

云计算环境下 Web 服务集成系统的研究

刘 菲 郝风杰

(72671 部队应用开发室 济南 250022)

摘 要 作为云平台提升应用性能的一种重要手段,Web 服务集成技术近年来受到了工业界和学术界的广泛关注。从云计算与 Web 服务集成技术的结合入手,分析设计了基于云计算的系统体系结构,并基于此提出了基于 Web Services 的异构数据集成方法和应用集成的总体架构。最后,给出了该系统的相关实现实例。实验表明,该系统架构的应用在降低构建成本的同时大幅提高了系统性能。

关键词 Web 服务,云计算,应用集成,数据集成

中图法分类号 TP319 **文献标识码** A

Research on Dynamic Web Service Integration System in Cloud Computing

LIU Fei HAO Feng-jie

(Application and Development Unit, PLA 72671, Jinan 250022, China)

Abstract As a kind of emerging business computational model, cloud computing distributes computation task on the resource pool which consists of massive computers, accordingly, the application systems can gain the computation strength, storage space and software service according to its demand. In this paper, we described what is cloud computing, summed up key techniques, such as virtual database, Web services, as well as programming model and ESB technology used in cloud computing, and then an example of dynamic Web service integration system was illustrated. Experimental results show that the proposed system architecture can reduce the building-cost significantly as well as improve the system performance remarkably.

Keywords Web service, Cloud computing, Application integration, Data integration

1 引言

云计算^[1]把网络上的服务资源虚拟化,整个服务资源的调度、管理、维护等工作由专门的人员负责,用户不必关心“云”内部的实现,因此云计算实质上是给用户提供像传统的电力、水、煤气一样的按需计算服务^[2,3],它是一种新的有效的计算使用范式;并且,云计算是分布式计算、效用计算、虚拟化技术、Web 服务、网格计算等技术的融合和发展,其目标是用户通过网络能够在任何时间、任何地点最大限度地使用虚拟资源池,处理大规模计算问题。目前,在学术界和工业界共同推动下,云计算应用呈现迅速增长的趋势,各大云计算厂商如 Amazon, IBM, Google, Microsoft, Sun 等都推出自己研发的云计算平台^[4]。而学术界也源于云计算的现实背景纷纷对模型、应用、成本、仿真、性能优化、测试等诸多问题进行了深入研究,提出了各自的理论方法和技术成果^[5,6],极大地推动了云计算继续向前发展。

Web 服务^[7]和 UDDI^[8]的出现为异构的系统间的交互提供了可能性,使各种信息可以以消息的方式跨域交互。各个部门提供的事务可以以内部应用组件的形式完成,然后向外暴露这些应用组件良好定义的接口,让外部应用可以调用,成为一个基本的 Web 服务,而这个 Web 服务的描述 WSDL 在 UDDI 上发布,并按要求提供给其他部门、组织共享。虽然

Web 服务的出现为跨域操作提供了可能,但现阶段它仍然存在很多问题。其中一个非常重要的问题,就是部门的事务大都不是单一的基本服务的任务。某个部门事务可能是多个单位在不同系统上的跨域工作流的协同工作,形成复杂的集成服务,因此就涉及到有关 Web 服务的集成问题。

云环境具有超大规模的存储和计算能力,资源和结构具有动态伸缩性,并且通过虚拟化技术和庞大的资源池按需提供服务。随着越来越多的云服务的出现,需要采用 SOA 技术^[9]来将其加以整合,尤其是在 SaaS 层面上的服务整合。本文将分析在云环境下如何将多个异构系统的服务集成整合起来。

本文采用云计算的设计理念,依托网络、存储、传感器等基础设施,综合运用中间件、系统集成、数据挖掘、信息采集等技术,构建了 SaaS、PaaS、IaaS 服务层,为各级、各类用户按需提供服务和资源。系统综合利用 ESB^[10]、WebService^[11]、虚拟数据库等多种系统集成方式,整合多个业务系统,将系统的数据、服务资源统一进行管理和调度,解决了部分信息孤岛、烟囱系统的问题,提高了管理信息的共享应用水平。

