第34卷 第17期 计 算 机 工 程 2008年9月
Vol.34 No.17 Computer Engineering September 2008
·博士论文· 文章编号：1000—3428(2008)17—0018—03 文献标识码：A 中图分类号：TP393

本地站点网格作业监控模块的设计和实现

樊滑翔[1,2]，孙功星[1]，许榕生[1]
（1. 中国科学院高能物理研究所，北京 100049；2. 中国科学院研究生院，北京 100039）

摘 要：为构建基于用户网格身份的本地站点网格作业监控系统，该文在分析和研究WSRF规范和WS-GRAM实现机制的基础上给出网格作业监控模块的设计方案和体系结构，并阐述在其原型系统中各个组成模块的实现细节。网格监控模块原型系统能与WS-GRAM整合以提供基于用户网格身份的作业状态监控，并可进一步为网格授权服务提供动态信息。

关键词：Globus工具集；WSRF规范；WS-GRAM服务；DBCP技术

Design and Implementation on Grid Job Monitoring Module for Local Site

FAN Hua-xiang[1,2], SUN Gong-xing[1], XU Rong-sheng[1]
(1. Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049;
2. Graduate University of Chinese Academy of Sciences, Beijing 100039)

【Abstract】 In order to build a grid identity based grid job monitor system for local site management, the paper presents a grid job monitoring module scheme on the analysis and research of WSRF specification and WS-GRAM implementation mechanism, and also describes the implementation details of each component in the prototype system. The prototype system can integrate with WS-GRAM to monitor job state as user's grid identity, and can further provide grid authorization service with dynamic information.

【Key words】 Globus Toolkit(GT); WSRF specification; WS-GRAM service; DBCP technology

网格是在动态、跨机构虚拟组织环境下的协同资源分享和问题解决[1]，为达到这一目的而研发的网格中间件(grid middleware)是在网格层面提供统一视图以屏蔽分布各地的系统和资源的不同本地工作机制。GT4[2]即是这样的网格中间件，而其作为网格计算资源门户的WS-GRAM组件是网格计算资源的统一接口，为不同的本地资源管理服务提供网格统一视图。虽然通过本地资源管理服务，如Fork, PBS, Condor, LSF等，管理员和用户可以容易地以本地用户身份监控和管理作业，但是在网格环境中，用户的网格身份是唯一和相对长期的，而其本地身份是在特定环境下相关和暂时的，所以，以网格身份的用户行为，即在网格层面的作业监控和管理有重要的意义。本文在分析WS-GRAM工作机制的基础上，设计和实现为了本地站点管理的网格作业监控模块(Grid Job Monitoring Module, GJMM)。

1 WS-GRAM

Globus工具集[3](Globus Toolkit, GT)自20世纪90年代末发展至今一直致力于支持面向服务的分布式计算应用和基础设施，其核心组件涉及安全、资源访问和管理、数据移动和管理、资源发现等基本方面。GRAM作为GT核心组件之一，是针对遵从Unix编程和安全模型操作系统的网格作业管理问题解决方案的实现软件。

1.1 WSRF标准[4]

2005年4月发布的GT第4版，即GT4遵从WSRF标准，其GRAM升级为WS-GRAM。WSRF标准为利用Web服务建模和访问有状态的资源定义了一个通用和开发的框架。WSRF是利用Web服务通过属性来管理有状态的资源。在WS-GRAM的设计中，每个作业的执行情况正是由一个有状态的资源以一组属性的方式来表示的。

1.2 WS-GRAM的体系结构

WS-GRAM服务体系结构的核心对外表现为一组Web服务，这组Web服务被设计为驻于GT4的WSRF核心容器(container)中。对这组服务描述如下：

(1)ManagedJob Web服务。每个被提交的作业被表示为一个ManagedJob资源。该服务提供监控作业状态和作业操作的接口。

(2)ManagedJobFactory web服务。每个通过一个特有的本地资源管理服务访问的计算元素，被表示为一个ManagedJobFactory资源。该服务提供一个创建ManagedJob资源的接口，以通过本地资源管理服务执行作业。

WS-GRAM组件联合其他驻于GT4的WSRF核心容器环境中的组件来协同解决有关问题，如通过委托(delegation)服务组件来委托作业属主的网格身份证书，通过可靠传输服务(RFT)组件引入(stage in)执行所需文件和引出(stage out)执行结果文件等。

