一种基于SOA的网格任务调度框架

易 侃,王汝传
(南京邮电大学 计算机学院,江苏 南京 210003)

摘 要:目前网格任务调度算法大都通过仿真手段进行验证,缺少在实际的网格任务调度系统中检验。通过在实施网格项目中的经验,提出了一种基于SOA的网格任务调度框架GTSF(Grid Task Scheduling Framework),该框架通过web服务技术将任务调度解耦为多个服务模块,不仅简化了算法设计人员的工作量,还使得网格任务调度系统更加稳定。最后基于GTSF设计并实现了一个图像渲染应用供其他网格应用的开发人员参考。实际的网格应用开发过程显示GTSF使得基于Globus中间件的网格应用系统能够更快、更好的开发和部署。

关键词:网格;任务调度;SOA;Globus
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2010)04-0155-04

A Task Scheduling Framework Based on SOA in Grid Computing

YI Kan, WANG Ru-chuan
(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: At present, task scheduling algorithms are almost verified by means of simulations, rarely by task scheduling systems in real grid environment. Provides a grid task scheduling framework totally based on SOA, GTSF, which came from our experiments in grid related projects. GTSF decouples a task scheduling system into some independent modules by web services. It can not only simplify coding workload of algorithm designers but make the whole scheduling system stable. An image rendering application base on GTSF will be described at the end of the paper. Experience has shown that GTSF makes the process of grid application development and deployment based on globus toolkit more simple and rapid.

Key words: grid; task scheduling; SOA; Globus

0 引 言

基于开放网格体系结构(OGSA)[1]的网格基础设施将异构、分布的资源组织在一起,提供大规模的计算和存储能力。网络上的任何资源通过安装和配置中间件如Globus[2],即可成为一个虚拟组织内的网格资源。Globus中间件提供安全服务、任务管理服务、数据传输服务、资源发现和索引服务等基本的网格服务。基于web的网格服务已被公认为网格的基本技术支撑[3],网格任务调度框架利用基本的网格服务为用户提供透明的网格应用。目前Globus官方提供的第三方工具中包含两个网格调度框架CSF和GridWay。

CSF[4]是与WSRF兼容的网格任务调度框架,利用Globus提供的基本服务,开发了作业服务、队列服务、资源预留服务等有状态的web服务。利用CSF,网格用户可以通过Globus的GRAM协议与各种本地资源管理器,如LSF[5]、PBS[6]、Condor[7]等协作。除了支持基本的作业管理服务外,CSF还支持可配置的作业调度机制;有限的资源预留机制;兼容Pre-WS-GRAM,WS-GRAM协议等。然而,基于CSF的网格应用只限制在分布式作业管理领域,无法支持其他类型的网格应用,此外,对于一个简单的调度算法,算法设计人员需要了解CSF底层的大部分设计,编写大量的代码才能实现。GridWay[8]参照了集群环境下的分布式资源管理系统的设计,通过Globus的基本服务管理网格内共享的、异构的资源。GridWay通过中间件访问驱动(MAD,Middleware Access Driver),包括信息管理器、传输管理器,执行管理器,屏蔽底层网格中间件之间的差异,支持多种网格中间件;通过解耦调度过程和派送过程,让用户编写自己的调度器,实现自己设

收稿日期:2009-08-09;**修回日期**:2009-12-15
基金项目:国家自然科学基金(60973139;60773041);江苏省自然科学基金(BK2008451);国家高科技863项目(2007AA01Z404;2007AA01Z478);江苏高校科技创新计划项目(CX08B-085Z;CX08B-086Z);江苏省六大高峰人才项目(2008118)
作者简介:易 侃(1981-),男,江苏南通人,博士研究生,研究方向为网格计算、信息安全、计算机软件等;王汝传,教授,博士生导师,研究方向是计算机软件、计算机网络和网格、对等计算、信息安全、无线传感器网络、移动代理和虚拟现实技术等。

