P2P技术在网格文件管理中的应用

杨 磊,姜 浩

(东南大学 计算机工程与科学系,江苏 南京 210096)

摘 要:目前把网格分为两类:计算网格和数据网格。计算网格目标在于通过大量计算节点的协作来减少应用程序的执行时间。数据网格提供解决大量数据管理的问题的方法。目前网格文件传输软件例如Grid FTP使用Client/Server结构,在性能和结构上存在问题。将网格和P2P结合起来,提出一种新的非集中式的、高效的数据网格文件管理协议。

关键词:数据网格;P2P;Globus Toolkit;Grid FTP

中图分类号:TP393.07 **文献标识码**:A **文章编号**:1673-629X(2006)12-0127-03

Application of P2P Technology in Grid File Management

YANG Lei, JIANG Hao

(Dept. of Computer Sci. and Eng., Southeast University, Nanjing 210096, China)

Abstract: Grid computing is moving into two ways. The computational grid focuses on reducing execution time of applications that require a great number of computer processing cycles. The data grid provides the way to solve large scale data management problems. Most of the grid file transfer software such as Globus GridFtp uses the Client/Server structure, which may arouse some performance issue. Combining grid with P2P technology, describe new data grid file management protocol, which is decentralized, high availability and high manageability.

Key words: data grid; P2P; Globus Toolkit; grid FTP

0 引 言

网格是一种大容量的,可伸缩的资源共享虚拟组织(virtual organizations),同时也是解决问题的机制[1]。目前把网格分为两类:计算网格和数据网格。计算网格目标在于通过大量计算节点的协作来减少应用程序的执行时间。数据网格则主要提供两种服务:一种有效的、可靠的、安全的数据传输协议(数据存取和元数据的存取)和复制管理[2,3]。

图1简单说明了数据网格的结构层次。

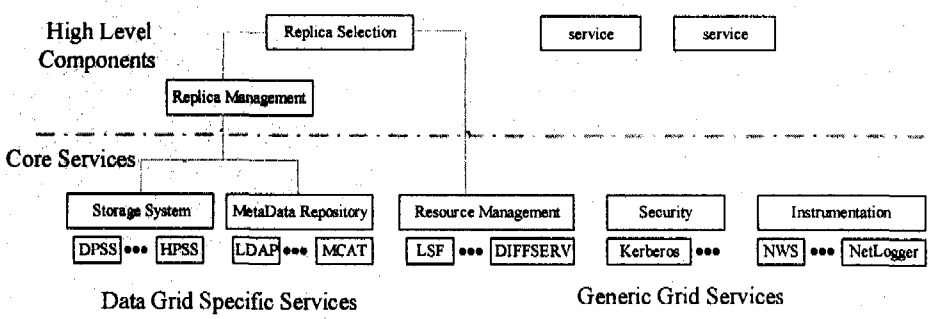

图1 数据网格的结构层次

从层次图中可知数据网格提供了存储系统、目录服务两个底层核心服务。在核心服务层之上根据核心服务提供的功能,数据网格提供了复制管理的服务。同时数据网格也需要调用其它通用的网格服务,例如安全身份认证、网络调度等服务。

目前的数据网格环境,大部分以Globus为基础,Globus Alliance以FTP为基础,增加了安全文件传输与第三方控制文件传输等功能,推出了网格数据传输协议Grid FTP,目的是在网格环境下提供安全有效的数据传输,是一种通用的数据存取传输协议,提供了对当前使用的网格存储系统的支持[4]。但是受到FTP协议与实现的限制,Grid FTP使用C/S结构,与现有服务网格体系结构不兼容,不能满足网格中服务互联的需求。GridFTP所使用的数据端口可用性差,为了防止非法的入侵,现有防火墙一般不允许除HTTP端口以外的其他端口传输数据,造成许多情况下数据资源存在却无法使用GridFTP访问。

随着目前网格计算平台的数量增长,节点的自组织和动态重规划的要求越来越重要。在这种情况下,自然会把网格计算和点对点计算P2P结合在一起。然而,网格平

收稿日期:2006-02-27

作者简介:杨 磊(1976-),男,湖北襄樊人,硕士研究生,研究方向为数据网格以及面向服务应用;姜 浩,博士,副教授,研究方向为工作流以及数据库应用。

台通常使用在由高带宽互联的基于 LAN 或者 SAN 的集群构成的层次结构的联邦模型下；而 P2P 则是通常运行在 Internet 上的通常节点是随机加入的，构成一个平面的网状拓扑。两者之间的模型不同。

