

网络计算系统的分类研究

徐志伟 廖华明 余海燕 查 礼

(中国科学院计算技术研究所 北京 100190)

摘 要 计算系统的分类是计算机科学的一项基础工作,对计算系统的设计、编程和优化都有直接影响和指导作用.近二十年来,众多的网络计算系统得到了广泛使用,但学术界尚未研究出一套有效的分类方法,影响了网络计算系统的研究和知识共享.文中借鉴并行计算系统分类的成功经验,从应用编程的角度出发,给出了一套基于(执行、控制、层次)三维坐标的分类方法,能够覆盖代表性网络计算系统,如网格、云计算、互联网服务等系统.

关键词 网络计算;分类;网格;万维网服务;云计算

中图法分类号 TP301

Notes on Classifying Network Computing Systems

XU Zhi-Wei LIAO Hua-Ming YU Hai-Yan ZHA Li

(*Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190*)

Abstract Classifying computing systems is a basic component of computer science, directly impacting the design, programming and optimization of computer systems artifacts. The past 20 years have seen many network computing applications. However, an effective classification of network computing systems has not emerged in the academia. This negatively impacted scientific research, knowledge accumulation and sharing in network computing. This paper presents a classification based on three dimensions of (execution, control, and level), and shows that the classification method is able to cover representative network computing systems. The proposed method borrowed from successful experiences in classifying parallel computing systems.

Keywords network computing; classification; grid; Web service; cloud computing

1 引 言

计算系统的分类(classification)是计算机科学的一项基本内容.从历史看来,一类计算系统的分类一旦形成,就会促进该类系统的程序设计、应用开发、系统设计、系统优化.同时,好的分类也有利于知识的积累、传播和重用.

并行计算系统的分类工作为我们提供了一个很

好的示例.早期的分类代表是 Flynn 的“指令流-数据流”分类(SISD、SIMD、MISD、MIMD)^[1].在 20 世纪 80~90 年代,人们又逐步将并行机中最广泛的一类,即 MIMD 系统,做了更加细致的分类(如文献[2]所给出的).直到今天,这些分类仍然覆盖了主流的并行计算系统,如并行向量机(PVP)、对称多处理机(SMP)、非均匀存储访问并行机(NUMA)等.并行计算系统分类工作的一个重要历史经验就是从程序设计者的角度划分系统.它的一个重要副产品就是

业界出现了几个广泛使用的、生命力比较长久的并行编程模式和软件,如线程模式(POSIX 线程与 Java 线程等)、共享内存模式(OpenMP 等)、消息传递模式(MPI 等)^[2].

与并行计算系统相比,网络计算系统的分类还处于非常初级的阶段.一方面,产业界和学术界提出并实现了各种各样的网络计算系统.下面仅列出了一部分已经流行的例子:

(1) 万维网(World Wide Web, WWW, 或简称 Web).

(2) 面向消费者的万维网应用系统(consumer Web).

(3) 面向企业的万维网应用系统(business Web).

(4) 服务计算系统,尤其是万维网服务(service computing, Web service).

(5) 网格(grid)系统.

(6) 云计算系统(cloud computing).

(7) 对等计算系统(Peer to Peer, P2P).

(8) 社会网络与社会计算(social networking, social computing).

(9) 软件服务(Software as a Service, SaaS).

(10) Web2.0 系统.

(11) 以数据为中心的超级计算系统(DISC).

另一方面,学术界对网络计算系统的科学分类工作还很少.文献[3]是最近的一篇.作者按拥有者(owner)个数和在系统之上运行的应用软件(application)个数这两个维度,将网络计算系统分为 4 类:

(1) 单拥有者单应用(Single Owner Single Applications, SOSA).

(2) 单拥有者多应用(Single Owner Multiple Application, SOMA).

(3) 多拥有者单应用(Multiple Owner Single Applications, MOSA).

(4) 多拥有者多应用(Multiple Owners Multiple Applications, MOMA).

后文将说明,这种分类是一个好的开端,但尚不能回答一些基本的问题.

