

网格 workflow 研究现状及存在问题^{*}

李维宏^{1,3} 张绍华² 祝精荟³

(复旦大学计算机科学与工程系 上海 200433)¹ (上海计算机软件技术开发中心 上海 200418)²

(浙江万里学院信息管理系 宁波 315100)³

摘要 网格计算成为当前业界研究的热点,网格 workflow 能够方便地构建、执行、管理和监控网格应用,使得网格应用能够自动、高效地实施。由于网格的动态性、分布性、异构性和自治性导致传统 workflow 的一些方法和技术不能有效地处理网格环境中的有关问题。本文首先比较了网格 workflow 与传统 workflow 的不同,指出了网格 workflow 研究的必要性,然后介绍了当前网格 workflow 研究的现状,归纳了三种网络 workflow 规范及四种网格 workflow 系统。最后,给出了网格 workflow 研究存在的问题及研究方向。

关键词 网格 workflow,组织资源模型,动态调度

Grid Workflow Research and Existing Problems

LI Wei-Hong^{1,3} ZHANG Shao-Hua² ZHU Jing-Hui¹

(The Department of Computer Science and Technology, Fudan University, Shanghai 200433)¹

(Shanghai Development Center of Computer Software Technology, Shanghai 200418)²

(The Department of Information Management, Wanli University, Ningbo 315100)³

Abstract Grid computing has become a hot point in the research area. Grid workflow can conveniently construct, execute, manage and monitor grid applications, and automate grid applications with great efficiency. Due to the dynamism, distribution, heterogeneity and autonomy of grid applications, conventional workflow technology can't effectively solve the relative problems of grid environment. In this paper, first we compare the difference between the grid workflow and the conventional workflow which illustrates that it is necessary to study the grid workflow. Then we summarize the specifications and systems of the grid workflow. At last we point the problems and directions of the grid workflow study.

Keywords Grid workflow, Organization and resource model, Dynamic scheduling

1 引言

随着互联网技术的迅猛发展,科学研究和相关应用借助网络环境获得了极大的进步,同时在互联网上形成了大量以计算机系统为主导的计算资源、存储资源、数据资源、信息资源、知识资源和仪器设备资源等^[1]。研究人员试图将这些大范围地理分布的异构计算机系统和资源整合在一起形成一个大规模的计算平台,其中每一台参与的计算机或者以一台计算机为中心的集合就是一个“节点”,而许多这样的“节点”便组成“网格”,网格正逐步成为一种新的技术和基础设施,可以充分利用集成的资源,形成一个大范围的计算池。利用网格的这种超级处理能力,能够解决以前一台高性能计算机无法解决或者不容易解决的大规模科学计算;利用网格中的共享资源,可以处理需要多种资源、多个系统和多门学科参与的重大应用研究问题;网格不但有助于解决物理、天文、生物、工程和模拟等领域的科学计算问题,同时在商业领域、社会生活和国民经济的各行各业方面具有潜在的重大影响。

网格(Grid)是借鉴电力网的概念提出来的,网格的最终目的是希望用户在使用网格计算能力时,就如同现在使用电力一样方便,为网格用户提供与地理位置无关、与具体计算设

施无关的通用计算能力。网格研究现在处于发展阶段,对于网格概念还没有达成一致, Ian Foster 和 Carl Kesselman 于 1998 年在网格的第一本专著《网格:一种新的计算基础设施蓝图》^[1]中定义如下:计算网格是一个包含硬件和软件的基础设施,它能为高端计算能力提供可靠的、一致的、普遍的和不昂贵的接入。Ian Foster 等人在 2001 年认为^[2]:网格关心的是在动态的、多机构的虚拟组织中协调资源共享和协同解决问题,核心思想是在一组参与节点(资源提供者 and 消费者)中协商资源共享管理的能力,利用协商得到的资源池共同解决一些问题。