2 系统总体架构

2.1 系统体系架构

政府面对的信息资源及应用多种多样,来自不同的操作

刘 菲(1978—),女,博士,工程师,主要研究领域为云计算、移动计算;郝风杰(1979—),男,硕士,工程师,主要研究领域为数据处理、网络分布式计算。

系统、不同的开发平台和不同的应用数据库。基于 Portal 平台的政府信息集成框架,试图以最小的代价在企业应用系统的构架层次上,为政府提供一个跨越多种分散的、内部和外部的信息处理过程的集成纽带,把这些信息整合到一起。

系统利用分布式存储、数据仓库等先进技术,通过综合集成的方法,将各部门的信息进行应用集成、数据整合,为用户 提供可靠、安全、准确的服务。

系统的总体结构基于云计算设计理念,采用层次化结构设计,主要由基础设施即服务 (Infrastructure as a Service, IaaS)、平台即服务 (Platform as a Service, PaaS)、软件即服务 (Software as a Service, SaaS) 以及安全防护与容灾备份体系组成。系统体系架构如图 1 所示。

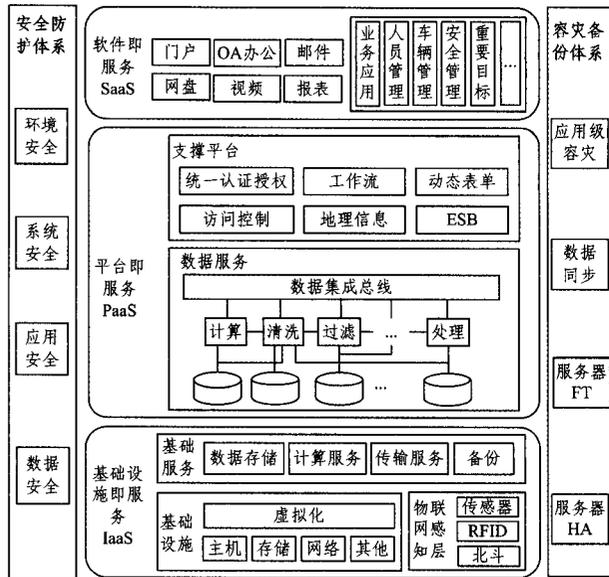


图 1 系统体系架构

基础设施层 IaaS,为管理提供随遇接入、按需交换的基础信息网络,以及模块互换、接口一致、规格统一的平台,为构建信息化安全管理综合集成平台提供基础支撑。经过把计算、存储和网络资源通过虚拟化形成资源池,为用户提供基础、灵活的运算和存储服务。

平台层 PaaS,是在基础设施层 IaaS 之上,提供了数据、统一认证授权、工作流、访问控制、地理信息、ESB 等服务。对于用户而言,平台层如同一台安装了地理信息、数据库、消息中间件等系统的计算机,在此基础上可直接开发软件应用。该层整合了管理的资源信息,并通过数据计算、清洗、过滤、存储以及处理等,实现异构数据的同步、共享和交换,设计数据集成总线,构建管理的资源服务体系,为业务支撑提供精确的数据和资源服务。

应用层 SaaS,主要提供共用和专用服务,实现对资源的动态规划和组织运用,并可以根据用户需要和 workflows 进行业务流程定制,为应用提供功能、业务和服务支撑。

安全防护与容灾备份体系,依据国标,提供用户认证、访问控制、网络防病毒、HA、FT 等安全防护手段和容灾备份策略,以提高服务和数据的安全性、可靠性、抗毁性。系统对用户资源进行统一管理,可以与其他应用系统的用户管理模块对接,并能随相关业务系统实时更新访问权限。通过完善的授权机制及存取控制,用户访问权限控制到字段级别,确保用户只能访问其具有权限的应用系统及相关信息。

2.2 异构数据集成

在系统综合集成的环境中,共享、维护、操作数据源是必须解决的问题,异构数据集成屏蔽各种异构数据间的差异,对各种异构数据提供统一的操作,使集成后的异构数据对用户来说是统一的和无差异的。

平台使用虚拟数据库法^[12]实现数据源的集成,其实现原理如图 2 所示。在虚拟数据库中,数据仍然保存在原来的各个数据源中,对数据集成系统的查询在运行时被分解为对各个数据源的查询。集成系统仅提供一个虚拟的集成视图以及对这个集成视图的查询处理机制。在这种方法中,数据不用复制到另外的数据仓库中,并且保证在查询时的数据均是“新鲜的”。

