

ERFT: 一种增强型的可靠文件传输工具^①

王圆春¹, 余芳², 管康萍³, 王先义¹

¹(中国电波传播研究所, 青岛 266107)

²(山东省农村信用联合社, 青岛 266107)

³(中兵光电科技股份有限公司, 青岛 266107)

摘要: 网格数据传输组件 RFT 提供了可靠文件传输服务, 但是 GT4 中的 RFT 仍存在不支持 xml 配置文件、不能进行全局优化配置、不能可视化操作等不足。在研究 GT4 RFT 组件的基础上, 设计和实现了一个增强型的可视化可靠文件传输工具——ERFT。对大型文件的传输实验表明, ERFT 继承了 RFT 的传统优势, 同时又改进了其不足, 为用户提供了更为可靠、友好的文件传输服务。

关键词: 网格; RFT; 可靠文件传输; ERFT

ERFT : An Enganced Reliable File Transfer Tool

WANG Yuan-Chun¹, YU Fang², GUAN Kang-Ping³, WANG Xian-Yi¹

¹(China Research Institute of Radiowave Propagation, Qingdao 266107, China)

²(Shandong Rural Credit Cooperatives, Qingdao 266107, China)

³(China North Optical-Electrical Technology, Qingdao 266107, China)

Abstract: RFT, grid data transfer component, provides a reliable file transfer service, but the RFT in GT4 is still deficient. It does not support the xml configuration file, and can not be globally optimal allocation, and has not visual manipulation. Based on the study of GT4 RFT component, we designed and implemented a visualized reliable file transfer tool - ERFT. Experimental transmission of large files show that, ERFT inherited the RFT's traditional strengths, while improving its shortcomings, provides users with more reliable and friendly file transfer services.

Key words: Grid; RFT; reliable file transfer; ERFT

文件传输是网格中最重要的功能之一^[1], 它与资源管理、作业调度、信息服务等构成一个完整的网-格系统, 为多个虚拟组织之间的资源共享与协同问题求解提供了支撑^[2]。为了提高传输的可靠性, Globus 提供了可靠文件传输工具 (Reliable File Transfer, RFT)。

RFT 提供了一些 API 和命令, 尽管可以满足网格中的大数据传输, 但是它仍然存在一些缺陷和不足, 主要如下:

1. 配置文件采用文本类型, 缺少良好的规范性和可读性。
2. 传输任务的配置文件, 所有的 URL 对都是采用

全局配置, 不能根据特定传输任务的情况设置最优参数。

3. 用户必须使用 Grid shell 命令执行相关操作, 或者在此基础上进行二次开发。而对一般用户而言, 屏蔽底层的实现细节和接口规范, 提供友好的可视化操作平台是很有必要的。

针对 RFT 工具的缺点及不足, 本文在研究 RFT 的基础上, 开发了一个增强型 RFT 工具——ERFT。ERFT 提供了文件传输和文件删除的集成功能, 配置文件支持 XML 类型, 允许对特定的传输和删除任务进行参数配置和优化; 提供了可视化操作界面, 用户只需要点击鼠标就可以完成文件传输。

^① 基金项目:国家重大专项基金(2010ZX03006-002-03)

收稿时间:2011-09-20;收到修改稿时间:2011-12-27

1 RFT介绍

在 GT4 中, 提供了 GridFTP 和 RFT 两种数据管理服务。GridFTP 是全球网络论坛基于 GFD.020, RFC 959, RFC 2228, RFC 2389 而定义的一个协议, 它提供了安全、可靠、快速和高效的网格数据传输, 但是有些方面却需要优化^[3]。首先, GridFTP 协议不是一个 Web 服务协议 (没有使用 SOAP 和 WSDL 等); 同时, 在整个传输期间, GridFTP 要求客户端始终维持一个开放的套接字连接到服务器, 对于长期传输任务而言, 这显然是不合理的。

为此, Globus 在 GridFTP 的上层引入了 RFT 服务和后台数据库的支持。RFT 是基于 Web 服务资源框架 (WSRF) 标准的网络服务, 提供一个永久、易调用、可靠的网格文件传输服务。它使用 HTTP 上的标准 SOAP 消息, 通过 Web 服务容器和非常简单的客户端, 为用户提供 GridFTP 的第三方数据传输和部分文件管理功能、传输动态及性能监测, 自动恢复传输等功能^[4]。