研究者从不同的角度和侧重点,对于网格多方面的特点和功能进行了说明,网格是面向问题领域,不同问题的要求和解决方法不一样,因此呈现出各种各样的定义和观点,同时网格还处于发展阶段,有关的概念也在不断变化和发展之中的,但是从网格的起源和目的看,网格的本质是“资源共享与协同工作”。

工作流技术由于能够使过程自动化和协同工作,提高工作效率,因此在企业的业务流程重组(BPR)、电子政务、协同科学研究等领域受了广泛的重视。对于网格研究和应用,不但要充分利用网格来实现资源共享,而且要在这些资源的基

^{*} 基金项目:国家自然科学基金项目(No. 60473062);国家 863 项目(No. 2002AA114010)。李维宏 博士生,从事软件工程,工作流技术等研究。张绍华 博士,从事网格、工作流技术和嵌入式系统研究。祝精荟 讲师,从事工作流技术研究。

础上开展协同工作,使得网格应用能够有条不紊地自动或者半自动执行,尤其是当网格应用的逻辑过程比较复杂,具有各种时间和因果约束的时候,更需要有相应的服务和技术来管理这些应用,因此网格工作流也就随着网格研究的深入和应用的发展而得到了重视。

2 网格工作流及其研究的必要性

工作流是一类能够完全或者部分自动执行的经营过程,它根据一系列过程规则,文档、信息或任务能够在不同的执行者之间进行传递与执行。国际工作流管理联盟 WfMC 将其定义为“工作流是为了实现某些标准或业务目标而进行的自动过程。在这些过程中文件、信息或任务根据标准或目标的要求在参加者之间传输。大多数工作流都由计算机化的信息系统来支持过程的自动执行。”

由此可以看到工作流是一个过程,这个过程由许多彼此相关的工作步(活动)组成,来完成特定的业务处理目标。在这个过程中涉及到人、工具和信息。这些资源按照一定的规则在分布的环境中流转。在工作流管理系统中有四个基本要素:工作流模型、工作流运行支持机制、人和外部程序。工作流模型定义了工作流运行的规则,工作流运行支持机制根据规则合理地在人和机器之间调度资源共同完成业务处理的目标。

2.1 网格应用需要网格工作流

网格研究和应用来源于元计算(Metacomputing)^[5],最初的应用处理模式相对比较简单,例如将几个大的超级计算节点联合起来,得到更强的处理能力或者存储能力,共同处理某项科学计算任务;或者类似 SETI@HOME 项目^[8],将任务划分为许多小的任务,每个任务处理自己的数据和计算,任务之间的关联较少,任务之间没有复杂的过程,尽管整个任务庞大,需要的计算存储资源多,数据量大,但是整个任务的逻辑过程相对比较简单,控制比较容易。因此随着应用的深入和推广,出现了大量的其它领域的网格应用,例如协同工作,具有过程复杂,涉及到多个步骤、资源和过程,不但需要大量的计算资源和其它资源,而且任务的过程也比较复杂,包括很多时间、空间和资源方面的约束条件,如果利用一般的处理方法,不但效率低下,而且导致某些应用无法完成。需要由网格工作流来对网格应用进行构建、执行调度、管理监控,使得网格应用能够自动化而且效率更高。

面对网格研究和应用对网格工作流的强烈需求,一些国际组织和研究机构提出了一些关于网格工作流的规范建议例如:GSFL(Grid Service Flow Language)^[9]、GGF(Global Grid Forum)^[10]的 Grid Workflow^[11]和 GWA(Grid Workflow Architecture)^[12]、OGSA 中的 Grid Workflow Services^[13];同时很多网格项目也采用工作流和具有工作流特征的服务来管理网格中的应用,提高网格应用的执行和管理效率。

网格的动态性、分布性、异构性和自治性导致了传统工作流的一些方法和技术不能有效地处理网格环境中的一些问题。虽然网格工作流与传统工作流中一样包含四个基本要素:工作流模型、工作流运行支持机制、人和外部程序,但是与传统工作流相比主要有以下不同:基于虚拟组织的工作方式,虚拟组织是松耦合的,包括多个管理领域和组织,缺乏集中的控制;基于网格服务的应用方式,根据 OGSA 规范,采用一致的网格服务方式来处理各种应用,有效地解决异构性、分布性和自治性等;网格资源的动态性,导致网格的计算和处理能力

