

基于后缀结构进行数据块优化的重复 数据删除系统^①

黄晨晖¹ 林泳琴² (1.广东女子职业技术学院 网络中心 广东 广州 510450;

2.广州大学网络中心 项目部 广东 广州 510006)

摘要: 为进一步提高重复数据删除系统的性能,提出基于数据分块的后缀数组 SA 和最长公共前缀 LCP 进行数据块优化的重复数据删除系统。系统首先将输入的数据流进行第一次分块,识别出相同的分块并给分块编号,创建分块编号序列的 SA 和 LCP 表,识别出最大重复队列和非重复数据块,进一步得出优化的超级块大小,然后以超级块为单元进行第二次数据分块并保存数据压缩结果。实验表明,相比于固定分块,该系统能实现给定输入流较好的压缩性和数据重构性。

关键词: 重复数据删除; 后缀数组; 最长公共前缀; 块优化

De-Duplication System Based on Suffix Structure for the Block Size Optimization

HUANG Chen-Hui¹, LIN Yong-Qin²

(1. Department of Network Center of Guangdong Women's Polytechnic College, Guangzhou 510450, China;

2. Department of Network Center of Guangzhou University, Guangzhou 510006, China)

Abstract: To further improve the performance of data de-duplication system, the paper brings out a de-duplication system based on data blocks' suffix array and longest common prefix for the block size optimization. The method first chunks input data into segments with a first size, then identifies the segments and create the indexes queue for the identifiers, and then create the suffix array and the longest common prefix structure from the indexes, next determines a second optimize size based on repeated indexes sequences and non-repeated indexes, finally chunks the input data into segments based on super chunks and saves the compression result. The result shows that it achieves a better compression ratio and object reconstruction for the given input data comparing the original fixed chunk segment size.

Keywords: de-duplication; suffix array(SA); longest common prefix(LCP); block size optimization

1 引言

面对日益爆炸的数据增长难题及用于备份和容灾的主存储成本压力,重复数据删除技术逐步被业界公认为数据中心中的必备技术^[1]。重复数据删除技术主要目的是实现在已有的磁盘设备上存储更多的备份数据,目前其实现理念是:在存储数据时检查和比较已存在的数据,如果它们是相同的,则过滤掉这部分数据的备份,然后通过指针引用已存在的数据,整个过

程可概括为五个阶段(1)数据收集(2)数据识别和比较,以字节或字节的倍数为单位对数据进行比较,识别出重复数据(3)数据重组,重复数据用已存在的数据指针所替代(4)可选的完整性检查(5)空间回收,释放之前重复数据占用的空间。

后缀树是一种数据结构,它能提供字符串的快速匹配和查找^[2],基于磁盘存储的后缀结构研究如文献^[3-5]取得了很大的进展,当前主要应用在人类基

^① 收稿时间:2010-03-03;收到修改稿时间:2010-04-06

可以转化为求一些后缀的最长公共前缀的最大值。表 1 lcp_{tab} 列给出了串 S=1211121314 的最长公共前缀数组。

S 的最大重复对(Maximal Repeated Pair): S 的两个子串 a 和 b, 在 S 中的开始位置分别为 p₁、p₂, 后缀串 a 和 b 匹配部分长度为 n', 也就是说, S[p₁..p₁+n] = S[p₂..p₂+n], 但 S[p₁+n+1] ≠ S[p₂+n+1]。最大对通过三元组(p₁,p₂,n')来表示, 其中 p₁,p₂ 为 S 中两个子串的开始位置, n' 为两个子串匹配部分的长度。并用 R(S)来表示 S 的所有描述最大对的三元组的集合。例如: S=xabcyiiiizabcqabcyrxar, abc 在 S 中出现了三次, 第一次和第二次出现的 abc 就组成最大对(2,10,3), 第二次和第三次同样也组成最大对(10,14,3), 但第一次和第三次出现的 abc 并不是最大对, 它们最大匹配长度为 4, 所以它们组成的最大对应该是(2, 14, 4)。

最大重复(Maximal Repeat): 是 S 的一个子串并出现在 S 中的一个最大重复对中的公共串称之为最大重复。若 x 是最大重复, 则存在(p₁,p₂,|x|)∈R(S), 且 x 在 S 中出现的位置是 p₁ 和 p₂。假定 R'(S)为 S 所有最大重复组成的集合。例如: 对于 S=xabcyiiiizabcqabcyrxar, abc 和 abcy 都是最大重复, 无论一个子串出现在 S 中的最大重复对多少次, 这个串在 R'(S)中且仅出现一次。也就是说, |R'(S)| ≤ |R(S)|, 通常 R(S)可能会比较大, 而 R'(S)远远小于 R(S), 并能很好地反映最大重复对。例如: 序列 S=1211121314 显示的 R'(S)={11,121,1,21}, R(S)={(2,3,2),(0,4,3),(6,8,1),(1,5,2)}, 在重复出现次数增加时, R'(S)将远远小于 R(S)。

