

数据网格中的数据复制技术研究^{*}

孙海燕 王晓东 肖 依 贾 焰 周 斌 邹 鹏

(国防科技大学计算机学院 长沙 410073)

摘 要 数据复制技术是数据网格系统中广泛采用的改善网格系统性能的关键技术之一。与传统分布式系统应用领域的复制相比,数据网格中的复制技术在复制目标、复制粒度、复制关键技术等方面表现出独特性质。本文将数据复制技术概括为副本创建、数据传输、副本删除、副本选择、副本一致性管理、安全管理等环节,深入分析、探讨了数据网格系统中数据复制的各项关键技术,为建立综合的数据网格复制策略和技术框架提供了全面的技术分析。最后,对数据复制技术下一步的研究方向作了分析和预测。

关键词 数据网格,复制,副本管理

A Survey on Data Replication Technologies in Data Grid

SUN Hai-Yan WANG Xiao-Dong XIAO Nong JIA Yan ZHOU Bin ZOU Peng

(School of Computer Science, National University of Defense Technology, Changsha 410073)

Abstract Replication technologies have been widely applied in data grid systems to improve the performance of grid applications. Compared with those technologies applied in distributed database, distributed object computing and mobile computing systems, the replication technologies applied in data grid systems hold many unique characteristics. In this paper, replication in data grid systems is summarized as 6 aspects including Replica Creating, Data Transferring, Replica Deleting, Replica Selecting, Replica Consistency Management and Security Management. The replication technologies related with each aspect are analyzed and discussed in details within some successful data grid application systems. Finally, the future work on Data Replication is given.

Keywords Data grid, Replication, Replica management

1 引言

数据网格^[1](Data Grid)是当前网格领域的研究热点,美国、欧洲等国家的多个组织都在开展重大数据网格项目的研究^[2~4]。数据网格的目标是使地理上广泛分布的团体能够共享数据资源,以完成对大规模海量数据集的复杂分析、处理任务。衡量不同存储设备和技术的一个重要指标就是数据访问时间。在分布存储技术中,网络延迟是影响数据访问时间的重要因素之一。虽然随着网络性能的提高,网络延迟将逐渐降低,但访问本地数据与通过网络访问远程数据相比,特别是广域网,仍存在很大的性能差异。在数据网格中,通过数据复制为广域网环境下的客户提供多个数据备份,可以有效减小数据访问时间,降低网络带宽负载^[5]。

复制技术在分布数据库、分布对象计算、移动计算等领域已得到广泛应用。由于不同的研究前提、研究目标和对网络环境的假定,现有的一些复制策略和技术不能很好地综合应用在数据网格系统中。因此需要在这些策略的基础上,根据应用特点和用户需求,综合设计、实现符合数据网格特性的数据复制策略和技术。本文首先通过对比分析数据复制技术在不同领域中的应用,总结了数据网格系统中数据复制的应用特点;通过深入分析数据复制技术的研究内容,总结了数据复制所面临的六大技术问题:数据副本创建、数据传输、副本删除、副

本选择、数据一致性管理、安全管理。在此基础上,分析了数据网格系统中各项数据复制关键技术,为构造综合的数据复制技术框架提供了全面深入的技术分析和总结。最后,对数据复制技术下一步的研究方向作了分析和预测。

2 数据复制技术

复制技术广泛应用于很多领域,如分布式数据库系统、分布对象系统和移动计算等。本节主要通过对比分析数据复制技术在各领域的应用,总结数据网格系统中数据复制技术的特点。

2.1 相关领域中的复制技术

复制技术广泛应用于多个领域,其中跟分布存储关系比较密切的应用领域包括分布数据库、分布对象和移动计算等。不同的应用领域,复制的目标、复制的粒度及其关键技术各不相同。

作为分布数据库系统的关键技术,复制技术可以提高数据库系统的容错能力、实现系统的负载均衡、改善系统的访问性能等。其关键技术包括多个数据库的一致性维护、负载均衡以及分布式事务管理等。