文中以数据网格的层次模型为基础，利用当前 P2P 技术出现的新项目技术，设计一种数据网格的文件传输协议的实现形式。其特点包含在如下几个方面：非集中式结构；结合到 Globus 工具包；动态的发布。

1 设计与介绍

1.1 小世界模型和 scale－free 模型

在网络中查找资源很像在连通图中查询节点，通过查找开始节点到结束节点的路径来定义这次查找过程。查找的耗费就是定位到目标节点所经过的所遍历的边的数量(即在查找过程中向网络发送的“消息”的数量)。在 Internet 或者实际世界中，网络的拓扑可以看成随机图的形式。随机图中，任意两个节点的平均路径长度近似于 $\log(n)/\log(k)$，其中，n 为节点数量，k 为平均连通度(参考文献[5])。

统计显示，如果一个网络拓扑是 scale－free 的，那么也同时符合小世界模型。网络的度的分布服从 Power－law 分布，统计表明网络的拓扑结构和小世界网络模型类似：网络的平均路径长度和随机网络一样比较小，但是网络的节点聚集度会远远大于随机网络的。这说明 scale－free 网络中包含很少的具有高连通度的超级节点或者集中节点；绝大部分的节点是弱连通的，节点连通度保持在一定的程度上。绝大部分的节点遍历在泛洪法查询下会在 7 次查询中找到(统计说明大约成功率为 95%)，这也说明了网络符合小世界现象。但是泛洪法查询也带来了巨大的消息耗费和迟缓的缺点。Random walk 方法是常用的对泛洪式查询的改进。

1.1.1 系统结构

系统结构基于 Globus，但是添加了 P2P 的功能，如图 2 所示。

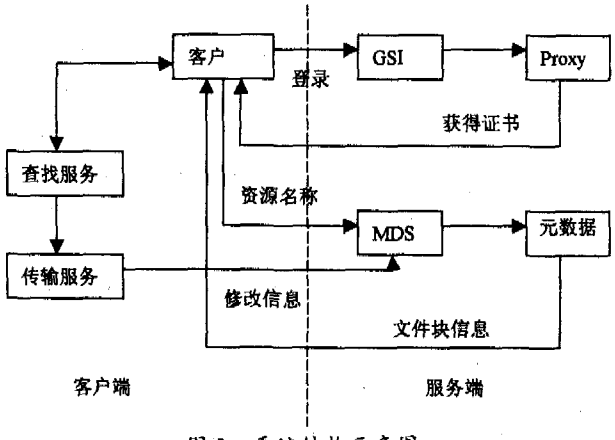

图 2 系统结构示意图

首先使用 Globus 的 GSI 获得单点登录机制，这样，在用户得到认证后，就由一个代理认证来创建证书以认证用户网格内的活动。然后，由 MDS 来提供网格内可用的资源和资源的相关情况。这里，在 MDS 中，仅仅保存的是文件块的信息，而其它的信息比如文件块的存放位置保存在客户端。如何查找文件块则采用了 P2P 的技术，客户节点向其它节点查询来定位资源的存储位置。

1.1.2 元数据的组织

元数据的组织按照功能的不同分别存放在不同的地方：

* 文件元数据：文件元数据说明了文件的信息，并存放在 MDS 中。在节点查询文件时由 MDS 返回给节点数据块的消息以供节点查询。

* 数据块元数据：数据块元数据说明了文件的数据块的数据，存放在 P2P 节点中，在节点查询时按照数据块的元信息定位数据块的位置。

(1)文件元数据。

文件的命名和文件的存储是分开的。文件的名称使用与 Java 兼容的本地文件命名格式命名网格文件，是一个虚拟的 URL 地址，它与一个数据服务的地址和一个路径对应，由 3 个主要部分构成：第 1 部分是协议的标识；第 2 部分是数据服务地址；第 3 部分是‘/’分隔的路径，代表文件在服务中的相对位置。实例如下：DCG:[grid address]/path。文件在网格内的存放是分成小的数据块存放的。每一个数据块以数据块的 HASH 值作为其唯一标志。