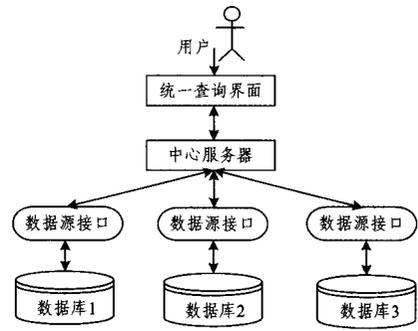


图 2 虚拟数据库法

当用户向中心服务器提交一个查询时,中心服务器把查询分送到每一个数据源,同时数据源接口把查询请求发送到相应的数据源中,事实上中心服务器可能给一个数据源接口发送很多查询,也可能不给该接口发送任何查询。从各个数据源中得到的结果返回给了中心处理程序,又由中心服务器集成这些结果数据,再把这些数据的局部模型转换成公共数据模型后呈现给用户。

在系统集成中,数据中心的中心服务器建立数据映射规则表,使用 Web Services 技术,以 SOAP 为传输协议,为用户 提供各个异构数据库的共享信息及统一的查询界面,能够很好地解决互操作问题,真正实现异构数据源的无缝集成。在各个异构数据源所属的服务器中,以 Web Services 组件的方式提供远程数据库连接、查询服务,如图 3 所示。

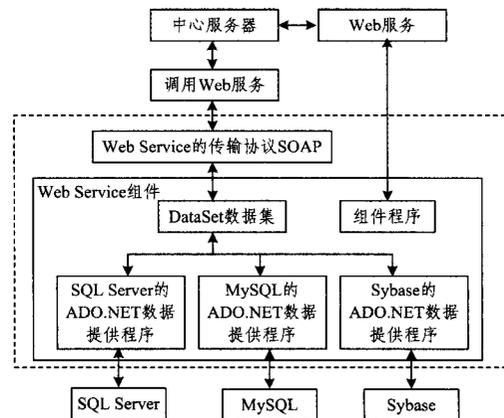


图 3 基于 Web Services 的异构数据集成

Web Services 组件利用 JDBC 技术存取异构数据源中的数据,再通过 Web Services 的 SOAP 将数据传输至中心服务程序的 DataSet 数据集,最后,利用字段映射表将数据的局部数据模式转换至公共数据模式。

2.3 应用集成

该平台通过企业服务总线(ESB)、XML 消息处理工具、流程集成(BPM)^[11]、企业门户(Portal)^[13]等工具来搭建集成环境。其中,实现集成能力的技术支撑组件包括部分软件产品和组件化应用,主要有统一身份认证组件、身份管理组件、统一授权组件、界面管理组件和访问控制组件等。平台的应用集成总体架构如图 4 所示。

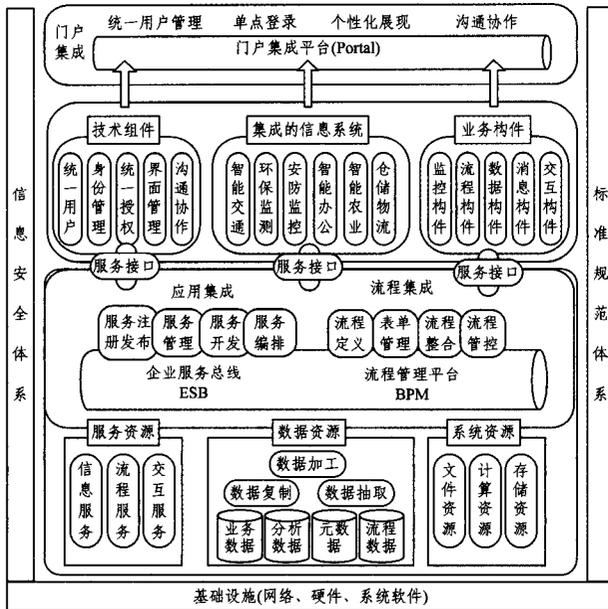


图 4 平台综合集成总体架构

应用系统集成以 Web Services 技术为核心,通过服务链接应用系统和平台及附属平台的技术组件和业务组件,从而形成“平台+应用”的整体架构。利用 ESB 技术将各种复杂的业务系统与上层应用分离,通过消息、适配器和传输管道搭建中间平台,完成应用系统间的整合和集成。

