远程文件快速同步方法

胡晓勤[1]，卢正添[1]，刘晓洁[1]，李 涛[1]，赵庆华[2]，赵 奎[1]

(1. 四川大学计算机学院 成都 610065; 2. 四川大学网络教育学院 成都 610065)

【摘要】远程文件快速同步在文件备份与恢复、Web缓存等应用场景有广泛的应用。该文给出了一种新的文件快速远程同步方法，在客户端计算文件新旧版本之间的状态差异集和在服务器端重放文件状态差异集实现文件内容差异同步和重命名优化；给出了文件状态差异集的生成、重放和优化方法。实验结果表明，与rsync相比，文件内容差异同步网络流量显著降低，同时，重命名优化网络流量为KB级，与文件长度无关，显著地降低了网络流量。

关 键 词 文件同步； 重命名优化； rsync； 快照

中图分类号 TP309.3 **文献标识码** A

A Remote File Fast Synchronization Method

HU Xiao-qin[1], LU Zheng tian, LIU Xiao-jie[1], LI Tao[1], ZHAO Qing-hua[2], and ZHAO Kui[1]

(1. School of Computer, Sichuan University Chengdu 610065; 2. College of Distance Education, Sichuan University Chengdu 610065)

Abstract Fast synchronization for remote file is widely used in many scenarios such as file system backup and restore, web mirroring, etc. A new method for remote file fast synchronization is proposed. The file state difference set is calculated at client end. The difference synchronization and rename optimization are achieved by replaying the file state difference set at server end. The generation, replay and optimization methods for the file state difference set are given. The experimental results indicate that this method offers signficant improvement over rsync. The network traffic of file content synchornization decreases remarkablely. And the network traffic of rename optimization is KB-level and independent of file length.

Key words file synchronization; rename optimization; rsync; snapshot

远程文件同步在文件备份与恢复、Web缓存等方面应用广泛。文件远程同步问题可描述为：给定一个文件f的两个版本，新版本f_{new}存储在客户端C上，旧版本f_{old}存储在服务器S上，通过网络，如何以最小的通信量更新文件的旧版本。特别是当两个版本之间的差异较小时，通信传输量应该显著小于文件长度[1]。目前，文件同步研究取得的主要研究成果包括rsync[2-3]、unison[4]及其改进方法[5-6]和应用系统[7-8]，国内研究侧重压缩算法[9]和复制策略[10]。它们的主要方法是计算客户端和服务器文件的差异，通过差异计算更新服务器文件。在实践中，往往需同步一组文件，文件变化可由多种操作引起，如创建、复制、删除、重命名、文件内容变化等。这些操作导致文件内容和名称的变化，可称之为文件状态变化。

对于同步一组文件，以下问题不能忽略：(1) 差异计算产生的网络流量不能忽略，就rsync来说，即便f_{new}和f_{old}相同，也会产生网络流量；(2) 重命名问题，当客户端文件由旧版本文件移动或复制产生，如何快速同步这类文件。对于文件复制，rsync作为新文件传送到服务器；而对于文件移动，rsync首先删除旧文件，再传送新文件到服务器，这造成网络带宽严重浪费，特别在文件较大时。

本文提出在客户端计算一组文件新旧版本之间的差异，生成文件状态差异集，统一描述文件内容变化和重命名，通过网络传输到服务器端后重放文件差异集，实现文件同步。实现时，可使用快照技术在客户端保存文件的新旧版本。

1 相关技术

1.1 rsync

为快速同步文件，文献[3]提出了rsync算法。该算法的核心是滚动校验和生成算法，当已知给定数据块的校验和，可快速计算下一个偏移量数据块的

收稿日期：2007－10－14; 修回日期：2008－03－19
基金项目：国家863计划项目(2006AA01Z435)；国家自然科学基金(60573130; 60502011)
作者简介：胡晓勤(1977－)，男，博士，讲师，主要从事网络安全、电子政务方面的研究.

