 算法进行性能仿真评价。

第六章总结全文并提出将来可能进一步的研究工作。

## 第二章 云计算技术

### 2.1 定义与特征

现在云计算仍然没有普遍一致的定义，许多研究人员和开发者尝试着从不同方面给予定义。美国著名科学家，“网络之父” Ian Foster 认为：“云计算是一种通过规模经济带，提供给互联网外部用户的一组抽象、虚拟化、动态可扩展，可管理的存储、计算资源能力、平台与服务的分布式计算集合体<sup>[4]</sup>”。UC Berkeley 对云计算的观点：“云计算不仅指由互联网提供的应用程序服务，也包括数据中心用来提供这些服务的系统软件和硬件。云分为公共云和私有云<sup>[5]</sup>”。IEEE 计算机协会集群计算工作组主席，墨尔本大学 Rajkumar Buyya<sup>[6]</sup>指出云计算是分布式和并行计算系统，依据预先协商定义的 SLA（服务等级协议）提供云用户动态计算资源。美国标准技术研究院 NIST 给出的定义使用较为普遍：云计算是一种网络访问的模型，可以使用户便捷且按需应变地访问共享、可配置的计算资源共享池（如服务器、存储器、网络、应用程序和服务）。该模型以最少的管理投入或服务供应商介入快速实现资源提供与发布<sup>[7]</sup>。

NIST 定义的云计算框架如图 2-1，包括四种部署模式、三种服务模型和五个基本特征。



图 2-1 云计算框架

Fig.2-1 The Framework of Cloud Computing

四种部署模式：（1）混合云。由公有云与私有云融合组成，用户在这种模型下通常将非关键性业务信息以及流程转移至公有云处理，而将业务关键性服务和数据自己掌控。（2）私有云。私有云以服务为基础，灵活性强，为用户和供应商提供更好的云基础架构控制，有较高的安全性和恢复能力。（3）社区云。一群共享利益（如拥有共同目标或特殊安全需求）的企业管理和共享使用云中数据和应用程序。（4）公有云。免费或以较低成本在特定访问控制机制下通过网络访问第三方供应商的云计算服务。

三种服务模型：

(1) 软件即服务(SaaS)

将某些特定应用软件功能统一封装成服务，互联网最终用户可以使用这些应用程序，但不能对此程序所需的基础设施或操作系统进行控制。如 Salesforce 提供的在线客户关系服务 (Client Relationship Magement)。

(2) 平台即服务(PaaS)

平台通常是一应用程序的框架，提供终端用户的运行环境。微软云计算操作系统 Microsoft Windows Azure 和 Google App Engine 可以归为这一类。

(3) 基础架构即服务(IaaS)

用户可使用诸如存储器、网络部件（如防火墙或负载均衡器）或处理能力，控制操作系统和已经部署的应用程序。封装硬件设备等基础资源为服务，允许用户动态申请，按量使用付费，具有更高资源使用效率。典型实例是 Amazon 云计算 AWS(Amazon Web Services)的 EC2(Elastic Compute Cloud)和 S3(Simple Storage Service)。

云计算五个基本特征：

(1) 按需自助服务：云计算按需与自助服务允许用户需要时可以直接使用云计算服务，单方面地部署资源。

(2) 无所不在网络访问：通过互联网获取供应商的资源，并且可以以标准方式访问，例如移动电话，笔记本电脑，PDA。

(3) 快速伸缩：云计算一个主要特征，资源可以根据需求迅速且弹性地部署，以便快速扩展和释放。

(4) 资源池：云计算供应商的资源被池化，便于多用户共享模式服务于不同需求的用户。由于资源池的地点独立性，客户一般无法控制或者知晓所提供资源准确位置，但是可以在高端提取层面（如地区、国家）指定位置。这些资源包括存储、处理器、虚拟机器和网络带宽。

(5) 服务可度量：可度量服务下，云供应商控制并监测云计算服务各方面使用情况。资源收费是基于计量一次一付，或者基于广告的收费模式。服务的度量对于计费、资源优化、访问控制、容量规划以及其他任务具有重要意义。

## 2.2 相关概念比较

云计算是多种技术的综合演进，以下对相关概念进行比较：

(1) 分布式计算

分布式计算 (Distributed Computing) 是研究将一个需要非常巨大的计算能力才能解决的问题分成许多小部分，之后把这些部分分配给许多计算机进行处理，最后把这些计算结果综合起来得出最终的结果的一门计算机科学。分布式计算这类问题一般是跨学科、极富挑战性、人类急待解决的科研课题，如比较著名的有寻找最大的梅森素数，研

究寻找最为安全的密码系统 RC-72（密码破解）等。分布式计算主要有资源共享、高性价比、应用分布性、高可靠性、分布式透明处理平台、可扩展性、灵活性的特点。

云计算把计算分布于大量的分布式计算机上，而非本地计算机或远程服务器中，以便企业能够将资源切换到需要的应用，按照需求访问计算机和存储系统。云计算作为分布式计算的一种，它更加侧重的是商业化实现，从而获取相应商业价值。

### （2）并行计算

并行计算<sup>[8]</sup>（Parallel Computing）指的是同时使用多种计算资源解决计算问题的过程。为了执行并行计算，计算资源应该包括一台配有多处理机（并行处理）的计算机、一与网络相连计算机专有编号，或者两者结合一起使用。并行计算的主要目的是快速解决大型且复杂的计算问题。此外还包括：利用非本地资源节约成本，使用多个“廉价”计算资源取代大型计算机，同时也克服由于单个计算机上存在的存储器限制。通常采用并行计算的问题具有以下特征：有助于同时解决工作分成的离散部分；随时且及时地执行多个程序指令；多计算资源下解决问题耗时要少于单个计算资源下耗时。

云计算是在并行计算之后产生的概念，由并行计算发展而来，两者在很多方面有着共性，但是并行计算并不等于云计算，云计算也不等同于并行计算。它们的区别<sup>[9]</sup>：①云计算萌芽应该从计算机并行化开始，并行机的出现是人们不满足于 CPU 摩尔定率的增长速度，因而希望把多个计算机并联起来，从而获得更快的计算速度。这是一种简单且朴素的实现高速计算的方法，此种方法被证明是相当成功的。②并行计算的提出主要是为了满足科学和技术领域专业需要，应用领域也基本局限于科学领域。传统的并行计算机使用是一个相当专业的工作，要求使用者有着较高的专业素质，多数是命令行操作。③并行计算时代，人们极力追求的是采用昂贵服务器实现高速计算，各国也因此不惜代价在计算速度上超越他国，所以并行计算时代的高性能机群是一个“快速消费品”，而云计算中心的计算力与存储力可随着需要逐步地增加，云计算基础架构支持这一动态增加方式，高性能计算将在云计算时代成为“耐用消费品”。

### （3）效用计算

效用计算把计算、存储和服务等计算资源打包，作为一种计量服务，就像传统的水、电、煤气等公共设施一样。这种模型的优点是以较低或不需要成本获取计算资源，本质上计算资源是可租用的，从而实现由先前购买产品（硬件、软件和网络带宽）到服务的转变。它是一种按照计算资源使用量付费的商业模式，用户可以从计算资源供应商获取并使用计算资源，根据实际使用的资源进行付费。

效用计算与云计算的以服务形式提供计算、存储、应用资源思想类似。两者的区别不是这些思想背后目标，在于组合到一块，使这些思想成为现实技术。云计算以虚拟化技术为基础，可以提供最大限度灵活性与可伸缩性。云计算的服务提供商可以轻松扩展虚拟环境，以通过提供者的虚拟基础设施提供更大的带宽或计算资源。效用计算一般需要类似云计算基础设施的支持，但并不是一定需要，在云计算之上同样可以提供效用计

算,也可以不采用效用计算<sup>[9]</sup>。

#### (4) 框计算

框计算<sup>[10]</sup> (Box Computing) 是由百度董事长兼首席执行官李彦宏在 2009 年 8 月 18 日“百度技术创新大会”上提出的全新技术概念。用户只需简单地在“框”中输入服务需求,系统就能非常明确识别并且将需求分配给用户最优内容资源或应用提供商处理,最终能够精准高效地返回给用户相匹配的结果。这种高度智能互联网需求交互模式,以及“最简单可依赖”的信息交互实现机制与过程,就称之为“框计算”。用户需求实现的过程:首先,用户在“框”里提交任意一个需求;之后,“框计算”经过一系列诸如语义、智能人机交互、行为分析等复杂需求分析与海量计算,“框计算”后台单个或多个对应的数据/应用接受用户需求并响应;第三,“框计算”大量即插即用的接口方便响应收集到的需求;最后,用户“即搜即得、即搜即用”地获得精准、可靠、稳定的信息或应用需求结果。

云计算中的“云”通常是包括宽带资源、存储和计算服务器的大型服务器集群,对虚拟计算资源进行管理和自我维护。实质来说,“框计算”强调前端用户,而云计算则强调后台资源整合,提供低成本的 IT 基础设施配置供用户使用。总体来看“云计算”仍处于探索之中,而百度的“框计算”从构想到最终成为全球有影响力的现实产品,同样需要经过一个深入的探索期。

#### (5) 物联网

物联网(Internet of Things),一种建立在互联网上的泛在网络,通过各种有线和无线网络与互联网融合,综合应用海量的传感器、智能处理终端、全球定位系统等,实现物与物、物与人,所有的物品与网络的连接,方便识别、管理和控制<sup>[11]</sup>,最终完成对“万物”的“节能、高效、安全、环保”的“管、控、营”一体化服务。

物联网与云计算是当前的热门的技术新词汇,它们之间有着密切的内在关系。物联网的发展需要云计算加以支撑。物联网利用巨大数量的传感器采集海量数据,借助无线传感网或宽带传输至指定的存储和处理地,云计算非常显著的性能价格比优势可以很好的完成这些任务。云计算可以非常准确、迅速且智能地对海量数据进行分析处理,从而进一步提高资源利用率。物联网的快速发展需要具有强大处理、存储与极高性价比的云计算支持,云计算要实现商业成功同时也依赖物联网,两者相互促进。

#### (6) 网格计算

网格计算概念是在动态的环境下实现对资源的共享和问题的求解,最近也进一步强调网格具备联合不同机构的资源来实现同一目标。表 2-1 将对云计算与网格计算的相关特征进行比较<sup>[12]</sup>。



表 2-1 云计算与网格计算特征比较

特征	云计算 (Cloud Computing)	网格计算 (Grid Computing)
目标	通用计算平台与存储空间、软件服务	资源共享, 协同工作
架构	用户选择	面向服务
标准化	尚无统一标准	OGSA/WSRF
可用性	用户友好	管理困难
虚拟化	硬件与软件平台	数据与计算资源
资源类型	同构资源	异构资源
自我管理	可重构, 自我修复	可重构
资源共享	已分配资源不共享	虚拟组织公平共享
付费模型	灵活	严格
集中化程度	集中控制	分散控制

云计算与网格计算都是基于用户的特定需求来提供便利计算资源这个构想<sup>[13]</sup>：网络是在一般的协议下多机构间共享硬件或软件资源，采取自底向上的设计方式，部署彼此协作的服务以便于获取物理资源（CPU, 内存，存储等）和软件服务。用户使用的是真实的机器，网络资源由他们各自的所有者管理，授权的用户可以不用花费和保障服务水平的前提下通过远程机器调用网络资源。网络中间件提供一整套的 API 来设计一异构地理分布式系统。云计算是自顶向下设计，旨在向其用户提供特定的高级服务，如存储，计算平台，专门服务等。用户从云中得到虚拟资源，隐藏基础设施细节。云用户请求服务时只需了解其 QoS, 付费并成为虚拟机的所有者。虚拟化技术的应用便于建立一个隔离环境来配置满足用户的需求。云使用方便，可扩展，准确提供用户所需，相比而言，网格使用不方便，不能保证性能和互操作性，一般局限于科学界解决科学计算。

中国云计算专家委员刘鹏教授认为云计算与网格计算不是取代关系，它们可以互补，预见将来云计算与网格将会融为一体<sup>[3]</sup>：

- 不同的云计算厂商解决方案互不兼容，云计算缺乏统一标准；网格技术可以跨越系统、平台和地域解决异构资源动态共享与集成，且已经形成了统一的标准。网格技术将能够帮助云计算平台之间实现互操作，达到云计算设施一体化目标。
- 主要面向松耦合型数据处理的云计算管理着由个人计算机和服务器构成的计算资源池，对于紧耦合型任务，由于不容易分解，节点之间频繁通信，显然云计算处理的效率很低。因此，云计算与网格一体化可以使它们充分发挥各自特点。
- 网格技术集成海量的科学数据，如物种基因数据，气象数据和海洋数据等；云计算主要面对商业数据。两者集成在一起，可以很好地扩大云计算的应用范畴。
- 使用云计算必须得将各数据，应用和系统集中到数据中心，改变系统的运行模式与迁移到云计算平台面临难度，考虑到成本和数据中心可能随时更新，假如

要不断向数据中心传送数据，则会造成大量网络带宽消耗。网络计算可以实现“物理分散，逻辑集中”的目标，有限解决应用系统分散运转，没能集中到云计算平台。

- 云计算涉及到敏感的数据安全问题，用户担心自己数据被非法利用和丢失，而网络中的数据保存在原来数据中心，由所有者掌控，但是提供给外界数据访问服务，数据使用权不丧失，数据资源使用范围进一步扩大，利用率提高。不同的所有者掌握着各自数据源头，有效避免云计算数据中心的敏感数据扩散。

## 2.3 体系结构

目前不同的云计算厂商提供各自云计算的解决方案，缺乏一个统一的技术体系结构。

### 2.3.1 一般体系结构

综合不同厂商的解决方案，构造了一个可供参考的云计算体系结构<sup>[14]</sup>。体系结构如图 2-2 示，较全面的概括了不同的方案主要特征。

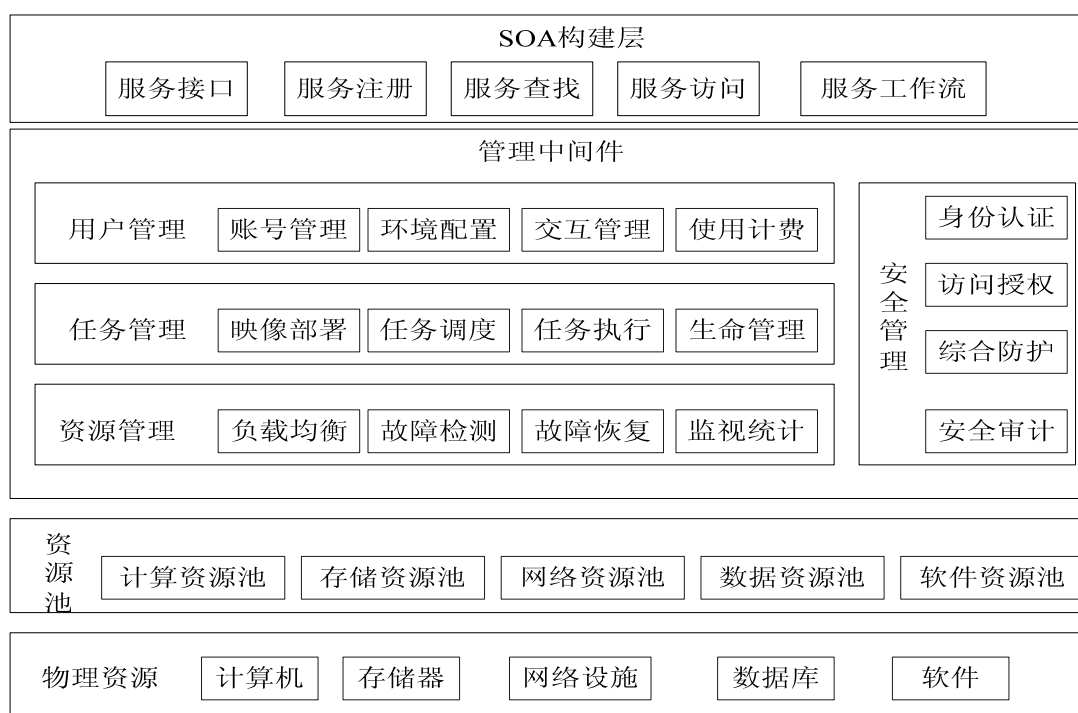


图 2-2 云计算体系结构

Fig.2-2 The Architecture of Cloud Computing

云计算技术体系结构分为：SOA（Service-Oriented Architecture, 面向服务体系结构）层、管理中间件层、资源池层和物理资源层。

#### （1）物理资源层

物理资源层是指分布在不同地理位置的资源,包括存储器、计算机、网络设施,软件和数据库等。物理资源层是云计算的底层基础设施,提供者负责对其运行、管理、维护和升级。

### (2) 资源池层

资源池层是将大量相同类型的资源构成同构或接近同构的资源池,如计算资源池、数据资源池等。构建资源池更多是物理资源的集成和管理工作。云计算的核心理念是资源池,池的规模可以进行动态扩展来满足应用和用户规模增长需要。

