Block Size Optimization in Deduplication Systems

Cornel Constantinescu, Jan Pieper and Tiancheng Li
{cornel, jhpieper}@almaden.ibm.com, li83@cs.purdue.edu
IBM Almaden Research Center
San Jose, California, USA

Data deduplication is a popular dictionary based compression method in storage archival and backup. The deduplication efficiency improves for smaller chunk sizes, however the files become highly *fragmented* requiring many disk accesses during reconstruction or chattiness in a client-server architecture. Within the sequence of chunks that an object (file) is decomposed into, sub-sequences of adjacent chunks tend to repeat. We exploit this insight to optimize the chunk sizes by joining repeated sub-sequences of small chunks into new *super chunks* with the constraint to achieve practically the same matching performance. We employ suffix arrays to find these repeating sub-sequences and to determine a new encoding that covers the original sequence. With super chunks we significantly reduce fragmentation, improving reconstruction time and the overall deduplication ratio by lowering the amount of metadata. As a result, fewer chunks are used to represent a file, reducing the number of disk accesses needed to reconstruct the file and requiring fewer entries in the chunk dictionary and fewer hashes to encode a file. To encode a sequence of chunks we proved two facts: (1) *any subsequence that repeats is part of some super chunk* (supermaximal in Figure 1) - therefore the dictionary contains just supermaximals and non-repeats, and (2) *maximals are the only repeats not covered (overlapped) by the containing supermaximals* - so once we discover the maximals with the suffix array, and encode them, there is no need for maintaining auxiliary data structures like bit masks, to guarantee the coverage (encoding) of the entire object. Our experimental evaluation (Figure 2) yields a reduction in fragmentation between 89%-97% and a reduction of dictionary metadata (number of entries) between 80%-97%, without increasing the total amount of storage required for unique chunks.

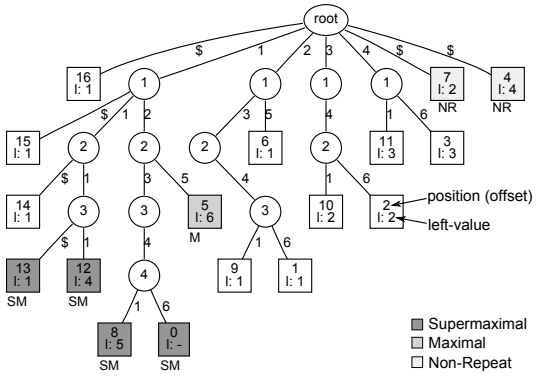

Figure 1: Suffix Tree Example

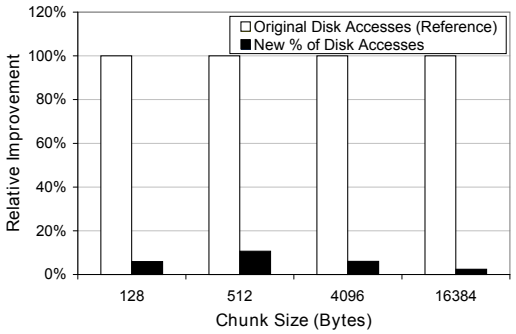

Figure 2: Disk Fragmentation Reduction.

1068-0314/09 $25.00 © 2009 IEEE
DOI 10.1109/DCC.2009.51