校验和，算法细节参见文献[2-3]。

rsync运行的简单例子如图1所示，图中ABCD、ACBA、HJWZ和XQNL均为文件内容。假设客户端C有文件f_{new}，服务器S上有文件f_{old}，初始时刻这2个文件相同。现文件f_{new}部分内容变化，为同步S上的f_{old}，采用的步骤为：(1) S将文件f_{old}以k字节分成一系列没有重叠的数据块，共n块。如果最后一块不足k字节，则填充为k字节。(2) S计算所有数据块的校验和，即滚动校验和R_i与强校验和M_i (MD4散列值)，表示为h_i<R_i, M_i >并传送给C。(3) C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，再计算其强校验和与M_i比较，如果强校验和相等，说明这两个数据块相同，记录数据块的索引，即指示值，继续比较下一个数据块；如果不相等，说明这两个数据块不同，则记录该数据块第一个偏移量的内容如T和F，然后向后移动一个偏移量，继续比较。(4) 比较完成后，S收到客户端C包含差异内容和指示值的数据流，更新f_{old}，生成f_{new}。

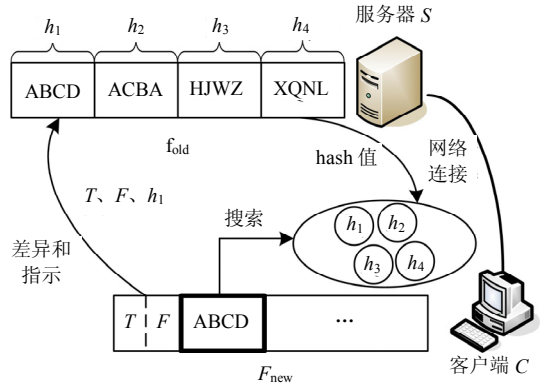

图1　rsync算法举例

rsync计算文件差异产生的网络流量为：滚动校验值、MD4散列值、更新数据和指示值。即使文件没有差异，也要发生流量。而当文件内容无变化，只是文件名或者路径改变，rsync将这种文件作为新文件处理，作删除和重传操作，产生多余网络流量。

1.2　快 照

快照是关于指定数据集合的一个完全可用拷贝，包括相应数据在某个时间点(拷贝开始的时间点)的映像。快照实现技术包括分割镜像、按需复制、虚拟视图等。虚拟视图快照在快照命令发出之后，只建立一份快照时刻源数据的逻辑副本，只需分配少量的存储空间，用于保存快照时间点之后源数据中被更新的数据。复制操作只发生在源数据变化时，因此系统开销分散，对系统性能影响小，也不产生网络流量，适合数据备份应用。目前主流操作系统已实现快照技术。

2　远程文件快速同步方法

2.1　术语定义

(1) 文件f_i，指需要同步的一个文件(目录名表示为文件)包括绝对路径和文件名。(2) 文件集F，指多个需要同步的文件组成的集合：$F=\{f_1,f_2,\cdots,f_i,\cdots,f_n\}$，在$t_1$时刻记为$F_1$，在$t_2$时刻记为$F_2$，其中$t_1$< t_2。(3) 文件差异集$\Delta F=(F_1-F_2)\cup(F_2-F_1)$，其中$F_1$、$F_2$为$t_1$和$t_2$时刻的文件集。文件集没有差异，即需要同步的文件和目录没有增减时，则$\Delta F=\Phi$。(4) 文件状态$s_i^{\mathrm{t}}<f_i^{\mathrm{p}},f_i^{\mathrm{c}}>$，一个文件在$t$时刻的状态，包括绝对路径和文件名$f_i^{\mathrm{p}}$、文件内容$f_i^{\mathrm{c}}$。(5) 文件状态集$S^t$，指一组文件在$t$时刻的状态，可表示为$S^t=\{s_1^t,s_2^t,\cdots,s_i^t,\cdots,s_n^t\}$。(6) 文件状态差异$\Delta s_i<\Delta f_i^{\mathrm{p}},\Delta f_i^{\mathrm{c}}>$，指文件集$F_1$和$F_2$中一个文件在$t_1$和$t_2$时刻之间的状态变化，表示为$\Delta s_i^{2,1}=s_i^{t_2}-s_i^{t_1}$。文件$t_2$时刻状态可由$t_1$时刻状态和文件状态差异计算得到，表示为$s_i^{t_2}=\Delta s_i^{2,1}+s_i^{t_1}$。(7) 文件状态差异集$\Delta S$，由文件集$F_1$和$F_2$中一个或一组文件状态差异$\Delta s_i$组成的集合$\Delta S=\{\Delta s_1,\Delta s_2,\cdots,\Delta s_i,\cdots,\Delta s_n\}$。(8) 运算$F_{\mathrm{R}}=O_c(f_i^{\mathrm{c}},f_k^{\mathrm{c}},F')$，比较文件$f_i$和每个$f_k\in F'$文件内容是否相同，若相同，输出文件名$f_k^{\mathrm{p}}$组成的集合$F_{\mathrm{R}}$，否则$F_{\mathrm{R}}=\Phi$。(9) 运算$f^{\mathrm{p}}=O_p(f_i^{\mathrm{p}},F'')$，判断文件$f_i$的文件名$f_i^p$是否属于$F''$。若$f_i^p\in F''$，则输出$f^{\mathrm{p}}=f_i^{\mathrm{p}}$，否则输出空。

