Java CoG Kits在GridFTP开发中的应用

应 宏,黄 河,鄢 沛
(重庆三峡学院 数学与计算机科学学院,重庆 404000)

摘 要:通过对Java CoG Kits软件包的研究,介绍了Java CoG Kits的结构和GridFTP的特性,讨论了Java CoG Kits中用于开发GridFTP客户端的FTP包的组件层次和类库,描述了GridFTP客户端类库用于编写GridFTP应用的函数,研究了GridFTP第三方控制数据传输模型,给出了实现的过程和步骤,最后基于Java CoG Kits设计实现了GridFTP第三方控制数据传输。

关键词:Java CoG Kits;GridFTP;第三方控制数据传输
中图分类号:TP39 **文献标识码**:A **文章编号**:1673-629X(2008)11-0211-03

Application of Java CoG Kits in GridFTP Development

YING Hong, HUANG He, YAN Pei
(College of Mathematics and Computer Science, Chongqing Three Gorges University, Chongqing 404000, China)

Abstract: Through the research of Java CoG Kits soft package, introduced the construct of Java CoG Kits and the characteristic of GridFTP; discussed component hierarchy and class library of Java CoG Kits to develop GridFTP client; described the functions of GridFTP client; researched GridFTP third-party control of data transfer model, gave the realization process; finally designed and implemented GridFTP third-arty control of data transfer based on Java CoG Kits.

Key words: Java CoG kits; GridFTP; third-party control of data transfer

0 引 言

商业分布式计算技术提供了快速构建复杂Client/Server应用的方法,网格技术则提供了在大范围跨区域多种异构环境中进行协作计算和资源共享的网格服务,显然两者应用开发的关注点不一致。前者主要考虑的是可扩展性、基于组件的封装、基于桌面的表示等,而后者考虑的重点在于端到端的性能、网格服务支持、动态自适应等问题,所以网格技术与现有的商业应用技术尚不能很好地融合在一起。Java CoG Kits (Java Commodity Grid Toolkit)就是在网格技术和现有商业开发技术(如Java Platform、CORBA等)之间建立桥梁,进行映射和交互,充分利用商品化技术的优点,建立各种网格应用。Java CoG Kits支持网格应用的开发,为GridFTP提供底层接口,方便实现GridFTP数据传输。

收稿日期:2008-02-08
基金项目:重庆市自然科学基金(2005BB2001);重庆市教委科研基金(KJ051101)
作者简介:应 宏(1962-),男,重庆万州人,教授,研究方向为网格计算和Web数据库。

1 Java CoG Kits

1.1 Java CoG Kits结构

Java CoG Kits的设计目标主要是方便开发基于Java和Web界面的网格应用,使用Java CoG Kits能方便地通过Java类和组件访问网格,它提供了大量的接口用于网络通信和端到端的访问。Java CoG Kits组件可分四个层次[1]:

(1)底层网格界面组件(Low-Level Grid Interface Components):提供了到网格服务资源的映射。这些网格服务资源包括:提供安全访问远端资源的GSI,基于LDAP获取网格服务资源状态信息的原计算目录服务Globus MDS,用于分配和管理资源的Globus GRAM,通过Globus GASS的数据访问服务等。

(2)底层实用工具组件(Low-Level Utility Components):主要用于加强Globus toolkit的功能。其中有使用MDS寻找计算资源的组件、定位计算资源的组件、对基于XML或者RSL的任务描述进行验证的工具组件、判断资源是否继续有效的组件。

(3)通用底层GUI组件(Common Low-Level GUI Components):提供了一组可重用的底层GUI组件,可用于LDAP属性编辑器、RSL编辑器、LDAP浏

览器等的开发。

(4)应用相关 GUI 组件(Application - specific GUI Components):用于简化应用程序和基本 CoG Kit 组件之间的差异。如股票监视器、图形化的天气数据显示组件、对天气数据的图形化搜索引擎等。

1.2 org.globus.ftp 包

1.2.1 GridFTP

GridFTP 是一个文件传输协议,它在标准的 FTP 协议上进行扩展,旨在为网格环境下分离的存储系统间的互操作提供一个通用的、可扩展的底层数据传输协议,并为应用程序提供统一的访问接口,从而实现网格环境下安全、高效、快速、可靠的数据访问和数据传输服务。相比于普通 FTP 协议,GridFTP 协议有以下特性[2]:

(1)支持安全基础设施(GSI)和 Kerberos 认证;

(2)支持数据的第三方控制传输(建立在 GSI 的安全机制上);

(3)支持并行(parallel)数据传输;

(4)支持条状(stripped)数据传输;

(5)支持部分(partial)文件传输;

