;login:

THE MAGAZINE OF USENIX & SAGE

April 2001 • volume 26 • number 2

inside:

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

mobile hoarding and dynamic grouping

by Ahmed Amer

Ahmed Amer is a PhD candidate at UC Santa Cruz. As a member of the Computer Systems Laboratory he is involved in systems research with a particular focus on data management for mobile systems and alternative storage architectures.

<amer4@cse.ucsc.edu>

The USENIX Scholars Program *<http://www.usenix.org/students/scholar.html>* provides support for student stipends, tuition, and other expenses for students with exceptional research ability and promise. This article is an example of the kind of work resulting from this program.

Motivation

Mobile computing is growing more practical with ever-decreasing hardware sizes and power requirements. We also see a growing dependence on larger and larger repositories of data, and with limited bandwidth and the possibility of frequent disconnection, it is still not a simple process for a user to "take their computer and go."

With improving wireless communication technology this problem remains far from being solved. Wireless bandwidth does not approach the bandwidth available with conventional wired networks and is more susceptible to unforeseen service disruption (e.g., driving through a tunnel, signal interference).

One promising approach to allow for more effective mobile computing is file hoarding, the caching of a user's data to their local system's hard drive, with the goal of allowing disconnected operation or simply better performance over slow and unreliable network links.

Despite significant developments in file hoarding algorithms, none are available in a form that allows widespread general use. Whether the actual hoarding process requires extensive user input, or whether parameter selection requires extensive experimental tuning, we feel that a major obstacle to general availability of file hoarding as a useful tool is an insufficient level of automation. Our work proposes a model for file clustering and grouping based on statistical predictors (relationship estimators) that is amenable to automatic parameter tuning.

Description

Our system is based on a two-phase process: interfile relationship estimation, and the grouping of files to produce minimal intergroup relationships (see Figures 1a and 1b).

Relationship estimation uses previous file access events to provide a statistical model of file access relationships. This is similar to the task performed by predictive file caching. This includes graph-based systems that use access windows, or more recent work utilizing context modeling techniques, which can also be seen as providing probability estimates for subsequent file access events.

In Figure 1 (see next page), a relationship estimator in our system is required to accept a stream of file access events and produce a weighted graph G(V, E, W) of file relationship estimates. Here V is the set of files, E is the set of significant edges, and W is the set of associated weights (relationship estimates). Relationship estimates are weightings that are proportional to the probability of a particular file being accessed after another file. It should be noted that the resultant graph will be of a fixed maximum degree, since it is infeasible to track relationship estimates for "all other files," which would require quadratic space in the number of files in the file system. Fixing the degree of the graph is equivalent to placing a limit on the length of the adjacency list for a particular file. This restriction reduces any algorithm linear in the number of relationships to being linear within the number of files. This allows us to produce a feasible solution for the grouping