本文在第 2 节说明网络计算系统的范围,并分析其分类的需求和难点;第 3 节借鉴并行计算系统的经验,提出了一种按照执行、控制、层次 3 个维度的分类方法;第 4 节将此方法用于界定一些典型的网络计算系统;第 5 节陈述结论.

2 网络计算系统及其分类问题

2.1 网络计算系统的范围

本节首先界定本文所讨论的网络计算系统的范围,即什么是网络计算系统,什么不是.

网络计算系统(network computing system)是指通过广域的因特网或万维网,利用网上的资源,为多个用户提供价值的系统.如图 1 所示,各地的用户通过自己的客户端设备,连到广域网,使用网络计算系统得到价值.图 1 中的 site 是数据中心,它们往往是网络计算系统的组成部分.客户端设备有时也构成了网络计算系统的组成部分.

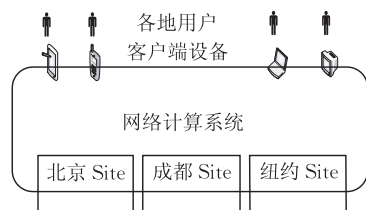


图 1 网络计算系统及相关环境

下面的系统则不是本文所考虑的网络计算系统,尽管它们有相关性:

(1) 单台微机、手持设备、嵌入式设备;

(2) 一台在局域网环境使用的高性能计算机;

(3) 存储设备或存储系统;

(4) 非数据服务的电信网,尽管它也是广域的;

(5) 传感器网络;

(6) 计算机网络 IP 层以下的网络系统本身,不论是有线、无线、固定、移动,这些网络只是网络计算系统的部件;

(7) 嵌入式网络和应用.

2.2 网络计算系统的分类问题

什么是网络计算系统的合理的、好的分类呢?人们能够使用这个分类做什么样的工作呢?如何评价分类的好坏和适用范围呢?本节提出 5 个判别条件,它们也是网络计算系统分类工作的研究目标.

条件 1. 划分. 一个好的分类应是所有网络计算系统集合的一个划分(partition),即互不相交的子集,这些子集之并集就是网络计算系统的全集.

条件 1 在自然科学领域分类工作中比较普遍.例如,“柳树”是一种植物,就不会是动物、微生物,也不存在 40% 植物 60% 动物的物种.但是,在计算机系统领域,这样完美的划分有时难以达到.比如,一套微机的主机系统也可以用作服务器;一台 MIMD

并行机也可含有 SIMD 加速部件. 认知科学领域将条件 1 界定的划分性分类称为经典分类法, 并指出很多事物难以这样分类^[4].

条件 2. 简单. 分类应该足够简单.

尽管划分性是一个理想的目标, 不一定强求, 但我们还是应该尽量使分类简单, 比如尽量使子类互不相交, 尽管我们不一定能排除实际系统是混合系统. 另外, 最初的几个大类应该严格控制它们的数量.

条件 3. 编程. 一个好的分类应该有利于网络计算系统应用的程序设计和开发使用.

一个类别的网络计算系统, 其应用软件和应用程序的开发应具有共同的特点, 人们可以利用这些特点发展出一套编程模式 (programming model), 甚至使得开发出的软件与服务可以在本类别的任一系统上运行. 比如, OpenMP 为 SMP 并行机提供了一套广泛使用的编程模式. 使用 C 和 OpenMP 开发出来的并行应用软件可以在很多 SMP 并行机上运行.

条件 4. 优化. 一个好的分类应该有利于网络计算系统本身的优化设计、实现与维护.

在网络计算系统的设计、实现与维护工作中, 同类系统有共同的难点, 一个系统的成功经验、技术、体系结构与部件可以用于同类系统中. 比如, NUMA 并行机系统必须考虑的一个难点是分布式存储中如何有效地应对缓存一致性.

条件 5. 学科. 一个好的分类应该有利于网络计算系统的学科建设, 特别是知识积累和教育.