WS-GRAM是本地资源管理服务的网格化抽象，对于不同的本地资源管理服务，WS-GRAM通过对应的作业包装程序(job wrapper)提交作业给本地资源管理服务，而通过为不同的本地资源管理服务启动对应的调度事件产生守护进程

基金项目：国家科学网络综合平台下高能物理网格试验床系统建设基金资助项目(904l2017)；中国科学院创新基金资助项目“网格计算在高能物理科学中的应用研究”(U-512)

作者简介：樊滑翔(1982－)，男，博士研究生，主研方向：网格安全技术；孙功星、许榕生，研究员、博士生导师

收稿日期：2007-10-30 **E-mail：**Huaxiang.Fan@ihep.ac.cn

(Scheduler Event Generator, SEG)来获取调度事件，如作业状态转换事件。WS-GRAM 的系统结构和所驻环境如图 1 所示。其中 GRAM 代表前述的 2 个 Web 服务。

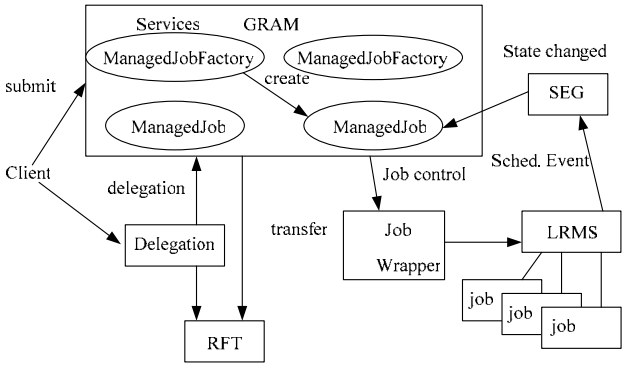

图 1　WS-GRAM 的体系结构和所驻环境

1.3　ManagedJob 资源的生命周期

如上节对 ManagedJob Web 服务的描述，一个 ManagedJob 资源承载一个被提交的作业的执行信息。其生命过程描述如下：当用户提交作业时，提交程序通过 Managed JobFactory web 服务提供的接口创建一个 ManagedJob 资源并设置其静态属性值，这些属性主要包括作业唯一号、作业属主网格身份、本地身份等。随即通过作业包装程序提交作业给本地资源管理服务。在作业执行过程中，WS-GRAM 通过 SEG 获取调度事件，在分析事件后，通过资源动态属性值更新机制来更新 ManagedJob 资源的动态属性值，这些属性主要包括作业当前状态等。

2　设计方案

GJMM 的设计目的是在网格层面对作业进行监控和管理。具体地说，其原型系统将主要获取两方面的信息：作业属主的网格身份与本地身份的映射信息和作业状态信息。GJMM 的设计涉及到以下几个问题：

(1)GJMM 以何种形式记录监控数据。因为 GJMM 记录的信息中包含本地站点范围内的所有运行作业的实时状态信息，信息增删、更新频繁，所以数据库更加适合管理这样的数据。

(2)GJMM 以怎样的方式采集信息。对于静态的信息，例如作业属主的网格身份和本地身份信息，可以通过资源属性规范主动查询 ManagedJob 资源的相应属性值；对于动态信息，例如作业状态信息，考虑到其对实时性的要求和频繁查询开销，采用通知(notify)方式，即每当作业状态改变时，GJMM 被通知更新其数据库信息。

(3)数据库的优化。鉴于(1)，GJMM 将无法避免频繁的数据库访问，所以，不能忽略这方面的开销。采用 DBCP 技术[5]，可以在一定程度上减轻这方面的负担。

本文方案是通过资源属性规范查询 ManagedJob 资源关于作业的静态属性值，如作业唯一号、作业属主的本地身份、网格身份等；通过修改 WS-GRAM 的资源动态属性值更新机制，使得当调度事件发生时，WS-GRAM 不仅更新对应 ManagedJob 资源的动态属性值，而且通知 GJMM 更新数据库信息。以上称为 GJMM 资源属性采集机制，主要负责对 ManagedJob 相关静态或动态属性值的获取。同时为了更好地让 GJMM 资源属性采集机制与 GJMM 数据库无缝整合，还需要有一个支持 DBCP 的数据库访问模块。由此，GJMM 将由数据库、数据库访问模块、资源属性采集机制组成并经由它们与 WS-GRAM 整合在一起。如图 2 为 GJMM 的组成部分，其中 GRAM 的黑色部分表示 GJMM 资源属性采集机制。