RFT 服务建立在 GridFTP 的客户端之上, 一方面它继承了 GridFTP 的特点, 同时还避免了故障后重新传输的缺点。在任务完成之前, 它将传输状态以永久的方式存储在数据库中, 当故障发生时, 从数据库中获取故障时的传输状态, 并从因故障中断的地方继续传输。

2 ERFT的设计

2.1 ERFT 服务架构

本文设计的 ERFT 整体架构如图 1 所示。主要包括客户控制 GUI、直接控制 GUI、性能 GUI、ERFT 服务、GridFTP 传输客户端、网络记录器和传输数据库等模块。

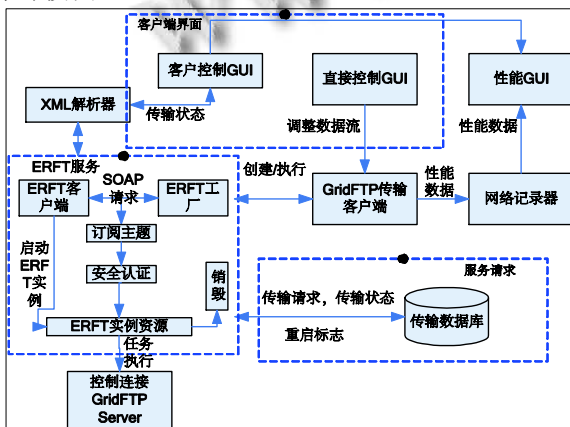


图 1 ERFT 服务架构

客户控制 GUI 用来管理传输服务请求和状态, 直接控制 GUI 用来调整一个活动传输的并行流数目及 TCP 缓冲区大小, 性能 GUI 用来显示吞吐量和时间之间的性能图, 其数据源自于网络记录器中传输客户端的性能标记, 传输过程中的状态数据存储在传输数据库中, 当故障发生时, ERFT 服务将会读取该数据库中的数据恢复。

ERFT 接收来自客户端 GUI 的传输请求, 并将其转发给 ERFT 工厂, ERFT 工厂负责创建一个 ERFT 服务实例。该服务实例调用传输客户端, 启动一个从源到目标 GridFTP 服务器的第三方控制数据传输, 将传输请求永久存储到传输数据库中, 并实时更新数据库中描述传输的状态信息。如果在传输过程中, 由于客户端或者某个资源失效导致任务失败, 那么当所有资源可用时, ERFT 服务将会根据传输数据库的状态信息, 启动故障恢复和重传。

2.2 主要的数据结构

(1) 服务请求:

BaseRequest: 存储服务请求的参数, 是传输请求类 **TransferRequest** 和删除请求类 **DeleteRequest** 的父类;

Transfer: 存储传输子任务的 **Source Url** 和 **Destination Url** 等参数;

RFTOptions: 存储可选的参数, 如: **binary** (是否是二进制传输)、**dcau** (是否是数据信道认证)、**blockSize** (数据块大小)、**tcpBufferSize** (tcp 缓冲大小);

TransferRequest: 继承 **BaseRequest**, 存储传输任务请求, 可选参数和子任务数组;

DeleteRequest: 继承 **BaseRequest**, 存储删除任务请求;

(2) 客户端界面:

RFTInputDialog: 选项界面, 有选取文件 GUI、xml 传输、帮助和退出四个选项;

TransferOptionFrame: 显示和修改 RFT 传输任务全局参数;

DeleteOptionFrame: 显示和修改 RFT 删除任务全局参数;

RFT Frame: 用户选取文件界面, 可以进行选择文件、设置参数、修改和删除文件任务等。RFT Frame 包含四个内部类: 按钮监听器 **ButtonActionListener**、

菜单项监听器 ItemActionListener、列表框监听器 ListMouseListener、输入文本框监听器 TextActionListener;

(3) XML 配置文件解析:

MyDefaultHandler: 解析和验证 xml, 继承自 org.xml.sax.helpers.DefaultHandler;

MySAXApp: 获取 xml 配置文件, 调用 MyDefaultHandler, 验证、解析并输出结果;

(4) 服务执行:

BaseRFTClient:

继承 org.globus.wsrft.client.BaseClient 并实现 org.globus.wsrft.NotifyCallback 接口, 由它派生两个子类 ReliableFileTransferClient 和 ReliableFileDeleteClient, 分别负责执行文件传输服务和文件删除服务。