### (3) 管理中间件层

云计算管理中间件负责的是用户管理、任务管理、资源管理和安全管理等工作。资源管理实现对云资源节点均衡使用,检测节点故障并试图对其恢复或屏蔽,监视统计资源使用情况;任务管理对提交的任务进行执行;云计算商业模式的实现不可缺少对用户的管理,涉及有用户身份的管理和识别、用户程序的执行环境配置、用户交互管理,对用户使用予以计费等;安全管理可以保障设施整体安全性,主要涵盖安全审计、综合防护、访问授权和身份认证等。

### (4) SOA 层

SOA 层封装云计算能力成标准 Web Services 服务,同时纳入 SOA 体系进行管理与使用,包括服务注册、查找、访问和服务工作流等。资源池层和管理中间件是云计算技术最重要的部分,SOA 层的功能更多依靠外部设施的提供。

## 2.3.2 面向市场的体系结构

面向市场的资源管理有必要用来指导云资源的供需以实现市场的均衡,向云用户与提供者反馈经济信息,提升基于 QoS 的资源分配机制满足用户的服务请求。澳大利亚墨尔本大学 Buyya 教授提出了一个面向市场的云计算体系结构<sup>[15]</sup>,如图 2-3 所示。

面向市场的云计算体系结构主要由以下部分组成:云用户/经纪人、资源管理器、虚拟资源池和物理资源组成。

- 云用户/经纪人

代表用户将世界各地的服务请求提交到相应的数据中心和云中。

- 资源管理器

资源管理器是体系结构的核心,包括用户驱动管理、计算风险管理和自治资源管理。它是数据中心/云服务提供者和外部用户/经纪人之间的接口,并通过以下机制交互来支持云资源的管理:(1)定价机制:定价机制决定服务请求采取何种方式收费,例如,提交时间(高峰期/非高峰期)、定价利率(如固定/变化)或资源的可用性(如供应/需求)。(2)审计机制:记录被请求资源的实际使用,计算最终成本。另外,所记录的历史使用信息

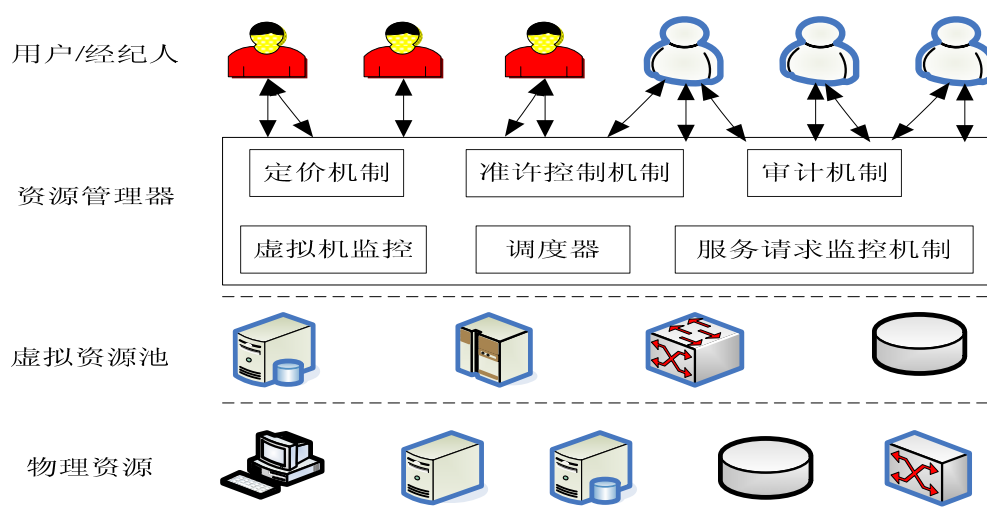


图 2-3 面向市场的云计算体系结构

Fig.2-3 Market-oriented Cloud Architecture

还可以被服务准许控制机制用来改进资源分配的决定。(3) 虚拟机监控：监视虚拟机的可用性及其资源所有权。(4) 调度器：分配相应的服务请求至虚拟机，并执行这些请求。(5) 服务请求监控机制：监视服务请求执行进展情况。(6) 准许控制机制：服务请求第一次被提交时，在确定是否接受或拒绝之前，为提交的 QoS 请求进行解释。有限的资源不能成功完成多个服务的请求，该机制可以保证资源没有超载。同时虚拟机监测机制中获得资源最新可用信息，从服务请求监控机制中获取工作量处理信息等，有助于更高效地进行资源配置。

#### ● 虚拟资源池

多虚拟机可在单一的物理机上动态启动或者停止以此满足接受的服务请求，这为应用程序提供了最大限度的灵活性。另外，由于每个 VM 在同一物理机上相互隔离，所以多个虚拟机可以同时运行在基于不同操作系统环境应用的单一物理机上。

#### ● 物理资源

云计算中心提供了多种计算服务器来满足不同用户的服务需求。

## 2.4 云计算应用

云计算将会是一场深刻改变 IT 格局的划时代变革，未来市场蕴含巨大的潜力，为此全球的 IT 巨头纷纷发布各自的云计算解决方案与战略，提供风格各异的云计算应用服务。下文对主流的云计算应用予以介绍，分析其中的关键技术和解决方案。同时对国内的典型云计算应用情况进行概述。

### 2.4.1 Microsoft 云计算

微软认为未来的互联网将是“云+端”的组合，以云为中心的世界，用户可以便捷

地使用各种终端设备访问云中数据与应用，微软希望依靠云计算为用户创造跨越不同设备的无缝体验。微软云计算战略包括三大部分：微软运营、伙伴运营，客户自建。微软云计算战略有三个典型特点：软件+服务；开放和安全的平台；用户自由选择。微软主要拥有三类云计算解决方案，即 Live 和 Online、动态云解决方案和 Windows Azure 平台解决方案。

### （1）Live 和 Online 方案

微软云计算应用有针对消费者和企业的两种服务，提供更多应用模式，允许应用软件服务的用户缩减投资、降低软件成本、随需随用。微软提供给消费者服务包括 Office Live、Windows Live、Live Messenger、Bing 等，对企业用户服务为 Microsoft Online Services。

### （2）动态云解决方案

微软提供的基于动态数据中心技术的云计算优化与管理方案。企业可以在此基础上快速构建内部使用的私有云平台，服务提供商也可以短时间内搭建云计算服务平台来对外提供服务。

### （3）Windows Azure 平台

微软云计算平台属于 PaaS 模式，提供可由互联网访问的基础设施（存储设备，处理器等），企业以及用户部署数据和应用程序在微软云计算平台上运行。此外，用户和企业开发和运行本地应用程序，也可利用云中存储数据或云计算基础设施服务。

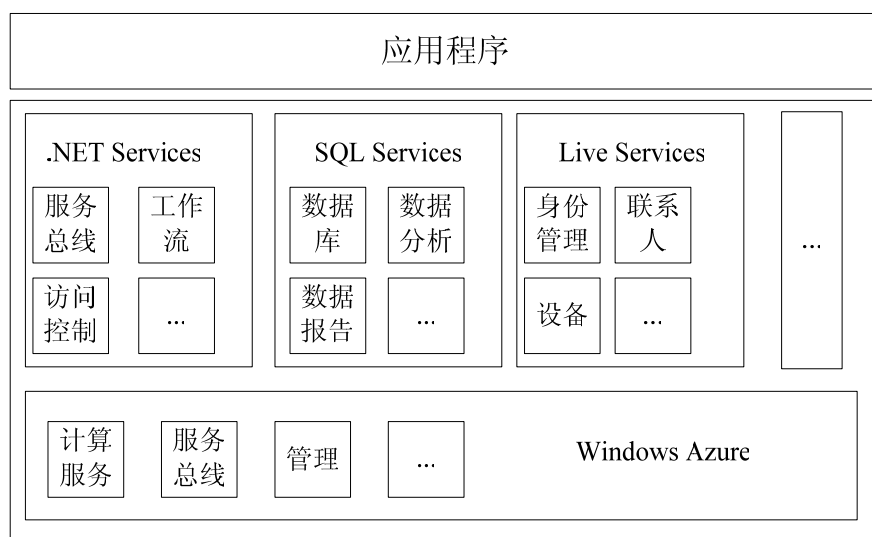


图 2-4 微软云计算服务平台

Fig.2-4 Microsoft Azure Services Platform

微软云计算服务平台主要包括四个组件<sup>[16]</sup>，如图 2-4 所示。位于最底层的 Windows Azure 是核心，提供一个在微软数据中心服务器上运行应用程序和存储数据的环境，是微软的云计算操作系统。为了更好构建应用程序，Windows Azure 提供 Blob、Table 存储应用程序中数据，Queue 被用来存储消息。Live Services 是微软基于软件加服务战略，

整合各种服务, 提供用户以便捷方式享受一站式不同类型在线服务。平台封装 Windows Live 应用程序功能, 并扩展至云端, 用户通过使用云平台获取 Live 提供的所有服务。SQL Service 为本地应用程序和云端提供可扩展、多租户、高可用的数据库服务。.NET Services 为本地和云端应用程序提供常用基础功能模块, 整合云平台与本地服务器上程序, 包括服务总线、访问控制和工作流服务。

### 2.4.2 IBM 云计算

IBM 提出的云计算战略是“智慧的云计算”, 经过多年探索与实践开发出先进的基础架构管理平台—IBM “蓝云”解决方案。该方案结合了业界最新技术, 充分体现云计算的理念。

该方案整合企业基础架构, 采用自动化与虚拟技术, 构建云计算中心, 使企业软硬件资源能够实现部署、分配、管理、备份与监控的统一, 改变应用独占资源的状况。“蓝云”向用户提供虚拟基础架构, 用户可自定义虚拟基础架构构成, 包括服务器、网络配置, 存储类型和大小等。用户借助服务界面提交请求, 每一请求的生命周期由平台来维护。图 2-5 是蓝云系统架构<sup>[17]</sup>。

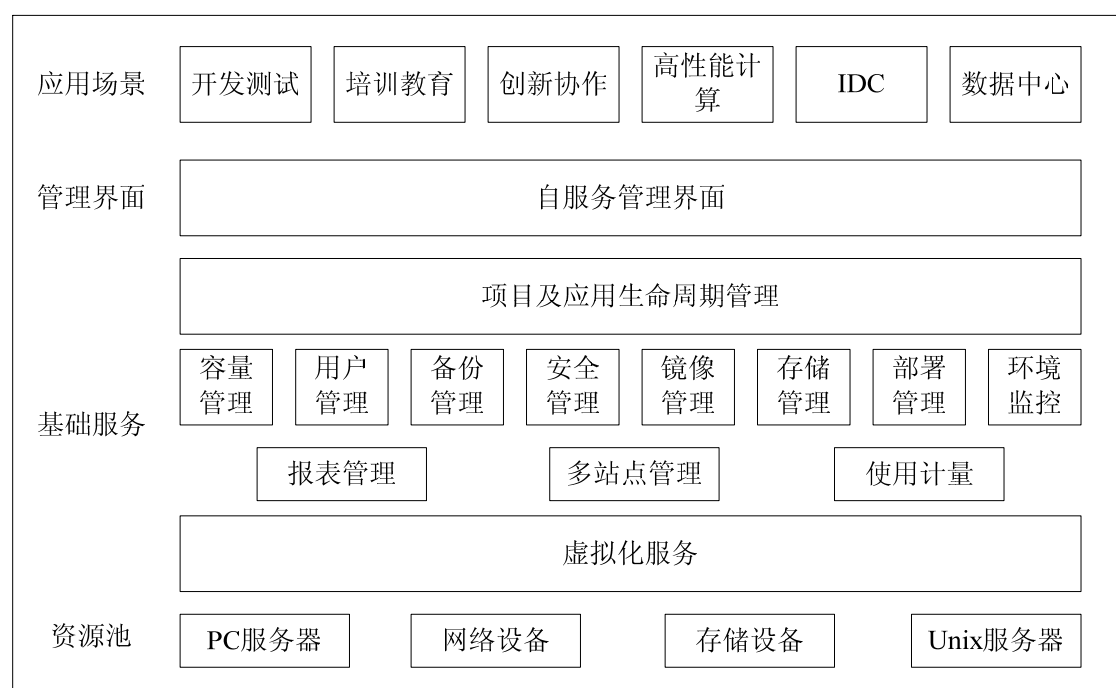


图 2-5 IBM 蓝云系统

Fig.2-5 IBM Blue Cloud System

“蓝云”解决方案由以下部分构成:

- 软硬件资源: 软件包括各种操作系统、中间件、数据库及应用, 如 Aix、DB2、Linux、Lotus、Rational 和 WebSphere 等。硬件有 x86 或 Power 服务器、存储服务器、交换机和路由器等网络设备。

- IBM Tivoli 管理软件：IBM 云计算中心开发，用于提供云计算服务。
- 咨询、部署及客户化服务：“蓝云”解决方案能够依照客户的需求与具体应用场景二次开发，实现云计算管理平台和客户已有软硬件有效整合。

### 2.4.3 Google 云计算

Google 不仅拥有强大的搜索引擎，还提供 Google Docs、Google Apps、Google App Engine、Chrome Browser、Gmail 等多种业务。这些应用需要实时向全球用户提供服务，数据量巨大，Google 云计算技术很好解决海量数据存储与快速处理的问题。Google 云计算技术架构如图 2-6 所示。



图 2-6 Google 云计算技术架构

Fig.2-6 The Architecture of Google's Cloud Technology

Google 云计算技术具体有：海量数据并行编程模式 MapReduce，海量数据数据存储与访问能力的 Google 文件系统 GFS，保证在分布式环境下并发操作同步问题的分布式锁服务 Chubby 以及分布式结构化数据存储系统 BigTable。

#### （1）MapReduce

MapReduce 是一个解决大规模数据并行运算的软件架构，向用户提供一个强大与便利的接口，允许把海量计算自动并发且分布执行，具备较好的通用性。MapReduce 执行流程<sup>[18]</sup>如图 2-7 所示。

- 1) 用户程序调用 MapReduce 函数，第一步是分割输入文件成 N 块，每块大小可由一定参数决定，进而在集群机器处理执行程序。
- 2) 主控程序选择空闲工作机来分配 Map（映射）或 Reduce（化简）任务。
- 3) 已经分配了 Map 任务的工作机读取相应输入并进行处理，Map 函数产生中间结果缓冲到内存。
- 4) 缓冲到的中间结果定时写入本地硬盘，分成相应区域，主控程序接收中间结果的位置信息，并发布给负责化简的工作机。
- 5) 化简的工作机远程读取中间数据，对 Map 函数的参数进行排序归类等操作。
- 6) 中间结果值集合递交给用户自定义的 Reduce 函数，输出 Reduce 函数结果。