并行机系统是一个好的例子. 经过 40 多年的发展, 作为服务器的并行机已经形成了比较成熟的学科以及年价值数百亿美元的产业. 并行机分类提供了一套比较规范、稳定的参照系, 为学科的积累和产业的发展产生了基础性的贡献.

但是, 在网络计算系统领域, 我们观察到相反的现象. 网络计算系统的应用和产业蓬勃发展, 但学科建设比较滞后. 涉及具体产品和工具的工作很多, 但网络计算科学的教材很少. 即使是本领域的从业人员和学生, 也往往被下述问题困扰:

(1) 为什么 Berners-Lee 要将 World Wide Web 刻画为一种分散信息系统 (decentralized information system) 或分散信息结构 (decentralized information structure)^[5]?

(2) Google 明明有数十万台服务器在全球分布式运行, 为什么要说它是一个集中式系统?

(3) 网络与云计算有什么区别?

(4) 网络计算中 REST 原理为什么重要?

(5) 并行计算 (parallel computing)、并发计算 (concurrent computing)、分布计算 (distributed computing) 有什么异同? 它们与新出现的服务计算 (services computing)、网格 (grid) 有什么关系?

(6) 为什么网络计算系统中部署、配置、维护日益重要?

(7) 网络计算系统有哪些关键问题和共性核心技术?

这些问题, 文献[3]中的 SOSA、SOMA、MO-SA、MOMA 分类法也不能解答.

2.3 分类的角度与主线

并行机分类的工作很多, 但 40 余年后沉淀下来的分类却只有少数几种. 这些分类成功有一条重要经验, 就是它们选择了程序设计的角度和主线, 即从应用开发者的使用模式入手.

后来对 MIMD 的细分也基本遵循了同样的方法学. MIMD 系统的应用开发者最关心数据变量的组织和使用模式, 对应到硬件就是地址空间. 因此, MIMD 系统的第一层划分 (UMA, NUMA, NORMA) 就采用了以地址空间使用模式为主线的思路.

对网络计算系统, 我们可以借鉴并行机的成功经验, 从应用开发者的使用模式出发划分系统. 这种应用开发工作需要考虑下列成分:

(1) 应用的需求规范.

(2) 应用的支撑环境.

(3) 应用代码和文档的设计、编写与集成.

(4) 应用的处理 (编译、链接、运行).

(5) 应用的部署、配置和维护.

在应用开发过程中, 需求规范起着重要的指导作用. 开发者需要使用到 4 类资源去实现应用需求, 即硬件 (服务器、存储、网络)、软件、数据和用户信息.

3 一种网络计算系统分类法

本节提出一种 (执行、控制、层次) 三维分类方法. 这 3 个维度与上述软件开发的 5 个成分都有关系. 但执行重点关心 (4), 控制重点关心 (3)、(5), 层次重点涉及 (2).

3.1 执行

为了保证程序的语义正确性和性能, 在编码过程中, 开发者必须理解网络计算系统的一些特点. 这些特点中最突出的就是单点还是多点执行. 所谓“点”

(地点、结点,英文称 site),我们主要指数据中心。

单点(single site)网络计算系统. 单点系统尽管仍然可能有很多广域网用户,但开发者需要使用的四种资源都在一个数据中心内. 程序的执行包括(1)通过广域网接收用户请求;(2)在一个数据中心内处理用户请求;(3)将请求的响应通过广域网传回用户的客户端设备.

多点(multiple site)网络计算系统. 在这种系统中,资源分布在多个结点(数据中心). 程序的执行一般也会涉及多个结点. 系统可能将多个结点暴露给用户,但往往提供单一系统映像(single system image). 一个特定的用户请求可能由一个结点处理,多个请求则根据某种调度方法分配到多个结点处理. 也可能一个请求的多个子任务被多个结点处理.

与单点系统相比,多点系统有更多的影响正确性和性能的困难需要考虑. 这些难点有些被网络计算系统屏蔽掉了,有些则仍然暴露给开发者,需要在开发程序的过程中解决. 比如:

(1)与结点内资源间通信相比,广域网连接的多点之间的通信性能往往差得多.