定的调度策略;通过丰富的 DRMAA 库和命令,让用户编写复杂的网格应用。然而,Gridway 不是基于 SOA 架构的,它是一个客户端程序,扩展性较差。

为此,开发了 GTSF 网格任务调度框架,该框架完全基于 SOA 架构[9],以 Globus toolkit4 为底层,框架主体完成了与用户通信,框架模块之间的通信,以及框架与 Globus 服务的通信的大部分通用的功能,算法设计人员只需要关注调度算法本身,忽略与算法无关的其他细节,并部署与算法相关的网格服务即可实现新的满足应用需求的调度算法,该过程极大地简化了算法设计人员编写的代码量,同时也有利于调度系统运行的稳定。

1 GTSF 的框架体系结构

基于 WS-GRAM 的作业管理使得用户远程提交、执行作业称为可能,它是基于 Globus 的网格任务调度框架的基础。由于网格应用的类型、对应用性能要求的不同,针对不同网格应用的调度算法也不同,如何让算法的设计者只关心算法本身而忽略用户、信息服务、调度服务和应用服务之间的相互调用是研究网格任务调度服务框架的关键。因此首先要将网格任务调度框架与调度算法模块分离,明确网格任务调度框架和调度算法模块所应有的功能和作用。

图 1 显示了基于 SOA 的网格任务调度框架(GTSF)的体系结构,其中灰色部分为任务调度算法设计者需要根据算法需求编写的服务部分,而网格任务调度服务、应用层服务和网格基础服务是网格任务调度服务框架应有的服务,完成绝大部分与算法无关的功能。

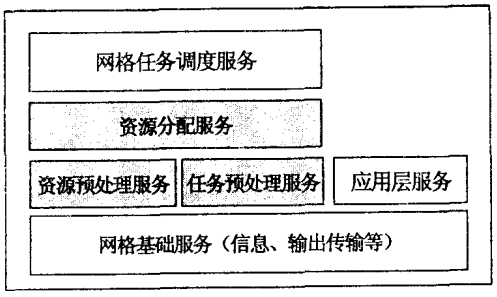

图 1 GTSF 体系结构

网格用户与网格任务调度框架的交互以及调度框架内其他服务的交互都是通过 Web Services 的方式实现的。网格任务调度服务是网格用户与网格平台交互的接口,通过该接口进行任务提交、任务监控以及获得任务调度结果等功能;应用服务是应用服务提供商将自己开发的网格应用以标准接口发布的网格服务,如客户流失分析服务、图像渲染服务等;而网格基础服务,如信息服务、输出传输服务,安全服务等可以由标准的网格基础设施 Globus、Legion 等中间件提供。图 2 显示了用户与网格调度服务以及网格任务调度服务框架中的其他服务之间的消息传递过程。