3 核心模块的实现

ERFT 服务的核心模块, 是采用 Web Service 的 WS-Resource 工厂模式进行开发, 主要涉及安全认证、创建资源、订阅主题、通知、任务执行和资源销毁。

3.1 安全认证

保证安全是文件传输中的重要问题, 在 ERFT 服务中, 用户通过 DFS (Delegation Factory Service) 的 URL, 从资源属性里获取 DFS 的证书链, 证书链的首节点就是代理服务的公开证书。用户把自己的代理证书和代理服务的公开证书进行绑定, 产生委托证书, 把该证书发送给 DFS, 委托证书的端点引用就会返回用户。这样, 用户就可以使用这个端点指引来调用经过代理认证的服务。

委托证书序列如图 2 所示。

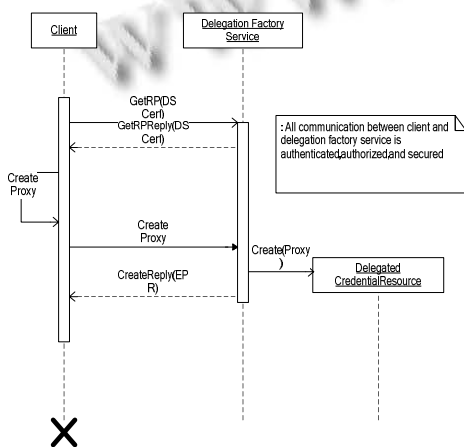


图 2 代理服务序列图

3.2 创建 ERFT 资源

在 WSDL 文件中预定义创建 ERFT 资源的操作 createReliableFileTransfer, 并定义 input 和 output 消息。在 API 中, 对应的两个类 CreateReliableFileTransfer InputType 和 CreateReliableFileTransferOutputType 对预定义的 input 和 output 消息进行详细描述。创建 CreateReliableFileTransferInputType 对象 input, 根据任务类型设置操作请求。调用 factoryPort.createReliableFile Transfer(input) 来创建文件传输的资源, 并在数据库上创建相应的表来存储传输的状态, 以保证文件的可靠传输。

3.3 订阅主题

创建一个 Subscribe 对象 request, 通过 setUseNotify 激活通知, 根据 PROTOCOL 协议, 通知 Notification ConsumerManager (消费者管理器) 来创建不同的实例 consumer, 创建之后 consumer 便进入消息监听状态。

根据 authz 授权类型的不同, 创建资源级别的安全描述符。此处, 我们定义了两类授权: 一类是 host 授权, 需要定义一个授权链名为“host”的 pdpConfig 对象, 该对象实现了 PDPConfig 的 ResourcePDPConfig 属性, 然后根据 ServiceAuthorizationChain 的初始化方法对 authz 初始化, 并将其作为资源安全描述符的授权链; 另一类是 self 授权, 直接把“self”授权链作为资源安全描述符的授权链即可。

对于资源安全描述符的认证, 通常需要根据 PROTOCOL 协议的类型来设置。根据 consumer.CreateNotificationConsumer(newBaseRFTClient(), resDesc) 返回的端点引用 consumerEPR.request, consumer 订阅主题 RFTConstants.OVERALL_STATUS_RESOURCE, 该主题即为文件传输和删除过程中的所有状态, 包括文件的完成数目, 失败数目, 重试数目等, 这样就完成了客户端的订阅。

3.4 通知

通知模块以 Notification Producer 身份向 Notification Consumer 通知其所订阅的主题, 该模块主要是实现了 GT4.0 标准接口 org.globus.wsrft.NotifyCallback 中的方法。用户发送消息获取资源属性变化的通知 change Message, 该消息经过类型转换为 Overall Status, 就可以得到服务完成、失败、挂起、重试等状态。

3.5 执行传输、删除

完成以上工作之后, rft.start(new Start())启动文件传输或删除, 并实时刷新服务状态, 一旦传输或删除结束, 即成功传输任务的值等于总的文件数量, 或者失败任务的值等于总的文件数量, 或者二者的和等于总的文件数量, 将结果状态返回给客户端。在此过程中, 用户也可以为 ERFT 设置生命周期, 一旦周期结束, 不管操作是否执行成功, 都将自动结束整个过程。