(6)支持 TCP buffer 大小的控制(自动调节尚未实现);

(7)支持数据的重传以确保数据的可靠传输。

1.2.2 FTP 包

org.globus.ftp 包(简称 FTP 包)是 CoG Kits 的一个子集,用于开发 GridFTP 客户端的工具包,它为 FTP 和 GridFTP 提供底层接口,支持 GridFTP 数据传输功能[3]。FTP 包由三部分组成,如图 1 所示。

第一层,该层是最接近用户的一层,是最高级的一层,它提供用户接口,FTP 类、GridFTP 类和其他 org.globus.ftp 下的类。

第二层,实现基本的控制协议:控制通道、发送命令、主机响应等,另外还为数据通道管理提供了一个接口(GridFTPServerFacade)。

第三层,实现底层数据通道和读写操作。GridFTPServerFacade 通过建立多条数据通道来处理并行传输和条状传输。读写操作与传输类型和传输模式联系在一起。

FTP 包的三个层次包含以下六个类:

(1)org.globus.ftp 下包含了第一层可以直接使用的用户接口类。

(2)org.globus.ftp.vanilla 下包含了第二层实现 vanilla FTP 协议的类。

(3)org.globus.ftp.extend 下包含了第二层实现 GridFTP 协议的类。

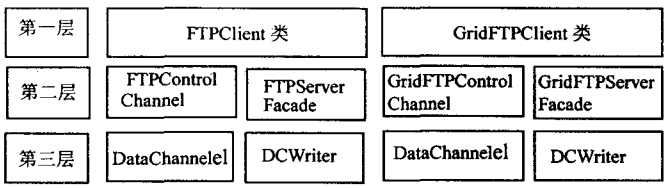

图 1 GridFTP 客户端库的层次结构图

(4)org.globus.ftp.dc 下包含了第三层实现数据通道的类。

(5)org.globus.ftp.exception 下包含了 FTP 的例外类。

(6)org.globus.ftp.test 下包含的是测试类,给用户提供参考。

1.3 FTP 函数

基于 Java CoG Kits 实现 GridFTP,以两大类 API 及相应库的形式提供给开发人员,分别是 globus - ftp - control 和 globus - ftp - client。其中 globus - ftp - control 用于管理 GridFTP 连接,包括安全认证、创建控制和数据信道,从数据信道读取和写入数据。具体包括 FTP 协议标准中定义的数据通道管理,并很好地支持了并行传输、分块传输和第三方控制传输等 GridFTP 新增功能。而 globus - ftp - client 则用于实现 GridFTP 的客户端,这个 API 提供在 globus - ftp - control 之上对 GridFTP 新增功能的支持,包括文件的 get 与 put 操作,调用和设置并行数据传输中的并行层次和部分文件传输操作,第三方控制文件传输操作和 TCP 缓冲/窗口大小的设置。globus - ftp - client 库主要用于编写 GridFTP 应用,它包含较多函数,表 1 列出部分函数[4]。

表 1 globus - ftp - client 部分函数

功能集	函数名称	功能描述
文件管理	exists(String filename)	检查服务器上的文件是否存在
	makeDir(String dir)	在服务器上创建目录
	deleteFile(String filename)	在服务器上删除目录
	list()	列举服务器上的文件
数据传输	get(String remoteFileName, File localFile)	从服务器上获取文件内容
	extendedGet(String remoteFileName, long offset, long size, DataSink sink, MarkerListener mListener)	从服务器上获取文件的部分内容
	put(File localFile, String remoteFileName, boolean append)	往服务器上传文件
	extendedPut(String remoteFileName, long offset, DataSource source, Marker Listener mListener)	往服务器上传文件的部分内容
	extendedTransfer(String remoteSrcFile, GridFTPClient destination, String remoteDstFile, MarkerListener mListener)	在两个服务器之间第三方传输数据
控制管理	setProtectionBufferSize(int size)	设置保护缓冲
	setOptions(Options opts)	设置并行传输的并行数
	setType(int type)	设置数据传输类型
	setMode(int mode)	设置数据传输工作模式

2　GridFTP 数据传输

GridFTP 支持两种工作模式:流传输模式(ASCCI)与扩展数据块模式(EBLOCK)。GridFTP 默认使用流传输工作模式,在流传输模式中 GridFTP 通过单个 TCP 连接顺序地发送和接收数据。在扩展数据块模式中数据将被分成不同的块,并通过建立多个连接发送数据,该模式支持并行数据传输、条状数据传输、第三方控制的数据传输等。