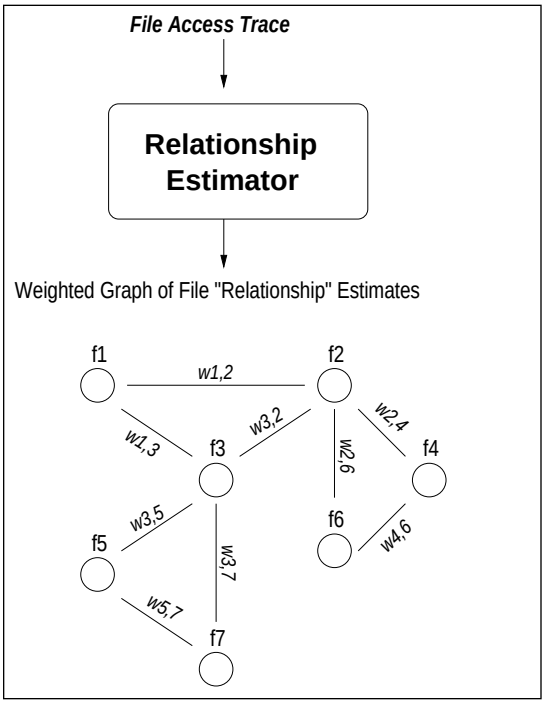

Figure 1a – Relationship Estimation

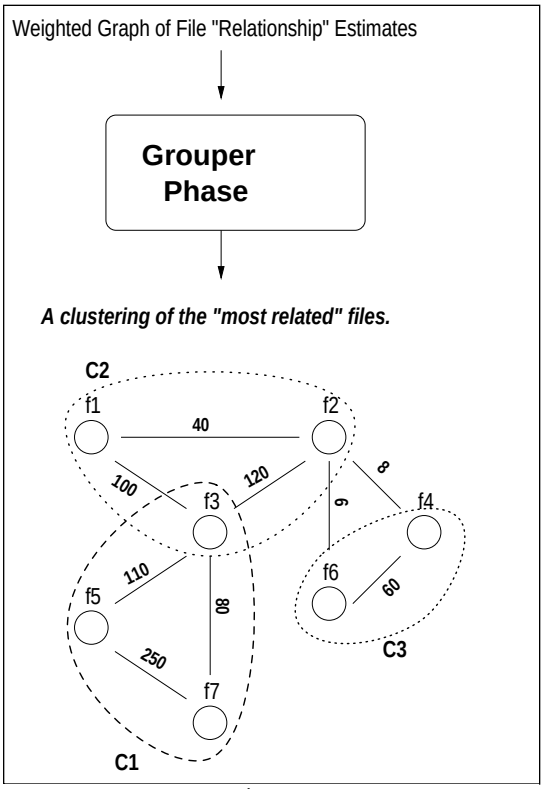

Figure 1b – Grouping

Given a stream of file access events, relationship estimation builds a graph (of strictly limited degree) estimating the level of interfile relationships, which is used by the grouping phase to divide the graph into minimally related groups

problem and also limits state-space requirements to a constant factor of existing file system metadata.

It is important to note that there are multiple mechanisms for measuring this "likelihood." We have considered several classes of such predictors and have produced a novel predictor, Noah, that identifies strong pairings among files with minimal state space and computational requirements.

In the second phase, grouping, we are given the graph of interfile relationship estimates, and we attempt to divide the graph into minimally related groups/clusters. Intuitively, we wish to group together the files most likely to be accessed within a short period of each other.

After dividing the weighted relationship graph into clusters of related files, the final step in the process of selecting files for hoarding is ranking the clusters. This is done by assigning a score to each cluster and is necessary when a single highly related cluster cannot fill the mobile store. In summary, we employ a ranking mechanism for the generated clusters, and use this ranking to select the clusters to be imported to the local system for hoarding purposes. There are several ranking metrics that can be used, varying in complexity from simple LRU up to more elaborate algorithms like file aging.

A final note on our approach involves the automatic tuning of operating parameters. We continuously attempt to avoid any manual adjustment of operating parameters for our algorithms. This is greatly facilitated by the strong machine-learning community within the University of California, Santa Cruz's School of Engineering.

Further Applications

We evaluate groupings generated by our approach against file access traces to verify their usefulness for mobile file hoarding. And yet, if we relax the requirement for cluster ranking, we can directly use the clusters we have generated in the grouping phase for purposes of data placement on storage media elements. In this scenario, the limit on cluster size would be directly proportional to the capacity of a single media "element." Media elements can be any units of a storage medium that invoke a high-latency operation for a switch between them. The limit on the total size of all combined clusters is analogous to the limit on total available storage space. (A subtle difference between placement and hoarding restrictions is that in hoarding, each file size is counted only once, whereas for placement, duplicate copies will use double the physical storage space, as opposed to simply state space for hoarding.)

Figures 2a and 2b. show the effect of increasing the latency/bandwidth gap on system throughput. For the tertiary store we assume typical access characteristics of a tape or M.O. disk media changer, while for the secondary store, throughput is increased to model a scenario where raw bandwidth can be increased to almost 1GBps – which is feasible given enough parallelism.

Examples include existing automated tertiary libraries that incur a heavy performance penalty for media replacement. A more general example is the trend in storage technology toward higher and higher bandwidth, with limited improvement in access speed. This is true of magnetic disks, optical disks, and even communication networks.

Placing data into clusters that have a minimal likelihood of intercluster transitions can ameliorate a wide range of growing systems problems. Constructing such clus-

ters is exactly the purpose of our two phases of relationship estimation and grouping. We focus on data hoarding applications which, thanks to the highly automated nature of our statistical predictors, promises to be performed with minimal user intervention.

Status

Work on relationship estimation is largely complete, and our most recent development, the Noah algorithm for low-cost online evaluation of file pairings, has generated two papers currently under review for publication. Noah maintains accurate pairings of files while tracking only two candidate successors for each file, and adapts to variations in access patterns on a per-file basis. Our most recent results for grouping, and optimality measures for online relationship detection, are currently being prepared for possible submission at the next USENIX annual technical conference in Boston.