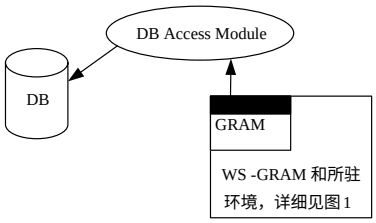

图 2　GJMM 的组成部分

3　实现细节

3.1　GJMM 数据库设计

GJMM 数据库数据内容以作业为中心，包括作业属主的网格身份和本地身份的映射关系以及作业的实时状态信息。所以，原系统设计 5 个数据库表：

(1)表 grid_users。反映作业属主的网格身份信息。表字段依次包括网格身份数据库编号(grid_id)和网格全球唯一名称(DN)。

(2)表 local_users。反映作业属主的本地身份信息。表字段依次包括本地身份的数据库编号(user_id)和本地 Unix 用户 uid。

(3)表 job_info。反映作业属主网格身份和本地身份的映射关系以及其他附属信息。表字段依次包括作业唯一号(uuid)、网格身份数据库编号(grid_id)、本地身份的数据库编号(local_id)、作业开始时间(start_time)、作业结束时间(end_time)、作业完成标记(is_finished)。

(4)表 job_state。反映作业的实时状态信息。表字段依次包括作业唯一号(uuid)和作业当前状态(current_state)。

(5)表 job_logs。记录作业各个状态转化的时间戳。表字段依次包括作业唯一号(uuid)、状态值(state)和状态转换时间戳(state_changed)。

GJMM 数据库表结构及其关系如图 3 所示。

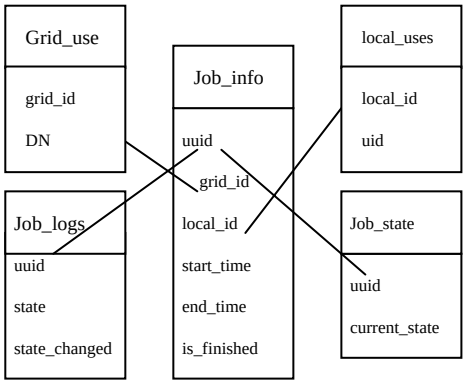

图 3　GJMM 数据库表结构及其关系

3.2　GJMM 数据库访问模块

GJMM 数据库访问模块是为了抽象数据库连接及 SQL 操作，提供访问 GJMM 数据库的业务层操作 API 和函数库。为了减少因频繁连接数据库而带来的开销，本模块采用 DBCP 技术，即在初始化时，建立一个由一定数量的连接句柄组成的连接池，连接池的实际数量会随着实际情况而消长，每次执行一次业务操作之前必须从连接池借得连接句柄或当连接池中所有连接句柄都借出时生成新的连接句柄，执行完毕后则归还连接句柄。执行一次业务层的操作即等效执行一个系列的 SQL 操作组。以下详细说明本原型系统提供的 2 个

业务层接口函数：

(1)startNewJob 函数。当创建新作业时，调用此函数创建 GJMM 数据库信息，为调用此函数，需要提供的参数包括作业唯一号 uuid、作业属主网格身份 DN 和本地身份 uid 以及作业创建时间戳 start_time。对应具体的一个系列的 SQL 操作组为：

1)若作业属主的网格身份或/和本地身份在 grid_users 或/和 local_users 表中不存在对应记录则执行本操作，否则直接跳到 2)。在所缺对应记录的表中插入新记录 {grid_id,dn} 或/和{user_id,uid}，其中，grid_id 和 user_id 分别是网格身份和本地身份的数据库编号标识。每次在这 2 个表中新增一条记录时，就将前一记录的数据库编号自动加 1 作为新记录的数据库编号；

2)分别从 grid_users 和 local_users 表中得到作业属主的网格身份数据库编号 grid_id 和本地身份的数据库编号 user_id. 在 job_info 表中插入新记录 {uuid,grid_id,user_id, start_time,null,false}；

3)在 job_state 表中插入新记录{uuid,'unsubmitted'},本原型系统的作业初始状态为 unsubmitted；

4)在 job_logs 表中插入新记录 {uuid, 'unsubmitted', start_time}。

(2)updateJobState 函数。当已创建作业状态改变时，调用此函数更新 GJMM 数据库信息，为调用此函数，需要提供的参数包括作业唯一号 uuid、作业改变后的状态 current_state、作业状态改变时间戳 state_changed。