分布对象系统中通过复制多个服务对象,实现系统的容错、负载均衡等,其关键技术包括对象组的生命周期管理、负载均衡、对象成员的出错恢复以及对对象成员状态一致性维护

^{*} Supported by National Natural Science Foundation of China under Grant No. 90104020(国家自然科学基金)。孙海燕 博士生,主要研究领域为分布计算,数据网格;王晓东 博士,副研究员,主要研究领域为协同工作,移动计算;肖 依 主要研究领域为网格计算;贾 焰 教授,主要研究领域为分布计算,数据库;周 斌 博士,副研究员,主要研究领域为分布计算;邹 鹏 教授,主要研究领域为分布计算。

等。

由于移动环境下网络带宽较低,根据当前移动用户的布局 and 访问情况,通过动态的数据复制,可以实现“数据跟着用户走”,使得移动用户可就近访问到所需数据,从而提高访问性能。移动计算的数据复制主要考虑移动环境下的网络带宽低,以及断接操作模式等特点。

2.2 数据网格中的数据复制技术

数据网格系统中,数据复制技术的主要目的是降低系统的访问时间以及节约网络带宽等。复制技术在运行时刻根据数据网格系统的应用特征以及用户的访问特征自动将数据项复制到不同网格节点,其实质就是通过数据在网格系统的流动,实现网格系统的整体性能优化。

2.2.1 数据网格中数据复制技术的特点 与分布数据库、分布对象等领域的复制技术相比,数据网格环境中的复制技术需要面对以下问题:

- 数据类型的多样性:在数据网格系统中,数据可以存储为普通文件,XML 文件以及各种不同的数据库等;
- 副本数量大:数据网格环境下,副本的数量可能达到几百,甚至更多,因此需要对副本提供有效管理;
- 副本创建策略:由于网格环境的动态性,以及数据网格系统巨大的数据量,只有适合具体应用的动态副本创建策略才能满足要求;
- 网络开销:数据库、分布对象系统选择副本时很少考虑网络开销,而在数据网格系统中,副本选择要充分考虑网络开销;
- 维护副本一致性的算法:数据库、分布对象系统中的副本同步仅仅考虑包括几个读、写操作这样的小事务,而数据网格系统中的数据量很大,且操作粒度很大,一个操作可能修改整个文件的内容。另一方面,数据网格中的数据经常分布和复制于广域网上,意味着传统的保持副本一致性的算法并不适于数据网格中副本的一致性管理。

2.2.2 数据网格对复制系统的要求 根据上面的分析以及网格系统的特点,数据网格环境下的数据复制系统应该满足以下要求:

- 动态性:复制系统能够根据需要动态创建和删除副本;
- 安全性:复制系统在创建副本时,必须注意安全问题,如避免信息泄漏给第三方等;
- 有效性:复制系统创建副本时必须考虑资源和时间的开销;
- 适应性:复制系统必须考虑各种网络速度、异构存储环境、各种处理速度以及失败等情况;
- 灵活性:副本能够根据需要加入和离开系统;
- 可扩展性:复制系统能够创建和管理大量副本。

3 关键技术

由于数据网格的特殊性,数据网格系统对于数据复制技术的要求不同于传统应用领域的复制技术。数据网格环境下实现数据复制的关键技术包括:副本创建、数据传输、副本删除、副本选择、副本一致性管理以及安全管理等。本节详细分析了各项关键技术的特点以及实现关键。

3.1 副本创建

副本的创建过程包括静态创建和动态创建两种。在网格环境下,由于数据量、用户量非常大且网格环境的动态性,只有动态创建副本才能够满足网格环境的要求。在运行时刻复

制系统根据网格系统的应用特征以及用户的访问特征等自动将数据项复制到不同节点上,以实现网格系统的整体性能优化。

在选择副本创建策略时,必须考虑以下因素:网格系统的拓扑结构、系统运行负载、存储终端效率、网络状况和数据副本尺寸等物理特性因素以及用户访问特征等。副本创建策略包括复制的粒度、复制的时机和副本的放置等关键问题。