(2)广域网的带宽和延迟不稳定,变化大.

(3)数据中心内的费用成本模式与广域网收费模式不同.

(4)广域网的安全考虑与数据中心内不同.

(5)广域网的可靠性与数据中心内不同.

一个具体的例子就是数据资源在多个结点的复制(replication). 如果网络计算系统本身不提供自动复制的功能,应用软件开发就要提供此项功能,以改善应用性能和可靠性. 单点系统较少需要复制.

还有一些网络计算系统采用了下述形态:在服务器端只有一个数据中心,但多个用户的客户端设备与该数据中心一起共同执行一个应用程序. 很多 P2P 和文件分享系统就是例子. 我们将这类例子也归类为多点系统,因为从开发者角度看,程序是在多点(包括客户端设备)执行的.

多点系统也就是传统的分布式系统.

当然,我们还可以在“单点-多点”基础上做更细致的分类工作. 比如,一个数据中心本身就可能是一个网络计算系统,甚至是多点系统,不过这里的“点”是机器.

3.2 控 制

谁控制一个应用的开发工作(包括部署、配置、维护等)? 需不需要得到别人的批准? 一个开发者能否自主地完成应用的所有工作,让应用运行起来

为用户提供价值?

从技术上看,控制主要通过三种形式体现(我们忽略了法律、知识产权等其它因素):

(1)名字(或命名,naming). 简单的名字包括变量名、常数名、子程序名,更为复杂的名字包括 schema、服务名、数据字典、名字空间等.

(2)管理域(administrative domain).

(3)共享范围,即应用提供的服务和价值能为哪些用户使用?

对一个网络计算系统上的程序开发者而言,他特别需要知道他是控制者,还是网络计算系统是控制者.

集中式(centralized)网络计算系统. 控制权集中在管理网络计算系统的某个组织手里. 应用的开发必须受该组织的统一规定、统一标准、统一管理、统一维护所约束.

分散式(decentralized)网络计算系统. 控制权分散在各个开发者手里,每个开发者可自行获取所需信息和知识,自主决定名字、管理域和共享范围.

分散系统提供的自由不是绝对的. 分散系统的程序也必须遵循一定的约定和控制. 但是,这些控制通常以开放标准甚至开源软件的方式体现,可以不经别人批准自主获得. 另外,这些开发者和用户往往会通过草根方式自主地形成社区. 社区的约定和非强制性的控制,社区成员一般都遵守.

3.3 执行与控制

将执行与控制组合起来,我们可以得到图 2 所示的 4 类网络计算系统,即集中单点系统、集中多点系统、分散单点系统、分散多点系统. 图 2 还给出了一些典型的例子.

很多服务型软件(Software as a Service, SaaS)系统采用集中控制、单点执行的模式,是集中单点系统. 比如,salesforce.com 就是此类系统的典型代表. 该系统在一个数据中心执行,集中地为很多用户统一提供 CRM 服务. 使用起来,每个用户似乎看到一套安装在用户自己计算机上的 CRM 软件. 事实上,用户客户机只运行通用的 WWW 浏览器软件,CRM 功能是由 salesforce.com 的数据中心运行的软件提供的. CRM 软件的控制掌握在 salesforce.com 手里.

为了提高性能和用户体验,集中单点系统可以扩展成为多点执行系统. Google 就是这样的集中多点系统(又称为集中分布系统,centralized distributed system),其数十万个服务器在全球多个数据中心执

行应用. 但是, Google 的应用都是集中开发、维护的, 并不是任何人都可以开发应用、部署到 Google 的服务器上运行. 只有 Google 自身才能升级维护应用.

WWW 则是分散多点系统(分散分布系统, decentralized distributed system). 它是分布的, 因为现在的 WWW 已有数亿台机器运行. WWW 技术允许任何人开发出一个 Web 网页, 部署在一个 Web 服务器上(当然该 Web 服务器要有一个域名), 不需要某个公司或组织批准. 之后, 任何人都可以访问该网页, 也不需要某个公司或组织批准. 所有开发者都需要遵循 WWW 技术规范(URL, HTML, HTTP), 但编写、部署、维护网页的控制权并不掌握在某个中心, 而是分散在各个开发者手中.