7) Map 和 Reduce 任务完成，主控程序激活用户程序，返回调用点。

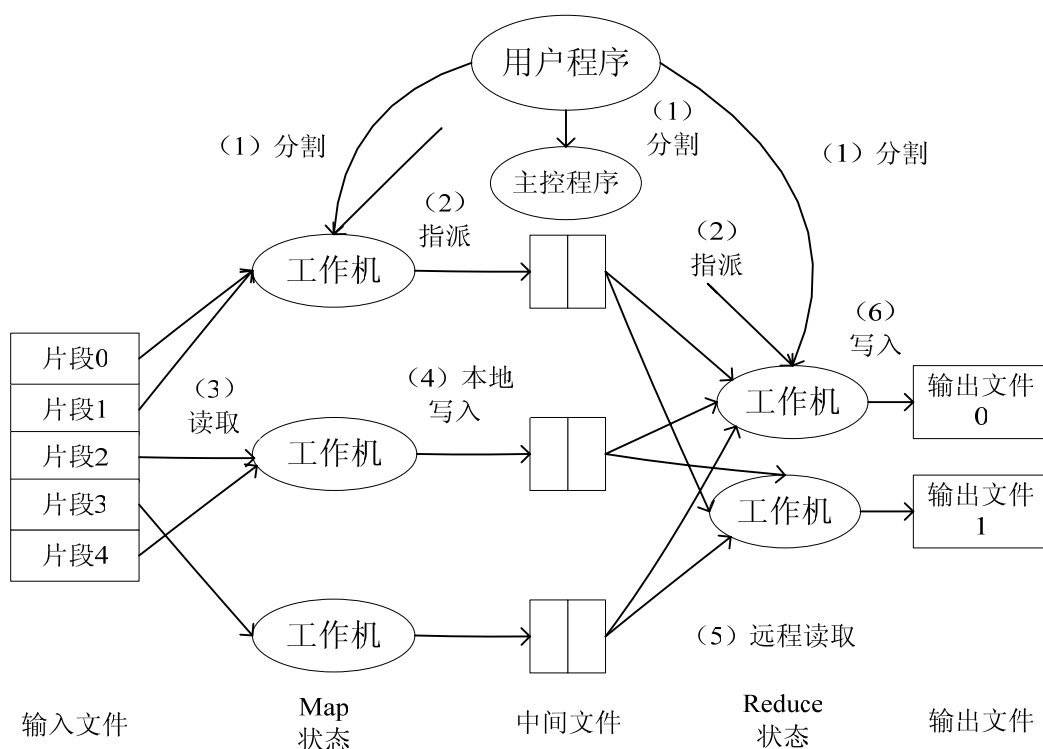


图 2-7 MapReduce 执行流程图

Fig.2-7 MapReduce Execution Overview

## (2) GFS

GFS(Google File System)是一大型分布式文件系统，为 Google 云计算提供海量存储而设计，是所有核心技术的底层。GFS 系统架构<sup>[19]</sup>如图 2-8 所示。

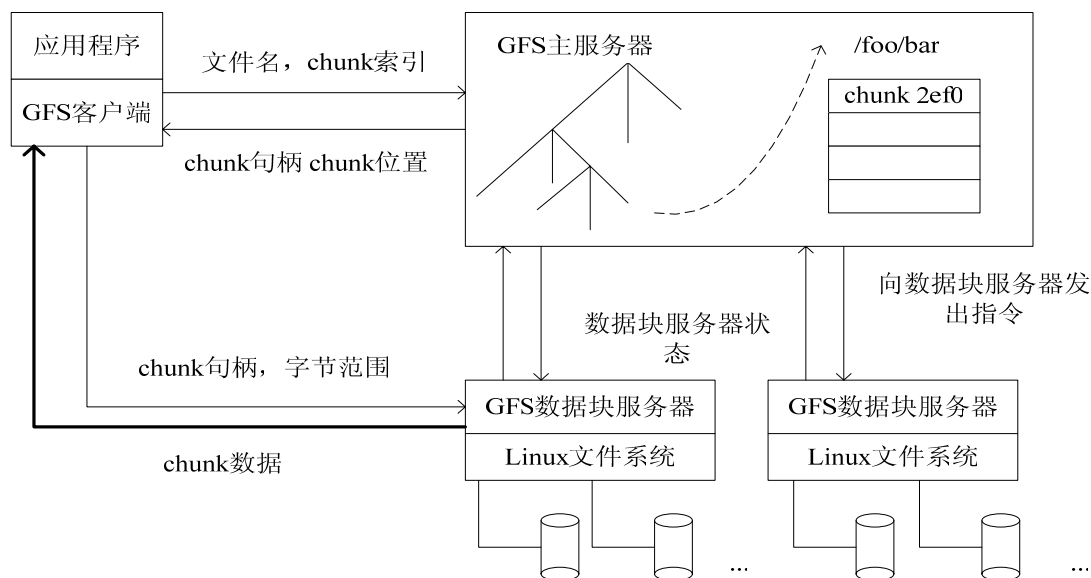


图 2-8 GFS 系统架构

Fig.2-8 Google File System Architecture



GFS 将整个系统节点分为三部分：提供专用应用程序访问接口的客户端；对系统元数据保存，管理整个文件系统的主服务器；负责存储的数据块服务器。GFS 把文件按照固定大小(默认 64MB)进行分块，每一块为一数据块(chunk)并且有对应的索引(index)。客户端首先访问主服务器节点，获得将要进行交互的数据块服务器的信息，之后则直接访问完成数据存储。这种设计思想有效分离了数据流和控制流，极大降低主服务器负载，同时由于文件分为多个数据块，客户端可以同时访问，提高了系统的整体性。

### (3) Chubby

Chubby 是基于松耦合分布式系统而解决分布一致性问题文件系统，用户使用 Chubby 锁服务可以达到数据操作中的一致性。但是它不是强制性的锁，只是一种建议，为了提高系统的灵活性。这种锁服务可以实现高可靠、高可用性以及文件的直接存储等特性。

### (4) BigTable

Google 开放众多服务，运行繁忙的系统，需要处理各种数据和接受海量服务请求，商用数据库无法满足这些需求。BigTable 是基于 Chubby 和 GFS 的分布式存储系统，能够很好满足 Google 需求，组织架构<sup>[20]</sup>如图 2-9 所示。

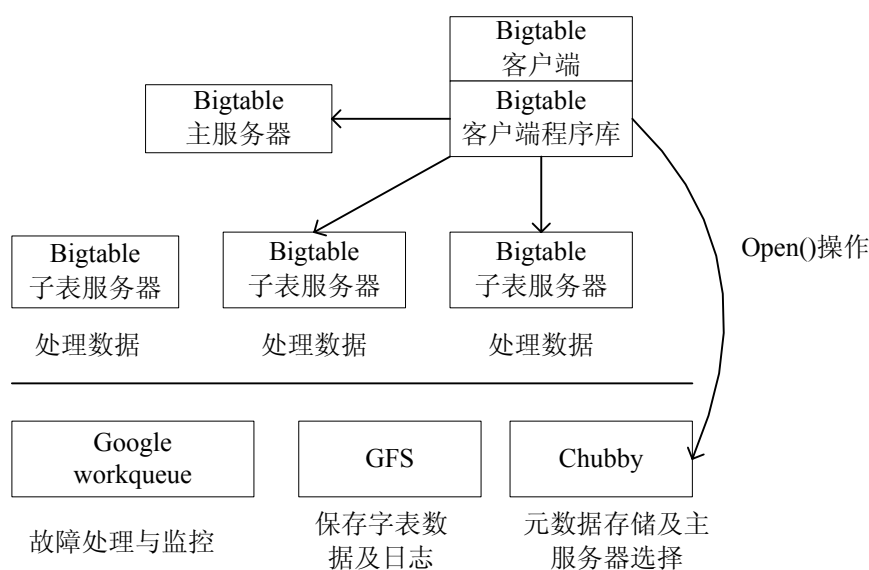


图 2-9 BigTable 系统的组织结构

Fig.2-9 Organization of BigTable System

BigTable 由三部分组成：一主服务器，客户端程序库以及多个子表服务器。workqueue 用于处理分布式系统的分组与任务调度，GFS 负责保存子表数据及日志，BigTable 需要锁服务 Chubby 的支持。客户端对 BigTable 服务访问时需要通过 Open()操作打开锁（获得文件目录），然后才可以和多个子表服务器通信。客户端主要和子表服务器通信，降低了主服务器的负载。主服务器管理元数据，负责对子表服务器的远程管理与负载调配。客户端利用编程接口和主服务器进行元数据通信，与子表服务器数据通信。

表 2-2 概述了其他一些典型云计算厂商的战略与解决方案。

表 2-2 云计算厂商战略与解决方案概述

厂商	云计算理念与战略	解决方案概述
HP	一切皆服务，重点关注企业私有云和下一代数据中心	HP Cloud Assure 管理云服务，HP Operations Orchestration 帮助企业自动配置服务；CloudStart(私有云快速部署方案)；HP Server Automation Server、HP Blade System 等
Oracle(SUN)	企业级私有云和公共云服务提供商，云服务主要是在线 CRM 等	基于 SaaS/PaaS 的 Oracle Platform 及 Oracle IaaS 服务；Oracle Enterprise Manager；Sun Cloud, 包含管理程序 (Sun xVM Server)、网络 (Crossbow)、存储 (COMSTAR, ZSF) 等
Salesforce	Salesforce 新的价格模式和新的特色功能就是，把平台作为一种服务	Force.com 是 Salesforce 的 PaaS 层面的云计算服务，提供用于创建应用程序的各种技术，包括 web 服务器，数据库等。目前有三个版本，免费版、企业版和无限版
VMware	虚拟化为基础，提供云计算解决方案，实现“IT 即服务”	基础架构和运营管理 VMware vCenter 等；应用平台解决方案 VMware vFabric 等；数据中心 VMware vSphere 等；安全产品 VMware vShield Product Family 等；桌面虚拟化产品等
Amazon	全球最早提出云计算概念，并将云计算应用于中小企业	弹性计算云 EC2、简单存储服务 S3、简单数据库服务 Simple DB、简单队列服务 SQS、内容推送服务 CloudFront、电子商务服务 DevPay、弹性 MapReduce 服务

#### 2.4.4 国内典型云计算应用

中国的云计算应用也在发展中。中国移动为满足低成本、高性能、可扩展、高可靠性 IT 计算与存储的需要，互联网业务和服务的需求，启动“大云”（BigCloud）研发计划。已经在 2010 年发布大云 1.0 成果。中国移动的“大云”实现以下关键功能<sup>[21]</sup>：1、弹性计算系统（BC-EC）。管理和调度计算、网络、存储资源，提供弹性计算服务。2、分布式计算框架(MapReduce)。采取 MapReduce 并行编程模式，把任务自动分成多个子任务，保证后台复杂并行执行和任务调度对用户和编程人员透明。3、分布式海量数据仓库(Hugetable)。数据管理采用列存储模式，保证海量数据存储和分析性能。4、云存储系统（BC-NAS）。提供多种便捷文件获取方式，支持文件共享。5、数据挖掘工具(BC-PDM)。提供基于 SaaS 的数据挖掘服务，支持低成本高性能的商务智能应用开发。

中国联通以 IDC 业务为突破，积极建设“互联云”项目。联通将逐步构建全球云、云联云和互联云架构，在相关的软件和协议下实现不同云的互联，集成软件、硬件、网络、应用和服务实现综合性平台。目前已经在进行互联云试点工作。

中国电信 e 云目标定位是为家庭与中小企业用户提供可靠安全在线备份的服务，提供给用户 2GB 的免费存储空间。“号码百事通”是基于中国电信 114 增值业务统称，延

伸并拓展了传统查号业务，满足现代用户现实和潜在的各类信息查询的需求，将 114 台打造成一个综合信息服务平台。

华为启动了“云帆”计划，与合作伙伴共同为各行各业提供解决方案。华为云计算战略包含三方面：开放与共赢；推动业务与应用云化；构建云平台，促进资源共享、效率提升与节能环保。华为云计算的解决方案有 SingleCLOUD 云平台解决方案和电信应用云解决方案，其中电信应用云解决方案包括云计算业务交付平台和 BSS 系统<sup>[22]</sup>。

阿里巴巴积极筹建“电子商务云计算中心”，期待打造“商业云”体系。与其他云计算中心比较，阿里云出租计算、存储和网络服务外，还推出适合国内的电子商务服务。

隶属于世纪互联的云快线科技有限公司成功推出国内首个商用云计算服务平台“CloudEx”（云快线弹性计算平台），帮助数以千计中小企业在线方便获取 IT 基础设施资源。业务范围涵盖云主机、云托管、云备份、云存储、云立方、CDN 加速等商业应用服务。

云安全指的是互联网连接用户和安全平台，构成一个对垃圾邮件、木马、病毒、恶意软件等内容的庞大监测网络，每个用户都为云安全贡献一份力量，同时也可以分享其他用户的安全成果<sup>[3]</sup>。目前 360 安全卫士、趋势、江民科技、瑞星、金山、卡巴斯基等都已经推出了自己的云安全解决方案。

中国云计算的产业链<sup>[23]</sup>如图 2-10 所示，在政府的监管下，云计算服务提供商和硬件提供商、网络基础设施提供商以及云计算的咨询、规划、交付、运维、集成服务商、终端设备厂商等共同构成了云计算产业生态链，为政府、企业与个人用户提供云计算服务。

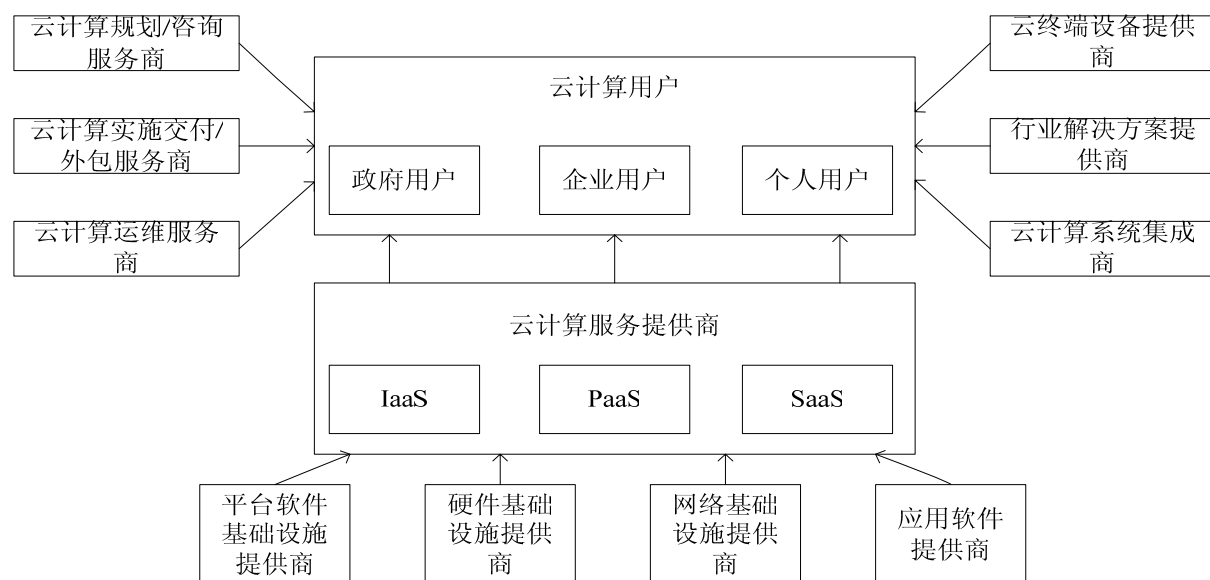


图 2-10 中国云计算产业链

Fig.2-10 China's Cloud Computing Industrial Chain

## 2.5 云计算标准化

相比网格计算有统一的国际标准，云计算的标准化势在必行。云计算的标准化是云计算推广并应用的前提和终极目标。现在国际、国内相关组织已经开始研究和探讨云计算标准化。

云计算标准化的主要内容包括：云计算互操作与集成标准、云计算服务接口与应用程序开发标准、云计算的不同层面间接口标准、云计算服务的目录管理、云用户提高资产利用率标准、资源与性能优化、评估性能价格比标准以及安全与隐私标准。