由于每个分系统的数据格式、传输协议、接口不尽相同,通过增加适配器将这些系统对外的接口进行包装并挂接到统一的服务总线上。如果现有系统本身提供了数据发布接口(如 Web Service、JMS 等),则可直接在接口层进行适配器的编码;如果系统是单机应用或不提供对外的数据接口,则通过 Web Services 技术将这些系统对外的接口进行包装。

利用 Web Services 能够很好地解决异构系统之间的通信、数据共享问题,Web Services 的基本应用模式就是所有的服务端组件都以 Web 服务的形式提供服务。从使用者的角度来看,不管底层是使用 EJB,还是 CORBA 或者 DCOM,最终面向用户的都是一个统一的 Web 服务调用接口,通过 Web 服务统一的接口调用,完全屏蔽了不同服务端组件之间的差异。

具体方法是:通过建立内部 Web 服务平台,将内部各种系统进行 Web 服务改造,即利用系统的逻辑建立相应的 Web 服务接口,并发布到内部的 UDDI 中,其他系统欲使用其相关功能时可以通过调用相关的 Web 服务接口来实现,使得原有的采用不同平台、语言、工具开发的各个分散的系统通过 Web 服务的模式得以重新应用,为系统提供了进一步扩展的开放平台。

图 5 描述了基于 Web Services 的应用集成的基本模式。

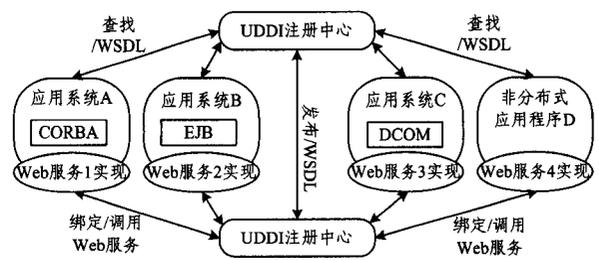


图 5 基于 Web Services 的应用集成

如图 5 所示的那样,基于 Web Services 实现应用集成,就是要使用 Web Services 封装、转变遗留系统(包括 CORBA 组件、EJB 组件、DCOM 组件等不同规范、语言实现的应用组件,以及非分布式的应用程序)。通过使用 Web Services 建立统一的调用接口,并通过内部的私有 UDDI 注册中心发布 Web 服务调用接口,这样其他的应用系统及 Web 服务就可以通过查询企业内部的 UDDI 注册中心发现并绑定所需的 Web 服务。

3 系统实现

3.1 平台技术架构

该平台采用 6 层的体系架构,如图 6 所示,其中:

第一层:Web(表示)层,通过 HTML 或 Ajax 的形式展现用户的输入界面并完成数据的采集;

第二层:Action Form 层,负责数据的采集,将 Web 层收集回来的数据通过 WebWork2(Struts 框架的核心)传递给 ValueObject 对象,并为进一步处理做好准备;

第三层:Proxy 层,负责将 Action 请求进行分发,通过这一层的处理,将 Web 前端处理和业务逻辑清晰地分开;

第四层:Process 层,利用 EJB 技术实现主要的业务逻辑;

第五层:DAO 代理层,通过相应的设置使得系统具备适应不同数据库及采用不同实现方式的能力;

第六层:DAO 实现层,实现具体的 DAO 接口,利用 Hibernate 实现 DAO 功能,通过实现不同的 JDBC 接口来兼容不同类型的数据库。

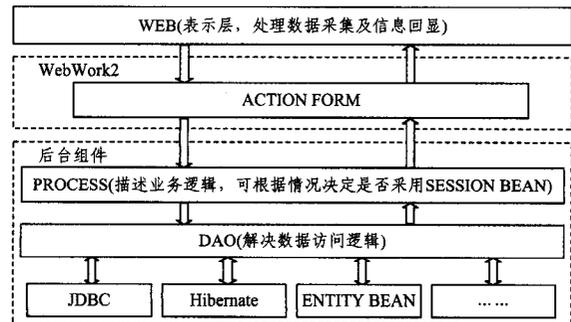


图 6 支撑平台技术架构

3.2 应用集成

Mule 是一个以 Java 为中心的 ESB 平台^[14,15]。它是可升级的、高分布式的对象代理,可以通过异步传输消息技术来无缝地处理服务与应用之间的交互。它支持 20 多种传输协议(FTP、UDP、TCP、SMTP、POP、HTTP、SOAP、JMS 等),可以方便、快捷地创建集成流程。Mule 的最大特色就是提供用于消息格式转换和特定消息格式传输的工具,把这些复杂的工作与应用逻辑、消息组件分离,用户只需编制消息组件的逻