目前对于副本创建策略的研究主要针对层次式网格拓扑结构^[6],比较常用的副本创建策略包括最佳客户副本创建策略、瀑布副本创建策略、基于缓存的副本创建策略、缓存加瀑布副本创建策略、快速传播副本创建策略、基于经济模型的副本创建策略。

系统的复制性能与用户的请求模式也有很大关系。一般地,用户访问数据的局部性越大,则通过复制得到的好处越多。用户对数据的请求可能具有多种访问局部性,包括:

- 时间局部性:最近被访问过的数据可能被再次访问;
 - 地理局部性:一个客户最近访问的数据可能被与该客户临近的客户访问;
 - 空间局部性:与刚被访问的数据临近的数据可能被访问。
- 从文[6]性能模拟的结果可以看出,合适的副本创建策略确实能够缩小数据网格系统的访问时间并节省网络带宽,用户请求模式影响到复制的性能。

3.2 可靠数据传输

实现不同网格节点间数据复制的关键是大型数据集合的可靠传输。在网格系统中,不同节点可能使用不同存储系统,且不同的存储系统间一般不兼容。为了支持多种存储系统并为用户提供统一的访问界面,Globus 小组基于标准 FTP 协议,提出了 GridFTP 协议。GridFTP 协议对标准 FTP 协议进行了全面扩展,跟标准 FTP 协议相比,GridFTP 协议具有以下特点^[7,8]:

- GridFTP 基于网络安全基本框架和 Kerbero 安全机制;
- GridFTP 提供了第三方控制的数据传输;
- GridFTP 采用并行的、多 TCP 流提供并行数据传输,以提高网络带宽;
- 在数据网格环境中,大规模数据可分布放置在多个存储点上,这种存储方式称为条状数据存储,GridFTP 通过支持条状数据传输以提高数据的传输速度;
- GridFTP 提供了传输一个文件的任意子集或某个区域的功能;
- 在 GridFTP 协议中,可以根据具体的文件大小和类型,手动和自动设置 TCP 缓冲/窗口大小;
- 在 GridFTP 协议中,支持可靠的数据传输以及数据重传等。

在 Globus 工具集中实现了 GridFTP 协议的大部分功能,并且在进一步完善中。目前,几乎所有的数据网格系统中的数据传输工具都采用 Globus 的 GridFTP 包。

3.3 副本删除

副本删除在数据网格环境下同样非常重要,这是因为:

- 由于节点的存储资源有限,在创建新数据项的副本时,可能需要删除其它数据项的副本;
- 某些存储副本的节点可能由于网络等原因而不可达,此时需要从副本目录中将该副本节点对应的项删除;
- 由于副本的动态创建,会导致一个数据项存在很多副

本,而另一方面网络 and 用户访问模式的变化,可能使得数据项存在“过多”副本。“过多”是指这些副本的存在并不能够提高系统性能,相反可能会由于管理代价的提高而降低系统性能,浪费存储空间。因此,需要将过多的副本删除;

- 用户可能要求删除某个逻辑数据项的全部副本。

在第 1、2 种情况下是否需要显式删除副本,要根据具体的副本目录来确定,若副本目录采用软状态协议,则无需显式删除。

3.4 副本选择

数据网格系统根据客户的请求定位相应数据项的副本,若存在多个副本,则需要从副本序列中选择最合适该客户的副本。因此,副本选择包括两个阶段:根据数据项的逻辑名称,得到数据项对应副本的物理位置即副本定位;从一系列存储在不同物理位置的副本中,选择“最好”的副本即副本优化,“最好”主要指副本的响应时间最短。