国外云计算相关标准开发组织在云计算标准化方面的工作如表 2-3 所示。从中看出，开发组织或联盟致力于研究云计算一个或一些技术或者是操作方面标准。

表 2-3 国外云计算相关标准开发组织工作概览

标准化组织	主要成员	工作概述
开放网格论坛 (OGF)	Sun、Oracle、IBM、Intel、HP、Microsoft、eBay、AT&T、Fujitsu 等	成立开放云计算接口 OCCI(Open Cloud Computing Interface)工作组, OCCI 是管理和共享云计算基础设施的应用程序接口, 已发布 4 项规划草案
云计算互操作论坛 (CCIF)	Cisco、IBM、Intel、RSA Thomson Reuters 等	组织制定《统一的云接口》规范, 旨在建立开放与标准化云接口, 屏蔽不同的基础设施接口间差异, 从而达到实现不同云之间互操作
分布式管理任务组 (DMTF)	Citrix、Cisco、HP、IBM、Intel、Microsoft、Sun Novel、Vmware 等	发布开放虚拟化格式 (OVF) 标准, 用于封装及分发给虚拟机上运行的虚拟工具 (或软件) 开放标准, 一种开放安全、可迁移、有效和可扩展格式
开放云联盟 (OCC)	芝加哥伊利诺伊大学、MIT、Yahoo、Cisco 等	研究并制定云扩展性、数据传输协议云计算技术和服务相关的测试规范, 成立大型数据云互操作、开放云测试、开放科学数据云、内部云测试工作组
云安全联盟 (CSA)	Qualys、PGP、zScaler eBay 等	2 项标准化研究报告, 2 项云计算相关标准; 同时还发布了云控制框架, 在 13 个领域提供安全认识和原则概念指导
欧洲电信标准化协会 (ETSI)	电信相关管理机构、网络运营商、设备制造商、用户研究机构等	强调对可扩展计算与存储资源的网络访问; 标准化方面, 关注云计算互操作性方法, 重点研究作为服务交付模型的基础设施 (IaaS) 相关的规范和标准
开放云计算宣言 (OCM)	300 多家公司和组织参与支持	收集云计算需求并发起讨论, 为其他标准化组织提供参考, 目前已经发布了《云计算案例白皮书》V4.0

国内已经具备开展云计算标准化工作条件和基础, 我国也积极开展云计算的标准化工作, 以期在未来云计算核心技术发展中掌握主动权。

全国信息技术标准化技术委员会 SOA 标准工作组成员有复旦大学、IBM、大唐软件、东方通、浪潮、北京邮电、上海软件行业协会、世纪互联等。工作组稳步有序开展云计算标准以及 SOA、虚拟化和中间件等技术标准研究与制定。2010 年 11 月 19 日,

由中国电子技术标准化研究所、SOA 标准工作组、ITSS 工作组联合举办“中国云计算标准化研讨会”。会议上发布了《云计算标准研究报告（征求意见稿）》，分别从云计算定义、应用情况、技术分析、标准化工作等方面详细介绍了云计算现状。

信息技术服务标准工作组秘书处组织中国移动、微软、百度、阿里巴巴、腾讯、万国数据、曙光、联想、清华、中搜等国内外从事云计算应用研发的相关企事业单位参与了云计算标准化预研工作，形成《云计算标准化研究报告》。我国基于该报告提出了云计算国际化工作建议，作为中国的贡献提交 SC38（分布应用平台与服务，Distributed Application Platforms and Services）秘书处。

云计算相关企业、科研院所、相关机构自发、自愿组建的开放式、非营利性技术与产业联盟中国云计算技术与产业联盟（China Cloud Computing Technology and Industry Alliance, CCCTIA）2010 年 1 月在北京正式宣布成立。产业联盟旨在为国家与政府相关部门提供云计算技术和产业相关合理化建议；积极引导用户正确应用云计算技术；主导、推动并参与国际、国家以及行业标准形成与制定；突破关键技术，提升云计算行业核心竞争力。

## 第三章 资源分配相关研究

### 3.1 网格资源分配与调度研究

云计算是网格进一步的发展,在一定层面上,云计算核心技术中的资源分配与调度和网格计算具有类似性,有着较强的借鉴意义。资源分配与调度在网格计算中进行了深入地研究与探讨,而云计算中的资源分配相关研究并不多,其商业性以及虚拟化技术的运用也使得云环境下的资源分配与调度有着新的特征。

#### 3.1.1 网格资源特征与调度目标

网格中资源具有以下特征<sup>[24]</sup>:

- (1) 分布共享 网格资源分散在不同的地理位置,使用者可以共享网格中的资源,分布式资源共享是网格计算需要解决的核心。
- (2) 资源多样性 网格计算环境下资源类型是异构与多样性的,系统需要很好的解决种类不同资源的互操作以及通信。
- (3) 自治和多重性 某组织或者是个人对其资源具有管理的权限,同时资源也接受网格统一管理。
- (4) 动态增减 网格资源与功能在某一个阶段可能是确定的,但随着时间变化,原有的资源与功能可能会动态增加或动态减少。

网格调度总体目标就是要最大限度发挥网格的性能来满足用户所提交任务的调度需求,性能评价标准有<sup>[25]</sup>:

##### (1) 高性能指标

任务完成时间是任务在特定的节点上执行完成时刻,任务开始执行第一条到最后一条指令所经历的时间为任务执行时间。调度目标就是使得任务总完成时间最短。

##### (2) 服务质量 QoS

用户希望尽量好的服务来满足其特定的需求,通过提交任务的 QoS 形式予以反映。网格环境涉及的不同实体在安全、管理以及费用等方面的 QoS 指标各异,甚至互相矛盾,有效保证服务质量极其必要。

##### (3) 负载均衡

负载均衡影响到网格节点进行协同计算,其目标是不仅解决节点间均衡,同时使远程通信与负载重分配产生最小新开销,提升网格计算的可用性与灵活性,实现用户优质服务和高资源利用率。

##### (4) 经济原则

资源提供者与网格用户在市场经济机制下实现互利共赢才能保证网格系统的长远发展。

### 3.1.2 网格资源管理模型

网络资源分配与调度是网络资源管理的一部分,在预先设定的标准下,通过管理软件保证网络资源合理分配与利用,实现协同共享资源与求解的目标。目前从体系结构层面资源管理模型有<sup>[26]</sup>:

### (1) 层次模型

整个资源管理系统被划分为若干功能层, 较低层组件提供服务实现较高层的功能, 这种设计思想较好解决底层异构性, 站点自治并在一定程度实现资源联合分配。该模型已经广泛应用于现今网格计算系统。

## (2) 抽象所有者模型

抽象所有者（资源经纪人）和用户交互协商，任务的提交与收集结果时依据订购、交货形式。该模型把每个资源甚至网格使用一个或多个抽象所有者加以表示，用户则签订合同或对话抽象所有者来获取资源。

### (3) 经济/市场模型

计算经济模型的采用极大促使资源拥有者不断更新资源与提供共享服务，进一步便利用户使用。网格中不同资源有着特定的管理机制，对用户收取不同的费用。用户与资源所有者有各自经济诉求，互惠互利很有必要。该模型综合运用层次与抽象所有者模型；实现面向用户为中心的资源分配和管理机制；提供所有资源给用户公平交易；提升计算服务质量，促进资源的升级。图 3-1 反映了基于计算经济的网格资源管理模型<sup>[27]</sup>。

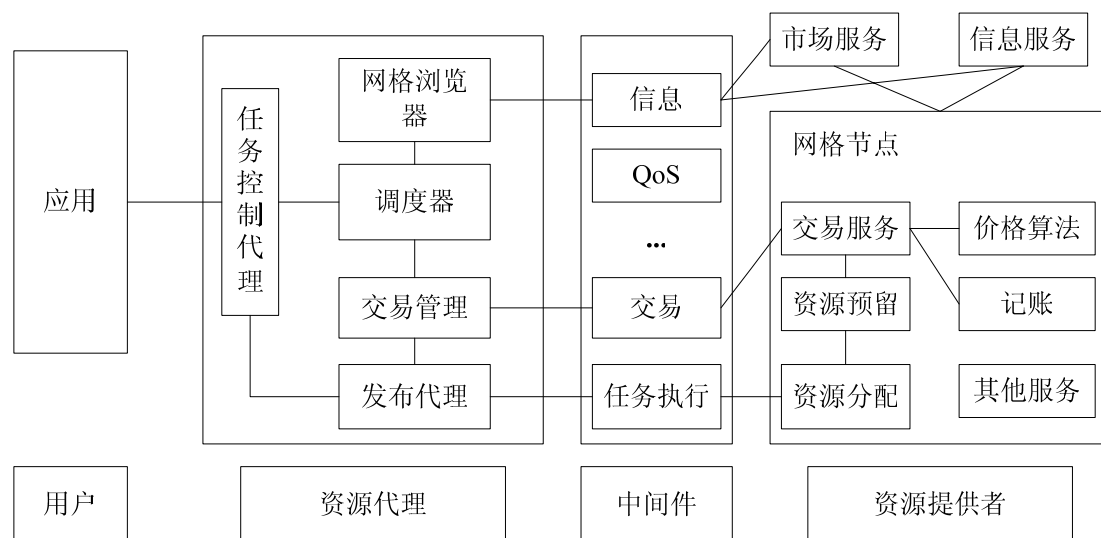


图 3-1 基于计算经济的网格资源管理模型

Fig.3-1 The Grid Resource Management Model for Computational Economy

### 3.2 云计算资源分配研究

云计算资源分配效率直接影响到整个云环境的性能,是云计算技术的核心组成。探讨云计算环境下的资源分配很有必要,相关研究还应进一步地深入。

Weiming Shi<sup>[28]</sup>对云计算环境下有预算约束的大量独立相同任务进行研究。相比于最小化任务总完成时间,作者更关注的是系统的稳定吞吐率,定义了预算约束和通信约束两种系统模式,相对应的最佳分配机制是利益优先和带宽优先。Shin-ichi Kuribayashi<sup>[29]</sup>发展了云计算环境下的资源分配模型,向每个服务请求同时分配处理能力和带宽;定义了云环境下多用户间资源分配的基本原理与公平性评估,提出公平性联合多资源分配方法,实现不影响资源效率的同时多用户间公平地资源分配。Bo An et al.<sup>[30]</sup>探索基于市场经济下资源提供者和用户自动协商资源租赁合同模式,考虑资源供需不确定、动态变化,提出分布式协商机制,由代理确定合同价格以及违约处罚金。Hesam Izakian et al.<sup>[31]</sup>运用面向类市场技术到网格计算中调节资源的供需,激发资源提供者和用户进行交易。翁楚良,陆鑫达<sup>[32]</sup>提出一种改进双向拍卖机制的网格资源分配方法,采取统一拍卖的方式,灵活调节交易的双方付费,分析了该双向拍卖机制特点,并定义拍卖机制效率。Ikki Fujiwara et al.<sup>[33]</sup>提出市场机制为参与者有效提供云服务:定制组合服务和预定现货市场服务。华夏渝等<sup>[34]</sup>提出基于蚁群优化计算资源分配算法,表明在响应时间和运行质量方面算法更适合云环境。Wei-Yu Lin et al.<sup>[35]</sup>提出动态拍卖机制解决云计算环境下计算性能的分配问题。

### 3.2.1 云资源特征与分配目标

为了实现资源共享,云计算提供给使用者诸如存储资源、计算资源、通信网络资源、数据资源等。资源特征主要有:分布多样、动态异构、互操作以及抽象层次等。资源的这些特性要求云计算应该屏蔽资源动态异构特点,评估资源性能,提供访问接口,保障用户的多样化需求,同时也要高度重视资源的安全性。

云计算环境包括服务提供者、用户和市场。服务提供者和用户通过市场买卖抽象成服务的资源。资源分配需要满足资源提供者和用户双方的利益,基于市场化的资源分配机制满足以下目标需求<sup>[33]</sup>:

#### (1) 经济效率

经济效率的资源分配是不存在不降低一方利益的情况下增加另一方参与者的利益,也就是没有浪费的资源,最大化整体的福利是获得经济效率的充足条件。

#### (2) 灵活可预测性

云计算环境下资源实时动态改变,用户希望事先得到可预测的资源分配并不断调整适应。

#### (3) 服务组合

多数云计算应用要求特定顺序的多种服务组合运用,因此,用户在不同时间内需要捆绑多层次服务。

#### (4) 公平竞争

为鼓励资源提供者(卖方)与用户(买房)进行公平交易,资源交易价格必须依据供需的状况。



### 3.2.2 面向市场的拍卖机制

云计算环境下实现资源的最优分配是充满挑战性的课题，传统的资源管理方式不能简单地应用于云计算环境中。新的资源分配方法应该充分考虑用户和资源提供者的双方利益，一个可行的解决方案是采取基于市场的资源分配。图 3-2 反应了分配资源的经济模型<sup>[36]</sup>。

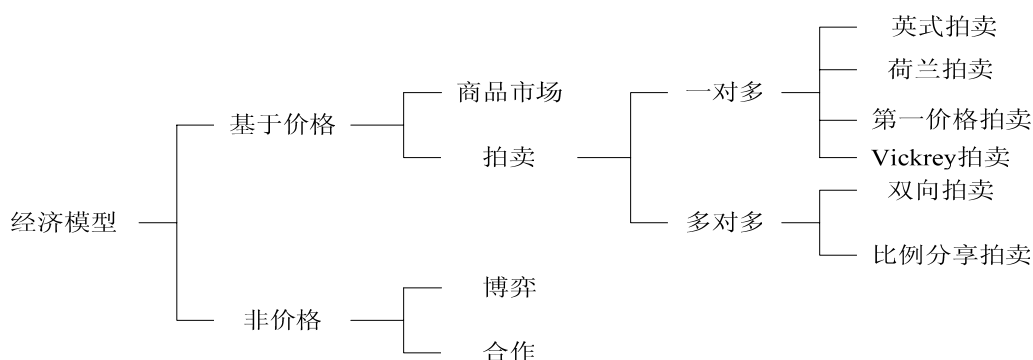


图 3-2 经济模型

Fig.3-2 Economic Models

有两种基于市场的模型适合分布式系统的资源管理<sup>[37]</sup>：一是商品市场模型，该模式下用户不参与而是由资源提供者对资源进行定价；二是拍卖模型，此模式下，用户与资源提供者双方进行协商确定最终交易成交的价格。

拍卖是由多个资源需求者提出不同的价格竞争资源,最终通过“拍卖人”根据不同拍卖的方式选定中标人并同时确定资源成交价格<sup>[38]</sup>。拍卖类型指的是拍卖活动过程中采取的制度规则，包含资源分配方法，出价方式和最终定价等标准。根据买与卖双方的交互方式分为单向与双向拍卖两类。单向拍卖（one-sided auctions）指多个买方和一个卖方或多个卖方与一个买方情况，用户提交竞价(bid)给拍卖商（auctioneer），拍卖商依据特定拍卖机制决定胜出者，如 First-price auction, Vickrey auction 等。双向拍卖（double auctions）指买卖双方都有多位参与者。资源提供者提出竞价，称为“要价”（ask price），用户也提出竞价（bid price），由拍卖商按照拍卖协议决定最终的成交价。三种最重要的双向拍卖是：普雷斯顿·麦卡菲双向拍卖（PMDA, Preston-McAfee Double Auction）、最低价格双向拍卖（TPDA, Threshold Price Double Auction）以及连续双向拍卖（CDA, Continuous Double Auction）。Kant<sup>[39]</sup>研究表明网格计算下运用 CDA 可以更好地为用户和资源提供者以高资源利用率。本文将积极探索在云计算资源分配与调度中引入连续双向拍卖模型,保障用户与资源提供者的利益，提高资源分配公平性，实现资源优化配置。

## 第四章 基于 CDA 的云计算资源分配

经济学中市场机制可以充分调动个体积极性, 激励资源共享, 实现合理资源配置。拍卖模型根据市场供求关系为资源提供者和用户协商适合的交易价格, 是快速可靠, 公平透明的市场价格设置策略, 能够揭示实际需求, 适应异构与分散用户, 具有灵活性。网络计算中的拍卖作为分布式计算资源分配策略已经得到了广泛研究。云计算的商业性质也特别适合拍卖机制, 有着广阔的研究前景。

对于用户与资源提供者, 采取固定的价格机制收费是不公平的, 因为不同的用户对不同资源有着与众不同的需要与需求, 而且会随着时间改变<sup>[40]</sup>。本章重点探讨研究基于连续双向拍卖的云计算资源分配。