随时间变化,应用性能估计困难;计算型任务多,用户型任务相对较少,网格应用很多是利用网格的计算资源分布并行运行,但是由于网格的动态性,因此并行任务的执行效率存在问题。

因此网格工作流不仅仅是传统工作流技术在网格环境中一个应用,而是有大量的基本问题需要重新考虑,尤其是针对网格的资源动态性特点和网格应用的特点,对网格工作流的过程建模问题,组织资源问题,调度算法问题都需要重新研究和考虑。

2.2 网格环境需要对工作流进行重新研究

围绕着工作流的过程建模、组织建模、资源建模、执行调度、监控管理、模拟分析、互操作、柔性适应性,工作流研究取得了许多研究成果,但是正如前面所述,网格特点使得传统工作流不能够在网格环境很好的工作,或者根本不适用。

从总体架构上,网格工作流实际上同样包含前面所述的功能模型,即网格工作流的建模功能,网格工作流运行阶段的流程控制功能,网格工作流运行阶段不同子任务的用户或 IT 工具的交互功能。从网格工作流管理系统上同样需要网格工作流流程定义、网格工作流引擎、网格工作流控制及相关应用数据、网格工作流用户接口。但是还有许多需要重新考虑的问题:

- 从宏观的系统架构看,网格工作流还需要有资源/服务管理,虽然网格工作流可以利用网格的资源/服务管理,但是还需要网格工作流能够自己有效地调度执行、管理监控网格的资源/服务;从系统架构的层次看,网格工作流的体系结构与网格具有相适应的层次属性,一个大的工作流系统可能由很多相互之间具有关联的子工作流组织,因此工作流的系统架构方面具有很强的开放性和可扩展性。

- 过程模型:网格环境的分布性动态性导致网格应用过程具有一定的动态性,因此网格应用的过程模型需要具有一定的适应性,能够根据网格资源的动态变化作出一定的反应,这样才能高效地完成工作。

- 组织资源模型:传统工作流的应用环境一般都在一个组织内部或者跨越几个组织,组织内部及组织和组织之间可以集中控制,因此对于相应的组织资源模型能够进行统一的建模和处理,但是在网格环境中,组织和网格都是异构分布和动态的,虽然网格工作流可以利用网格提供的组织资源管理,但是网格工作流同样需要相应的组织资源机制来保障网格应用的顺利执行。

- 工作流调度:传统的工作流调度算法一般在组织内部执行,有关组织内部的资源实时信息能够集中获取,调度比较容易顺利实施,但是网格环境可能需要处理动态资源信息、复杂组织、多种策略和选择等问题。

3 网格工作流研究现状

网格工作流方面的研究主要包括两个方面的内容,一是有关研究组织和联盟提出的关于网格工作流的建议和规范,二是一些实际的网格项目和系统采用网格工作流或者具有工作流特征的服务来构建和管理复杂网格应用。

3.1 网格工作流规范

3.1.1 GSFL(Grid Service Flow Language) GSFL^[14]分析了 WSFL(Web Service Flow Language)技术,力图利用它们结合网格服务来解决网格服务流程;GSFL 利用已有的网格服务集成新的网格服务,或者成为其它网格服务中的组

成部分,描述网格的执行顺序和网格服务之间的交互。采用 WSFL 的思想,描述 Web 服务的流程语言,采用流程模型和全局模型。流程模型指定了行为的执行顺序和行为间的数据交换。WSFL 全局模型将 Web 服务操作绑定到由聚集服务定义的操作,包括导出的操作和生命周期操作。网格服务自身的特点,采用 Web 服务的流程方法不能很好地解决点对点的通信问题,在 WSFL 中流程引擎作为桥梁与每一个 Web 服务通信形成 Web 服务的通信,在 Web 服务中通信量不大,没有什么影响,但是在网格环境中,有可能通信量很大,Web 服务引擎将成为通信的瓶颈。因此在 GSFL 中要引入点对点的通信,使单个网格服务能够在网格服务引擎协同下直接通信,减少网格服务与引擎之间的通信量。

