第 36 卷 第 2 期 计 算 机 工 程 2010 年 1 月
Vol.36 No.2 Computer Engineering January 2010
·开发研究与设计技术· 文章编号：1000—3428(2010)02—0247—02 文献标识码：A 中图分类号：TP391

基于 Internet 的高效文件备份方法

赵昊汉，胡晓勤，邓洪敏，马晓旭
(四川大学计算机学院，成都 610065)

摘 要：针对当前文件备份领域中存在备份效率低、速度慢的问题，提出一种基于差异的远程文件备份方法。改进传统 Rsync 算法，实现差异文件的计算功能，从而减少网络数据传输，保证传输的高效性。实验结果表明，与传统文件备份方法相比，该方法能减少网络流量，提升备份与恢复的效率。

关键词：文件备份；差异计算；摘要文件

High-efficiency File Backup Method Based on Internet

ZHAO Hao-han, HU Xiao-qin, DENG Hong-min, MA Xiao-xu
(School of Computer Science, Sichuan University, Chengdu 610065)

【Abstract】 Aiming at the problem of low efficiency and slow speed in file backup area, this paper proposes a remote file backup method based on files difference. Through improvement of traditional Rsync algorithm, it achieves the differences of document in calculation functions, reduces the data transmission network to ensure the transmission efficiency. Experimental results show that this method can significantly reduce network traffic and greatly enhance efficiency of backup and recovery compared with traditional file backup method.

【Key words】 file backup; difference calculation; abstract file

传统文件备份业务大都面向企业级，基于资金的考虑，个人用户能够获得的备份服务相对较少。若能在远程构建一个安全稳定的文件备份中心，定时的通过 Internet 自动将用户的重要文件备份上去，则用户本地计算机即使遭遇灾难，文件也不会丢失[1]。传统的远程文件备份工具以 Ftp 和 Rsync[2-3]为主。其中，Ftp 只支持完全备份，效率偏低；Rsync 算法虽能实现文件的差异同步，但是并不适合直接用来构建文件备份中心，它需要改进以适应本文件备份系统。

1 高效文件备份方法

1.1 Rsync 算法

在低带宽、高延迟的链路上传输文件的步骤如图 1 所示。

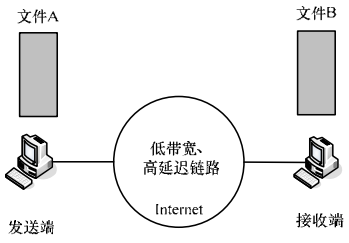

图 1 在低带宽、高延迟的链路上传输文件

由图 1 可知，Rsync 算法提供了一种快速的文件传输方法，即仅传送服务器上文件的不同之处，而不需要把 2 端的文件预先拷贝到一台服务器上进行比较，从而达到很快的文件传送速度。

该算法具体过程如下[4-5]：

(1)服务器 Server 将文件 f_{old} 进行块长为 k 个字节的分块。

(2)服务器 Server 计算每个数据块的滚动校验和(弱校验和)与强校验和。

(3)服务器 Server 建立哈希表，对于每一对校验值，以滚动校验值为关键值进行哈希排序，表示为 $h_i<R_i, M_i>$，并将哈希表传送给客户端 Client。

(4)客户端 Client 从头搜索文件 f_{new}，产生滚动校验和与 $h_i<R_i, M_i>$中的 R_i 比较，发现与 R_i 相匹配的数据块，再计算其 MD4 哈希值与强校验和 M_i 比较，如果强校验和相等，则说明这 2 个数据块相同；如果不匹配，则说明这 2 个数据块不同，记录该数据块的第 1 个字节，然后向后移动一个偏移量，继续比较，直到文件末尾。