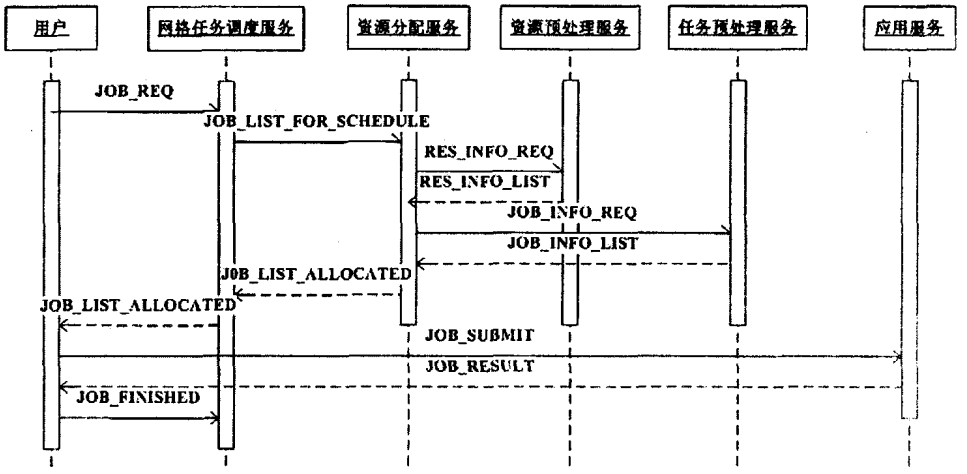

图 2 服务之间消息传递关系

1.1 网格任务调度服务

网格任务调度服务(SchedulerService)是用户与资源分配服务之间的接口,用于接收用户的任务请求,向资源分配服务发送资源分配请求并将资源分配结果返回给用户,同时可以控制任务的执行、跟踪任务状态等。该服务属于框架中的核心服务,它使得调度算法设计人员无需关心用户与网格任务调度服务之间的交互功能,减少这部分功能的重复开发。图 3 显示了调度服务的类图,其基本功能有:

1)用户安全策略管理。包括用户的认证和授权;

2) 任务队列管理。包括任务入队策略,如先来先服务策略、优先级入队策略等;任务出队策略,如任务完成出队策略、超时等待出队策略等;

3)任务的状态管理。包括任务执行状态的控制和监控,为实现该功能需要网格应用服务集成 ApplicationService 接口,该接口提供了控制和监控应用状态的方法;

4)任务调度性能统计。包括对任务平均等待时间、任务平均完成时间等其他指标的统计。

1.2 资源分配服务

资源分配服务由网格任务调度服务触发,并调用任务预处理服务和资源预处理服务分别获取任务信息列表和资源信息列表,然后调用任务调度算法为每个

任务分配一个资源。

资源分配服务(ResAllocateService)提供一个标准的调度算法的接口 SchedPlugin,与应用相关的调度算法实现该接口中的 scheMatch(List resInfoList, List jobInfoList)方法即可实现资源分配功能,其中资源信息列表(resInfoList)是通过资源预处理服务的客户端存根(ResInfoClient)获得,资源信息由基类 BaseResInfo 描述,特殊的资源信息要求,如声誉、信任、故障率等可以通过扩展该基类来实现;同样任务信息列表(jobInfoList)是通过任务预处理服务的客户端存根(JobInfoClient)获得的,任务信息由基类 BaseJobInfo 描述,特殊的任务信息要求,如信任等可以通过扩展该基类来实现。图 4 显示了任务调度算法插件类图。

1.3　任务预处理服务

任务预处理服务接收网格任务调度服务的请求、分析任务描述文档,将任务的信息列表返回给资源分配服务使用,包含以下两个功能:

1)分析任务执行的信息。任务执行信息包括任务描述文档中的基本信息,如任务名、任务执行的空间、上传的文件、执行该任务的服务、需要的资源能力等;另外还包含调度算法需要的信息,这些信息与具体的应用要求有关,如任务执行的估计时间、任务所需数据的估计传输时间等,这些信息通常需要通过分析任务描述信息,和应用相应的预测机制获得;

2)构造任务信息对象。任务信息对象是描述任务所有信息的对象,它所对应的类是扩展自 BaseJobInfo 类,该类的实例将返回给资源分配服务。

1.4　资源预处理服务

资源预处理服务接收网格任务调度服务的请求,从资源信息相关服务中获取资源基本信息,并构造资源信息列表返回给资源分配服务,包括以下三个功能:

1)获取资源信息。资源的信息一般有三类,获取的方式有所区别:

a)硬件信息,软件(服务)信息,例如 CPU 个数、最

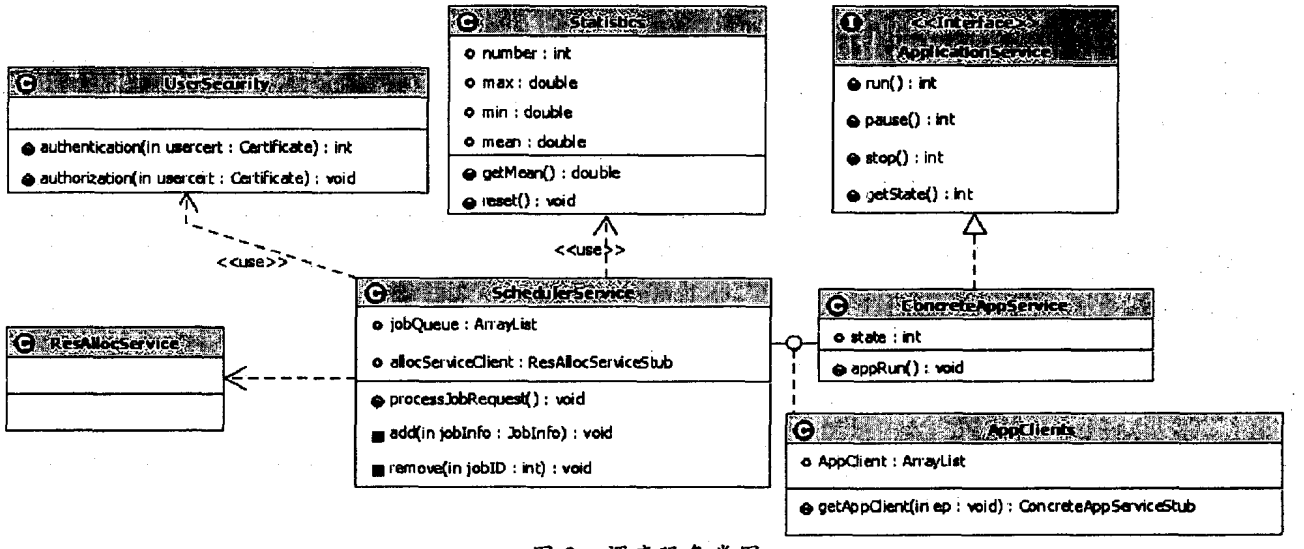

图 3　调度服务类图

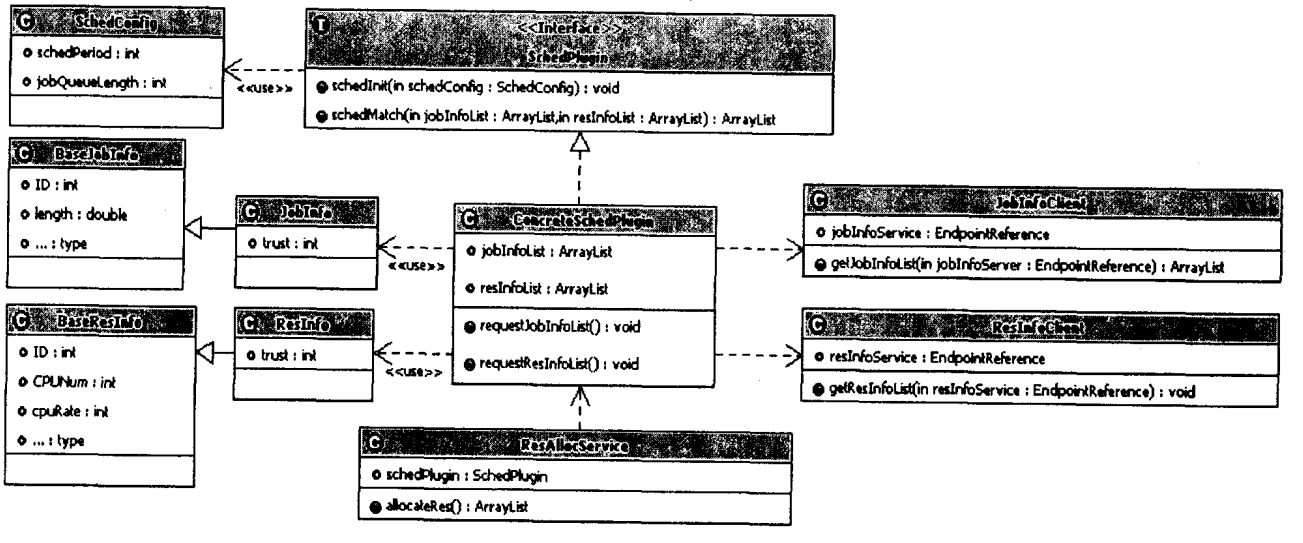

图 4　任务调度算法插件类图

大和可用内存大小、最大和可用硬盘大小、虚拟硬盘大小等，它们可以通过调用资源信息服务如 GT4 的 MDS 获取；

b)可以通过预测等手段获取的信息，例如 CPU 的计算能力、CPU 的平均使用率、内存的平均使用率等，它们可以通过统计的方法预测近似值；

c)资源的其他特殊信息，例如网格资源的信任、声誉等信息，它们可以通过设计专门的资源信任、声誉系统，并以服务的信息发布，供资源预处理调用。

2)筛选资源。筛选资源的过程是根据任务的需求信息对资源进行初次筛选，对于不能满足任务所需服务、性能要求的资源首先过滤掉。进行资源筛选的前提是假定资源分配问题是组合优化问题，不考虑服务组合的情况。

3)构造可用资源信息对象。资源信息对象是描述资源相关信息的对象，它所对应的类是扩展自 BaseResInfo 类，该类包含资源直接可获取的信息，如硬件和软件信息，而对于需要预测的信息以及特殊的信息，需要扩展该类。

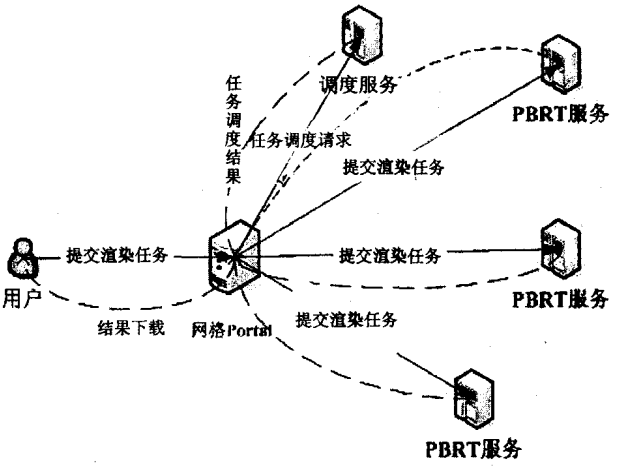

图5 图像渲染应用体系结构