由于 Globus 和 OGSA 已经成为事实上的标准,网格服务之间的通讯是 notificationSources 和 notificationSinks,可以实现异步信息的发布,网格服务实例的生成和定位是 registers 和 factories 服务。

GSFL 是基于 XML 在 OGSA 框架下描述 workflow 规范的语言,采用 XML Schemas 定义。整个框架如图 1 所示。

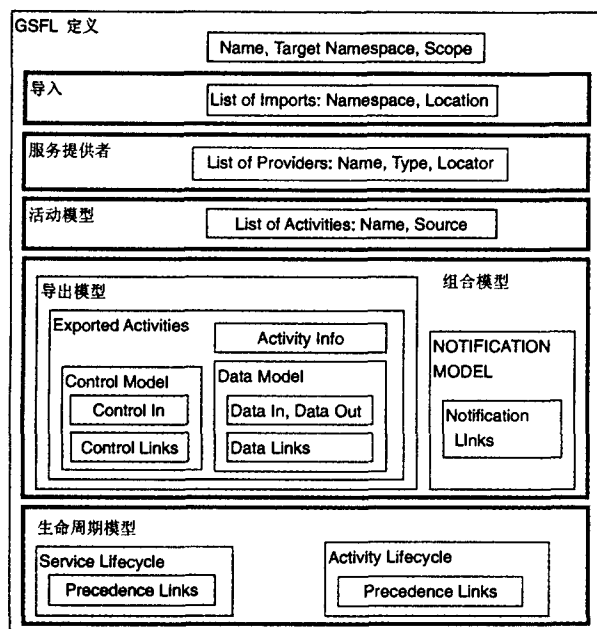


图 1 GSFL 框架

3.1.2 Grid Workflow GGF(Global Grid Forum)提出了关于 workflow 的推荐规范《Grid Workflow》^[15],针对网格环境下的特征,采用 XML 定义了 workflow 当中的属性和内容;利用 XML 的规范性、工具支持、环境无关性、支持多种字符集和易读性,在动态异构、分布自主的环境中用 XML 来定义网格 workflow。对于网格 workflow 中的词汇进行了定义,详细如下:

- 控制流(Flow of control):任务执行的顺序。
- 网格输出变量:采用@variable name@形式来定义网格过程中的输出变量。
- 工作流:标识一个工作流,包含属性 id 和 content。
- 计算:计算任务:包含属性 input, id 和 content。
- 资源查询:查询网格活动所需的资源,包含属性 id 和 content。
- 数据传输:在网格环境中进行数据传输。
- 重新启动:根据一系列条件启动一个活动。
- 分支任务:采用分支的方法来执行一个大的网格任务。

- 软件请求:完成任务需要的软件。
- 描述:网格活动的描述。
- 参数:指定应用实例的命令行参数。
- 资源请求:资源约束。
- 资源管理者请求:指定本地的资源管理器。
- 计算属性:为“计算”和“分支任务”指定参数。
- 环境参数:指定远程过程的环境变量。
- 客户端钩子(ClientHook):指导 workflow 引擎返回给客户端的控制。
- 监控:指示 workflow 引擎监督网格活动的状态。

3.1.3 GWA(Grid Workflow Architecture) GCE(Grid Computing Environment)和 GSM(Grid Service Management)研究小组提出了一个网格 workflow 管理系统的体系结构^[16],指出了网格 workflow 的生命周期包含 workflow 过程描述和验证,workflow 实例的执行。GWA 的目标是:①定义了一个网格 workflow 公开架构,与 GCF 和 WFMC 一致,采用已经存在的标准和实践来实现;②确定了在 GCF 和其它组织的工作流管理系统的公共特征和区别;③确定了一套研究网格 workflow 管理的公共问题。

