Research and Implementation of Service-Oriented Grid Workflow System

Zenggang XIONG[1,2], Xuemin ZHANG[1], Wencai GUO [2]

1. School of Computer and Information, Xiaogan University, Xiaogan, China

2. School of Information Engineering, University of Science and Technology Beijing, Beijing, China

Email: gavinwill2008@gmail.com

Abstract: Based on the analysis of the characteristics of grid workflow, this paper firstly presented the concept of grid service workflow, and then described a grid service workflow system architecture, in which the specific functions of each module was discussed in detail. Finally, a Grid Workflow Language (GSWL) was proposed and a grid service workflow application was provided to illustrate our approach.

Keywords: grid computing; workflow; workflow language

面向服务的网格工作流系统设计与实现

熊曾刚 [1,2]，张学敏 [1]，郭文彩 [2]

1. 孝感学院计算机与信息科学学院，孝感，中国，432000

2. 北京科技大学信息工程学院，北京，中国，100086

Email: gavinwill2008@gmail.com

摘　要：在分析网格工作流特点的基础上,本文首先提出网格服务工作流的概念，给出一个网格服务工作流系统的组成架构，详细讨论架构中每一模块的具体功能。最后，提出了一种网格工作流语言（GSWL），并给出了一个网格服务工作流应用实例。

关键词：网格计算；工作流；工作流语言

1 引言

随着现代高科技的发展，以网格为基础的科学活动环境成为当前国际计算机技术研究的热点领域，网格计算是目前网络计算、分布式计算和高性能计算技术研究的热点，而网格工作流属于网格的一个新的研究领域，与网格中的其他技术相比，网格工作流方面的研究相对较少，现有的工作流产品和原型系统的设计主要面向普通的办公自动化应用，不是面向复杂的网格应用。

网格工作流系统大都是将传统的工作流系统直接移植到网格环境中[1]，一方面这些系统或模型没有充分考虑网格的特点和发挥网格的优点；另一方面，这些网格工作流模型均与特定的业务系统绑定，不能作为一个普适性的参考模型。本文针对网格工作流研究中存在的问题，结合工作流管理联合组织(WfMC)的工作流参考模型和其他一些模型[2,3]的基础上，给出了一个网格服务工作流系统的组成架构，并且提出了一种网格工作流语言（GSWL），最后针对这种架构给出了一个网格工作流应用实例。

资助信息：湖北省教育厅重点项目（No.2010）;湖北省十一五规划课题(2009B105); 孝感学院自然科学基金(Z2009017, Z2010016)

2 网格服务工作流的定义

定义 1 网格服务工作流（GSW）

网格工作流 W 由过程单元 P 和过程单元间变迁关系 $Trans \subseteq P \times P$ 组成，其中，过程单元包括 or-汇聚 O_j 、and-汇聚 A_j 、or-分支 O_s、and-分支 A_s 节点以及由网格服务集 S 提供的活动集 A 组成，若存在双射 $f : A \leftrightarrow S$ ，则称 W 为网格服务工作流，简称为 GSW。

在网格服务工作流 W 中，or-分支的输出变迁由函数 $\Pr ed : Trans \cap (O_s \times P) \to \Pr edicate$ 确定，网格活动使用偏序函数 $Name : A \mathbf{f} Name$ 进行命名，无名活动定义为空活动，and-汇聚和 or-汇聚的出度最大为 1，and-分支和 or-分支的入度最大为 1，所有的网

978-1-935068-10-5

格活动的入度和出度最大为 1，在 GSW 中，入度为 0 的活动称为初始活动 $(I \subseteq P)$ ，出度为 0 的活动则称为终止活动 $(F \subseteq P)$ 。

定义 1 表明，网格服务工作流是网格工作流的一个特例，即组成网格工作流的任一活动是由网格服务集合中的唯一一种服务提供，而任意服务仅能提供一种活动，从而简化网格服务间的合作关系，便于分析和解决网格工作流应用问题。