(5)所有比较完成后，服务器 Server 收到客户端 Client 包含差异内容和哈希指示值的数据流，更新 f_{old}，生成 f_{new}。

1.2 基于 Rsync 的文件备份改进方法

传统的 Rsync 算法每找到新旧文件的一处差异或者相同块，就会发指令给服务器端。这样就增加了客户端与服务器之间的交互，在网络状况不是很稳定的 Internet 上，这样的行为会影响文件备份的效率。如果在计算差异时，客户端与服务器端的交互尽可能少，那么就可以将网络带宽以及网速的影响降至最低。

改进的文件备份方法采用的算法流程为：在客户端计算差异，生成差异文件，保存差异计算的结果，把计算结果一次性发给服务器，其减少了网络开销。

1.2.1 计算差异

对于第 2 个子任务计算差异，可以分为 4 个步骤来完成：

基金项目：国家自然科学基金资助项目(60573130, 60502011, 60873246)；国家“863”计划基金资助项目(2006AA01Z435)；教育部博士点基金资助项目(20070610032)

作者简介：赵昊汉(1984－)，男，硕士研究生，主研方向：网络安全技术及应用；胡晓勤，讲师、博士研究生；邓洪敏，博士后；马晓旭，博士研究生

收稿日期：2009-06-24 **E-mail**：wojiaoawen@163.com

分块，摘要文件计算，滚动校验和排序，搜索和差异文件对生成。

(1)分块：将文件 $f_j \in F_1$ 以 k 字节为单位分成一系列没有重叠的数据块。由于差异计算的复杂度和精确度很大程度上取决于分块大小，因此 k 值的大小一般是动态变化的，主要由文件大小决定。

(2)摘要文件计算：对于每个分块进行滚动校验计算和 MD4 校验计算，计算结果表示为 $h_j\langle R_j, M_j\rangle$，其中，$R_j$ 是滚动校验和；M_j 是 MD4 校验值。

(3)滚动校验和排序：对计算出的滚动校验和进行哈希排序，将排序结果放入一个 16 bit 的哈希表中，然后创建一个 16 bit 的索引表，每一项索引值指向哈希表的每一个入口点。

(4)搜索和差异文件对生成，对该步骤分解描述如下：

1)对文件 f_i 中偏移量为 l、文件大小为 k 的数据块计算滚动校验和 R_i，并对 R_i 进行哈希计算，然后搜索索引表和哈希表，查找想匹配的校验和。

2)如果没有匹配的校验和，则把 f_i 的第 l 个字节记录到 f_D 中，得到 $l = l+1$，转到 1)。

3)如果找到匹配的校验和，则计算该数据块的 MD4 校验值 M_i，并与 f_j 相应数据块的 M_j 相比较。若 $M_i = M_j$，则说明 2 个数据块完全一样，记录 f_j 的数据块序号到 f_P 中，得到 $l = l+k$，转到 1)。

4)若 $M_i \neq M_j$，则把 f_i 的第 l 个字节记录到 f_D 中，得到 $l = l+1$，转到 1)。

5)当对文件 f_i 搜索完毕后，会得到匹配数据块的序号和不匹配的数据，它们都保存在 f_P 和 f_D 中。

例如：文件 f_i 的内容为 gacdhef，文件 f_j 的内容为 abcdefgh，分块大小设为 2。通过搜索和差异文件对生成步骤，可得到差异数据文件 f_D 内容为 gah，而差异指示文件 f_P 内容为 2,-2,1,-3，如图 2 所示。