GWA 严格限制讨论架构和机制而不是实现。

GWA 确定了网格 workflow 生命周期的里程碑:workflow 过程描述(WPD)的创建,WPD 的验证,workflow 实例描述(WCD)创建,实例的执行。GWA 的体系结构和各部分的关系如图 2 所示。

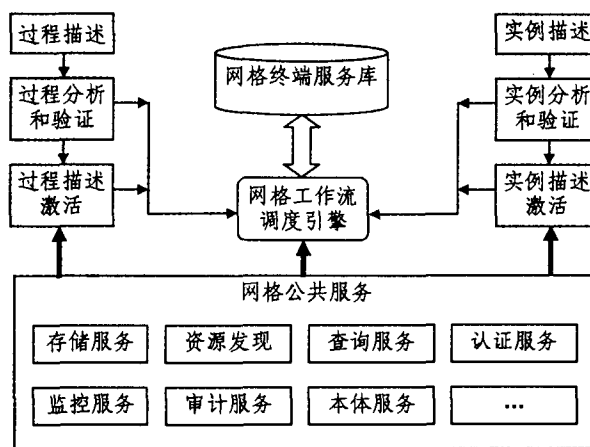


图 2 网格 workflow 体系结构

3.2 网格 workflow 系统

一些项目采用网格 workflow 作为整个网格应用中的一个服务,使得网格应用的构建更加方便和高效;另外网格 workflow 是一个单独的系统,作为网格应用的一个支持平台。

3.2.1 GridFlow GridFlow^[17]包含全局网格 workflow 管理系统和局部网格子 workflow 系统。其中全局层提供模拟、执行和监控功能,每个局部网格处理子 workflow 的调度和冲突。提出的网格计算的 workflow 管理系统 GridFlow 包括一个用户门户和全局网格 workflow 管理系统服务、局部网格子 workflow 调度服务。模拟、执行和监控 workflow 在全局网格层提供,在基于代理的网格资源管理系统层上工作。在每个局部网格,子 workflow 调度和冲突管理在任务调度系统的性能预测上处理,模糊时间技术用来解决跨领域和高动态的网格环境中的 workflow 挑战。局部和全局网格 workflow 管理能够协调 workflow,优化 workflow 执行时间和解决利益冲突。

3.2.2 GALE GALE^[6]是美国能源部网格项目 ASCI^[7]中研究的工作流系统,主要对网格中计算和数据传输等工作进行管理,但是流程比较简单,没有考虑到工作流过程中任务的分解和合并、条件和循环执行,缺乏对复杂工作流的支持。

3.2.3 McRunjob McRunjob 是在高能物理中进行大量数据处理的工作流管理系统(图 3),包含提交任务的核心元数据,核心元数据描述语言包括大量的方法将元数据转换成持久的表、工作描述、步骤控制和数据源信息,相当于工作流的建模功能。

McRunjob 是网格工作流管理器,是管理大量处理业务工作的产生器,它转换核心元数据将工作流提交到各种环境中,强大的核心元数据描述语言转换元数据成为表单和工作描述,多步工作流和数据来源信息。该语言允许结构化元数据中包含丰富的表达式、命名空间、函数依赖和本体的定义。对大型的并行化工作也有简单的控制结构,采用模块还设计允许对新的应用任务进行新的工作描述扩充。