3.4.1 副本定位 副本定位是网格环境下资源定位的关键问题。系统需要为用户提供副本定位所需的逻辑数据项标识,以获取该数据项的一个或者多个副本。在网格环境中,副本目录是副本定位的关键。网格主要面向高能物理和天气建模等大型科学应用,这类应用对副本定位系统的要求如下^[9]:

- 大小:必须能够支持最少几百个复制节点,约 5000 万个逻辑文件,5 亿个物理文件;

- 性能:服务必须能够每秒处理 1000 个查询和 200 个修改,其平均响应时间应低于 10 毫秒,最大查询响应时间应不超过 5 秒;

- 安全:副本定位应关注数据的私有性、数据的存在及位置知识的完整性等;

- 可靠性:副本定位系统能够透明处理某个节点失效或者不可访问的情况。

数据网格系统 SRB^[2]中的副本定位是通过集中式的中央元数据服务器实现的。集中的元数据服务器会带来单点失效、可扩展性、性能等方面的问题。Globus 工具集中提供了副本目录服务来实现副本定位,但目前其实现采用了集中式目录,限制了系统的可扩展性和可靠性。Ripeanu 和 Foster 提出了一种分布的副本定位方法,利用 Bloom Filter 技术在各定位节点上都压缩存储了系统中全部的副本定位信息,有着良好的查询性能^[10],但带来了较大的存储开销和更新开销。Giggle 体系结构为 Globus 项目组和欧洲数据网格项目组联合提出的副本定位系统^[11]。在 Giggle 中,系统可以根据需要,通过选择合适的参数来构建所需的副本目录。

3.4.2 副本优化 从副本目录查找到一系列副本后,需要从多个副本中选择最适合客户请求的副本,该过程即副本优化。在这里主要指访问数据的响应时间最短,副本的响应时间可用下面公式表示^[12]:

$$\text{responsetime}_i = \text{filetransfer_time}_i + \text{fileaccess_time}_i$$

其中, responsetime_i 为副本 i 对特定客户请求的响应时间, $\text{filetransfer_time}_i$ 为将副本 i 从源节点传输到请求节点的传输时间, fileaccess_time_i 为访问副本 i 所花费的时间,包括传输前的等待时间以及传输到本地节点后访问数据所花费的时间等。

副本优化的过程即选择副本 i ,对于副本序列中任意副本 j ,都有 $\text{responsetime}_j \geq \text{responsetime}_i$ 。在具体实现时,副本优化并不一定选择响应时间最短的副本,只要在允许的时间

内选择出响应时间能够满足客户要求的副本即可。

在广域网环境下,一般 $\text{filetransfer_time}_i$ 远远大于 fileaccess_time_i 。而 $\text{filetransfer_time}_j = \text{size}_{file} / \text{transfer_rate}$,其中 size_{file} 为数据项的大小, transfer_rate 为网络传输速率,而 size_{file} 可从副本元数据服务中得到。由上面公式可以看出,网络性能是影响副本选择的一个关键,因此在作副本选择时,需要对网络性能做出预测。

3.5 副本一致性管理

副本一致性在很大程度上依赖于应用语义,如网络文件系统中实现的松散一致性对广域网的用户来说非常强,但对数据库管理系统而言仍无法满足应用要求。在考虑数据网格系统中副本一致性时,必须充分考虑具体应用需求^[13]。

只有写操作会影响副本的一致性,而不同的数据网格应用具有不同的读写特征,因此需要根据应用的读写特性选择副本一致性管理策略。保持数据网格中副本的一致性,必须充分考虑数据网格系统的以下特点:

- (1)数据网格系统分布于广域网环境,其访问延迟大,且经常存在网络失败;

- (2)数据网格中的数据量可能很大,有时可为 terabyte 的数据量级;

- (3)数据网格环境中数据项副本的数量可能非常大;