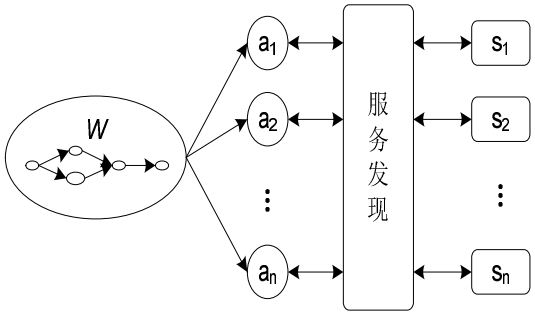

Figure 1. The mapping between grid activities and grid services
图 1. 网格活动与网格服务间的映射

图 1 表明组成网格服务工作流 W 的网格活动集 $\{a_i\}_{i=1...n}$ 与网格服务集 $\{s_i\}_{i=1...n}$ 之间的映射关系，这种映射关系的建立是通过服务发现机制来实现的。

3 网格服务工作流系统的架构

网格服务运行于广域、多组织的计算环境中，其中每个计算节点都有其管理策略和控制机制，这些节点包括多处理器计算机、工作站集群、或 PC 机，网格服务工作流系统则是根据服务节点的功能和特性，为完成某种应用，实现不同服务间的整合，并按照服务工作流描述规范顺序执行。

如图 2 所示，网格服务工作流系统分为区域级和网格级两层管理系统，前者针对区域内的网格应用，使用统一的服务查询方式，易于实现网格服务组合，并采用集中式工作流调度架构，调度的主要目的是提高网格资源的利用率，完成域内服务与域内资源间的映射；而网格级服务工作流管理系统则用于满足跨组织（域）的网格工作流应用的需求，网格资源存在更大的异构性，需实现跨域的网格服务组合，采用混合式的工作流调度架构，调度的主要目的是把用户的任务序列映射到多个域内的不同的服务节点上，以满足用户的服务质量要求，因此，网格级工作流调度器需要与区域调度器协作才能完成工作流任务。

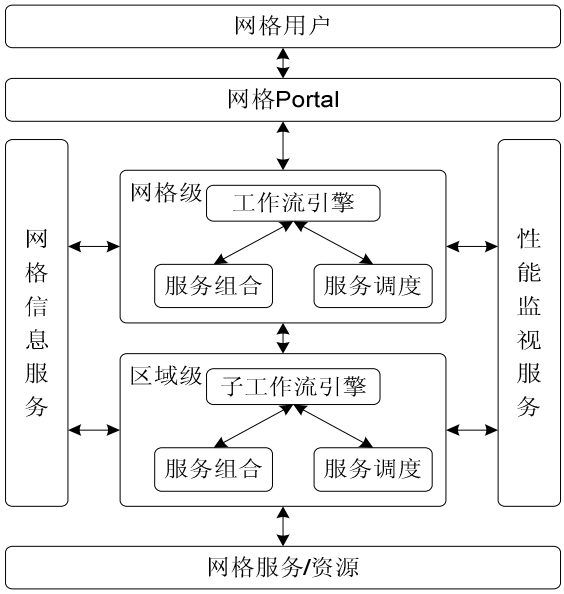

Figure 2. Grid service workflow system architecture
图 2. 网格服务工作流系统架构

在网格服务工作流系统中，用户通过网格 portal 提交一个网格工作流作业，工作流引擎使用工作流描述语言定义工作流过程模型，确定构成工作流的网格活动，然后利用网格信息服务获取可用服务信息，并使用服务组合模块检查和验证网格服务间的协作关系，服务工作流通过验证后，工作流引擎则通过服务调度模块选择提供服务的节点，开始执行网格工作流应用。