推论 1　如果$\Delta S=\Phi$，则文件状态无变化或文件状态差异已全部更新。

推论 2　如果$\Delta F\neq\Phi$，则$\Delta S\neq\Phi$，当文件集发生变化，则一定会产生文件状态差异集。

2.2　文件状态差异集生成

文件状态差异集生成方法如图2所示。对于任何$f_i\in F_2$，Δs_i生成过程如下：(1) 在文件集F_1中搜索是否存在内容相同的文件，结果记为F_{c}。(2) 若F_c不为空，则判断f_i^{p}是否属于F_{c}，若$f_i^{\mathrm{p}}\in F_{\mathrm{c}}$，则文件状态无变化，若$f_i^{\mathrm{p}}\notin F_{\mathrm{c}}$，则文件由复制、移动产生。(3) 若$F_{\mathrm{c}}$为空，则文件内容发生变化或新文件产生，则判断$f_i^{\mathrm{p}}$是否属于$F_1$。(4) 若$f_i^{\mathrm{p}}\in F_1$，则说明文件名未变，$f_i$文件内容差异为$\Delta f_i^{\mathrm{c}}=\mathrm{rsync}(f_i^{\mathrm{c}},f_j^{\mathrm{c}})$，可由经典rsync算法计算，其中$f_i^{\mathrm{c}}$为$f_i$新文件内容，而$f_j^{\mathrm{c}}$为$f_i$旧文件内容；若$f_i^{\mathrm{p}}\notin F_1$，则为新文件，文件差异为$f_i$。

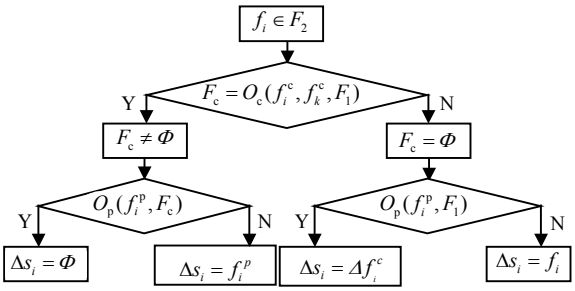

图2 文件状态差异集生成方法流程图

当计算完F_2中的所有元素后，应计算集合F_1和F_2的差$\Delta F' = F_1 - F_2$，若$\Delta F' \neq \Phi$，说明F_1中部分文件已经删除。

2.3 运算F_R的优化

$F_R = O_c(f_i^c, f_k^c, F')$用于判断任意$f_i^c$和集合$F'$中某个文件的内容是否相同。计算这两个文件的散列值是最常用的方法，但这种方法的性能较差。f_i^c可表示为$< f_i^l, f_i^t, f_i^h >$，其中f_i^l为文件长度；f_i^t为文件最后修改时间，f_i^h为文件散列值。

快速判断方法为：(1) 比较文件长度f_i^l，如果f_i^l不等，则两个文件一定不同。(2) 如果长度相等，则比较f_i^t，如果f_i^t不同，则不同，否则相同。

精确判断方法为：(1) 比较文件长度f_i^l，如果f_i^l不等，则两个文件一定不同；(2) 如果长度相等，则比较f_i^h，如果f_i^h不同，则不同；否则相同。

2.4 文件状态差异集存储和传输

使用多个文件(FS_r,FS_c,FS_n,FS_d)保存文件状态差异集。其中FS_r保存文件内容相同的文件名，即$\Delta s_i = \Delta f_i^p$，每个$\Delta f_i^p$表示为$< f_k^p, f_i^p >$，其中$f_k^p$表示源文件名；$f_i^p$表示目标文件名。如(\Pb\b3.dat, \Pa\b2.dat)，表示\Pb\b3.dat的文件内容来自\Pa\b2.dat。FS_c保存发生文件内容差异的文件名和差异内容，即$\Delta s_i = \Delta f_i^c$，可按照rsync定义的格式保存。$FS_n$表示新增文件的文件名和文件内容，即$\Delta s_i = f_i$。$FS_d$表示被删除文件，即$\Delta F' = F_1 - F_2$，每条记录$r_d$为被删除文件的文件名。

文件状态差异集传输可用ftp协议或者http协议。在实现中，本文选择了ftp协议实现。

2.5 文件状态差异集重放

当文件状态差异集传送到服务器端后，重放文件状态差异集，即执行$S^2 = \Delta S + S^1$，完成一组文件同步。重放顺序为：FS_r→FS_c→FS_n→FS_d。