- (4)数据网格中数据的操作粒度大,一个写操作可能修改整个副本的内容;

- (5)网格环境具有一定的动态性。

3.6 安全管理

用户访问数据时,系统需要对用户进行授权和验证,特别是判断一个用户是否具有创建、删除、读、写数据项的权限。不同网格系统有其自身的安全策略,同一个网格系统中每个节点都可能采用不同的安全设施。在考虑数据网格中的复制时,安全同样为一个不可忽视的关键问题。复制技术的安全主要包括认证、授权和权力代理三个方面。

认证:用户通过复制系统访问数据时必须通过相应认证。在具体实现时,一般采用第三方服务进行用户认证。

授权:一旦用户得到认证后,复制系统通过身份和成员资格等信息来检查被访问数据项的授权状态。若用户或者应用属于某个虚拟组织,则用户或者应用可以继承虚拟组织的权限,并具有相应的角色。认证一般也通过第三方工具如社区认证服务实现。

权力代理:用户得到授权后,复制系统将代表用户完成所有任务。这意味着用户所有的权力都将代理给复制系统。

结束语 复制技术作为提高数据网格系统性能的重要技术,为数据网格领域中的一个热门课题。本文详细讨论了数据网格环境下复制技术的特点、复制系统的实现要求等,同时分析了数据网格系统中实现数据复制的关键技术。

数据网格中的复制技术虽然已经得到各方的广泛关注,但是目前还有许多不足之处。我们认为在数据网格的复制领域,还需要在以下几个方面做进一步的研究工作:

- 提供一个灵活通用的副本管理框架:通过副本管理框架集成副本管理的各项功能,为用户提供统一的接口。同时该管理框架能够灵活配置,以满足不同数据网格应用的需求;

- 针对特定应用的高效副本创建策略:目前的副本创建策略主要是针对层次式、高带宽以及单个节点存储能力强的数据网格系统进行研究的,而很多具体的数据网格应用并不具有以上特点。因此在具体数据网格应用中,需要针对具体应

用特点开发高效的副本创建策略;

• 快速、高可扩展的副本定位算法:在很多数据网格系统中,节点、文件以及每个文件对应的副本都可能很多,因此研究快速、高可扩展的副本定位算法将非常必要;

• 副本一致性策略:针对具体数据网格的应用特点,提供高效的副本一致性管理策略;

• 安全管理:在数据复制技术中安全是数据完整性、机密性以及有效性的保证,包括认证、授权和权力代理三个方面。需要针对网格系统的特点,在权力代理方面做进一步的研究工作。

参考文献

- 1 Foster I, Kesselman C. The Grid: Blueprint for a New Computing Infrastructure. July 1998
- 2 <http://www.npaci.edu/DICE/SRB/>
- 3 <http://www.globus.org/>
- 4 <http://cern.ch/grid-data-management>
- 5 Stockinger H. Database Replication in World-wide Distributed Data Grids: [Ph. D thesis]. U. of Vienna, 2001

- 6 Ranganathan K, Foster I. Design and Evaluation of Dynamic Replication Strategies for a High Performance Data Grid. In: Intl. Conf. on Computing in High Energy and Nuclear Physics, 2001
- 7 Allcock B, et al. Secure, Efficient Data Transport and Replica Management for High-Performance Data-Intensive Computing. In: Mass Storage Conf. 2001
- 8 <http://www-ftp.mcs.anl.gov/dsl/GridFTP-Protocol-RFC-Draft.pdf>
- 9 Mathematics and Computer Science Division, Argonne National Laboratory. Response from Globus Team to EDG WP2's Proposed Design for a Distributed Replica Catalog. <http://grid-data-management.web.cern.ch/grid-data-management/docs/responsetoEDGReplica5.doc>, 7 Sep. 2001, CERN
- 10 Ripeanu M, Foster I. A Decentralized, Adaptive Replica Location Mechanism. In: Proc. High Performance Distributed Computing (HPDC), 2002
- 11 Chervenak I A, et al. Giggles: A Framework for Constructing Scalable Replica Location Services. In: Proc. of SC2002 Conf., Baltimore, MD, 2002
- 12 Stockinger K, et al. Access Cost Estimation for Unified Grid Storage Systems. In: Intl. Workshop on Grid Computing (Grid2003), Phoenix, Arizona, IEEE Computer Society Press, 2003
- 13 Dullmann D, Hoschek W, Jaen-Martinez J, Segal B. Models for Replica Synchronization and Consistency in a Data Grid. IEEE HPDC-10, 2001