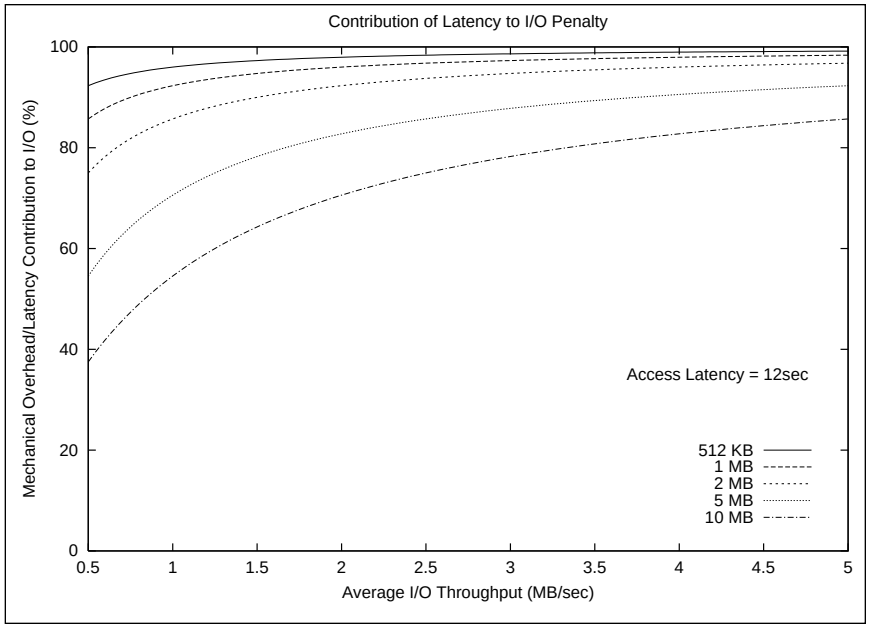

Figure 2a – Tertiary Store

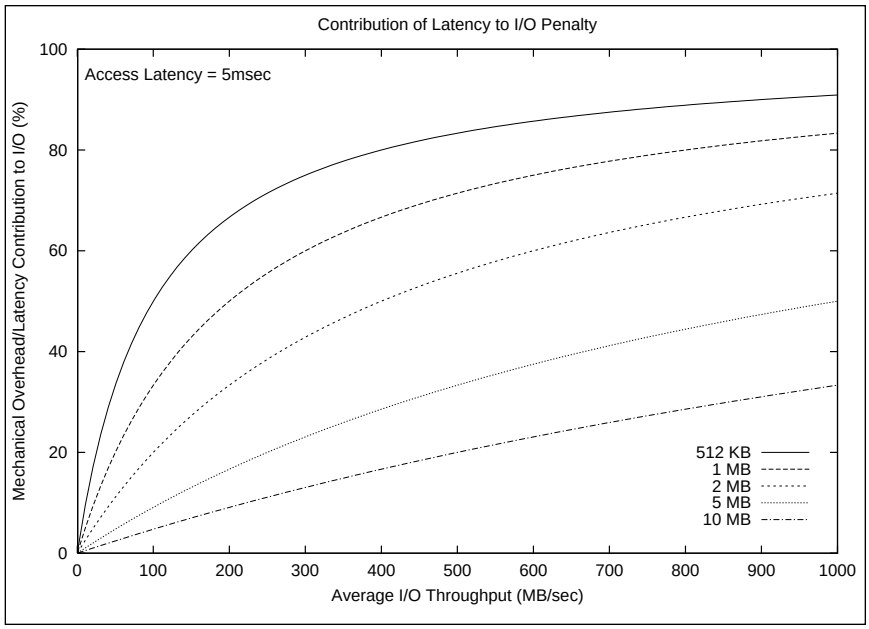

Figure 2b – Secondary Store

Acknowledgments

We are grateful to all the members of the Computer Systems Laboratory for their continuous feedback, support, and valuable discussions. We are also grateful to all those who provided us with detailed storage system traces, including G. Kuenning, D. Roselli, and especially J. Wilkes and the Hewlett-Packard Company, and R. Golding of Panasas. Our most extensive multi-year traces were made available by M. Satyanaryanan of Carnegie Mellon University, through the greatly appreciated efforts of T. Kroeger in processing and conversion.

This project is being conducted under the supervision and guidance of Professor Darrell D.E. Long, in the Computer Science Department of the University of California, Santa Cruz. This is one of several systems projects under investigation in the Computer Systems Laboratory, which has recently undergone significant expansion.

Further Information

Links to Web pages relating to this research can be found at
<http://www.cse.ucsc.edu/~amer4/research/>.

Further links to the Computer Systems Laboratory at the University of California, Santa Cruz, including summaries for research projects, members, and affiliates can be found at: *<http://csl.cse.ucsc.edu/>*.