组成网格服务工作流系统的各模块功能描述如下：

1）工作流引擎

工作流引擎的主要功能是根据网格用户的应用请求与业务过程参数创建工作流实例，调用服务组合模块生成网格服务执行序列，并通过服务调度模块选择实现服务的最佳节点，激活并监视工作流实例的执行，工作流引擎是网格服务工作流系统的核心。

2）服务组合

服务组合模块用于检查不同的网格服务间所提供的接口描述、服务功能、消息传递等属性是否匹配，根据工作流过程模型生成松耦合的服务工作流，并避免服务死锁的出现；为满足工作流应用动态变化的要

求，该模块应能够实现服务的动态组合；服务组合模块是通过服务协调器实现的。

3）服务调度

在网格环境中，由于每种服务都有可能由多个节点（资源）提供，因此需要解决服务与资源间的映射关系，即网格服务调度问题。在网格服务工作流系统中，服务调度模块采用遗传算法解决该问题，并且根据性能监视服务实现动态调度。

4）网格信息服务

网格信息服务模块为网格服务工作流系统提供可用的服务信息，具有服务查询、注册、销毁等功能。

5） 性能监视服务

性能监视服务用于监视远程或本地站点的服务实例执行状态，并将状态变化信息报告给工作流引擎，引擎据此确定服务实例的最终状态，为网格服务的动态调度提供依据。

4 网格服务工作流语言(GSWL)

目前网格工作流多采用 XML 作为描述语言，但是仍缺乏统一的描述规范，而基于网格服务的网格工作流描述的文献也比较少。由 S. Krishnan[4]等人提出的 GSFL 给出一种网格服务工作流描述机制，能较好的反映出网格服务、网格活动、服务提供者之间的协作关系，但是缺乏动态的描述机制和服务组合检查机制，因此，有可能产生服务死锁或服务失配的问题。本文借鉴 GSFL 的某些特点，提出一种可扩展的网格服务工作流语言—GSWL。

与 Web 服务技术相似，网格工作流规范应该允许每个服务导出组成工作流的活动，并且导出的活动能够激活其它的活动链，Web 服务的 WSFL 规范有效地实现了这一要求，因此，在 GSWL 中也包括了这部分功能，此外，采取这种方式导出的活动也应该能够描述成服务，即 GSWL 规范应该足以描述工作流使得服务协调器根据规范能够自动生成相应的 WSDL。

尽管从理论上讲，在 WSFL 规范中，通过使用 plugLinks 能够实现 Web 服务间的对等交互，但是 WSDL1.1 并没有定义请求-响应操作和通知操作，使得这种交互难以实现。因此，Web 服务工作流的服务间交互必须通过工作流引擎来实现，而工作流引擎的使用造成服务间通信延时，从而成为 B2B 应用的瓶颈。对于网格服务，服务间传递的数据量更大，因此有必要实现服务间的直接通信，如图 3 所示。

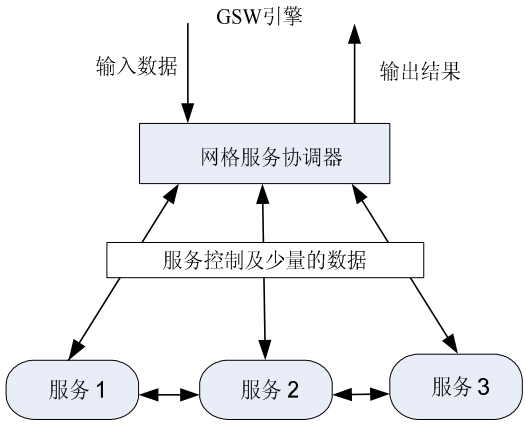

Figure 3. Grid service coordinator

图 3 网格服务协调器

在 OGSA 规范中，使用 notificationSource 和 notificationSinks 接口实现服务间的直接交互，因此，在 GSWL 中提供了连接通知源服务和通知目标服务间的实现机制，从而避免了 WSFL 中的工作流引擎的频繁参与。