(上接第7页)

$(y, \psi_1)) \wedge (\beta \wedge (Poss(\varphi_1, \psi_1) \wedge \gamma(x, \varphi_1, \psi_1) \rightarrow Knows(y, \psi_2) \vee Kwhether(y, \psi_2) \vee Kref(y, \psi_2))) \wedge \dots \wedge (\beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma(x, \varphi_1, \psi_{n-1}) \rightarrow Knows(y, \psi_n) \vee Kwhether(y, \psi_n) \vee Kref(y, \psi_n))) \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_{n-1}) \wedge \gamma(x, \varphi_1, \psi_{n-1}) \rightarrow Knows(y, \psi_n) \vee Kwhether(y, \psi_n) \vee Kref(y, \psi_n)))$, 其中 $\psi_1 = do(\varphi_1, s)$, $\psi_2 = do(\varphi_1, do(\varphi_1, s))$, $\psi_3 = do(\varphi_1, do(\varphi_1, do(\varphi_1, s)))$ 等;

③ 组合服务描述 φ 的前提的形式公式为 $(\beta \wedge (Poss(\varphi_1, s) \rightarrow \pi_{11} \wedge \pi_{12} \wedge \dots \wedge \pi_{1n})) \wedge (\beta \wedge (Poss(\varphi_1, \psi_1) \rightarrow \pi_{21} \wedge \pi_{22} \wedge \dots \wedge \pi_{2n})) \wedge \dots \wedge (\beta \wedge (Poss(\varphi_1, \psi_{n-1}) \rightarrow (\pi_{n1} \wedge \pi_{n2} \wedge \dots \wedge \pi_{nn})) \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_{n-1}) \rightarrow \pi_{n1} \wedge \pi_{n2} \wedge \dots \wedge \pi_{nn}))$, 其中 $\psi_1 = do(\varphi_1, s)$, $\psi_2 = do(\varphi_1, do(\varphi_1, s))$, $\psi_3 = do(\varphi_1, do(\varphi_1, do(\varphi_1, s)))$ 等;

④ 组合服务描述 φ 的输入的形式公式为 $(\beta \wedge (Poss(\varphi_1, s) \rightarrow Kref(\varphi_{1t}, s) \wedge \dots \wedge Kref(\varphi_{1w}, s))) \wedge (\beta \wedge (Poss(\varphi_1, \psi_1) \rightarrow Kref((\varphi_{2t}, \psi_1) \wedge \dots \wedge Kref(\varphi_{2w}, \psi_1))) \wedge \dots \wedge (\beta \wedge (Poss(\varphi_1, \psi_n) \rightarrow Kref(\varphi_{2t}, \psi_n) \wedge \dots \wedge Kref(\varphi_{2w}, \psi_n))) \wedge (\neg \beta \wedge (Poss(\varphi_1, \psi_n) \rightarrow Kref(\varphi_{2t}, \psi_n) \wedge \dots \wedge Kref(\varphi_{2w}, \psi_n)))$, 其中 $\psi_1 = do(\varphi_1, s)$, $\psi_2 = do(\varphi_1, do(\varphi_1, s))$, $\psi_3 = do(\varphi_1, do(\varphi_1, do(\varphi_1, s)))$ 等。