文件元数据是存放在 MDS 中的，当用户需要获得一个文件时，在获得认证后，用户向 MDS 发送文件的相关信息，例如文件名称、类型等属性来定位文件的服务名称。这样，MDS 就可以返回客户该文件每一块的 HASH 值，用户利用 HASH 值定位文件块的位置，最后开始数据传输。

(2)数据块元数据。

每一个客户节点自己动态维护一个文件块的元数据信息，以供节点相互查找时使用。文件块的存储以堆栈形式存储(见表 1)，采取类似 FIFO 的策略进行管理：

表 1 元数据堆栈示例

BHK	Block Data Path	Address	LINK
Dtiddfhfjghmjlker8r5nkdhkdfgjl	/GridData/block/blk1.blk	Tcp/202.119.11.225:8477	Tree1
Gdfgdfgr98fgr436ghuf789856jgj	/GridData/block/blk2.blk	Tcp/202.119.11.34:6543	Tree2
3p;ads045lfgi5lfguoiert8gfhldop		Tcp/202.119.9.108:3455	

各 BHK 代表数据块的唯一标示，地址栏里说明该数据块的来源地址和端口，对于一些节点本地没有的数据块，则数据块路径栏为空。对于经常查询的数据块，则会逐步将该数据块移到栈的顶端，并缓存该数据块到本地使用。

对于本地已有数据块情况，LINK 栏里维护一个该节点传输的节点地址，这样，可以通过本栏查询到所有具有本地数据的节点构成的一个节点树，利于传输优化和数据修改。

1.1.3 数据传输过程

传输过程分为两个阶段：首先，客户得到认证后，向

MDS 传输所要求的资源的上下文信息来得到需要的数据块的 HASH 值;第二阶段,使用 HASH 值向其它客户节点查询来找到拥有本地数据块备份的节点,定位数据块的实际位置,得到节点的地址后,使用 HTTP 协议传输数据,避免像 Grid FTP 使用 FTP 端口而无法穿透防火墙的情况。

在第一个阶段,主要调用的是通用的网格服务,而在第二个阶段,则采用的非集中式的 P2P 的结构。获得文件块 HASH 值(BHK)后,查询节点检查本地元数据是否该 HASH 值在栈中而且具有本地备份,这表明该数据块刚好在本地留有数据,这样传输服务就成功完成。如果 DATA 栏中为空,那么可以从 ADDRESS 栏中得到拥有本地备份的节点地址,然后如同一般的 P2P 网络一样向这些节点发送消息开始并行的数据传输,传输完成后,修改 DATA 栏并通知其它节点修改 LINK 数据。当栈中没有该 BHK 时,向邻接的节点发送查询 BHK 的消息,如果邻接节点中保存有该 BHK 的信息,返回消息到源节点,如果邻接节点仍然没有该 BHK 信息,选择拥有最相似的 BHK 的节点,以源节点的身份继续查询,依次在节点之间扩散,直到最后找到对应的存储节点。从而获得需要的数据,同时在源节点和目标节点保留对方的地址和身份等相关信息以使得信息能够回送到源节点。为了减少查询的次数,在消息中如同网络中的 TTL 定义了一个 NTL(消息节点活动数),当一个节点选择下一个节点发送消息时,消息中的 NTL 减去一,当 NTL 为零的时候,查询结束,返回源节点失败的信息。

在传输文件的同时,按照 LRU 算法调整各节点堆栈的次序,如果堆栈中的顺序被调整到了栈顶,而没有本地备份,说明这个文件需要的次数很多,而且该节点属于超级节点,该节点也将自动保存该文件块以供将来传输。

当需要修改文件时,计算需要修改的文件块的 HASH 值,通知 MDS 修改文件元数据,在将修改后的文件块写入本地数据块元数据后,标记旧的 BHK 失效标记以和不常被查询的 BHK 区分开。其它节点的这些失效的文件块会被 GC(Garbage Collection)服务自动删除。