### 4.1 云计算资源分配模型

云计算环境中, 有效分配资源并获取各自经济利益是云用户和资源提供者的共同目标。云计算资源分配模型中, 假定有  $m$  个云用户和  $n$  个资源, 其中每个云用户都含有一个独立的任务需要提交执行并付出相应的费用, 同时资源提供者出租计算资源获取利益。

云用户集合为  $U = \{u_1, u_2, \dots, u_m\}$ , 资源集合设为  $R = \{r_1, r_2, \dots, r_n\}$ 。设资源  $r_i$  ( $1 \leq i \leq n$ ) 的特征五元组为  $s_i = (id_i, cpu_i, st_i, rp_i, mp_i)$ , 其中  $id_i$  为资源唯一标识。  $cpu_i$  为资源  $r_i$  的计算速度, 用每秒百万条指令 MIPS (Million Instructions Per Second) 表示。  $st_i$  是资源  $r_i$  的工作量, 指的是对于已经接受的任务资源需要  $st_i$  秒执行完成, 或者是接受下一个新任务执行开始的时间。  $rp_i$  是资源的保留价格 (reserve price),  $mp_i$  是使用资源的最高价格 (maximum price), 单位是 \$/MIPS。资源最高价格的设置可以通过资源提供者或资源代理间的协商决定。

源于互联网性能的参数 QoS (Quality of Service) 在云计算中被用于评判用户对云服务的满意程度的指标, 具有商业性质的云计算需要不断提供给多样化用户满意的服务质量<sup>[41]</sup>。本文关注截止时间和预算, 并定义云用户  $u_j$  ( $1 \leq j \leq m$ ) 的任务特征四元组是  $t_j = (id_j, l_j, b_j, d_j)$ 。  $id_j$  指的是云任务  $t_j$  的唯一标识,  $l_j$  代表任务  $t_j$  长度, 单位是 MI (Millions of Instructions)。  $b_j$  是分配给任务  $t_j$  的预算,  $d_j$  意指任务的截止时间, 是云用户希望任务完成的时间。云用户希望在相应截止时间与最小化费用的情况下执行各自的任務。图 4-1 简单表示了调度方案, 通过拍卖商合理分配云用户的任务给相应的资源执行处理。



图 4-1 调度方案示意图

Fig.4-1 General View of Scheduling Scheme

## 4.2 CDA 拍卖模型

相比于云用户代理关注执行任务的截止时间和花费，资源代理期望获得更多利益，为了这个目的，资源提供者试图以更高的价格出让资源并和其他资源竞争执行更多的任务。本文的连续双向拍卖模型中，在每个时间单位内分别设置了云用户代理和资源代理的定价策略，云用户与资源分别提交竞价和要价给拍卖商。拍卖商管理当前的竞价列表与要价列表，当最高的竞价不低于最低的要价时，拍卖商匹配这两个出价。最终交易将会以匹配的竞价与出价的平均值来完成。竞价与要价分别由用户与资源提供者按照自身的目标独立决定。拍卖商匹配竞价与要价实现了云计算环境下用户性能 QoS 和资源提供者经济层面的 QoS。

### 4.2.1 云用户代理策略

用户代理在每个时间单位内独立决定竞价值基于以下两个参数：剩余资源数与平均剩余时间，参数可以很好反映资源的供求与价格关系<sup>[42]</sup>。

任务  $t_j$  竞标资源  $r_i$  需要满足：资源在任务截止时间内可以完成执行任务，见式 (4-1)；资源预留价格不高于任务可支付的最高费用，见式 (4-2)：

$$d_j - st_i - \frac{l_j}{cpu_i} \geq 0 \quad (4-1)$$

$$\frac{b_j}{l_j} \geq rp_i \quad (4-2)$$

(4-1) 式中：  $\frac{l_j}{cpu_i}$  一任务  $t_j$  在资源  $r_i$  上的执行时间。

(4-2) 式中：  $\frac{b_j}{l_j}$  一任务  $t_j$  每 MI 可支付最高费用。

在用户提交任务时，此刻剩余的资源数最大，设为  $n_j^{\max}$ ，随着资源不断接受新任务，剩余的资源数逐渐减少。在每个时间单位 T 内，任务  $t_j$  基于剩余资源数的竞价可以由以下多项式获取。

$$bid_j^{resource}(t) = \left[ \frac{\overline{rp}}{\varphi_j} + \left( 1 - \frac{\overline{rp}}{\varphi_j} \right) \left( 1 - \frac{n_j^t - 1}{n_j^{\max}} \right)^{\frac{1}{\alpha}} \right] \cdot \varphi_j \quad (4-3)$$

$$\varphi_j = \frac{b_j}{l_j} \quad (4-4)$$

$$\overline{rp} = \frac{1}{n} \sum_{i=1}^n rp_i \quad (4-5)$$

$\overline{rp}$  表示剩余资源的平均保留价格,  $n_j^t$  是任务  $t_j$  在时间  $t$  时剩余的可选资源数。参数值  $\alpha \in [0.01, 100]$ , 可以改变多项式形状曲线:  $\alpha < 1$  时, 用户保持一个较低的竞价值直至剩余资源数接近于 0;  $\alpha > 1$  时, 用户一开始竞价值就较高。接近  $\varphi_j$ 。

在时间单位  $T$  内, 用户代理根据竞标资源的平均剩余时间确定竞价值。假定任务  $t_j$  在时间  $t (0 \leq t \leq T)$  被提交, 用户  $u_j$  的任务竞标资源  $r_i$  的剩余时间是:

$$rt_{ij}(t) = \left( d_j - st_i - \frac{l_j}{cpu_i} \right) \cdot \delta_{ij}, \delta_{ij} = \begin{cases} 1, & \text{if } \varphi_j \geq rp_i \\ 0, & \text{otherwise} \end{cases} \quad (4-6)$$

式中:  $\delta_{ij}$ —限制参数。当  $rt_{ij}(t) < 0$  时, 表示资源  $r_i$  不能满足在截至时间内执行完成任务  $t_j$ , 竞价平均剩余时间:

$$\overline{rt_j}(t) = \sum_{i=1}^n rt_{ij}(t) \cdot \frac{\phi_{ij}}{n_j^{\max}}, \phi_{ij} = \begin{cases} 1, & rt_{ij}(t) \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (4-7)$$

任务  $t_j$  基于平均剩余时间的竞价可以由以下多项式获取:

$$bid_j^{time}(t) = \left[ \frac{\overline{rp}}{\varphi_j} + \left( 1 - \frac{\overline{rp}}{\varphi_j} \right) \left( 1 - \frac{\overline{rt_j}(t)}{rt_j^{\max}} \right)^{\frac{1}{\beta}} \right] \cdot \varphi_j \quad (4-8)$$

式中:  $rt_j^{\max}$ —在任务提交时, 最大的平均剩余时间;  $\beta \in [0.01, 100]$  与上述  $\alpha$  类同。综合式 (4-3) 与 (4-8) 可以得到基于剩余资源数和平均剩余时间的最终竞价值:

$$bid_j(t) = \alpha_0 \cdot bid_j^{resource}(t) + \beta_0 \cdot bid_j^{time}(t), \alpha_0, \beta_0 \in [0, 1] \quad (4-9)$$

其中  $\alpha_0$ 、 $\beta_0$  分别表示剩余资源数与平均剩余时间在最终竞价值中的相应比重, 且  $\alpha_0 + \beta_0 = 1$ 。 $\alpha_0 = 1$  表示在最终竞价值中只考虑剩余资源数的约束, 同样,  $\beta_0 = 1$  时, 最终竞价值只关注平均剩余时间的约束。 $\alpha_0, \beta_0 \in (0, 1)$  时, 两者约束同时考虑。

由于云用户有着各自不同的需求, 本文考虑极少数用户对截止时间特别重视, 这类用户的云任务紧迫程度特别高。当云用户提交了紧迫任务后, 为了在截至时间内尽快执

行完成,用户根据自身能力选择预定适合的资源提供者并协商付费价格,交易价格为(设第  $j$  个云任务预定的是第  $i$  个资源,  $k$  为协商后的比例常数):

$$bid_j^{reservation} = k \cdot mp_i, k \in (1, 2) \quad (4-10)$$

#### 4.2.2 资源代理策略

云计算环境中资源  $r_i$  初始工作量  $st_i$  设为 0, 此时资源为保留价  $rp_i$ 。资源接受新任务后, 更新  $st_i$ , 并设定价格为最高价  $mp_i$ 。随着资源负载(新任务的开始时间)不断减少至 0, 资源价格相应降至保留价。资源提供者向拍卖商提出要价请求:

$$ask_i(t) = rp_i + (mp_i - rp_i) \cdot \frac{st_i^t}{wl_i^t} \quad (4-11)$$

式中:  $ask_i(t)$ —第  $i$  个资源在时间  $t$  提出的要价;  $st_i^t$ —资源接收下一个新任务的开始时间;  $wl_i^t$ —资源在上一次的分配后负载。

#### 4.2.3 拍卖商策略

在每个时间单位内, 云用户代理与资源提供者代理分别向拍卖商提供各自的竞价值和要价请求值。拍卖商整理成竞价值列表与要价值列表, 并将用户竞价值按照升序排列, 资源要价请求值以降序排列。如果最高的用户竞价值不低于最低资源要价值, 交易成功。拍卖商决定最终交易价格<sup>[43]</sup>:

$$trade\_price = kbid + (1 - k)ask, k \in [0, 1] \quad (4-12)$$

文中取  $k=1/2$ , 则交易价格:

$$trade\_price = \frac{1}{2}(highest\_bid + lowest\_ask) \quad (4-13)$$

### 4.3 SLA

SLA (Service Level Agreement) 是正式定义服务层次的服务合约, 记录着对服务一般理解诸如优先权、责任和保证等。云计算利用服务等级的概念控制对计算资源的使用, 其中 SLA 管理包括: (1) 含有 QoS 参数的概要; (2) SLA 协商; (3) SLA 监控; (4) SLA 执行<sup>[44]</sup>。云计算市场里, 服务提供者与用户代理参与市场买卖服务或资源, 云用户发现了满足其需求的资源时, 双方建立协商。当两方达成共识时, 签署服务等级协议, 用户付费使用相应资源<sup>[45]</sup>。云用户与云服务提供者之间签署 SLA, 指定云用户实现不同 QoS 方面的服务水平<sup>[46]</sup>。SLA 中还包含定义可对服务进行客观衡量条件的服务等级目标 (SLO, Service Level Objectives), 描述了资源提供者需要完成的用户 QoS 条款。

云用户提交具有 QoS 约束请求, 如截止时间、预算等, 当资源提供者接受请求后, 双方正式签订协议 (SLA) 以保障 QoS 请求的实现。本文仅考虑 SLA 为截止时间 (Deadline) 与预算 (Budget)。

#### 4.4 算法伪代码

本文设计的基于连续双向拍卖的云计算资源分配算法（S-CDA）伪代码：

Input: cloud consumer set  $U$ , resource set  $R$

Output: allocation strategy of task to resource

Initialize  $U = U_1 + U_2 = \{u_1, u_2, \dots, u_m\}$ ,  $R = \{r_1, r_2, \dots, r_n\}$

For  $j=1; j \leq m; j++$

Submit  $t_j = (id_j, l_j, b_j, d_j)$  to auctioneer

End for

For  $i=1; i \leq n; i++$

Update  $s_i = (id_i, cpu_i, st_i, rp_i, mp_i)$  in real time on auctioneer

End for

If  $(t_j \in U_1)$  then // a set of tasks with very urgent deadline

Select and reserve suitable resource, establish SLA mechanism and the trade price is  $k \cdot mp_i$

End if

If  $(t_j \in U_2)$  then

For  $j$  such that  $1 \leq j \leq m$  do

Calculate  $bid_j(t)$  by (4-9)

End for

For  $i$  such that  $1 \leq i \leq n$  do

Calculate  $ask_i(t)$  by (4-11)

End for

Sort the list of bid values in increasing order

Sort the list of ask values in decreasing order

While lowest  $(ask_i(t)) \leq$  highest  $(bid_j(t))$  do

Calculate  $trade\_price$  by (4-13)

End while

Establish SLA mechanism and finish the trade

End if

Update  $U, R$

在一个拍卖周期内，输入云用户集合  $U$  与资源集合  $R$ ，输出结果时云任务-资源分配方案。考虑极少数云用户对截止时间特别重视，其任务紧迫程度特别高。这类云用户一般愿意付出更多的预算去预定合适的资源，尽快执行完成任务。用户依据自身需求选择预定适合的资源，与资源提供者协商付费价格，双方达成一致时签订 SLA 协议。其他云用户与资源代理分别向拍卖商提出相应的竞价与要价值，拍卖商对竞价与要价进行

排序，匹配用户与资源提供者，达成 SLA 协议，完成交易。

## 第五章 仿真实验与结果分析

CloudSim<sup>[47,48]</sup>是由澳大利亚墨尔本大学 Rajkumar Buyya 教授领导的云计算与分布式系统实验室（CLOUDS, Cloud Computing and Distributed Systems）开发的云计算仿真工具，目标是基于云基础设施（包括硬件、软件与服务）上对不同应用和服务模型的调度以及分配策略性能进行量化比较，有效控制云计算资源的使用。用户使用 CloudSim 可以不必关注云基础设施和服务相关的具体细节，反复测试实验他们的特定需求，能够实现部署实际服务之前调控性能瓶颈，整个过程无需耗费大量资金，这给用户的开发工作带来了极大方便。

### 5.1 CloudSim 介绍

云计算提供安全可靠、容错、可持续、易扩展的计算服务，这些服务在私有云、公共云或混合云中应用。云基础设施上应用与服务模型负载、能量耗费以及系统规模动态变化，对应用及服务模型中调度与分配策略性能研究充满挑战性。墨尔本大学 CLOUDS 实验室简化这个过程，提出云计算仿真工具 CloudSim。CloudSim 是一通用、可扩展的仿真框架，支持无缝建模、模拟，能够进行新兴云计算管理服务与基础设施实验。具有以下特性<sup>[49]</sup>：

- （1）支持在物理计算节点建模和大规模云计算数据中心实例化。
- （2）可定制协议的提供主机资源到虚拟机的建模仿真。
- （3）能量意识的计算资源仿真。
- （4）云联合的仿真。
- （5）动态插入仿真元素，暂停、重启仿真。
- （6）提供数据中心、服务代理、调度与分配策略进行建模的平台。
- （7）提供虚拟引擎，创建在空间与时间共享灵活转换的虚拟服务。

#### 5.1.1 CloudSim 体系结构

CloudSim 分层体系结构如图 5-1 所示。

本文使用的 CloudSim 版本是 CloudSim 2.1.1（2011 年 2 月发布）。云计算作为新兴分布式计算模型，现阶段还缺乏相应的标准，工具和方法来有效处理基础设施及应用层复杂性。相信将来学术与企业界都会依据执行环境对核心算法、策略与应用管理予以研究，而扩展 CloudSim 的基本功能，研究人员就可以基于特定环境配置测试实验，实现对云计算的关键放方面开发实践。

##### （1）用户代码层

提供主机、应用、用户数、虚拟机、调度策略等基本实体。扩展相应实体，能够实



现:

- 应用配置与工作负载分配请求。
- 云可应用场景及稳健性测试。
- 云和云联合的调度技术。

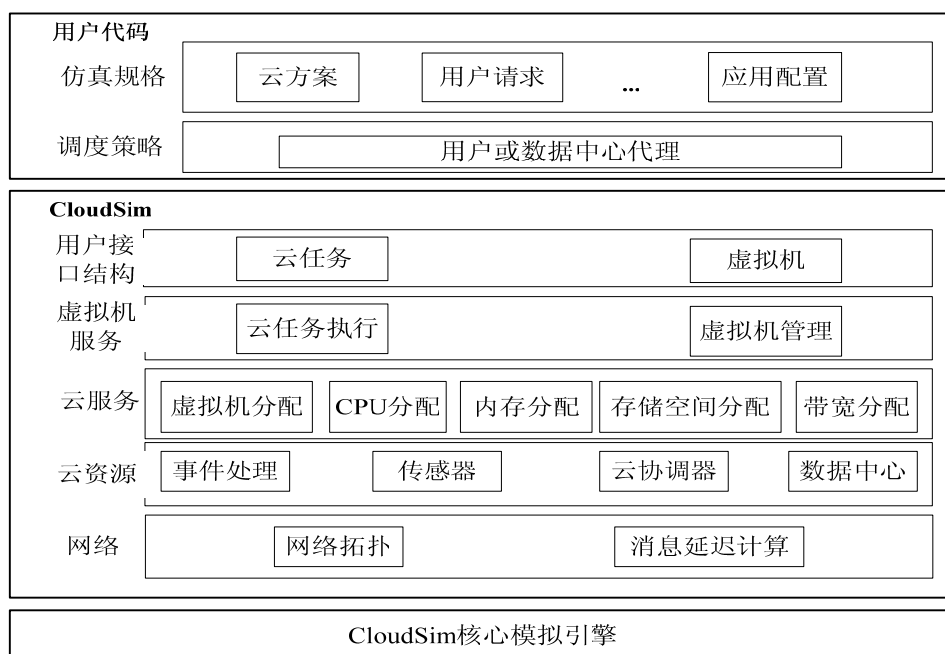


图 5-1 CloudSim 分层体系结构

Fig.5-1 Layered CloudSim Architecture

## (2) CloudSim 层

支持对云计算数据中心的仿真，负责诸如虚拟机调度、应用程序执行、系统动态监控等基本问题。具体可以分为以下五层：

**网络层：**连接云计算仿真实体需要考虑全面网络拓扑。同时消息的延时影响到用户对云提供商服务的满意度。因而云系统仿真框架有必要模拟真实的网络拓扑模型，CloudSim 的网络抽象概念通过延时矩阵信息模拟实体间的网络延时。

**云资源层：**一系列主机组成的数据中心实体模拟与云相关基础设施，每个主机代表一个物理计算节点，管理虚拟机的创建、销毁等操作。云协调器负责与终端用户和其他数据中心通信，查询传感器监控、管理数据中心实体内部状态。

**云服务层：**云计算基础设施不同网格的一个关键方面在于大量虚拟化工具的部署。虚拟机分配协议把特定应用虚拟机按照硬件需求，如处理核数、内存及存储器等分配给主机。

**虚拟机服务层：**虚拟机生命周期的管理。

**用户接口结构层：**虚拟机与任务的接口。

## (3) CloudSim 核心模拟引擎

早期的 CloudSim 由 SimJava 库实现事件处理与实体间消息传递，而在可伸缩仿真环境时暴露出不足：不支持重置仿真、不支持运行时新实体创建、系统性能严重下降以及系统调试更加复杂。CLOUDS 实验室开发新的离散事件管理框架，如图 5-2 所示。

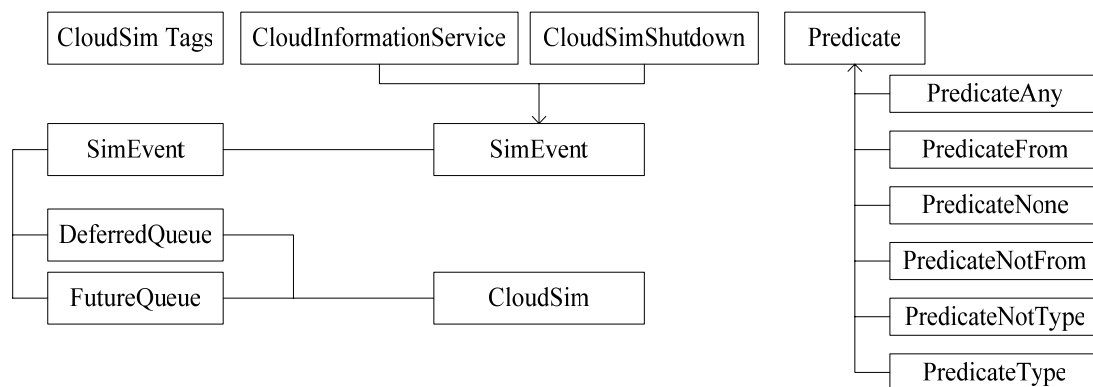


图 5-2 CloudSim 核心模拟引擎类图

Fig.5-2 CloudSim Core Simulation Framework Class Diagram

**CloudSim**: 主类，管理事件队列和控制仿真事件队列顺序执行。CloudSim 实体在运行时产生事件并存储在将来事件队列，调度过程中仿真事件会从未来事件队列删除，转移到延时事件队列。每个实体从延时事件队列中选择事件执行相应操作。

**DeferredQueue**: 延时事件队列。

**FutureQueue**: 未来事件队列。

**CloudInformationService (CIS)**: 提供资源注册、索引和发现的实体，支持两基本操作：**publish()**, 允许实体注册；**search()**, 允许类似于 Brokers 和 CloudCoordinator 实体发现其他实体的位置和状态。实体同时通知其他实体仿真结束。

**SimEntity**: 抽象类，代表仿真实体，不仅处理接受的消息，还能向其他实体发送消息。该类三个核心方法：**startEntity()**，实体初始化；**processEvent()**，事件处理；**shutdownEvent()**，实体销毁。实体建立时需要在 CIS 自动注册。

**CloudSimTags**: 包含静态的命令或时间标签，CloudSim 实体根据标签采取对应操作。

**SimEvent**: 两个或多个实体间传递的仿真事件实体，存储事件的信息，如类型、初始化时间、结束时间、资源标识、目标实体等。

**CloudSimShutdown**: 等待所有代理实体与终端用户，通知 CIS 仿真结束。

**Predicate**: 抽象且需扩展的类。

### 5.1.2 CloudSim 技术实现

CloudSim 的类设计图如图 5-3 所示，这些类是构成仿真工具的基础，主要类的功能描述如下。

**BwProvisioner**: 模拟虚拟机带宽分配的抽象类，扩展此类实现特定应用需求（服务质量，优先级），BwProvisionerSimple 允许虚拟机在主机总带宽限制下尽量保留带宽。

**CloudCoordinator:** 周期性监控云数据中心资源运行状态并实施动态负载均衡。

**Cloudlet:** 基于云应用服务的类，通常也称为云任务，预先设定指令长度和应用执行的数据传输量。

**CloudletScheduler:** 解决虚拟机共享问题，支持时间共享与空间共享策略。

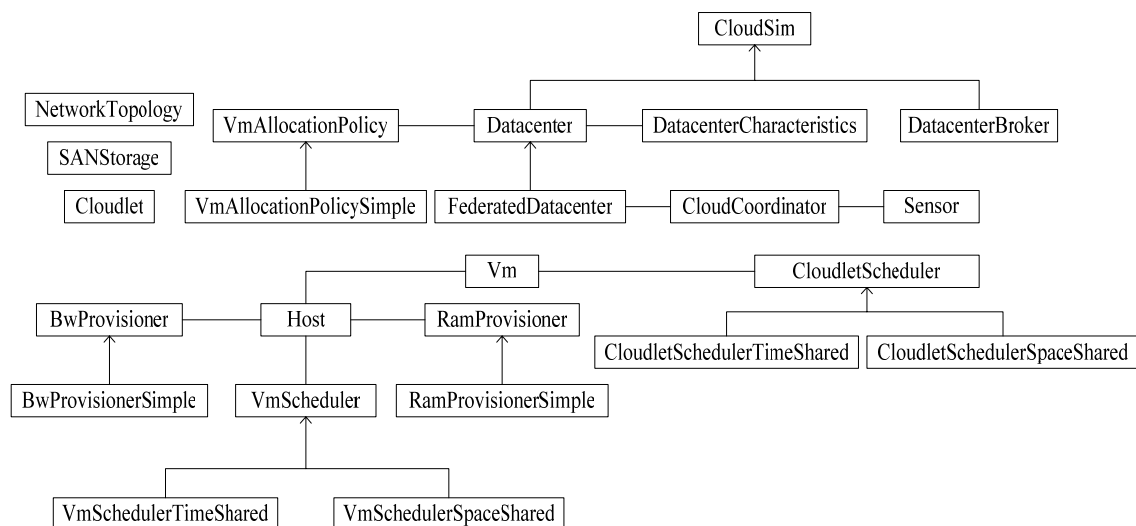


图 5-3 CloudSim 类设计图

Fig.5-3 CloudSim Class Design Diagram

**Datacenter:** 模拟核心硬件基础设施服务。封装支持异构与同构资源配置的主机（内核、内存、存储和容量）。数据中心实例化的通用调度组件实现为虚拟机和主机分配内存、带宽与存储设备的策略。

**DatacenterBroker:** 数据中心代理，查询云信息服务（Cloud Information Service）获取合适云服务提供者，依据服务质量进行资源与服务的分配。

**DatacenterCharacteristics:** 数据中心配置信息。

**Host:** 模拟计算或存储服务器的物理资源，封装了内存与存储总量，处理内核列表与类型以及分配内存与带宽协议等相关信息。

**NetworkTopology:** 由 BRUTE 拓扑生成器生成网络拓扑信息模拟网络延时行为。

**RamProvisioner:** 模拟虚拟机主存储器分配，虚拟机的执行部署操作需要 RamProvisioner 证实主机存在空闲主存。

**SanStorage:** 存储区域网，模拟受网络带宽可用性限制地存储、获取数据。

**Sensor:** 监控特定性能参数（资源利用率、能量消耗），接口定义设置性能参数的最小、最大值，模拟主流云提供商真实服务，周期性更新测量。

**Vm:** 模拟主机管理托管的虚拟机。

**VmAllocationPolicy:** 该抽象类实现在数据中心选择合适的可用主机并提供给需部署的虚拟机。

**VmScheduler:** 虚拟机调度策略。

云任务处理更新过程如图 5-4 所示。处理任务单元由各自的虚拟机负责完成，因而在仿真的每个步骤需要持续更新并监控虚拟机进展情况。数据中心实体为其管理的主机调用 `updateVMsProcessing()` 方法，相应虚拟机更新处理过程。此方法输入当前仿真时间，返回虚拟机上运行的下一个任务期望完成时间。下一个内部事件时间是返回的最小完成时间，反馈给数据中心实体。

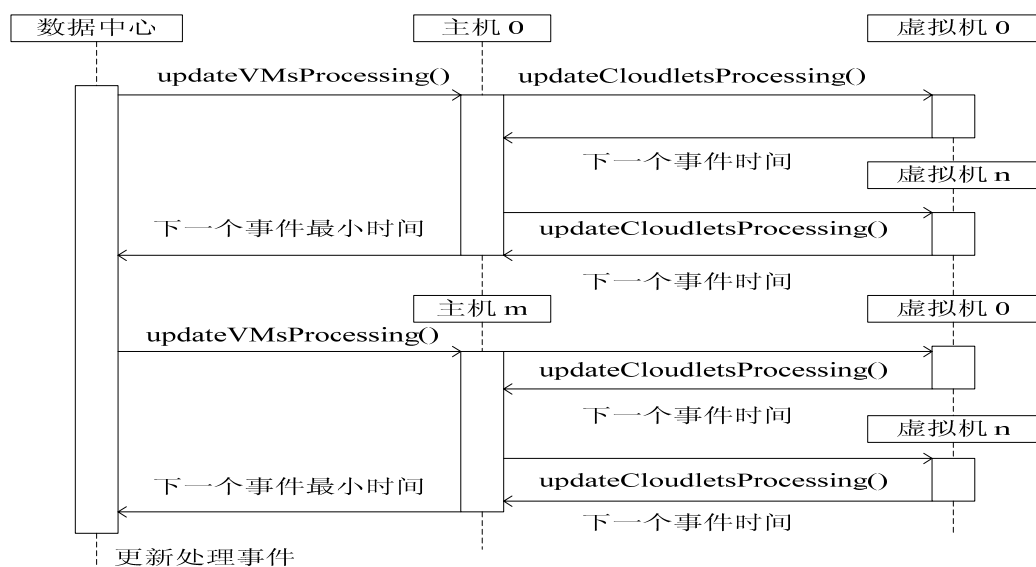


图 5-4 云任务处理更新过程

Fig.5-4 Cloudlet Processing Update Process

图 5-5 显示了仿真数据流，反应 Datacenter、CIS、DatacenterBroker 信息交互过程。仿真初期，数据中心实体通过 CIS 注册，用户发出请求时，CIS 依据特定应用需要，选择适合的云服务提供商。

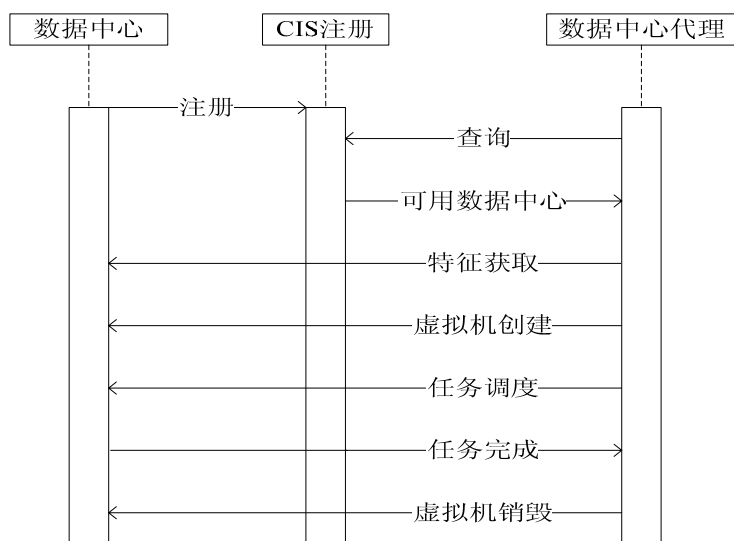


图 5-5 仿真数据流

Fig.5-5 Simulation Data Flow

## 5.2 CloudSim 使用方法

### 5.2.1 环境配置

#### (1) JDK

下载安装最新版 JDK<sup>[50]</sup>, CloudSim 的软件需求是 jdk1.6 以上版本。以 jdk1.6.0\_17 为例, 安装目录: C:\Program Files\jdk1.6.0\_17。环境变量设置: 新建变量 JAVA\_HOME, 变量值设置为 C:\Program Files\jdk1.6.0\_17; Path 中添加 %JAVA\_HOME%\bin; ClassPath 中添加 %JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar。

#### (2) CloudSim

下最新 CloudSim2.1.1 (2011 年 2 月发布) 解压其至磁盘, 以 CloudSim2.1.1 为例, 环境变量设置: ClassPath 中添加 C:\cloudsim-2.1.1\jars\ cloudsim-2.1.1.jar; C:\cloudsim-2.1.1\jars\ cloudsim-examples-2.1.1.jar。