分散系统不一定是多点执行. 比如, 一个数据中心可以为多个用户提供多台虚拟主机, 每一台都由特定用户自主控制. 这种虚拟化工作由 IBM 大型机(mainframe)的逻辑分区(LPAR)首倡, 今天已经用到了很多平台中. 虚拟化在近几年重新成为研究热点^[6], 部分因为它能够在一个数据中心或一台物理计算机上提供多台分散控制的虚拟机器.

图 2 揭示了分布系统(distributed system)和分散系统(decentralized system)的区别. 我们现在可以回答“为什么 Google 是一个集中系统”这样的问题了. 它是一个集中多点系统(集中分布系统), 即集中控制、分布执行的系统.

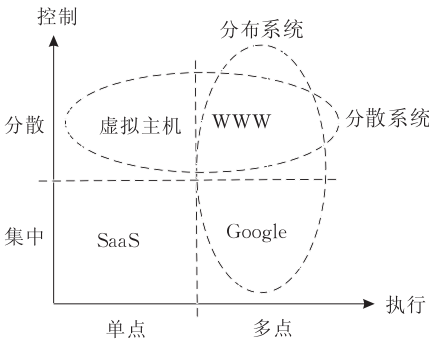


图 2 按照(执行,控制)划分的 4 类系统

反过来, 单点执行(有人称为集中执行)系统也可以有分散控制, 如虚拟主机.

3.4 层 次

应用开发者在网络计算系统上开发应用时, 首先需要了解系统提供了什么层次的开发平台. 比如, 在 Linux 平台上开发与在 J2EE 平台上开发的层次显然不同. 但是, 编程接口的区别并不是本质. 我们通过网络计算系统提供的资源类别不同划分层次.

第 1 个层次称为硬件层. 在这一层次, 网络计算系统提供硬件资源, 或硬件能力, 主要包括计算、存储和网络资源. 不论什么编程接口, 只要是提供硬件资源的网络计算系统, 都在这个层次. 比如, 通过一个文件系统提供存储, 或者像 Amazon Simple Storage Service(S3)那样通过 WSDL Web 服务接口提供存储, 都是硬件层.

用户可以在硬件层上开发并部署网络计算系统的应用和服务. 也可以部署一套应用支撑平台, 帮助其他用户开发和部署新的网络应用. 系统为这两类用户提供计算、存储和网络的硬件资源或硬件能力.

第 2 个层次称为平台层. 网络计算系统为开发者提供应用支撑平台, 包括 LAMP, J2EE, Ruby on Rails 等通用开发平台, 以及应用部署、配置、维护等支撑功能. Google App Engine、Heroku、force.com 可以看作提供应用平台的网络计算系统.

用户利用系统提供的平台环境, 可以降低应用开发、部署、维护的工作量. 例如, 在 force.com 系统上开发应用时, 可以通过 App Exchange 平台查找并集成其他用户开发的应用. Heroku 平台则提供即时部署、在线编辑、应用监控、弹性能扩展等功能, 可以在很大程度上降低应用的部署和升级维护工作量.

第 3 层称为应用层, 或应用服务层. 在这个层次, 网络计算系统为用户提供的主要是具有业务功能的应用(软件和数据)或服务.

用户可以通过专用客户端或 Web 浏览器, 使用网络计算系统提供的应用或服务, 也可以利用已有的应用或服务开发新的应用功能.

网络搜索(Google、百度等)是应用类网络计算系统的典型例子. 搜索软件和被搜数据库运行在搜索厂商的数据中心, 一个用户的客户端设备只需要运行 Web 浏览器. 用户提交需要搜索的词语, 搜索引擎返回搜索结果.

应用层上的“开发”向网络计算系统贡献两类新的资源: 新的数据和新的应用(或服务). 比如, 用户可向一个 P2P 文件共享系统贡献新的文件; 各种 mashup 方法则可以贡献新的服务.