2 基于 GTSF 的网格图像渲染应用

PBRT(Physically Based Rendering)[10]是一个开源的图像渲染引擎，它能将三维场景的描述渲染成一副具有真实感的图像。由于图像数据是非保密数据，数据易于分割、关联性小，因此 PBRT 常用于测试多核 CPU 的并行处理能力。现将多核 CPU 环境扩展为网格环境，即在多个网格资源内署基于 PBRT 的图像渲染服务，在 GTSF 框架下并行进行图像渲染。

图5显示了基于 PBRT 图像渲染应用的体系结构，包括两个关键的网格服务：

1)网格调度服务。该服务属于网格任务调度框架服务，已部署在 globus 容器中，它调用的资源分配服务、资源预处理服务、任务预处理服务在图中没有标出，它们之间的调度关系已在 GTSF 的体系结构中描述。调度算法采用经典的 MinMin 算法[11]用于测试。

2)PBRT 服务(pbrtService)是图像渲染应用的核心服务，它调用 pbrt 渲染引擎将 pbrt 文件渲染成图形。该网格服务主要提供调用图像渲染引擎的接口。用户、调度服务和 PBRT 服务的交互过程如图6所示。

用户首先通过用户名、密码登陆南京邮电大学网格门户，网格门户根据用户名对应的用户证书确定用户权限。用户选择图像渲染应用 Portlet 之后，就可以提交多幅渲染文件，进行并行的图像渲染，过程如下：