### 5.2.2 程序运行

为了便于查看修改程序代码, 一般选择在 Eclipse 下执行, 具体操作步骤如下。

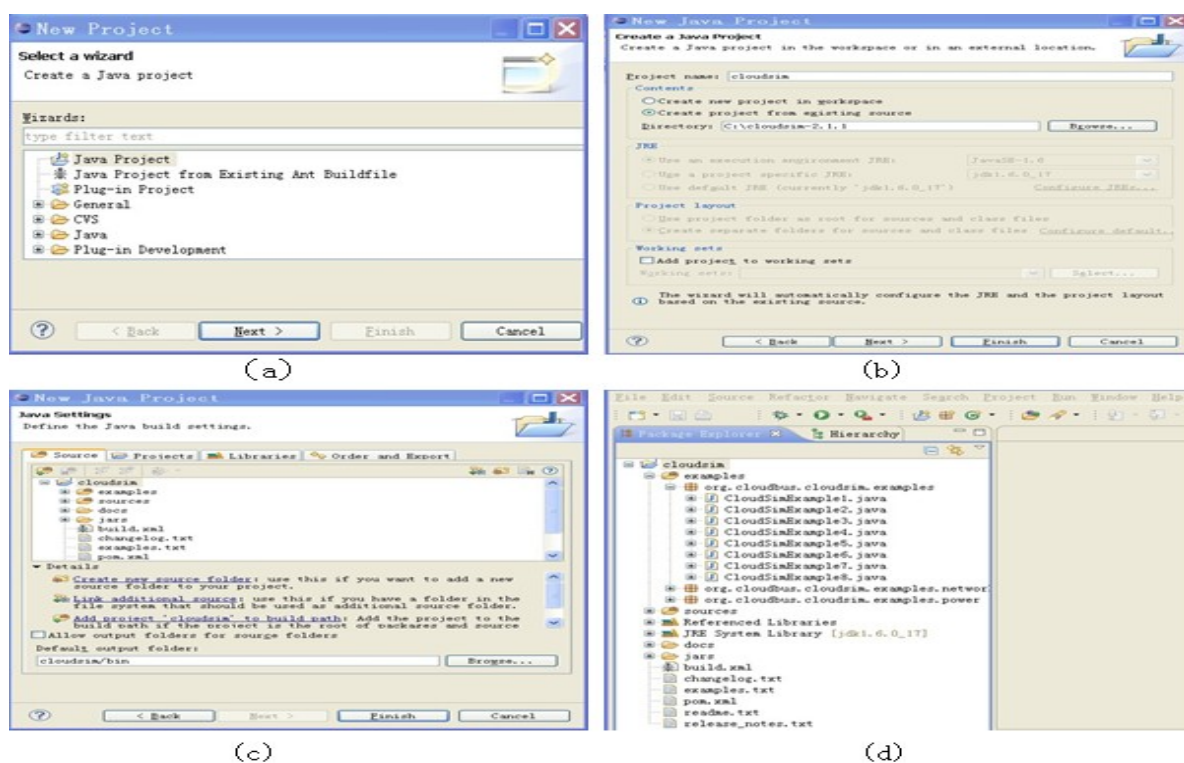


图 5-6 Eclipse 环境下 CloudSim 的运行  
Fig.5-6 Running the CloudSim in the Eclipse

(1) 启动 Eclipse, 执行 File→New→Project 命令, 打开 New Project 窗口, 新建工程, 如图 5-6 (a) 所示。

(2) 选择 Java Project, 单击 Next, 创建一个 Java 工程, Java 工程名为 cloudsim,

浏览选定源代码所在目录 C:\cloudsim-2.1.1, 如图 5-6 (b) 所示。

(3) 单击 Next, 显示 Java 工程配置界面, 含有源代码, 工程与库等信息, 其中源代码则有 examples、sources、docs、jars 等。如图 5-6 (c) 所示。

(4) 单击 Finish 完成 Java 工程创建, 如图 5-6 (d) 所示, 选择实例运行获得运行结果。

### 5.2.3 仿真基本步骤

用户使用 CloudSim 进行云计算仿真, 依据各自特定需求对主机参数、虚拟机、用户、云计算应用、应用类型数量以及代理调度策略进行测试, 支持云计算资源管理与调度模拟。

CloudSim 仿真一般包括以下基本步骤:

(1) CloudSim 包初始化

CloudSim.init(num\_user, calendar, trace\_flag) ;//CloudSim 参数初始化

(2) 数据中心创建

Datacenter datacenter0 = createDatacenter("Datacenter\_0");//数据中心是在 VM 生命周期管理 VM 的一组主机。

数据中心创建分以下几个步骤:

(a) 创建机器列表对象 List<Host> hostList = new ArrayList<Host>();

(b) 创建 PE 列表 List<Pe> peList = new ArrayList<Pe>();

(c) PE 添加到列表里 peList.add(new Pe(0, new PeProvisionerSimple(mips)));

(d) 创建主机, 并将其添加到机器列表

```
hostList.add(
    new Host(
        hostId,
        new RamProvisionerSimple(ram),
        new BwProvisionerSimple(bw),
        storage,
        peList,
        new VmSchedulerTimeShared(peList) ;//虚拟机时间共享
```

(e) 创建数据中心特征对象

DatacenterCharacteristics characteristics = new DatacenterCharacteristics(arch, os, vmm, hostList, time\_zone, cost, costPerMem, costPerStorage, costPerBw) ;//数据中心属性

(f) 创建数据中心对象

```
datacenter = new Datacenter(name, characteristics, new
VmAllocationPolicySimple(hostList), storageList, 0) ;
```

(3) 数据中心代理

```
DatacenterBroker broker = createBroker()
int brokerId = broker.getId();//负责根据用户 QoS 协调用户与服务供应商并部署任务
```

(4) 虚拟机创建

```

private static List<Vm> createVM(int userId, int vms, int idShift) {
    //创建虚拟机容器，相应代理提交虚拟机列表
    LinkedList<Vm> list = new LinkedList<Vm>();
    //虚拟机参数，按照特定需求设置参数
    long size = 10000;
    int ram = 512;
    int mips = 250;
    long bw = 1000;
    int pesNumber = 1;
    String vmm = "Xen";
    //创建虚拟机
    Vm[] vm = new Vm[vms];
    for(int i=0;i<vms;i++){
        vm[i] = new Vm(idShift + i, userId, mips, pesNumber, ram, bw, size, vmm, new
CloudletSchedulerTimeShared());
        list.add(vm[i]);
    }
    return list;
}

```

#### (5) 云任务创建

```

private static List<Cloudlet> createCloudlet(int userId, int cloudlets, int idShift){
    // 创建云任务容器
    LinkedList<Cloudlet> list = new LinkedList<Cloudlet>();
    //云任务参数，依据实际需要设置参数
    long length = 40000;
    long fileSize = 300;
    long outputSize = 300;
    int pesNumber = 1;
    UtilizationModel utilizationModel = new UtilizationModelFull();
    Cloudlet[] cloudlet = new Cloudlet[cloudlets];
    for(int i=0;i<cloudlets;i++){
        cloudlet[i] = new Cloudlet(idShift + i, length, pesNumber, fileSize, outputSize,
utilizationModel, utilizationModel, utilizationModel);
        cloudlet[i].setUserId(userId);
        list.add(cloudlet[i]);
    }
    return list;
}

```

#### (6) 仿真开启

```
CloudSim.startSimulation();
```

#### (7) 结果统计

```

List<Cloudlet> newList = broker.getCloudletReceivedList();
printCloudletList(newList);

```

### 5.2.4 CloudSim 扩展

仿真将对CloudSim进行扩展，Users和Datacenters模拟拍卖机制中的买方（提交云任

务)与卖方(资源提供者), CIS (CloudInformationService) 模拟拍卖商角色。

扩展Cloudlet类, 增加预算(budget)与截止时间(deadline)两个成员变量, 调用 Random random = new Random()实现云任务长度在一定幅度内的随机变化。

CloudSim允许用户根据自身需求调用适当的API, 如cloudlet.setVmId(vmId)或者 broker.bindCloudletToVm(cloudlet.getCloudletId(),vm.getId())实现将任务指定给特定虚拟机运行, 扩展 DatacenterBroker 类中 DatacenterBroker::processResourceCharacteristics()、DatacenterBroker::createVmsInDatacenter()和 DatacenterBroker::submitCloudlets()等实现自定义的策略算法。

### 5.2.5 平台重编译

用户实现自定义算法, 扩展 CloudSim 后可利用 Ant<sup>[51]</sup>工具实现 CloudSim 平台的重编译, 测试发布新平台。下载 Ant, 设置环境变量: Path 中添加: C:\apache-ant-1.8.0\bin。命令行切换到 build.xml 所在目录, 输入: C:\cloudsim-2.1.1>ant, CloudSim 源文件批量编译, 编译成功生成 cloudsim-new.jar,至此扩展的 CloudSim 平台生成, 如图 5-7 所示。ClassPath 中添加 C:\cloudsim-2.1.1\jars\cloudsim-new.jar, 新平台下即可编写扩展的仿真程序。

```
C:\Documents and Settings\Administrator>cd C:\cloudsim-2.1.1
C:\cloudsim-2.1.1>ant
Buildfile: C:\cloudsim-2.1.1\build.xml

prepare:
  [mkdir] Created dir: C:\cloudsim-2.1.1\classes

build:
  [javac] C:\cloudsim-2.1.1\build.xml:54: warning: 'includeantruntime' was not set, defaulting to
  build.sysclasspath=last; set to false for repeatable builds
  [javac] Compiling 89 source files to C:\cloudsim-2.1.1\classes

makejar:
  [echo] Compiling a new jar file, named: "cloudsim-new.jar".
  [echo] This jar file contains CloudSim classes only.
  [jar] Building jar: C:\cloudsim-2.1.1\jars\cloudsim-new.jar

BUILD SUCCESSFUL
Total time: 6 seconds
C:\cloudsim-2.1.1>
```

图 5-7 CloudSim 重编译  
Fig.5-7 Recompile for CloudSim

## 5.3 实验仿真与结果分析

仿真实验通过扩展 CloudSim 实现, 创建 8 个数据中心, 每个数据中心的机器数是 5, 每个机器的 PE 数量在区间[1,4]随机取值, 数据中心特征: 系统架构为 x86, 操作系统选为 Linux, 虚拟机是 Xen。

考察 500 个云任务, 每个云任务以泊松分布提交, 云任务长度是[1000, 30000] MI 随机产生的一整数。选取所有任务数的 5%作为模拟紧迫程度特别高的云任务, 其他云任务的预算按照  $b_j = \text{random}(3,9) \times l_j$  设置, 截止时间由



$d_j = \text{random}(\frac{l_j}{1.1 \max(pe)}, \frac{l_j}{0.9 \max(pe)}) + st_i$  决定，其中  $pe$  是资源的处理能力（MIPS）。

对于所有的资源提供者假设其资源保留价格为 4\$/MIPS，最高价格 8\$/MIPS。实验取  $\alpha = \beta = 0.2$ ， $\alpha_0 = \beta_0 = 0.5$ 。

为了比较本文提出算法（S-CDA）的有效性，选取先来先服务算法（FCFS，First Come First Serve），最短作业优先算法（SJF，Shortest Job First），最小完成时间算法（CloudSim 调度策略之一）做为比较。在 FCFS 算法中，任务按照到达时间排序，SJF 算法中，任务根据其长度排序。任务将分配给能够在其截止时间内完成的资源执行，当没有资源满足任务截止时间时，移除任务；对于满足任务的多个可用资源时，将随机选中一个资源执行。

仿真实验将从云任务执行成功率、SLA 违约率，公平性偏差和平均交易价格等方面进行考察。

#### （1）云任务执行成功率

云任务执行成功意指云任务能够在截至时内被完成，云用户为使用资源而付费，关心其任务是否能在截至时间内完成，云任务执行成功率由式（5-1）决定，其中  $T_j$  表示云任务  $t_j$  的执行完成时间：

$$K = \frac{\sum_{j=1}^m \theta_j}{m}, \theta_j = \begin{cases} 1, & \text{if } T_j \leq d_j \\ 0, & \text{otherwise} \end{cases} \quad (5-1)$$

云任务的执行成功率如图 5-8 所示，图中反应随着云任务数的增加本文提出的 S-CDA 策略算法有着更优的任务执行成功率，原因是本文的 S-CDA 很好反应资源供需与价格关系，云任务在拥有较多剩余时间和可用资源的情况下，通常以较低的价格竞标资源，时间的不断增加，剩余时间和可用资源逐渐减少，云任务相应增加其竞价值，所以对于具有更短剩余时间与更少可用资源的云任务将会以较高的竞价值获取资源并在截至时间内执行完成任务。

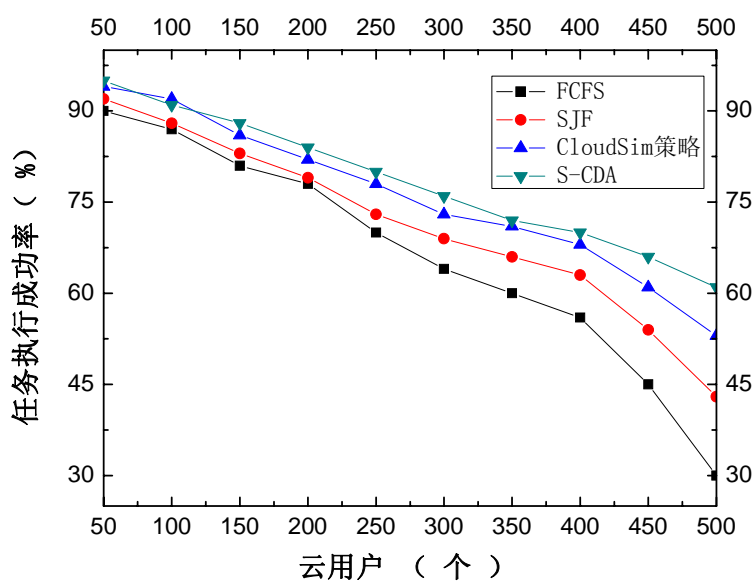


图 5-8 任务执行成功率

Fig.5-8 Successful Execution Rate

SLA 定义服务层次的服务合约，用来保障云用户提交 QoS 约束请求的实现，本文针对 S-CDA 定义 SLA 违约率来表示未能实现云用户的 QoS 请求的比率，如图 5-9 所示。随着云任务数的增加，SLA 违约率呈增加趋势。在以后进一步研究中，针对 SLA 违约很有必要引入对资源提供者的处罚比率来补偿云用户的损失以符合实际需求。

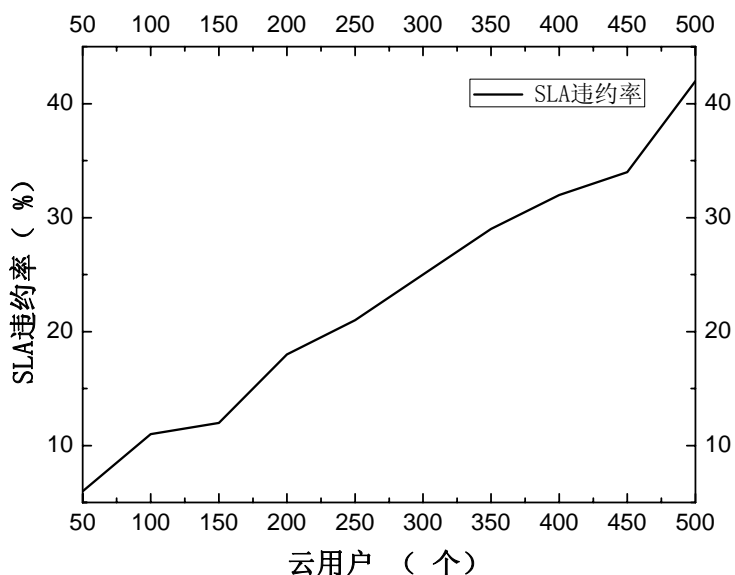


图 5-9 SLA 违约率

Fig.5-9 The Violation Rate of SLA

市场的公平性是指每个资源提供者拥有平等的机会出租其资源并根据自身性能获

得公平收益，资源公平性偏差越小，代表资源分配算法的越公平，资源公平性偏差<sup>[52]</sup>(std dev, Standard Deviation)由式 (5-2) 得到，其中  $P_i$  为资源  $r_i$  的收益。仿真实验使用系统负载概念表示提交云计算系统的云任务总长度与云计算系统在仿真周期  $T$  内能够处理的云任务总长度，系统负载由 (5-3) 得到。公平性偏差比较如图 5-10 所示。

$$\sigma = std\_dev(\Delta_1, \dots, \Delta_n), \Delta_i = \frac{P_i / \sum_{1 \leq u \leq n} P_u}{cpu_i / \sum_{1 \leq v \leq n} cpu_v} \quad (5-2)$$

$$\omega = \frac{\sum_{j=1}^m l_j}{T \times \sum_{i=1}^n cpu_i} \quad (5-3)$$

图 5-11 反应了系统负载与平均交易价格的关系，随着系统负载的增加，平均交易价格也在逐渐增加。本文提出的 S-CDA 能够很好反应资源供需与价格关系，云任务数的增加，即系统负载增加，云用户会以较高的竞价来竞争获得相应资源，同时实验设置的紧迫程度特别高的云任务也进一步提升了交易的平均价格。