下面介绍一下网格服务工作流语言(GSWL)组成。

网格服务工作流语言（GSWL）是一种基于 XML 语言和 OGSA 框架的网格服务工作流描述规范，其模式采用 XML schema 定义，它的简化结构如图 4 所示，主要包括：服务提供者, 网格活动, 服务组合, 生命周期。

1）服务提供者

服务提供者列表定义了组成工作流的所有的网格服务，列表使用 GSWL 文档定义服务提供者的名称、服务类型等属性，通过定位器（locator）元素实现服务提供者的查找与定位。指向正在运行的服务的 URL 可以静态的定位网格服务，对于通过 Factory 接口创建的临时服务，可以通过 Registry 服务查找它的服务句柄，以实现临时网格服务的定位。

2) 网格活动

网格活动列出了服务提供者提供的形成工作流的所有操作，该模型包含一个活动列表，表中的每个活动包含一个用于身份识别的名称和活动源，活动源是 Web 服务中 endPointType 元素定义的操作引用，endPointType 元素包括操作的名称、portName 及提供该操作的 serviceName。

3) 服务组合

服务组合 compositionModel 定义了不同的网格服

978-1-935068-10-5 © 2010 SciRes.

务如何形成一个新的网格服务，描述不同服务的操作之间的控制流和数据流如何实现，并定义了服务间的直接的对等通信，以及服务组合验证机制，该模型包括导出服务、消息传递和组合验证。

4) 导出服务

导出服务 exportModel 包含了组成工作流过程的所有导出活动列表，任意用户使用标准机制都可以激活工作流实例的这些操作，由于工作流实例也可以看作一个标准的网格服务，因此它也可以用作其它网格服务过程的一部分而递归使用，对于导出的每一个活动，它的控制流和数据流都可描述成相应的控制模型（contrlolModel）和数据模型（dataModel）。

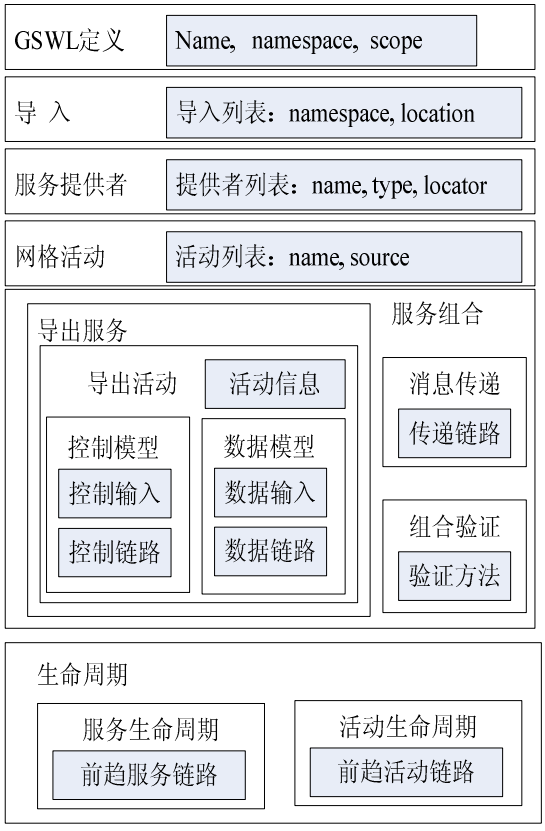

Figure 4. GSWL architecture
图 4. GSWL 的体系结构

导出活动的控制模型描述了该活动触发后激活的活动链，每个控制模型元素包含一个属性 controlIn，这个属性是导出活动触发后将执行的第一个活动的引用，每个控制模型也包括一系列的控制链路（controlLink），这个属性包括了导出活动所有前趋活动的列表。

导出活动的数据模型描述了活动触发后的数据流，每个数据模型包括一个 dataInTo 属性，该属性表明该活动将接受输入数据，而数据模型的 dataOutFrom 属性表明该活动将向调用者返回数据。