3.6 销毁资源

因为网格中涉及数据传输或者删除的任务极为频繁, 而传输的文件数一般较多, 导致存储服务状态的数据表条目迅速增长。为了保证网格传输中有足够的空间来存储服务状态, 每次服务完成之后都要销毁这些资源。服务状态存储在 PostgreSQL 数据库系统的 rftDataBase 数据库中, 任务执行完之后, 系统自动销毁相关记录。

4 系统测试

客户端是基于 GT4 的数据管理组件进行开发的, 操作系统选择 Fedora Linux 6.0, 开发平台使用 JDK 1.6、Eclipse 3.0 和 Apache ANT 1.7, 数据库采用 PostgreSQL, 开发语言使用 java。

利用 8 台服务器节点搭建网格平台, 节点通过 1000Mbps 以太网连接, 搭建 Globus Toolkit 4.2 网格中间件, 并部署本文设计实现的 ERFT 服务和客户端。

利用上述平台, 进行节点间大文件数据传输实验。如图 3 所示, 用户通过图形界面进行文件的选取。

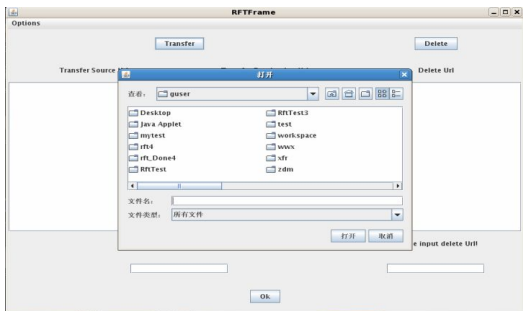


图 3 文件选择界面

针对不同的传输和删除任务, 允许用户根据特定的任务请求设置最优参数, 这些参数保存在 XML 类型的配置文档中, 如图 4。

服务结束之后, 将服务执行状态返回给客户端, 此处的服务结束可以是正常结束, 也可以是出现故障而中断, 见图 5。

整个测试过程表明, ERFT 在原有 RFT 基础上, 实现了可靠文件传输中对 XML 配置文件的完美支持, 实现对传输文件过程中启动、认证、传输和销毁等任务的全局优化配置, 并增加了可靠文件传输可视化操作, 实现了最初的设计目标。

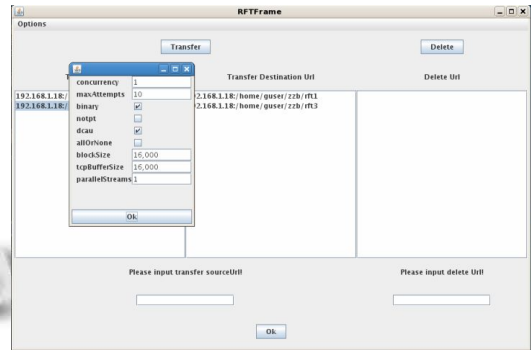


图 4 用户设置参数界面

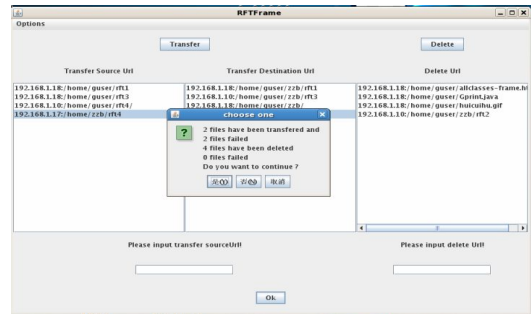


图 5 服务状态反馈界面

5 结语

本文在研究 RFT 工具的基础上, 分析了它存在的不足和缺陷, 设计并实现了增强型网格可靠文件传输工具 ERFT。ERFT 增加了对 XML 配置文件的支持, 实现了根据特定任务请求设置最优参数的功能, 并且为用户提供了可视化操作界面。下一阶段将研究 ERFT 如何在 RFT 的基础上, 进一步提高数据传输效率的问题。

参考文献

- 1 徐志伟,冯百明,李伟.网格计算技术.北京:电子工业出版社,2004.
- 2 Foster I,Kesselman C.网格计算第二版.金海,袁平鹏,石柯,译.北京:电子工业出版社,2004.
- 3 应宏,李梦蛟,刘福明,钟静,都沛.GridFTP 数据传输模型研究.计算机科学,2006,33(5).
- 4 The globus' alliance. http://www-unix.globus.org/toolkit/docs/