1)在 job_state 表中定位对应 uuid 的记录，更新其状态值字段值为 current_state；

2) 在 job_logs 表中插入新记录 {uuid,current_state, state_changed}；

3)若 current_state 为完成(done)或失败(failed)时执行本操作，否则就此结束。在 job_info 表中定位对应 uuid 的记录，更新其作业结束时间字段值为 state_changed，更新其作业完成标记字段值为 true。

3.3 GJMM 资源属性采集机制

WS-GRAM 的主体部分是由 Java 实现的，类 Executable ManagedJobResource 实现了 ManagedJob 资源的相关数据和访问接口，其中类函数 start()的作用是将提交作业排入开始队列，等待被 WS-GRAM 进程池中的空闲进程处理。类函数 setState()函数则被 SEG 调用来改变 ManagedJob 资源的作业当前状态动态属性值。由此，增加了 2 个类函数以分别采集静态和动态属性值，并更新 GJMM 数据库信息：

(1)startGJMM 函数,通过资源属性规范查询 WS-GRAM 的静态资源属性值，例如作业唯一号、作业属主的网格身份、本地身份等信息，连同生成的当前时间戳，提供给 GJMM 数据库访问模块的 startNewJob 接口更新数据库信息。修改原类函数 start()函数，确保此函数被 start()调用。以保证作业在创建时，GJMM 获取其静态信息并更新数据库信息。

(2)updateGJMM 函数，通过 SEG 获取作业当前状态，连同作业唯一号和生成的当前时间戳，提供给 GJMM 数据库访问模块的 updateJobState 接口更新数据库数据。修改原类函数 setState()，确保此函数被 setState()调用。以保证当作业状态改变时，WS-GRAM 不仅更新对应 ManagedJob 资源的动态属性值，而且通知 GJMM 更新数据库信息。

4 结束语

为了以网格身份监控和管理本地站点作业，本文在分析 WS-GRAM 的基础上给出了 GJMM 的设计路线，阐述了详细的实现细节。对 GJMM 还可以从以下几个方面进一步地完善和加以利用：

(1)对 GJMM 历史监控信息的打包和处理。本原型系统，只是定期将历史数据删除，但是为审计的目的，可以将这些信息进一步打包和处理。

(2)GJMM 数据库数据内容可以根据实际的情况得到扩展，例如 grid_users 表可以增加角色、组等属性字段，local_users 表也可以增加组 id、附加组 id 字段等字段。

(3)目前原型系统的 GJMM 数据库访问模块只支持数据库数据的创建和更新，但是可以将其扩展以支持数据的查询，如查询一个网格用户同时有多少作业分别以不同的本地账号在运行。从而可以进一步跟网格授权系统整合，为授权系统提供授权所需信息。例如，本地站点不希望同一网格用户占用太多的资源，由此可以根据 GJMM 实时信息作出相应授权。

参考文献

[1] Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations[J]. International Journal of Supercomputer Applications, 2001, 15(3): 200-222.

[2] Foster I. A Globus Primer or, Everything You Wanted to Know About Globus, but were Afraid to Ask Describing Globus Toolkit Version 4[Z]. (2005-08-05) http://globus.org/toolkit/docs/4.0/key/. 5/8/2005.

[3] The Globus Alliance[Z]. (2007-07-15). http://www.globus.org 15/7/2007.

[4] OASIS Consortium. Web Services Resource Framework-primer v1.2. Committee Draft 02[Z]. (2006-05-20). http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-02.pdf.

[5] Commons DBCP[Z]. (2007-07-15). http://jakarta.apache.org/commons/dbcp/15/7/2007.

（上接第 17 页）

参考文献

[1] Ali Y D B. Complex Concurrent Engineering and the Design Structure Matrix Method[J]. Concurrent Engineering Research and Applications, 2003, 11(3): 165-177.

[2] Eppinger S D, Nukala D, Whitney M. Generalized Models of Design Iteration Using Signal Flow Graphs[J]. Research in Engineering Design, 1997, 9(2): 112-123.

[3] 张汉鹏, 邱菀华. 基于 DSM 的产品开发进度规划模型及应用[J]. 北京航空航天大学学报: 社会科学版, 2007, 20(1): 22-24.

[4] Cooper D. Heuristics for Scheduling Resource-constrained Projects: an Experimental Investigation[J]. Management Science, 1976, 22(11): 1186-1194.

[5] Browning T R, Eppinger S D. Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development[J]. IEEE Transactions on Engineering Management, 2002, 49(4): 428-443.