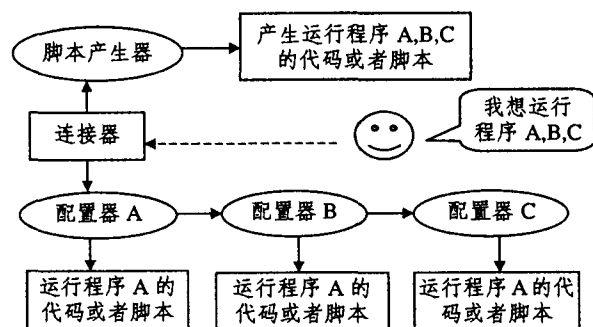


图 3 McRunjob 工作流程

3.2.4 Pegasus 在 PhyGridN 中通过 Chimera 和 Pegasus 产生和管理工作流,从而获得相应的数据,其中 Chimera 产生一个抽象的工作流(AW),Pegasus 是一个配置系统,在网格中映射和执行工作流,可以将两者分别看作工作流建模机制和工作流调度执行机制。

Chimera 和 Pegasus 是项目 GriPhyN 虚拟数据系统(Virtual Data System)的一部分,能够实现高效及时的数据导出。允许用户和应用根据抽象工作流描述数据产品,并且在网格中执行哪些工作流。不同于其它的工作流映射系统,Pegasus 可以容易利用已经存在的中间步骤数据。

利用 Chimera 虚拟数据语言 VDL (Virtual Data language),用户描述数据的转换和导出。数据的转换是应用到数据上的一般描述,导出是数据转换在特定数据集上的实例操作。转换列举出输入输出参数文件,导出实际的文件填充参数,从而产生希望的文件结构;数据的转换是一个程序和参数的模板,能构造希望的工作流。当用户和应用要求一个数据结果或者数据文件,Chimera 便产生一个抽象的工作流,只是根据描述逻辑文件名称和转换描述所需数据,而没有具体的资源来执行工作流。

Pegasus 是一个配置系统,在网格中映射和执行工作流,具体是 Pegasus 从 Chimera 接受抽象工作流描述,产生一个具体工作流(CW),然后提交给 Condor-G/DAG 执行。

工作流采用有向无环图 DAG(Directly Acyclic Graphs)来表示,抽象工作流 AW 根据逻辑名称描述转换和数据,有关执行实例需求的数据信息,Pegasus 请求管理器发送工作

流给具体的工作流产生器,为了在网格环境中定位输入数据,Pegasus 采用 RLS (Globus Replica Location Service),输入文件由 DAG 中的逻辑文件名称指定,采用 RLS,Pegasus 为这些文件发现具体的物理位置列表。其中某些获取数据的信息可以用来优化具体的工作流,优化由抽象 DAG Reduction Component 来完成,如果在 AW 中描述的某些数据存在,则可以直接利用它们来减少 CW 的复杂性。

4 网格工作流研究的不足及方向

网格的动态性、分布性、异构性和自治性导致传统工作流的一些方法和技术不能处理网格环境中的相关问题,虽然网格工作流的研究取得了一些初步成果,但在网格工作流的体系结构,网格工作流的过程模型、组织资源模型和动态调度算法等需要进行深入的研究,具体地说网格工作流的研究还存在以下不足:

4.1 过程模型问题

以前的网格应用主要是针对大数据量处理或者科学计算,涉及到关联应用相对较少,网格工作流中没有提供较强的建模和验证工具,通常采用基于有向无环图 DAG(Directed Acyclic Graph)的建模工具,缺乏形式化和精确的工作流描述语言。随着网格应用的复杂化,在网格工作流中同样需要有较强的建模和验证功能,如果在流程中存在一些错误,则会造成资源的损失和浪费,并可能造成严重的后果。虽然 Petri Net 具有图形化、形式化和多种分析工具等优点,受到了重视,但是考虑到网格环境对于动态性的要求极高,而 Petri Net 对动态性支持不足,例如基于 Petri 网的工作流过程建模语言 WF-Net^[18]得到的过程模型缺乏动态性,不能根据具体的运行情况动态地产生任务,只能重新建模、调度执行,但是代价太大。

网格工作流的过程模型。需在 WF-Net 基础上增加动态元素,需要解决网格工作流过程模型中的规模问题、动态性问题和选择性问题,有助于网格工作流的过程建模和过程进化。