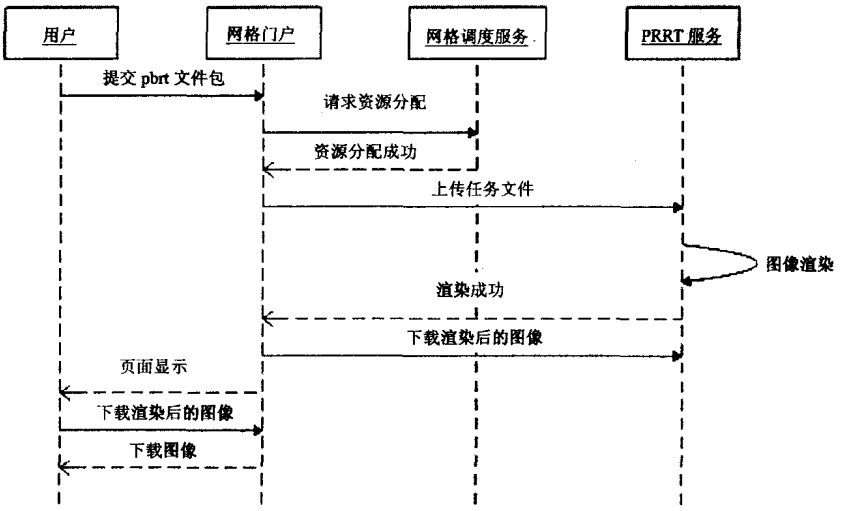

图6 基于网格的图像渲染过程

1)用户选择需要渲染的 pbrt 文件包，并上传至网格门户；

2)图像渲染应用 Portlet 向网格调度服务请求为每个任务(pbrt 文件)选择渲染服务所在的计算资源位置；

3)图像渲染应用 Portlet 通过 Globus 的数据传输服务将 pbrt 文件传输到分配的资源中，然后调用 pbrt 渲染服务进行渲染；

4)最后返回渲染后的图片供用户浏览和下载。

图像渲染应用包括四个模块：

* 图像渲染任务 PBRT 文件的选择和提交；
* 渲染图像的资源分配情况；
* 渲染后图像的缩略图显示；
* 单机和网格渲染的性能比较。

实验结果表明多幅图像在网格环境下渲染的完成

(下转封三)

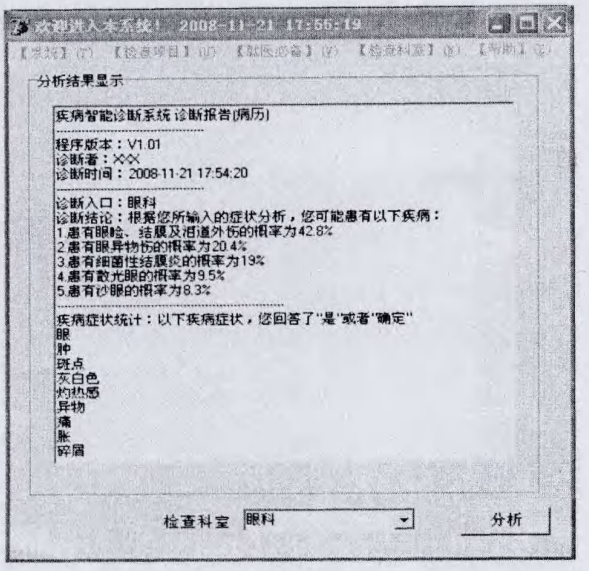

图 6 疾病智能诊断系统应用界面和疾病分析模块运行界面

系统的设计和开发，并给出了算法流程图及系统运行结果。

此方案在实际运行过程中，操作方便，运行稳定，性能良好，很好地解决了传统的疾病诊断方式中存在的不足，减轻了医生的工作负担，提高了工作效率，满足了对疾病诊断工作信息化、智能化的要求。

参考文献：

[1] 张红梅，王永成. 一个仿人疾病诊断专家系统模型[J]. 计算机应用研究，2000，17(1)：41－43.