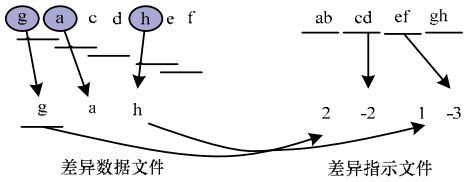

图 2　文件差异集生成过程

1.2.2　重构差异

在服务器端，现有旧文件 F_{old} 和 2 个差异文件 F_{ind}, F_{data}，重构生成新文件 F_{new} 的过程如下：

(1)建立临时文件 F'；

(2)从 F_{ind} 中读取数 n，若已到文件尾，则重构结束；

(3)如果 $n>0$，从 F_{data} 中读取 n 个字节到 F'中，则转到(2)；否则转到(4)；

(4)如果 $n<0$，从 F_{old} 中读取第-n 个数据块到 F'中，则转到(2)；

(5)将 F'取代 F_{old}，重构完成。F'即为新文件。

1.2.3　备份策略

上文介绍了基于 Rsync 的改进算法[6-7]，解决了单个文件带差异的备份。现在，需要定义有效的备份策略，解决文件集合之间带差异的备份。

定义 1 文件：$f(f^{path}, f^{size}, f^{time})$ 分别用来表示文件(路径、大小、最后修改时间)；

定义 2 文件集合：$F = \{f_1, f_2, \cdots, f_i, \cdots, f_n\}$；

定义 3 文件之间差异：$\Delta f = f_i - f_j$ 表示 2 个文件之间的差异；

定义 4 文件差异集合：$\Delta F = F_2 - F_1$ 表示在 t_1 时刻、t_2 时刻文件集合 F_1，文件集合 F_2 之间的差异。

假设文件集合 F_1 已经在时刻 t_1 备份，现在需要在 t_2 时刻对文件集合 F_2 进行文件备份，备份策略过程为：

(1)对于 $\forall f_i \in F_2$，$\exists f_j \in F_1$，有 $f_i^{path} = f_j^{path}$，表明 f_i 在时刻 t_1 已备份，转到(2)，否则表明为新增文件，将 f_i 加入文件差异集 ΔF。

(2)若 $f_i^{size} = f_j^{size}$ 且 $f_i^{time} = f_j^{time}$，则表明 2 个文件相同，不需要备份，转到(1)，继续处理文件集合 F_2 的其他文件，否则转到(3)。

(3)利用差异算法，计算文件之间的差异，生成差异文件，加入到文件差异集 ΔF 中。

(4)将文件差异集合 ΔF 由客户端传输到服务器端；

(5)在服务器端根据原文件和 ΔF 进行差异重构，生产新文件。

2　文件备份实验

2.1　实验环境

客户端操作系统为 MS Windows2003，文件系统为 NTFS，实验测试的 Rsync 系统版本为 cwRsync 2.0.10，网络流量的统计工具为 Net Meter v1.1.2。

2.2　实验内容

本实验测试的对象为 Rsync 备份方法和本文所述文件备份方法，针对不同大小的文件，当其改变内容大小、不大时，测试它们实现同步操作时所传输的网络流量。根据对用户日常改变文件的统计，把改变文件的大小统一为 5 KB 和 50 KB。

2.3　实验结果

实验结果如表 1 所示，实验结果的网络流量对比如图 3 所示，其中，L 为测试文件大小。

表 1　实验结果　　KB

文件大小	文件内容变化大小			
	5		50	
	Rsync	本方法	Rsync	本方法
200	30.3	11.6	75.1	65.5
600	62.5	13.1	116.6	76.1
1 800	94.1	14.5	169.2	89.2
5 400	183.2	18.4	239.9	108.4
16 200	315.0	31.9	388.5	115.6

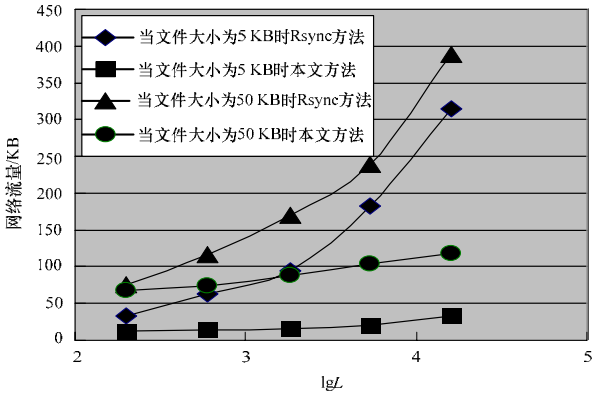

图 3　网络流量对比　　(下转第 251 页)