4.2 组织资源问题

目前网格工作流主要考虑计算型任务,基本没有考虑组织问题,对资源问题也只采用网格的资源管理方式。网格主要有两种资源管理方法。第一种是以 Globus 为代表的全局-本地两层资源管理方法。Globus 的资源模型^[21]主要由 4 部分组成:资源代理、并发分配器、资源信息服务组件以及资源分配管理器(简记 GRAM),另外 Globus 还提供了一种资源描述语言 RSL,用来在以上各个组件之间传递资源请求。另外一种是以 Condor-G^[22]为代表的资源池模型,在这种模型中,在一台中心服务器上记录计算环境中所有资源的信息,对这些信息进行集中式的管理,在为计算问题分配资源时,简单地从服务器的资源列表(资源池)中选择空闲资源来进行计算,资源池中对资源的描述是无序的,并没有对资源进行有效的结构化组织,因而在为计算问题分配计算资源时会比较盲目。

而网格工作流是工作在虚拟组织环境和资源当中,因此不能简单地使用这两种模型来进行网格工作流的资源管理。应该有进一步的网格工作流组织资源管理。

网格工作流的组织资源模型。要求解决网格工作流中基于虚拟组织的组织建模,定义虚拟组织中组织成员的各种关系;介绍了网格的资源管理模型,结合虚拟组织框架提出了网格工作流中集成的组织资源模型,给出了有关组织和资源中

的动态查询、动态加入、退出算法,有利于网格 workflow 引擎对于网格 workflow 任务的动态调度。创新的将网格组织和资源集成考虑,有利于两者的建模和管理。

4.3 调度问题

网格 workflow 中的调度问题是最重要的问题,不仅仅影响网格 workflow 执行成功与否和效率的高低,同时也涉及到网格 workflow 的资源管理情况,调度引擎中非常重要的功能是根据 workflow 模型生成相应的工作流实例任务,然后根据定义好的调度策略进行调度。当前网格调度中大部分调度策略和算法都是针对元任务的,即调度任务相互之间没有关联,显然不满足网格 workflow 中任务之间是否具有时序或者因果的约束条件,另外网格中考虑具有关联任务的调度采用基于 DAG 的调度方法,对算法缺乏优化,同时调度单一,没有提供多种策略,难以满足网格 workflow 应用的不同需求。

4.4 基于网格 workflow 的网格服务合成

单一的网格服务不一定能够满足各种网格应用需求,如果针对所有的应用都专门开发一个与之对应的网格服务可能代价很高,也是不现实的,因此基于网格 workflow 的网格服务合成能够为网格应用带来极大的方便。随着 WSRF^[21] 的推出,网格研究和 Web Service 研究正走向融合,现在网格服务合成很多采用 Web Service 合成方案,例如 BPELWS^[22]、WSC^[23],利用这些标准虽然能够较方便地定义企业内部、企业之间的业务流程是如何相互协作的,但是这些流程需要用户手工来明确定义,自动化程度不高,因此需要考虑如何利用网格 workflow 来方便网格服务的合成。

结论 网格还在飞速地发展变化之中,网格研究和 Web Services 正走向融合,WSRF 的规范即将颁布,从目前的草案看,WSRF 对于网格应用会带来很大的影响,对于网格 workflow 的相关技术的影响和调整也是业界需重点关注的问题,具体有网格应用的模式会产生什么样的变化,导致网格 workflow 的过程模型会有什么的变化,在 WSRF 框架下,网格的组织和管理如何体现等问题。