辑,并把消息组件、转换器和传输管道搭建起来即可完成应用系统间的整合和集成工作。在该系统中主要通过 Mule 来实现系统的集成,分为现有系统的集成和新研系统的集成两种情况。

1. 现有系统的集成

现有业务系统各自维护一部分统一、权威的数据,通过 Mule 把数据的变更信息传递给共享数据中心,共享数据中心中存放着标准化后的数据结构和数据,再由共享数据中心通过 Mule 进行变更数据的发布,需要该权威信息的系统接受数据,并更新本地数据库,以确保系统中存在数据的唯一性和有效性。现有系统的集成示意图如图 7 所示。

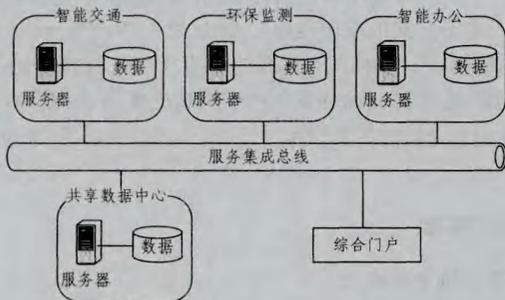


图 7 现有系统的集成示意图

现有系统的集成主要分以下几类。

(1) B/S 系统的集成

数据发布:可通过调用 Mule 提供的 Web Service 数据输入接口,将更新数据传递到 Mule 的服务组件中,再由服务组件对数据进行加工、存储、发布。

数据接收:由 Mule 直接调用系统的数据输入接口即可。

(2) C/S 系统

数据发布:如果系统有对外分发数据的功能,可直接用 Mule 接收其分发的数据,然后转交给 Mule 服务组件进行处理。若无,则可在 Server 端增加一个数据库监视服务,定时抽取更新数据,并传递到 Mule。

数据接收:可由 Mule 直接向系统的输入接口(或数据库监视服务)发送数据。

(3) 单机数据库系统

需增加单机库的监视服务,完成数据的发布和接收。

2. 新研系统的集成

新研系统的集成要方便很多,只需在开发时满足两个条件,一是按照共享数据中心的数据格式标准进行设计;二是提供数据发布、接收接口。Mule 通过系统的数据发布接口接收数据,无需进行转换,可直接将标准格式的数据存储到共享数据中心,这样不仅提高了系统执行效率,还减少了系统集成时的工作量。

4 应用实例

基于该平台开发了一个综合信息集成系统,运用云计算和异构数据集成技术,对人员、车辆等实体进行综合监控,提供面向用户的综合管理信息服务,以提高日常管理信息化水平。

该系统综合利用 ESB、Web Service、虚拟数据库等多种系统集成方式,整合了现有 6 类 21 个业务系统,将系统的数据、服务资源统一进行管理和调度,采用 RIA 技术,通过企业

服务总线、异构数据整合等技术,实现各类信息的聚合、展示。

在智能交通子系统中,车辆信息页面主要显示车辆属性信息、车辆实时监控信息等,如图 8 所示。选择单位可查看该单位车辆情况,并在地图上显示车辆位置。选择车辆,可查看该车辆的详细信息。

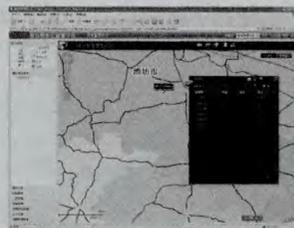


图 8 综合信息集成系统

与同类系统相比,本系统还具有以下优点:

1. 系统基于云计算、物联网架构体系,构建了柔性云计算,实现了一个灵活、健壮的基础框架,不仅能在该平台上进行各种服务构件的开发、安装,且能实现第三方应用的集成。

2. 系统通过统一的业务开放模式,以标准接口对外提供软件平台层面的应用服务引擎,将信息服务总线、身份认证管理、数据挖掘、访问控制、基础地理信息等功能作为服务提供给客户,为用户构建一个灵活的、模块化的和可扩展的综合应用平台。