GC 是数据复制管理服务的一个部分,是建立在数据网格核心服务之上的。在文件被删除或者文件块被修改之后,并不是立即回收可用的空间,而是使用 MDS 记录删除或者修改的信息,延迟到 GC 服务启动之后通知 GC 服务正式删除,采用这样的方法,系统更加简单、更加容错。

1.2　性能分析与评价

文中模拟了网络的规格和平均路径长度的关系,这个模拟说明了查找时通讯的耗费,结果如图 3 所示。

从图 3 可以看出,当网络规模在 1000 以下时,平均的路径长度小于 6,这说明能在很少的消息传递中获得资源的地址。初始的平均路径长度很小是因为网络的规模很小,这样不通过消息转发也很容易找到文件,但是随着网络规模的增长,路径长度的增长保持很小的幅度。

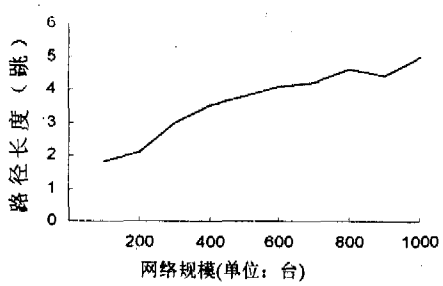

图 3　网络规模和平均路径长度

同时,文中也模拟了系统的可扩展能力和防失败能力。可扩展能力的模拟用来说明这种方法也适用于一些大规模的网格。防失败能力的模拟说明在节点失效的情况下(例如节点断电或者网络故障的情况),网络的健壮程度,如图 4 和图 5 所示。

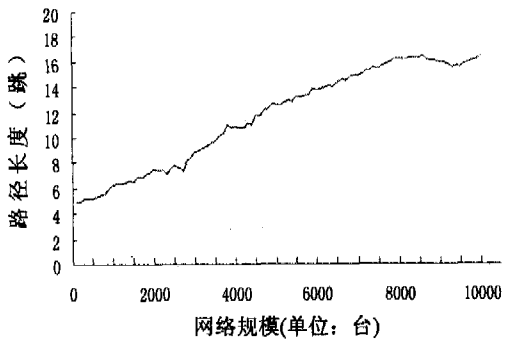

图 4　网络扩展能力和平均路径长度

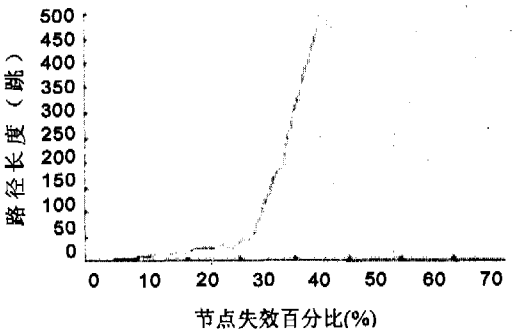

图 5　节点失效和平均路径长度

从图 4 中可以看出这种查询方法的可扩展能力适合大规模的网络结构,当网络规模增长时,平均路径长度保持近似线形增长,由图中可以看到网络的规模增加一倍,平均路径长度仅增加 4。

图 5 显示了查询路径长度的增长和失效节点的关系。即使有 30% 的节点失效,平均的路径长度仍然保持在 20 以下,说明网络具有比较好的鲁棒性。

2　结　论

文中利用 P2P 技术,构建了一种新的网格文件管理协议。相比较 Grid FTP,结合 P2P 技术的文件管理协议具有更多的自治性。同时,采用P2P技术,大大减轻了

(下转第 132 页)

Process 类的 setContentDescriptor(cd)设置输出的内容描述为 RAW_RTP,限定合法的 RTP 格式;

(5) 最后对每一个轨道,选择一种 RTP 支持的传输格式,完成初始化。

用户代理在完成了媒体的初始化后就可以创建 RTP 会话,开始媒体流的发射与接收了。媒体流的发射与接收由该类的 startTransmitting(),startReceiving()这两个方法来实现。

1.4 JAIN SIP 协议栈

JAIN SIP 对 SIP 通信实体的实现采用模块化处理,每个实体都由 SipStack,SipProvider 和 SipListener 构成。整个架构以事件为基础,采用了监听者(Listener)/提供者(Provider)的事件模型,其中事件封装了 SIP 实体所接受到的 SIP 消息。主要是向上层提供 SIP 消息的封装、解析、监听和发送功能。JAIN SIP 协议栈结构如图 2 所示。