[2] 张增强，刘 成. Delphi7 数据库开发完全手册[M]. 北京：清华大学出版社，2003.

[3] 王化玲，王玉洁. 判别分析——计算机看病[J]. 郑州铁路职业技术学院学报，2002，14(3)：62－63.

[4] 袁庆锋，景朋森. 基于 Delphi 下 ADO 技术应用技巧的探索与实践[J]. 淮海工学院学报：自然科学版，2005，14(3)：27－31.

[5] 李 金，吕汉兴. 医疗诊断专家系统推理机的设计与实现[J]. 微机发展(现名：计算机技术与发展)，2004，14(9)：43－44.

[6] 胡碧松，冯 丹，曹务春，等. 基于贝叶斯算法的移动式疾病智能诊断系统[J]. 计算机应用，2008(S1)：16－17.

[7] O'Neill P D, Roberts G O. Bayesian inference for partially observed stochastic epidemics[J]. Journal of Royal Statistical Society A，1999，162：121－129.

[8] Basanez M G, Marshall C, Carabin H. Bayesian statistics for parasitologists[J]. Trends in Parasitology, 2004，20(2)：85－91.

(上接第 158 页)

时间比在单台高性能计算机上渲染要快的多。

3 结束语

文中详细描述了一种基于 SOA 的网格任务调度框架 GTSF，并基于此框架实现了一个图像渲染应用。该框架完全是面向服务的，便于重用和灵活的构建与应用相关的调度系统。同时该框架为网格任务调度算法设计者屏蔽了网格任务调度中很多可复用的功能，使得它们只需要关心调度算法本身。目前 GTSF 功能和性能依然不够完善，需要通过更多的应用来进一步使之趋于稳定。

参考文献：

[1] Kishimoto H, Treadwell J. Defining the Grid: A Roadmap for OGSA™ Standards[M]. US: Open Grid Services Architecture Working Group, University of Virginia, 2005.

[2] Foster I. Globus Toolkit Version 4: Software for Service－Oriented Systems[C]//IFIP International Conference on Network and Parallel Computing. [s. l.]: Springer－Verlag, 2006：2－13.

[3] 胡春明，怀进鹏，孙海龙. 基于 Web 服务的网格体系结构及其支撑环境研究[J]. 软件学报，2004，15(7)：134－145.

[4] Mausolf J. Use Community Scheduler Framework to implement grid meta－schedulers, IBM Web Report, 2004, url[EB/OL]. 2004. www－128. ibm. com/developerworks/grid/library/gr－meta. htm.

[5] 王 涛，杨志义，周兴社. 基于 LSF 的集群管理系统的设计与实现[J]. 微电子学与计算机，2005，22(7)：73－75.

[6] 杨 洋，李菁菁，王庆官. PBS 作业调度研究[J]. 苏州大学学报：自然科学版，2009，25(1)：42－46.

[7] 周振宇，余丽琼，程东年. 浅析 Condor 和 Globus 在网格计算中的应用技术[J]. 信息工程大学学报，2004，5(1)：80－82.

[8] Huedo E, Montero R S, Llorente I M. A Recursive Architecture for Hierarchical Grid Resource Management[J]. Future Generation Computing Systems, 2009, 25(4)：401－405.

[9] Newcomer E, Lomow G. Understanding SOA with Web Services[M]. Toronto, Ontario: Addison Wesley2004.

[10] Lafortune E. Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering[D]. Brussels, Belgium: Katholieke Universiteit Leuven, 1996.

[11] Maheswaram M, Ali S, Siegel H J, et al. Dynamic Matching and Scheduling of a Class of Independent Tasks onto Heterogeneous Computing Systems[C]//8th Heterogeneous Computing Workshop (HCW'99). San Juan, Puerto Rico: IEEE Computer Society, 1999.