FS_r的重放重现客户端文件的复制、移动操作及依赖关系，重放步骤为：(1) 生成复制、移动脚本。(2) 执行脚本。

FS_r脚本必须处理交换文件名现象，即一个记录的目标文件名同时也是另外一条记录的源文件名。如$< f_i^1, f_k^2 >$和$< f_k^2, f_i^1 >$两条记录同时出现在FS_r中。设FS_r文件中所有的记录组成的集合为R，每条记录为$r < r_s, r_t >$，其中$r_s = f_k^p$为源文件名，$r_t = f_i^p$为目标文件名。脚本生成算法首先对FS_r进行预处理：(1) 扫描文件FS_r的所有记录，计算每个r_t的路径深度，并按照路径深度降序排列。(2) 生成源目标文件名r_s集合F_s。(3) 生成目标文件名r_t集合F_t。

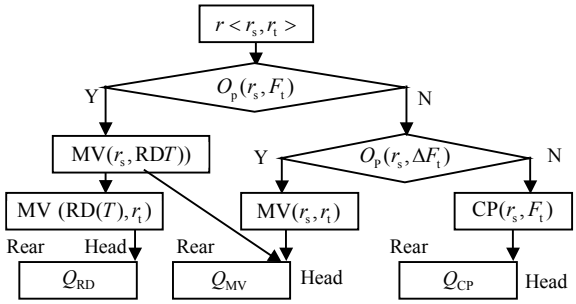

图3 重命名脚本生成方法流程图

重命名脚本生成方法如图3所示，具体步骤如下：(1) 声明双向队列Q_{CP}、Q_{MV}和Q_{RD}，其中Q_{CP}保存复制操作记录、Q_{MV}保存重命名操作和文件名交换前半部分操作记录；Q_{RD}保存执行文件名交换后半部分操作记录；(2) 对于按照r_t降序排列的每个记录$r < r_s, r_t >$中的r_s，在集合F_t中搜索，若$r_s \in F_t$则发生交换文件名，利用随机数生成函数RD、系统时钟T生成唯一文件名，产生重命名操作脚本$MV(r_s, RD(T))$插入到Q_{MV}队列头，产生脚本Q_{CP}插入到Q_{RD}队列头；(3) 若$r_s \notin F_t$，则在集合ΔF中搜r_s，若$r_s \in \Delta F$，则产生重命名操作脚本$MV(r_s, r_t)$插入到Q_{MV}队列头，否则生成复制操作脚本$CP(r_s, r_t)$插入到Q_{CP}队列头。

FS_r脚本执行的顺序为：(1) 从尾至头遍历队列Q_{CP}，执行每个复制操作；(2) 从尾至头遍历队列Q_{MV}，执行每个重命名操作；(3) 从尾至头遍历队列Q_{RD}，执行每个重命名操作的后半部分。

FS_c重放步骤：(1) 从文件FS_c中取出一条文件差异记录，(2) 利用rsync的重放算法，更新文件；(3) 重复步骤(1)～(2)，直到取出FS_c中的所有记录。

FS_n重放步骤：(1) 从文件FS_n中取出一条生成新文件的记录；(2) 根据新文件名和文件内容，产生新文件；(3) 重复步骤(1)～(2)，直到取出FS_n中的所有记录。

FS_d重放步骤：(1) 扫描文件FS_d的所有记录，计算每条记录的路径深度，并按照路径深度降序排列；(2) 按照排序先后，删除每条记录指示的文件。

3　实　验

客户端、服务器操作系统为MS Win2003，文件系统为NTFS，快照服务为VSS(Volume Shadow Copy Service)。采用SDK版本7.2、cwRsync版本2.0.10和协议版本29，100 Mb/s网络环境。实验数据集包括文件和目录总共995个，占用8 161 MB存储空间。散列算法为128 bit的MD4算法。文件内容更新方法为：选择一个文件随机修改一个字节。文件重命名产生方法为：选择一个文件改变其文件名。变化文件长度从6 KB～92.8 MB。实验结果如表1所示。表中，L为文件长度，lb(L)为对文件长度取2为底的对数，UT为文件更新流量；RT为文件重命名流量；lb(RT)为对文件重命名流量取2为底的对数。