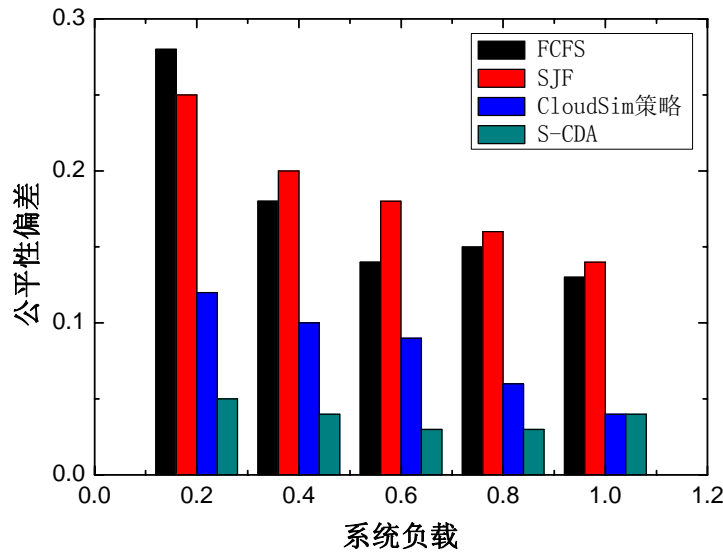


图 5-10 公平性偏差

Fig.5-10 Comparison on the Fairness Deviation

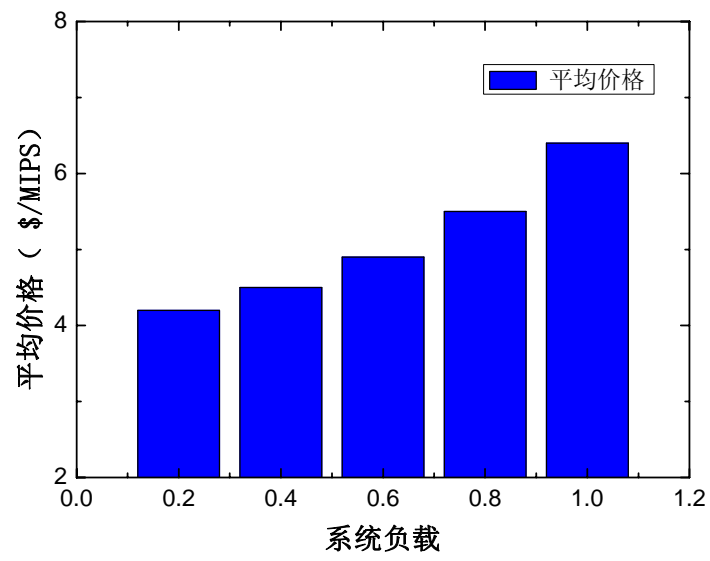


图 5-11 不同系统负载下的平均交易价格

Fig.5-11 Mean Price in Different System Loads

## 第六章 总结与展望

### 6.1 全文总结

云计算由多种技术混合演进，有着较高的成熟度，随着业界诸如 Google、Amazon、Vmware 和 IBM 等公司的全力推动，发展极其迅速。众多新技术的运用，规模经济的带动，很高的资源利用率与惊人的性价比都使得云计算成为一种全新的商业计算模型。

云计算分配计算任务在大量的计算机构成的资源上，采取按需索取，按量计费的模式，商业化的特性要求不断满足用户需求多样化，因而云计算的资源分配与调度显得尤为关键。云计算目标旨在优化使用资源，更好完成用户提交的任务请求，提供优质服务。

本文阐述云计算技术，对基于连续双向拍卖的云计算资源分配进行探讨研究，主要工作有：

(1) 对云计算技术进行阐述。总结归纳了几个典型的云计算定义与较普遍的特征、体系结构。分析比较云计算相关概念。对主流云计算应用中的关键技术与解决方案概述，概括云计算标准化现状。

(2) 云计算具有商业特性，实现资源的最优分配充满挑战，传统的资源管理方式不能简单地应用于云计算环境中。采取基于市场的资源分配方法充分考虑用户和资源提供者的双方利益，论文探讨了基于连续双向拍卖的云计算资源分配方法，实验结果显示算法有着较好性能。

(3) 介绍云计算仿真工具 CloudSim 的体系结构、技术实现与具体使用方法。扩展 CloudSim 平台，以 Users 和 Datacenters 模拟拍卖机制中的买方（提交云任务）与卖方（资源提供者），CIS（Cloud Information Service）模拟拍卖商角色，扩展重编译后对本文策略算法进行仿真实验与性能分析评价。

### 6.2 展望

云计算技术发展迅猛，得到学术界的高度重视。参考借鉴其他研究成果并结合自身的对相关技术的学习提出了本文的资源分配算法。实验结果表明该算法有着良好的性能，但未来还存在有待改进和进一步探讨研究之处：

(1) 为拍卖的双方引入纳什均衡概念，以期获得资源分配的整体最优状态。

(2) 本文仅考虑了基用户层面的 SLA，包括截止时间（Deadline）与预算（Budget）进一步的研究可以考虑资源提供者层面的 SLA。同时对于本文算法中云用户与云服务提供者之间签署的 SLA 在得不到满足的情况下，可以考虑引入处罚比率以对云用户进行补偿。

(3) 对云计算资源的信任度以及计算节点可信性进行进一步地研究。

## 参考文献

- [1] 第七框架计划[EB/OL]. [http://cordis.europa.eu/fp7/home\\_en.html](http://cordis.europa.eu/fp7/home_en.html), 2011-04-28
- [2] 位置云[EB/OL]. <http://baike.baidu.com/view/4180132.htm>, 2010-09-18
- [3] 刘鹏. 云计算（第二版）[M]. 北京：电子工业出版社，2011.
- [4] Foster,I.,Zhao,Y.,Raicu,Ioan,Lu,S., Cloud Computing and Grid Computing 360-Degree Compared [C]. IEEE Grid Computing Environments Workshop (GCE'08),2008:1-10.
- [5] Michael Armbrust et al., Above the Clouds:A Berkely view of Cloud Computing [R]. Technical Report No.UCB/EECS-2009-28,Feb.2009
- [6] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities[C]. 10th IEEE International Conference on High Performance Computing and Communications,2008:5-13.
- [7] NIST Definition of Cloud Computing [EB/OL].  
<http://csrc.nist.gov/groups/SNS/cloud-computing/>, 2011-4-30
- [8] 吴吉义, 平玲娣, 潘雪增, 李卓. 云计算：从概念到平台[J]. 电信科学, 2009: 25-26.
- [9] 雷万云. 云计算：企业信息化建设策略与实践[M]. 北京：清华大学出版社，2011.
- [10] 框计算[EB/OL]. <http://boxcomputing.baidu.com/>, 2011-5-2
- [11] 林闯. 《物联网关键理论与技术》专题[J]. 计算机学报, 2011:761-762.
- [12] Luis M.Vaquero et al., A Break in the Clouds:Towards a Cloud Definition [J]. ACM SIGCOMM,2009,Vol.39 (1): 50-55.
- [13] Rajkumar Buyya, James Broberg, Andrzej M. Goscinski. Cloud Computing: Principles and Paradigms[M]. John Wiley&Sons,Inc.,Hoboken,New Jersey,2011.
- [14] 赵春燕. 云环境下作业调度算法研究与实现[D]. [硕士学位论文]. 北京交通大学，2009.
- [15] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic, Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility [J].Future Generation Computer Systems, 2009,Vol.25(6): 599-616.
- [16] DAVID CHAPPELL,Introducing the Windows Azure Platform,Final PDC10 [R],2010
- [17] IBM Cloud Labs and HiPODS, 云计算策略及 IBM 蓝云方案[R], 2008
- [18] J.Dean,S.Ghemawat, MapReduce:Simplified data processing on large clusters[J].Communications of the ACM,2008,Vol.51(1):107-113.
- [19] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, The Google File System[C] . Proceedings of the nineteenth ACM symposium on Operating systems principles,2003:30-31.
- [20] 陈康, 郑纬民. 云计算：系统实例与研究现状[J]. 软件学报, 2009 , Vol. 20 (5) :1337-1348.
- [21] 移动 Labs. BigCloud[EB/OL]. <http://labs.chinamobile.com/focus/bigcloud>, 2011-5-10
- [22] 华为. 华为云计算[EB/OL]. <http://www.huawei.com/>, 2011-5-11
- [23] 赛迪顾问. 中国云计算产业发展白皮书[EB/OL]. <http://www.ccidnet.com/>, 2011-5-14
- [24] 杨炼. 网络资源三层调度模型与算法研究[D]: [硕士学位论文]. 中南大学，2008.
- [25] 刘宴兵, 尚明生, 肖云鹏. 网络高性能调度及资源管理技术[M]. 北京：科学出版社，2010.
- [26] 黄智维, 倪子伟. 网络计算环境下资源管理的研究[J] . 计算机技术与发展, 2009, Vol. 19(3) :200-203.

- [27] Rajkumar Buyya, Economic-based Distributed Resource Management and Scheduling for Grid Computing [EB/OL]. <http://arxiv.org/ftp/cs/papers/0204/0204048.pdf>
- [28] Weiming Shi, Bo Hong, Resource Allocation with a Budget Constraint for Computing Independent Tasks in the Cloud [C]. The 2nd International Conference on Cloud Computing, 2010: 327-334.
- [29] Shin-ichi Kuribayashi, Optimal Joint Multiple Resource Allocation Method for Cloud Computing Environments [J]. International Journal of Research and Reviews in Computer Science (IJRRCS), 2011, Vol. 2(1): 1-8.
- [30] Bo An et al., Automated Negotiation with Decommitment for Dynamic Resource Allocation in Cloud Computing [C]. Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010): 981-988.
- [31] Hesam Izakian, Ajith Abraham, Behrouz Tork Ladani, An auction method for resource allocation in computational grids [J]. Future Generation Computer Systems, 2010, Vol. 26 (2): 228-235.
- [32] 翁楚良, 陆鑫达. 一种基于双向拍卖机制的计算网格资源分配方法 [J]. 计算机学报, 2006, Vol. 29(6): 1004-1008.
- [33] Ikki Fujiwara, Kento Aida, Isao Ono, Applying Double-sided Combinational Auctions to Resource Allocation in Cloud Computing [C]. Proceedings of the 2010 10th IEEE/IPSJ International Symposium on Applications and the Internet: 7-14.
- [34] 华夏渝, 郑骏, 胡文心. 基于云计算环境的蚁群优化计算资源分配算法 [J]. 华东师范大学学报, 2010 (1): 127-134.
- [35] Wei-Yu Lin, Guan-Yu Lin, Hung-Yu Wei, Dynamic Auction Mechanism for Cloud Resource Allocation [C]. 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid): 591-592.
- [36] B. Pourebrahimi, K. Bertels, G. M. Kandru, S. Vassiliadis, Market-based Resource Allocation in Grids [C]. Proceedings of the Second IEEE International Conference on e-Science and Grid Computing, 1996, Vol. 122 (6): 80-88.
- [37] Umesh Kant, Daniel Grosu, Double Auction Protocols for Resource Allocation in Grids [C]. Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05): 366-371.
- [38] 陈冬娥, 杨扬, 刘丽. 基于反拍卖的计算网格资源分配方法 [J]. 北京科技大学学报, 2007, Vol. 29 (5): 549-552.
- [39] U. Kant, D. Grosu, Auction-Based Resource Allocation Protocols in Grids [C]. Proceedings of 16th International Conference on Parallel and Distributed, 2004: 20-27.
- [40] Chee Shin Yeo et al., Autonomic Metered Pricing for a Utility Computing Service [J]. Future Generation Computer Systems, 2010, Vol. 26 (8): 1368-1380.
- [41] Xu B et al. Job scheduling algorithm based on Berger model in cloud environment [J]. Adv Eng Softw (2011), doi: 10.1016/j.advengsoft.2011.03.007.
- [42] Anthony, P. and Jennings, N. R. Developing a bidding agent for multiple heterogeneous auctions [EB/OL]. <http://eprints.ecs.soton.ac.uk/8572>, 2011-06-04
- [43] Simon Parsons, Juan A. Rodriguez-Aguilar, Mark Klein, Auctions and Bidding: A Guide for Computer Scientists [J]. ACM Computing Surveys (CSUR), 2011, Vol. 43 (2): 1-59.
- [44] Service level agreement [EB/OL]. [http://en.wikipedia.org/wiki/Service\\_level\\_agreement](http://en.wikipedia.org/wiki/Service_level_agreement)
- [45] M. Macías, J. Guitart. Maximising revenue in Cloud Computing Markets by means of Economically

- Enhanced SLA Management [R] . Technical Report UPC-DAC-RR-CAP-2010-22.
- [46] Sara Bouchenak. Automated control for SLA-aware elastic clouds[C] . Proceedings of the Fifth International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks (FeBiD '10) , Paris, France, Apr. 2010.
- [47] Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros, Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities[C].Proceedings of the 7th High Performance Computing and Simulation Conference (HPCS 2009):1-11.
- [48] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya, CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms[J].Software: Practice and Experience (SPE),2011, Vol.41(1) : 23-50.
- [49] CLOUDS Laboratory, <http://www.cloudbus.org/cloudsim/>
- [50] Java SE. <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- [51] Apache Ant. <http://ant.apache.org/>
- [52] L.Xiao,Y.Zhu,L.M.Ni,and Z.Xu,Incentive-Based Scheduling for Market-Like Computational Grids [J]. IEEE Transactions on parallel,2008, Vol.19(7) : 903-911.



## 个人简历 在读期间发表的学术论文

### 个人简历:

刘鹏，男，1987 年 1 月生。

2009 年 7 月毕业于华东交通大学通信工程专业，获学士学位。

2009 年 9 月入华东交通大学读硕士研究生。

### 已发表论文:

- [1] Juefu Liu, Peng Liu, The research of load imbalance based on min-min in grid [C]. 2010 International Conference on Computer Design and Applications (ICCDa 2010) :1-4.
- [2] Juefu Liu, Peng Liu, Status and key techniques in cloud computing [C] . 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE 2010) :285-288.
- [3] Yan Yin, Peng Liu, Market-Oriented Resources Trading Mechanism for Cloud Computing [J]. Procedia Engineering (ISSN: 1877-7058, ELSEVIER),2011.

## 致谢

论文完成之际，谨向给予我指导、关心、支持和帮助的老师、同学和亲人致以深深的敬意和真挚的感谢！

首先，我要感谢我的恩师刘觉夫教授。刘老师有着严谨的学术态度、高度的责任感、开放的学术思想、广博的专业知识。在攻读硕士学位的二年半时间里刘老师不但关心我的生活和学习进展情况，更给予了许多及时并且必要的人生指导，他一直强调“先做人后做学问”让我深受启发，终身受用。作为导师，感谢您对我在科研上的严厉与指导；作为长辈，感谢您在为人处世上对我的谆谆教诲；作为朋友，感谢您知心好友般的促膝而谈。师恩如父，永刻于心，惟愿师生情谊一生延续，祝愿老师和家人健康快乐，万事如意！

其次，我要感谢柴亚辉老师、谢昕老师、李波老师和莫佳老师的建议！感谢信息学院刘建辉班主任，正是刘老师两年半来井井有条的管理，我们的学习和毕业论文才得以顺利进行。

我也要非常感谢澳大利亚墨尔本大学 Clouds 实验室的 Rajkumar Buyya 教授，博士生 Anton Beloglazov、Suraj Pandey 和 Rodrigo N. Calheiros，感谢你们给我在 CloudSim 方面建设性意见和帮助！

同时感谢我的师兄师姐对我在课题研究、专业学习和日常生活的帮助。感谢我们班全体同学，在我学习的各个阶段给予了我许多帮助，陪伴我愉快地度过这一段美好时光。特别感谢我的室友，在我意志消沉时给了我支持和鼓励，并提出了许多宝贵意见。

最后，感谢我的家人，他们对我默默的支持、关爱，为我创造了和谐的学习环境，为我解除了后顾之忧，使我能够全身心、毫无顾忌的投入到科研学习当中。再次对你们表示深深的感谢！