3. 基于柔性定制技术,研发了业务流程软件模型,实现了流程动态改变、模块按需定制、服务动态部署、应用动态重构。

4. 综合采用多种实用性设计,提高系统易用性、可视性和规范性。

结束语 云计算具有广阔的应用前景,云计算的数据具有海量、异构、非确定性的特点。与此同时,随着云计算越来越流行,预计有新的应用场景出现,数据管理方面也会面临新的挑战。例如,可能会出现一些需要预载大量数据集(像股票价格、天气历史数据以及网上检索等)的特殊服务。可以看出云计算和云数据管理平台服务本身在适当场景下巨大的优势,同时还有所面临的技术难题亟待解决。动态 Web 服务集成是 Web 服务和语义 Web 在云计算环境中发展的必然要求。在文中,我们基于云计算架构模型和 Web Service 技术进行服务的集成,形成复杂的业务服务,并给出了系统的原型;将该原型用于某个综合信息集成平台中,并取得了初步成效。

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。

参考文献

- [1] International Data Corporate(IDC). Ranking of issues of Cloud Computing model[OL]. <http://blogs.idc.com/ie/?p=730>
- [2] Armbrust M, Fox A, Grith R, et al. Above the clouds: A Berkeley View of Cloud Computing; UCB/EECS-2009-28[R]. Berkeley, USA; Electrical Engineering and Computer Sciences, University of California at Berkeley, 2009
- [3] Vaquero L, Rodero-Marino L, Caceres J, et al. A break in the clouds: towards a cloud definition [J]. SIGCOMM Computer Communication Review, 2009, 39(1): 50-55

(下转第 430 页)

节约了成本,尤其与 ICPCP 算法相比,优势非常明显。

随着 θ 增大,本文算法的优势更加明显,如 θ 取 0.3 时,对比结果如图 10 所示。

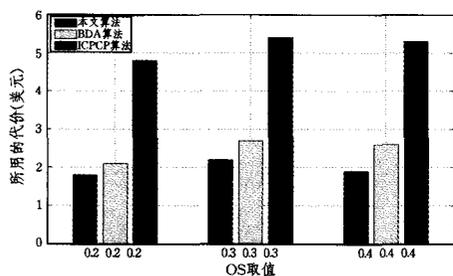


图 10 θ 取 0.3 时本文算法在执行费用上与其它算法的对比

由图 10 可知,随着 deadline 限制的逐步减小,本文算法与 BDA 以及 ICPCP 算法进行比较,平均降低 20%~60% 的成本,体现出了本文方法节约成本的明显优势。

结束语 由于虚拟机按周期调度,而当任务执行时间小于虚拟机周期时,将会因产生剩余时间而出现虚拟机浪费成本问题。为减少当前云计算中 workflow 任务调度成本及等待时间,提出了一种双向调度算法。该算法包括了 Backward 以及 Forward 两部分调度,实现将任务调度到具体类型的虚拟机,该算法充分考虑到云计算环境下的虚拟机按小时收费特点以及任务之间传输时间限制,在调度具体虚拟机时,不仅要满足任务按照用户所规定的 deadline 完成,还要满足用户支付费用最小的要求。通过实验可以看出,本文算法比 ICPCP 以及 BDA 算法进一步节约了费用。

参考文献

[1] Dastjerdi A V, Buyya R. An autonomous reliability-aware negotiation strategy for cloud computing environments [C] // 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). IEEE, 2012; 284-291

[2] Abrishami S, Naghibzadeh M, Epema D. Deadline-constrained workflow scheduling algorithms for IaaS clouds [J]. Future Generation Computer Systems, 2013, 29(1): 158-169

[3] Zhou A C, He B S. Transformation-Based Monetary Cost Optimizations for Workflows in the Cloud [J]. IEEE Transactions on Cloud Computing, 2014, 2(1): 85-98

[4] Rimal B P, Jukan A, Katsaros D, et al. Architectural requirements for cloud computing systems: an enterprise cloud approach [J]. J Grid Comput, 2011, 9(1): 3-26