参考文献

- 1 Foster I, Kesselman C. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publisher, 1999
- 2 Kesselman F C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Intl. J. of Supercomputer Application, 2001, 15(3): 200~222
- 3 Foster I. What is the Grid? A Three Point Checklist. GRIDTo-day, July 20, 2002
- 4 徐志伟. 织女星网格总体思路. 织女星网格文档, VGD-1, 2001
- 5 Messina P. Metacomputing and Data-Intensive Applications. In: Proc. of the Intl. Conf. on Worldwide Computing and Its Applications, March 1997
- 6 Bivens H P, Beiriger J I. GALE: Grid Access Language for HPC Environments. <http://sass3186.sandia.gov/hpbiven>, 2002
- 7 Beiriger J, et al. Constructing The ASCI Grid. In: Proc. of 9th High Performance and Distributed Computing Conf. Pittsburgh, PA, August 2000
- 8 SETI@HOME. <http://setiathome.ssl.berkeley.edu>
- 9 Krishnan S, Wagstrom P, von Laszewski G. GSFL: A Workflow Framework for Grid Services. ANL/MCS-P980-0802, Argonne, 2002
- 10 <http://www.gridforum.org/>
- 11 Bivens H P. Grid Workflow. Global Grid Forum, 2001
- 12 Marinescu D C. A Grid workflow Management Architecture. GGF white paper
- 13 Foster I, Kesselman C, Nick J, et al. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, 2002
- 14 Krishnan S, Wagstrom P, von Laszewski G. GSFL: A Workflow Framework for Grid Services. ANL/MCS-P980-0802, Argonne, 2002
- 15 Marinescu D C. A Grid workflow Management Architecture. GGF white paper
- 16 Graham G E, Evans D, Bertram I, McRunjob. A High Energy Physics Workflow Planner for Grid. Computing in High Energy and Nuclear Physics, La Jolla, California, March 2003
- 17 Aalst J, van der W M P. The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers, 1998, 18(3): 21~66
- 18 张绍华. 网格 workflow 关键技术研究, [复旦大学博士论文], 2004. 6
- 19 Czajkowski K, Foster I, Karonis N, et al. A Resource Management Architecture for Metacomputing Systems. In 4th Workshop on Job Scheduling Strategies for Parallel Processing, Springer-Verlag, 1998. 62~82
- 20 Frey J, Tannenbaum T, Foster I, et al. Condor-G: A Computation Management Agent for Multi-Institutional Grids. Cluster Computing, 2002, 5(3): 237~246
- 21 WSRF, <http://www.globus.org/wsr/>
- 22 Arkin A, et al. Business Process Modeling Language (BPML), Version 1.0. 2002
- 23 Arkin A, Askary A, Fordin S, et al. Web Service Choreography Interface (WSC^I) 1.0. Standards proposal by BEA Systems, Intel, SAP, and Sun Microsystems, 2002
- 24 Beckert B, Hahnle R. Analytic tableaux. In Bible W, Schmitt. Automated Deduction: A basis for Application. Kluwer, 1998. 11~41
- 25 Bryant R E. Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers, 1986, C35: 677~691
- 26 Brace K S, Rudell R L, Bryant R E. Efficient implementation of a BDD package. In: Proc. 27th ACM/IEEE Design Automation Conference, Los Alamitos, 1990. 40~45
- 27 Burch J R, Clarke E M, Dill D L. Sequential circuit verification using symbolic model checking. In: Proc. 27th Design Automation Conf. (DAC'90), 1990. 46~51
- 28 刘全, 孙吉贵. 非经典逻辑的语义 tableau 方法. 计算机科学, 2002, 29(5): 72~75

(上接第4页)

- 20 Brown F M. Towards the automation of set theory and its logic. Artificial Intelligence, 1978, 10(3): 281~316
- 21 Manthey R, Bry F. SATCIMO: A theorem prover implemented in prolog. In: Lusk E, Overbeek R, eds. In: Proc. 9th Conf. on Automated Deduction. New York: Springer-Verlag, 1988. 415~434
- 22 Fujita H, Hasegawa R. A model generation theorem prover in KL1 using a ramifiedstack algorithm. In: Furukawa K, ed. Proc. 8th Intl. Conf. on Logic Programming. France: MIT Press, 1991. 535~548
- 23 Bry F, Torge S. A deduction method complete for refutation and finite satisfiability. In: Dix J, Del Cerro L F and Furbach U, ed. Proc. 6th European Workshop on Logics in AI (JELIA). Springer-