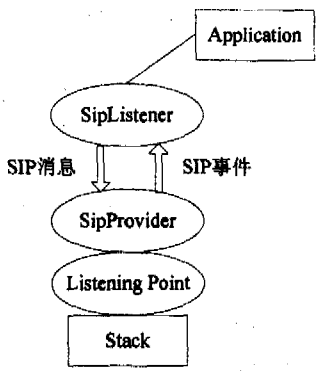

图 2 JAIN SIP 体系结构

SipStack 是 SIP 协议栈,是整个 JAIN SIP 的核心实现部分,所有操作都在这里实现,包括地址产生、消息编解码、消息事件的生成、会话和传输的建立,以及地址的解析等。SipStack 是 SipListener 和 SipProvider 的基础,其实现的 SIP 服务由 SipProvider 统一向 SipListener 提供。

SipProvider 是一个事件的提供者,通过 Listening Point 封装了对 Stack 进行操作的方法,将 SipStack 接受到的 SIP 消息加工成事件交给 SipListener 处理,并将上层应用生成的 SIP 消息交由 SipStack 处理。SipProvider 提供了客户端事务(Client Transaction)和服务器端事务(Server Transaction)的创建方法,对有状态的请求/应答消息提供了良好的支持。SipProvider 同时支持无状态的请求/应答消息,这是 SIP 的默认消息类型。一个应用程序中可以有多个 SipProvider,每个对应一个 Stack。

SipListener 负责管理一个或多个 SipProvider。SipProvider 的行为依赖于应用程序的业务逻辑。当一个 SipProvider 建立之后,将被注册到一个 SipListener;应用程序访问某个 SipProvider,就通过 SipListener 向该 SipProvider 发送消息;如果 SipProvider 收到了其他 SipProvider 的 SIP 消息,需要将其传递给应用程序进行解析,此时它将产生一个事件,SipListener 理解事件的含义并能根据不同的事件触发相应的响应。

2 结束语

SIP 协议作为下一代网络的核心协议,以其易于扩展、开放的业务生成环境、对移动性的支持等特点而具有广泛的应用前景,而基于 Java 技术制定的 SIP 协议栈规范 JAIN SIP 将对 SIP 消息的处理转化为对消息和事件的处理,极大地提高了开发的效率。对基于 JAIN SIP 的用户代理进行了设计分析,介绍了各个模块的功能及软件设计。本设计的实现对开发基于 JAIN SIP 协议栈的增值业务将起到一定的促进作用。

参考文献:

[1] Rosenberg J, Schulzrnne H, Camarillo G, et al. RFC3261 SIP: Session Initiation Protocol[S]. IETF, 2002.

[2] SunMicrosystems. JAIN SIP Tutorial[EB/OL]. 2004 - 06. http://java.sun.com/products/jain/.

[3] AIN - SIP - APPLET - PHONE[EB/OL]. 2003 - 11. http://snad.ncsl.nist.gov/proj/iptel/.

[4] Lewis J, Loftus W. Java 程序设计基础[M]. 第 3 版. 王锦全译. 北京:清华大学出版社, 2004.

[5] Java Media Framework[EB/OL]. 2004 - 02 - 10. http://java.sun.com/products/java2media/jmf/index.html.

(上接第 129 页)

服务器的处理负荷,在容错方面,在节点随机失效的情况下也提高了系统的鲁棒性。最后,系统可扩展方面更加的简单。

参考文献:

[1] Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations[J]. International J Supercomputer Applications, 2001, 5(3): 200 - 222.

[2] Foster I, Kesselman C. The Grid: Blueprint for a New Computing Infrastructure[M]. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998.

[3] Chervenak A, Foster I, Kesselman C, et al. The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets[J]. Journal of Network and Computer Applications, 2001, 23: 187 - 200.

[4] Allcock W. GridFTP Protocol Specification[EB/OL]. 2003 - 03. GGF GridFTP Working Group Document, http://www.globus.org/alliance/publications/papers/GFD - R.0201.pdf.

[5] Watts D, Strogatz S. Collective dynamics of smallworld networks[J]. Nature, 1998, 393: 440 - 442.