[5] Singh S, Chana I. QoS-aware resource scheduling framework in cloud computing [J]. Journal of Supercomputing, 2015, 71(1): 241-292

[6] Toosi A N, Calheiros R N, Thulasiram R K, et al. Resource Provisioning Policies to Increase IaaS Provider's Profit in a Federated Cloud [C] // Proceedings of the 13th IEEE International Conference on High Performance Computing and Communications (HPCC'11). IEEE Computer Society, 2011: 279-238

[8] Kwok Y K, Ahmad I. Static scheduling algorithms for allocating directed task graphs to multiprocessors [J]. ACM Computing Surveys (CSUR), 1999, 31(4): 406-471

[8] Bajaj R, Agrawal D P. Improving scheduling of tasks in a heterogeneous environment [J]. IEEE Transactions on Parallel and Distributed Systems, 2004, 15(2): 107-118

[9] Poola D, Garg S K, Buyya R, et al. Robust Scheduling of Scientific Workflows with Deadline and Budget Constraints in Clouds [C] // 2014 IEEE 28th International Conference on Advanced Information Networking and Applications. 2014: 858-865

[10] Yuan Y, Li X, Wang Q, et al. Deadline division-based heuristic for cost optimization in workflow scheduling [J]. Information Sciences, 2009, 179(15): 2562-2575

[11] Pandey S, Karunamoorthy D, Buyya R. Workflow engine for clouds [M] // Cloud Computing. 2013: 321-344

[12] Byun E K, Kee Y S, Kim J S, et al. Cost Optimized Provisioning of Elastic Resources for Application Workflows [J]. Future Generation Computer System, 2011, 27(8): 1011-1026

[13] 苑迎春, 李小平, 王茜, 等. 成本约束的网格 workflow 时间优化方法 [J]. 计算机研究与发展, 2009, 46(2): 194-201

[14] 李强, 郝沁汾. 云计算环境中虚拟机放置的自适应管理与多目标优化 [J]. 软件学报, 2011, 34(12): 2253-2266

[15] 张伟哲, 张宏莉, 张迪. 云计算平台中多虚拟机内存协同优化策略研究 [J]. 计算机学报, 2011, 34(12): 2266-2277

[16] 周景才, 张沪寅, 查文亮, 等. 云计算环境下基于用户行为特征的资源分配策略 [J]. 计算机研究与发展, 2014, 51(5): 1108-1119

[17] Cai Z C, Li X P, Chen L, et al. Bi-direction adjust heuristic for Workflow Scheduling in Clouds [C] // 19th IEEE International Conference on Parallel and Distributed System. 2013: 94-101

(上接第 420 页)

[4] 刘鹏. 云计算 [M]. 北京: 电子工业出版社, 2010

[5] 陈康, 郑纬民. 云计算: 系统实例与研究现状 [J]. 软件学报, 2009, 20(5): 1337-1348

[6] Lenk A, Klems M, Nimis J, et al. What's inside the Cloud? An Architectural Map of the Cloud Landscape [C] // Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing. 2009: 23-31

[7] Martin D, Burstein M, Hobbs J, et al. OWL-S: Semantic Markup for Web Services [OL]. <http://www.w3.org/Submission/OWL-S/>

[8] Clement L, Hatley A, von Riegen A, et al. UDDI Version 3. 0. 2 [OL]. <http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>

[9] Papazoglou M P, Traverso P, Dustdar S, et al. Service-oriented Computing: State of the Art and Research Challenges [J]. IEEE

Computer, 2007, 40(11): 38-45

[10] Wang Pei, Jiang Chao-hui, Kang Zhi-qian. Research on ESB-based Enterprise Application Integration [C] // Second Asia-Pacific Conference on Information Processing. 2010

[11] Papazoglou M P. Web Services-Principles and Technology [M]. Prentice Hall, 2008

[12] Weske M, et al. Business Process Management [M] // Concepts, Languages, Architectures. Springer, 2007

[13] Bures M, Jelinek I. Framework for Easy and Effective Implementation of Adaptive Features in Web Portal [C] // International Conference on Computers and Advanced Technology in Education. Crete, Greece, 2008

[14] Peter D, Antoine B. Mule 2: A Developer's Guide [M]. Apress, 2008

[15] Mule [OL]. <http://www.mulesoft.com/>