4 讨论

研究语义 Web 服务描述本体 OWL-S 的形式语义的主要目的是从理论上仿真、验证服务的执行过程,本文用带认知流的情景演算理论对 OWL-S 进行了严格的形式化描述,即用带认知流的情景演算理论来刻画 OWL-S 的形式语义。

从理论上仿真、验证服务执行的过程是:将非形式化的用 OWL-S 描述的服务(原子服务或组合服务)转化为情景演算形式化理论,其中包括后继状态公理、动作前提公理,以及情景演算的基本公理、描述初始情景的公理、动作的唯一命名公理等。对于后三种公理,遵循传统情景演算理论的要求;而对于后继状态公理和动作前提公理,由下列方法给出:

由 OWL-S 的形式语义描述中的条件效果和条件输出可以表达情景演算理论中的后继状态公理,它的形式为: $F(x, do(\varphi, s)) \equiv \gamma_F^+(x, \varphi, s) \vee (F(x, s) \wedge \gamma_F^-(x, \varphi, s))$ 。

由 OWL-S 的形式语义描述中的前提和输入可以表达情景演算理论中的动作前提公理,它的形式为: $Poss(\varphi, s) \equiv \pi_1 \wedge$

$\pi_2 \wedge \dots \wedge \pi_n \wedge Kref(\varphi, s) \wedge \dots \wedge Kref(\varphi_w, s)$ 。

将非形式化的服务描述转化为用情景演算理论刻画的形式化描述后,就可以利用情景演算理论的推理机制从理论上求解服务执行问题,即得到一个求解服务的动作序列。

本文的工作只是用情景演算理论来刻画 OWL-S 的形式语义,是在 Srinu Narayanan^[7]的基础上进行的(他只研究了 OWL-S 中的原子服务描述的形式语义,没有研究 OWL-S 中组合服务描述的形式语义)。至于如何利用情景演算从理论上验证服务执行等问题,有待进一步深入研究。

结束语 本文分析了目前语义 Web 服务的研究现状和存在的问题,特别是语义 Web 服务描述本体 OWL-S 的形式语义研究中存在的问题,在 Srinu Narayanan 等人研究的基础上,用情景演算理论进一步研究了 OWL-S 中组合服务描述的形式语义,从而完善了 OWL-S 的形式语义,为语义 Web 服务提供了合理的理论基础。进一步工作主要研究如何利用情景演算从理论上验证服务执行等问题。

参考文献

- 1 Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Scientific American, 2001, 284(5): 34~43
- 2 岳昆,王晓玲,周傲英. Web 服务核心支撑技术:研究综述. 软件学报, 2004, 15(3): 428~442
- 3 蒋运承. 基于主体的智能 Web 中的服务研究: [博士学位论文]. 北京: 中国科学院研究生院, 2004
- 4 McIlraith S, Son T C, Zeng H. Semantic Web Services. IEEE Intelligent Systems. Special Issue on the Semantic Web, 2001, 16(2): 46~53
- 5 The OWL Services Coalition. OWL-S: Semantic Markup for Web Services, OWL-S 1.0 Release, 2003. (<http://www.daml.org/services/owl-s/1.0/>)
- 6 Anupriya A, Frank H, Katia S. Concurrent Execution Semantics of DAML-S with Subtypes. In: I Horrocks, J Hendler, et al. The Semantic Web - ISWC 2002, First Intl. Semantic Web Conf Sardinia, Italy, Springer Verlag, 2002. 318~332
- 7 Narayanan S, McIlraith S. Analysis and simulation of Web services. Computer Networks: The International Journal of Computer and Telecommunications Networking, 2003, 42(5): 675~693
- 8 Reiter R. Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. MIT Press, 2001
- 9 Giacomo G D, Lespérance Y, Levesque H J. ConGolog: A Concurrent Programming Language Based on the Situation Calculus. Artificial Intelligence, 2000, 121: 109~169