2.1　第三方控制传输

GridFTP 支持建立在 GSI 的安全机制上的第三方控制的数据传输,在经过鉴别的第三方控制下,可以方便地允许某个站点的用户或应用,启动、监控和管理在其他两个站点之间的数据传输过程。

第三方控制的数据传输模型如图 2 所示[5],它含有 GridFTP 客户端和两个 GridFTP 服务器。GridFTP 客户端与两端的服务器分别建立控制通道和进行用户认证和审核,在认证审核通过后,控制指令在两个通道中进行传输并控制两端的 GridFTP 服务器进行数据传输操作,传输的控制由第三方的客户端来做。

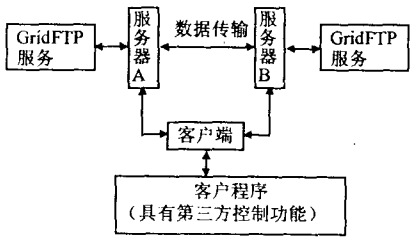

图 2　GridFTP 第三方控制模型

2.2　数据传输实现

GridFTP 数据传输的一般步骤是,建立连接→安全认证→设置传输模式和类型→进行数据传输→断开连接。即先创建 GridFTPClient 实例建立连接,然后调用 Authenticate 方法进行安全认证,接着设置传输模式、类型,进行数据传输,传输完毕后断开和主机的连接。

采用 Java CoG Kits 工具包,较容易开发具有并行和条状传输功能的第三方控制数据传输,其主要代码片段如下:

```
public void thtransfer(String host1, int port1, String sourceFile,
  String host2, int port2, String destFile,
  GSSCredential cred, int parallelism
  ) {
  try {
    //建立一个 GridFTPClient 实例,代表 FTP Server1 的接口,建立起控制通道
    GridFTPClient source = new GridFTPClient(host1, port1);
    //建立另一个 GridFTPClient 实例,代表 FTP Server2 的接口,建立起控制通道
    GridFTPClient dest = new GridFTPClient(host2, port2);
    setParams(source, cred); //调用自定义函数 setParams(),针对 Server1 进行设置
    setParams(dest, cred); // 调用自定义函数 setParams(),针对 Server2 进行设置
    source.setOptions(new RetrieveOptions(parallelism)); //设置并行数
    HostPortList hpl = dest.setStripedPassive(); //设置目标机为被动模式,并设置为条状传输
    source.setStripedActive(hpl); //设置源机器为主动模式,并设置为条状传输
    source.extendedTransfer(sourceFile, dest, destFile, null);//第三方控制传输
  }catch(Exception e)
  {
  e.printStackTrace();
  }
}
    private void setParams(GridFTPClient client, GSSCredential cred)
  throws Exception{
    client.authenticate(cred); //进行安全认证
    client.setProtectionBufferSize(16384); //设置保护缓冲
    client.setType(GridFTPSession.TYPE-IMAGE); //设置传输类型为映像
    client.setMode(GridFTPSession.MODE-EBLOCK); //设置主机模式为扩展模式
  }
```

3　结束语

Java CoG Kits 对 GridFTP 的支持仅仅是 Java CoG Kits 功能的一个部分,而 GridFTP 也仅仅是网格数据管理的一个部分。事实上,Java CoG Kits 定义了网格和实际商业框架之间的映射和界面,提供了一系列的通用构件及函数,很大程度上方便了网格应用开发者的工作。深入全面地研究 Java CoG Kits 软件包的技术和应用,将有助于快速实现各种网格应用。

参考文献:

[1] Laszewski G V, Foster I, Gawor J, et al. Peter Lane: A Java Commodity Grid Kit[J]. Concurrency and Computation: Practice and Experience, 2001, 13(8): 643 - 662.

[2] Allcock W, Bester J, Bresnahan J, et al. GridFTP: protocol Extensions to FTP for the Grid[EB/OL]. 2002 - 01. http://www - fp. mcs. anl. gov/dsl/GridFTP - Protocol - RFC - Draft. pdf.

[3] Laszewski G V, Gawor J, Lane J, et al. Features of the Java

(下转第 216 页)

R_{LCR} 对页面访问时间阈值 δ 的影响因子，α 是平滑系数，实验表明 α 选择为 1.25 比较合适。

计算公式如下：

$$R_{LCR} = 2(0.618L_I + 0.382L_O)/S_{DS} \quad (1)$$

$$\beta = 1 - \exp(-\mathrm{sqrt}(\mathrm{sqrt}(R_{LCR}))) \quad (2)$$

$$\delta = \alpha t(1 + \beta) \quad (3)$$

1.3.2 基于时间阈值 δ 生成用户会话集

要设置每个页面的访问时间阈值 δ，首先要获得统计后的页面的访问时间 t，并结合页面的 R_{LCR} 影响因子 β 调整 δ。统计后页面的访问时间 t 的集合记为 $S_t = \{t_1, t_2, \cdots, t_n\}$，页面的影响因子 β 集合记为 $S_\beta = \{\beta_1, \beta_2, \cdots, \beta_n\}$。