4 分类实例

现在, 我们可以根据(执行, 控制, 层次)3 个维度将网络计算系统分为 12 类了(见表 1). 我们首先将系统按层次分为 3 类, 每一类再与(执行, 控制)

4 类组合.

表 1 网络计算系统分类及实例

系统类别		缩写	举例
应用类	单点集中应用系统	ASC	Salesforce.com
	单点分散应用系统	ASD	一些博客服务
	多点集中应用系统	AMC	Google Search
	多点分散应用系统	AMD	一些 mashup
平台类	单点集中平台系统	PSC	一些 Intranet 平台
	单点分散平台系统	PSD	Heroku
	多点集中平台系统	PMC	Google 平台
	多点分散平台系统	PMD	Google App Engine
硬件类	单点集中硬件系统	HSC	超级计算中心
	单点分散硬件系统	HSD	虚拟主机托管
	多点集中硬件系统	HMC	TeraGrid
	多点分散硬件系统	HMD	Amazon S3/EC2

表 1 中系统类别的缩写规则是:第 1 个字母表示系统类别的(层次)维度,第 2 个字母表示(执行)维度,第 3 个字母表示(控制)维度.例如,HMD 表示(硬件,多点,分散),即多点分散硬件系统.

下面举例说明每一类系统的特征.在实际运行中,虽然一些示例系统可能同时提供多个层次的资源,但本文以这些系统目前主要的使用模式作为分类标准.

例如,目前用户使用超级计算中心的主要方式是登录到计算中心的主机上,以批处理作业的方式执行计算任务.超级计算中心为用户分配存储空间和计算节点,同时对用户程序的运行情况进行严格管理.TeraGrid^[7]则可以看作是一台具有多个中心的超级计算机.TeraGrid 包含 11 个计算中心,具备统一的安全认证体系,通过监测工具对各中心资源的使用情况进行控制.因此,上述系统都属于典型的集中式系统.

传统的虚拟主机系统在一个数据中心上为不同的用户提供相互独立的虚拟机器.每一台虚拟主机都具有独立的域名和 IP 地址(或共享的 IP 地址),拥有自己的一部分系统资源(文件存储空间、内存、CPU 时间等),由用户自行管理和使用.因此,虚拟主机虽然一般在一个数据中心上运行,但属于分散式系统.

Amazon S3 和 Elastic Compute Cloud(EC2)是 Amazon 公司发布的云计算系统.S3 为用户提供低成本的网络存储.EC2 则为应用执行提供可调整大小的网络计算能力.S3 和 EC2 系统允许用户在系统上自由地开发、部署和运行应用,将控制权完全交给应用开发者掌握.因此,S3 和 EC2 同样属于分散式系统.

目前,一些实际运行的硬件类网络计算系统可能同时提供多个层次的资源.例如,Amazon S3/EC2 在提供硬件能力的同时,也为用户提供支持应用开发或包含特定应用功能的 Amazon Machine Image(AMI).TeraGrid 同时提供 C/C++ 等语言环境 and 专业应用软件.但它们主要为应用开发者提供存储、计算、网络通信等硬件资源.因此,我们将它们仍然归属为硬件类系统.

平台类系统中,Intranet 平台系统为数据集成、数据分析、工作流等企业应用提供消息传输、数据管理等平台软件.这些企业应用的开发和维护一般都由所属企业统一管理,谁能够使用这些应用由系统管理员决定.

类似的,支撑 Google Search 的 Google 内部平台也是平台层的网络计算系统.该平台在多个数据中心运行,由 Google 统一控制.

同样作为应用平台的 Heroku 和 Google App Engine 则完全由用户对应用的开发和使用进行控制.应用的命名、功能、管理域由开发者自主决定.应用的用户与平台系统可以没有任何联系.因此,这两个系统是比较典型的分散式平台系统.

应用类系统中,Salesforce.com 是一个比较典型的集中式系统.只有 Salesforce.com 的注册用户才能够使用 Salesforce.com 提供的 CRM 服务,CRM 功能的升级维护由 Salesforce.com 统一控制.Google Search 也与此类似,系统的升级维护由 Google 公司集中控制.