表1　网络流量

L/ lb(L)	UT/KB		RT/lb(RT)	
	rsync	本文方法	rsync	本文方法
6/2.58	37.26	0.39	38.22/5.26	8.23/3.04
153/7.26	39.12	2.62	188.75/7.56	9.78/3.29
587/9.20	42.47	5.23	624.35/9.29	7.56/2.92
1 050/10.04	44.25	7.05	1 085.96/10.08	6.73/2.75
7 146/12.80	55.63	20.06	7 185.88/12.81	8.43/3.08
11 999/13.55	61.11	26.31	12 009.18/13.55	5.32/2.41
92 827/16.50	113.33	75.13	92 900.89/16.50	7.82/2.97

图4为文件内容同步时rsync和本文方法产生的网络流量对比图。结合表1和图4可知，本文方法和rsync相比，显著降低了网络流量。

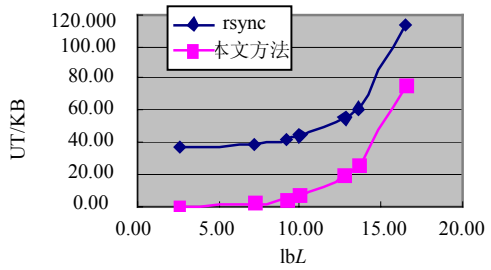

图4　文件内容同步流量对比图

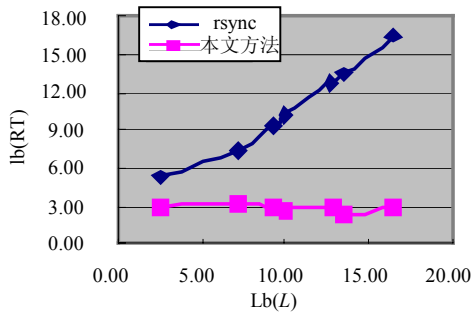

图5　文件重命名流量对比图

图5为文件重命名时rsync和本文方法产生的网络流量对比图。结合表1和图5可知，rsync产生的网络流量同文件长度成正比。本文方法产生的网络流量和文件长度无关，只与重命名前后的文件名长度相关，通常情况下，每个重命名文件的网络流量小于10 KB，通常为KB级，显著降低了网络流量。

4　结 束 语

远程文件快速同步在文件备份与恢复、Web缓存等应用广泛，给出了一种文件快速远程同步方法，在客户端计算文件新旧版本之间的状态差异集，在服务器端重放文件状态差异集实现文件差异同步和重命名优化。实验结果表明，与rsync相比，文件内容同步网络流量有显著降低，文件重命名网络流量为KB级，与文件大小无关，显著降低了网络流量。

参　考　文　献

[1] SUEL T, NOEL P, TRENDAFILOV D. Improved file synchronization techniques for maintaining large replicated collections over slow networks[C]// Proceedings of the 20th International Conference on Data Engineering. New York: IEEE, 2004: 153-164.

[2] TRIDGELL A. Efficient algorithms for sorting and synchronization[D]. Canberra: The Australian National University, 1999.

[3] TRIDGELL A, MACKERRAS P. The rsync algorithm[R]. Australia: Australian National University Canberra, 1996.

[4] PIERCE B C, VOUILLON J. What's in unison? A formal specification and reference implementation of a file synchronizer[R]. Pennsylvania: University of Pennsylvania, 2004.

[5] IRMK U, MIHAYLOV S, SUEL T. Improved single-round protocols for remote file synchronization[C]//Proceedings of the 24th International IEEE Infocom Conference. New York: IEEE, 2005, 3: 1665-1676.

[6] AGARWAL S, CHAUHAN V, TRACHTENBERG A. Bandwidth efficient string reconciliation using puzzles[J]. IEEE Trans on Parallel and Distributed Systems, 2006, 17(11): 1217-1225.

[7] SAI S, JOHN C. Flexible consistency for wide area peer replication[C]//Proceedings of 25th IEEE International Conference on Distributed Computing Systems. UT, USA: IEEE, 2005: 199-208.

[8] LIN W, YE C, CHIU D. Decentralized replication algorithms for improving file availability in P2P networks[C]//Proceedings of 15th IEEE International Workshop on Quality of Service. Hong Kong: IEEE, 2007: 29-37.

[9] 傅　彦, 周俊临, 吴　跃. 快速神经网络无损压缩方法研究[J]. 电子科技大学学报, 2007,36(6): 1245-1248.
FU Yan, ZHOU Jun-lin, WU Yao. Lossless data compression with neural network based on maximum entropy theory[J]. Journal of University of Electronic Science and Technology of China, 2007, 36(6): 1245-1248.

[10] ZHOU X, LU X, HOU M and etc. Research on distributed dynamic replication management policy[J]. Journal of Electronic Science and Technology of China, 2005, 3(5): 97-102.

编　辑　熊思亮