算法步骤如下：

1) 根据日志文件 log 统计得到 t 的集合 S_t，按用户将日志文件划分，即用户识别，根据 1.3.1 节的方法统计页面的访问时间得到 S_t；

2) 根据影响因子集合 S_β 和 α 值调整 S_t 得到页面访问时间阈值集合 S_δ，按式(3) 结合 α 和 S_β 计算得到 S_δ；

3) 根据 S_δ 重新划分日志文件得到用户的会话集合。

2 实验数据及分析

本实验数据来源于池州学院网站：211.86.192.12，服务器日志数据为 2007 年 10 月 24 日～11 月 3 日。实验中将基于固定时间阈值的会话识别算法与基于页面访问时间阈值的会话识别算法这两种算法结果进行比较分析(见表 1)。

表 1 训练数据与测试数据实验统计结果

	训练数据	测试数据
实验数据中共有访问记录数	708912	69852
数据清洗后共有有效页面数	55442	4265
页面过滤后共有有效页面数	31369	2536
基于 ip、agent 和 referer 方法划分的用户数	21492	1739
基于固定页面时间阈值(10min)方法划分的会话数	24627	1990
基于访问页面阈值方法划分的会话数	25667	2075
页面过滤后的会话数	24605	2038

从表 1 统计结果中得到以下结论：

1) 通过框架页面的过滤，会话个数基本不变，但是大大减少了有效页面数，提高了会话识别的效率，更重要的是提高了挖掘结果的兴趣性。

2) 相对于使用固定会话长度的识别方法，改进的方法可以识别长会话。因为会话长度固定的识别方法会话最长为 θ，一般设置为 25.5 min，而实际应用中会有较多的长会话存在，改进的方法就可以识别出这些长会话。

3 结束语

采用基于过滤框架网页与页面访问时间阈值相结合的会话识别方法，发现 Frame 页面极大地影响了挖掘结果的兴趣性。文中在现有数据预处理技术的基础上提出先消除 Frame 页面对挖掘结果的影响，再使用页面访问时间阈值来进行会话识别，经过实验数据的检验，改进后的数据预处理技术使得挖掘结果的兴趣性显著提高。

参考文献：

[1] Han Jia - Wei, Meng Xiao - Feng, Ang Jing. Research on Web Mining[J]. Journal of Computer Research & Development, 2001, 38(4): 405 - 414.

[2] Fu Y, Sandhu K, Shih M. A generalization - Based Approach to Clustering of Web Usage Session[C]// Proc 1999 KDD Workshop Web Mining, LNCS 1863. [s.l.]: Springer - Verlag, 2000: 21 - 28.

[3] Cooley R, Mobasher B, Srivastava J. Data Preparation for Mining World Wide Web Browsing Patterns[J]. Knowledge and Information system, 1999, 1(1): 5 - 32.

[4] Spiliopoulou M, Mobasher B, Berendt B, et al. A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis[J]. Informs Journal of Computing, 2003, 15(2): 171 - 179.

[5] Chen M S, Park J S, Yu P S. Data Mining for Path Traversal Patterns in a Web Environment[C]// Proc 16th Int'l Conf Distributed Computing System(ICDCS96). [s.l.]: IEEE CS Press, 1996: 385 - 392.

[6] 熊忠阳，周亚峰. Web 访问挖掘的预处理技术的研究[J]. 计算机技术与发展，2007，17(8)：11 - 14.

[7] Srivastava J, Cooley R, Deshpande M, et al. Web Usage Mining: Deiscovery and Applications of Usage Patterns from Web Data[J]. Proc ACM SIGKDD, 2000, 1(2): 12 - 23.

[8] 金松河，钱慎一，张素智. Frame 页面过滤算法在 Web 日志挖掘预处理中的应用[J]. 云南民族大学学报：自然科学版，2006，15(1)：63 - 65.

(上接第 213 页)

Commondity Grid Kit[J]. Concurrency and Computation: Practice and Experience, 2001, 13(15): 1045 - 1055.

[4] The globus 联盟. Package org.globus.ftp[EB/OL]. 2004 - 10. http://www.globus.org/cog/distribution/1.2/api/org/globus/ftp/package - summary.html.

[5] Foster I, Kesselman C. 网格计算[M]. 第 2 版. 金 海，袁平鹏等译. 北京：电子工业出版社，2004：233 - 237.