相比而言,博客服务和 mashup 系统的控制则非常松散.比如,我们不需要注册即可访问博客中的文章,并发表评论.我们可以将已发布的 Web 服务组合为新的 mashup 服务,而不需要通过某个管理机构的批准.

5 结 论

本文借鉴并行计算系统的经验,提出了一种按照(执行、控制、层次)3 个维度的分类方法,并用于一些典型的网络计算系统.

从近期的发展历史看,很多成功的网络计算系统都遵循了集中单点,再到集中多点,最后到分散多点的发展路线.WWW 是一个重要例外.

网络计算系统的分类工作还有很大的改进空间.一个真实的网络计算系统可能在几个层次都有体现,并有不同的(执行,控制)模型.

参 考 文 献

- [1] Flynn J. Some computer organizations and their effectiveness. *IEEE Transactions on Computers*, 1972, C-21(10): 948-960
- [2] Hwang K, Xu Z. *Scalable Parallel Computers: Technology, Architecture, Programming*. New York: McGraw-Hill, 1998
- [3] Maheswaran M, Ali S. A taxonomy of network computing systems. *IEEE Computer*, 2004, 37(10): 115-117



LIAO Hua-Ming, born in 1972, associate professor.

XU Zhi-Wei, born in 1956, Ph. D., professor. His research interests include parallel computing and network computing system.

- [4] Medin D, Ross B, Markman A. *Cognitive Psychology*. Orlando, USA: Harcourt College Publisher, 2001
- [5] Berners-Lee T, Hall W, Hendler J, Shadbolt N, Weitzner D. *Creating a science of the Web*. Science, 2006, 313(5788): 769-771
- [6] Smith J, Nair R. *Virtual Machines*. San Francisco, USA: Morgan Kaufmann, 2005
- [7] Foster I, Kesselman C. *The Grid 2: Blueprint for a New Computing Infrastructure*. San Francisco, USA: Morgan Kaufmann, 2003

Her research interests include grid computing and distributed system.

YU Hai-Yan, born in 1974, Ph. D., associate professor. His research interests include distributed system and programming environment.

ZHA Li, born in 1974, Ph. D., associate professor. His research interests include distributed computing and grid computing.

Background

In the past 20 years, many network computing systems have emerged due to rapid increases in market demands and enabling technologies. As these systems evolve by combining the best features of existing systems, differences among network computing systems are blurring. Research in classification is needed to assess new features and contributions of variant network computing systems, and to accelerate scientific research, knowledge accumulation and sharing in the field of network computing systems.

In the applications and industry fields, network computing systems are grouped in an ad hoc fashion. Examples include single-site clusters, multisite clusters, peer-to-peer networks, grids, computing utilities, service-oriented computing, cloud computing, business Web, consumer Web, etc. There is a lack of commonly accepted classification or taxonomy.

In academia, little research work has been done in classifying network computing systems. Only a four-way taxonomy has been found in the literature. Maheswaran et al proposed to differentiate network computing systems into four types, using the number of applications and number of owners as the grouping criteria. It is a good start, but still leaves

many systems difficult to classify, such as grid, cloud computing, and decentralized systems. This paper presents a new classification method, based on three dimensions of execution, control, and level. It shows that the classification method can cover representative network computing systems. For instance, we now have a class of systems featuring distributed execution, central control, and at application level. Many Internet search services belong to this class.

This research is supported in part by the National Basic Research Program (973 Program) of China Program of China (grant No. 2005CB321807) and the National High Technology Research and Development Program (863 Program) of China (grant Nos. 2006AA01A106, 2006AA01A121). The former is research on Internet virtual computing environment for scientific activities, while the later aims at research and development of middleware level software as well as applications on the China National Grid environment. A similar part of those projects is to analyze the characteristics of corresponding applications and optimize the architecture of each system. The research in this paper is a basic component of application analysis and architecture study, directly impacting the design, programming and optimization of such systems.