GSWL 文档使用 dataModel 和 controlModel 元素足以动态生成导出活动的 WSDL，并支持这些导出活动的动态激活。

5) 消息传递

消息传递 notificationModel 解决了工作流引擎与活动间的频繁协商的问题，如前所述，网格服务采用 notificaitionSources 和 notificationSinks 实现服务间通信，notificationSources 向目的服务建立链路以提供消息源，而 notificationSinks 则建立了从源服务获取消息的链路，这样服务间可以大量的交换数据而不需要与工作流引擎进行交互，用户也可以使用控制模型和数据模型传递控制消息和少量的数据。

6) 组合验证

为保证构成 GSW 的各网格服务间能成功组合，在服务工作流运行之前，需要对其组合机制进行检查，避免服务失配、服务死锁等问题。在 GSWL 中，使用 verificationModel 描述了 GSW 实例的过程模型，其中过程定义为 process，网格活动定义为 activity，顺序组合表示为 sequence 等，然后调用相关验证方法对 GSW 实例进行验证。

7) 生命周期

生命周期 lifecycleModel 用于解决服务和活动执行顺序的问题，网格服务的生命周期 serviceLifecycleModel 包括了优先执行服务的列表，因此，在初始阶段不需要启动所有的服务实例，只有在所有的前驱服务结束时才执行当前的服务。

生命周期使用工作流的 scope 属性，该属性值可以是 session(会话)或 application(应用)，若值为会话，则调用之间的状态信息不保留在工作流引擎内，所有的引擎调用操作都是合法的，使用 serviceLifecycleModel 进行调用实现服务实例化，对这些服务进行调用时它们处于激活状态。若 scope 的值为 application 时，则调用之间的状态信息保留在工作流引擎内，通过使用 serviceLifecycleModel，每个工作流只实例化一次服务，由于实现这些活动的服务可能并未处于激活状态，使得工作流引擎的调用有可能不合法，因此，加入一个 activityLifecycleModel 元素，用于描

述活动激活的顺序，即一些活动只有在其他活动触发后才能激活，例如，在线购物系统中的结账操作只有在一个或多个购买活动结束后才能启动，使用 activityLifecycleModel 属性可以保证按照正确的顺序调用服务时，它们能够处于激活状态。

5 网格服务工作流应用实例

下面给出一个网格服务工作流的例子，一个图像处理工作流应用由 ImageRead、ToGrey 和 ImageView 三个服务组成，ImageRead 服务用于提供原始图像数据，ToGrey 服务用于彩色图像转为灰度图像，ImageView 服务提供图像的浏览服务，这三个服务间数据传输、消息传递以及服务控制的关系如图 5 所示。