超级最大重复(Supermaximal Repeat): 那些不作为任何其他最大重复的子串出现的最大重复称为超级最大重复。序列 S=1211121314 的超级最大重复为 11 和 121, 因为 1,21 是它们的子串, 因而不是超级最大重复。

最大重复和超级最大重复可以反映在前缀数组 SA 跟 LCP 值的对应曲线图, 如图 2 所示, 图中的最高顶点 A, B 是超级最大重复; 顶点 D, C 是最大重复, 这为通过 LCP 的值来判断最大重复、超级最大重复以及不重复数据提供了依据。在本系统中, 判定规则如下(设 x 为当前项的 LCP 值, 它的前一项的 LCP 值假设为 p, 后一项的 LCP 值假设为 f):

(1) 若 x 满足 p>x>f, 或者 p<x<f, 则当 x 非零时当前项为最大重复。

(2) 若 p<x 且 x>f, 通过比较当前项 x 的值与其所有左边出现的项的值, 当其所有左边出现的项的值都小于当前项 x 的值时, x 是最大重复。

(3) 若是不重复结点 NR, 则有 LCP[NR]=0 且 LCP[NR+1]=0, 如图 1 的结点 7 和结点 9。

3 数据分块及其后缀结构在系统中的应用

3.1 重复数据删除系统结构图

系统实现对给定输入流, 对第一次分块后得到的后缀结构, 通过后缀数组 SA 和最大公共前缀 LCP 表决定出超块的分块大小, 然后对输入流以超块为单元进行第二次分块, 并保存重复数据删除后的最终压缩结果。系统结构示意图如图 3 所示。

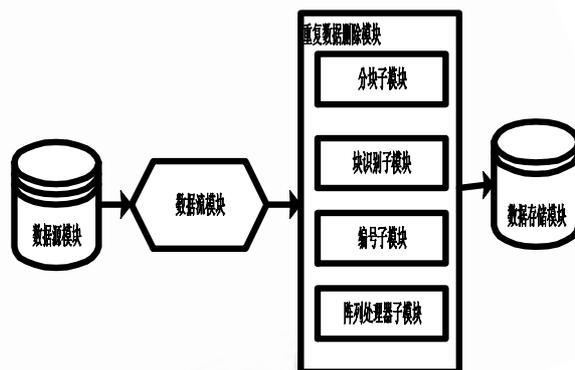


图 3 重复数据删除系统结构示意图

其中数据源模块负责文件(数据文件、音频视频文件、流媒体文件等)的上传和下载; 数据流模块实现数据的路由或转换功能, 将从数据源模块中获取的数据以数据流的形式交由重复数据删除模块进行处理。重复数据删除模块由分块子模块、相同块识别子模块、编号子模块和阵列处理器子模块组成, 进行块优化后的压缩数据最终由数据存储模块保存到永久性设备或磁盘等, 以备数据恢复用。

3.2 数据块优化流程

在本系统中数据分块的优化流程可概括为图 4 所示。

首先, 数据分块子模块对输入流的数据进行第一次分块后, 通过基于块内容的 hash 值识别出相同的块, 并对数据块进行编码, 即可得到数据块对应

参考文献

- 1 周永群.重复数据删除——数据中心“必备”技术. 华南金融电脑, 2009(9):8—8.
- 2 Gusfield D. Algorithms on Strings, Trees, and Sequences. Cambridge university press, 2005.
- 3 Cheung C, Yu JX, Lu H. Constructing suffix tree for gigabyte sequences with megabyte memory. IEEE Transactions on Knowledge and Data Engineering, 2005,17(1):90—105.
- 4 Phoophakdee B, Zaki MJ. Genome-scale disk-based suffix tree indexing. Proc. ACM SIGMOD International Conference on Management of Data. ACM Press, 2007:833—844.
- 5 Tian Y, Tata S, Hankins R A, Patel M. Practical methods for constructing suffix trees. The VLDB Jour., 2005,14(3):281—299.
- 6 Sinha R, Puglisi S J, Moffat A, Turpin A. Improving Suffix Array Locality for Fast Pattern Matching on Disk. Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data,2008: 661—672.
- 7 袁哲,赵永哲,张文睿,朱祥彬,赵东伟.利用水平分割法计算给定串中的所有 Maximal(NE/SNE) Repeats. 吉林大学学报(理学版), 2008,46(5):915—924.