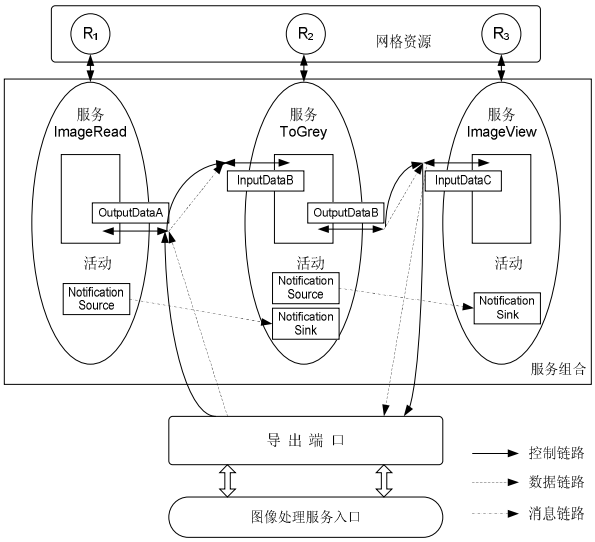

Figure 5. Grid workflow application example
图 5. 网格工作流应用实例

图 5 所示的工作流应用从图像处理服务入口接受用户请求，从而启动一个 GSW 应用实例，ImageRead 服务接受用户提供的原始图像数据并存储至 R1 中的缓冲区内，根据工作流引擎生成的 GSWL 规范，将这些数据传递给 ToGrey 服务，同时通过 NotificationSource 接口向 ToGrey 服务发送处理请求等消息，ToGrey 服务利用 NotificationSink 接口接收这些消息。ToGrey 服务利用网格资源集 R2 对接受到的图像数据进行灰度化处理，处理后的结果将被 ImageView 服务用于图像显示。

上述网格服务工作流应用实例的具体描述如下：

```
<definitions name="GSWLforImagProc" scope="session"
targetNameSpace="http://grid.ustb.edu.cn/gswl">
<!--List of Service Providers -->
<serviceProvider name="ImageRead"
        type="ImageReadType">
<serviceProvider name="ToGrey" type="ToGreyType">
<serviceProvider name="ImageView"
        type="ImageViewType">
<!--List of activities -->
<activityModel>
<activity name="OutputDataA">
<activity name="InputDataB">
<activity name="OutputDataB">
<activity name="InputDataC">
</activityModel>
<!-- The compostion model-->
<compositionModel>
<exportModel>
<exportedActivity>
<exportedActivityInfo name="FlowOfABC" port-
Type="exportType"/>
<controlModel controlIn="OutputDataA>
<controlLink label="link0" source="OutputDataA" tar-
get="InputDataB"/>
<controlLink label="link1" source="OutputDataB" tar-
get="InputDataC"/>
</controlModel>
<dataModel dataInTo="OutputDataA" dataOut-
From="InputDataC"/>
<dataLink label="link0" source="OutputDataA"
sink="InputDataB"/>
<dataLink label="link1" source="OutputDataB"
sink="InputDataC"/>
</dataModel>
</exportedActivity>
</exportModel>
<notificationModel>
```

<notificationLink label="link0" source="ImageRead" sink="ToGrey" topic="AtoB"/>

<notificationLink label="link1" source="ToGrey" sink="ImageView" topic="BtoC"/>

</notificationModel>

<verificationModel>

<process>

<sequence>

<activity name="OutputDataA" operation="Send">

<activity name="InputDataB" operation="Receive">

<activity name="OutputDataB" operation="Send">

<activity name="InputDataC" operation="Receive">

</sequence>

</process>

</verificationModel>

</compositionModel>

<!-- Lifecycle for the services -->

<lifecycleModel>

</definitions>

6 结束语

针对网格工作流研究面临的问题，结合已有的网格服务和 Web 服务的研究成果，本文首先提出网格服务工作流的概念，然后给出一个网格服务工作流系统的组成架构，详细讨论架构中每一模块的具体功能。最后，为能很好的描述网格服务工作流，本文在 GSFL 语言基础上进行改进并提出了一种网格工作流语言（GSWL），研究了该语言的组成与使用，并给出了一个网格工作流应用实例。

References (参考文献)

[1] Wil van der Aalst, Kees van Hee. Workflow Management Models, Methods, and Systems. The MIT Press.Mar.2004：210-250.

[2] Amin K, Laszewski G V, Hategan M. *et al.* GridAnt: a client controllable grid workflow system. In: Proceedings of the 37th Hawaii International Conference on System Sciences. IEEE Computer Society, 2004:1-10.

[3] Chen Jinjun, Yang Yun. Key research issues in grid workflow verification and validation. In: Proceedings of 4th austrolation Symposium on Grid Computing and e-Research. Australian computer society, 2006:97-104.

[4] Patrick Wagstrom, Sriram Krishnan and Gregor von Laszewski. GSFL: A Workflow Framework for Grid Services. In Supercomputing Conference (SC 2002)：23-34.