Adaptive File Cache Management for Mobile Computing

Jiangmei Mei Rick Bunt

Department of Computer Science
University of Saskatchewan
Saskatoon, SK, Canada

Abstract. By performing file system operations on cached copies of files mobile users reduce their need to access files across the network. Effective cache management strategies will limit use of the network to a small number of read misses and periodic reintegration of changes (updates), and thus can shield the mobile user from changes in bandwidth that will have a significant impact on performance. This paper addresses adaptive approaches to file cache management in mobile environments. The results of a trace-driven simulation study demonstrate that adaptive adjustment of the cache block size can improve performance for mobile users at acceptable cost.

1 Introduction

File systems that support mobile users, such as Coda [3][5] and AFS [2], rely on optimistic file caching to alleviate the impact of mobility. Cache management in such a file system addresses issues such as hoarding files to prepare for mobility, servicing file system requests in cache where possible when mobile, retrieving requested files from the home file server where necessary, and propagating updates to the home file server for timely reintegration. In a mobile environment the bandwidth of the connection to the home file server can change suddenly and dramatically. Since bandwidth has a large impact on the time to complete these file operations, this can have a significant impact on performance. A cache management strategy that assumes that the same bandwidth is available throughout the connection may not perform well in a dynamically changing mobile environment. Cache management in a mobile environment must react dynamically to changes in resource availability – to make effective use of increased resource, and to keep the impact of decreases to a minimum.

The long-term goal of our research is to examine the extent to which cache management in a mobile environment can be designed to adapt to changes in resource availability. This paper extends some earlier work on file cache management for mobile computing [1]. We used the same system model, performance metrics, file system traces and simulator to study the performance benefits of adaptive file cache management in a dynamically changing mobile environment.

2 File Caching for Mobile Clients

Optimistic caching of file replicas at clients is a technique commonly used to improve performance in distributed file systems. This not only reduces latency in servicing file requests, but also avoids traffic on the network.

In a distributed file system that contains mobile computers, caching plays an even more important role in providing mobile users with data availability and performance. When mobile, the user may connect to the home file server through different communication methods at different places (at different times), and the connections may have quite different characteristics. Much of the time, there may be only a "weak" connection to the home file server over a slow and perhaps error-prone link, and sometimes the client may be totally disconnected. Maintaining copies of actively used files in local cache allows the mobile user to continue to work even while the client is disconnected from the file server, and improves the performance when weakly connected since retrievals over the network are less frequent.

3 The Need for Adaptation

Mobility and portability pose significant restrictions on resources such as memory and disk space and it is necessary to cope with these limitations in the design of support software [6]. Mobile clients may also experience considerable variations in connectivity, such as changes in bandwidth and error rate. The quality of service may vary dramatically and unpredictably because of communication noise, congestion, or environmental interference, and disconnections may be frequent. Mobile systems and applications should react to changes dynamically – to take advantage of increased resource availability when that happens, and to avoid the negative impact of decreases when they happen. Being able to adapt successfully to changes in resource availability is widely recognized as a central support issue for mobile computing [6].

Our focus in this short paper is on the amount of data brought into the cache in a block caching implementation – in other words, the cache block size. When bandwidth is limited, fetching smaller blocks conserves bandwidth and reduces access time. When bandwidth is plentiful, it is possible to fetch a large block of data so that the likelihood of satisfying future references in cache is increased.

4 The Experiments

4.1 The Simulated Environment

For this study the mobile environment is defined as follows. After a period of time for hoarding, a mobile user attempts to work at several locations (at different times). While mobile, he/she connects to the home file server over different connection bandwidths and performs work that involves file system activities. The mobile client contacts a single logical home file server for file system requests.

While the file server is always connected to the internet through a fast and reliable link, the mobile client is either weakly connected to the internet when mobile or disconnected.

In the mobile client, the file cache contains copies of file blocks that the user either hoarded prior to going mobile or referenced recently. When notified of any change in the effective bandwidth of the connection, the adaptive cache manager adapts to the changes by modifying its behaviour. For this study, the block size for different bandwidths was predetermined: 16 kB, 8 kB, 4 kB, 2 kB, and 1 kB for bandwidths of 2 Mbps, 512 kbps, 64 kbps, 9.6 kbps, 2 kbps, respectively.

Detailed traces of real user file system activity were collected in the Distributed Systems Performance Laboratory in the Department of Computer Science at the University of Saskatchewan. Each trace is a complete record of an individual user's real day-to-day file system activities in a typical academic research environment over a seven-day period.

The characteristics of the 3 traces we selected for this study are given in Table 1. The number of *Active Hours* in each trace is computed by subtracting the amount of "idle" time in the trace from the time between the first and last references in the trace. *Total Requests* is the number of file system events in the trace, *Unique Files* is the number of different files referenced, and *Unique Bytes* is the total size of those files. *Read Transfers* and *Write Transfers* are the sum of the sizes of all read and write requests in the trace, respectively, and *Write Percentage* is the percentage of all read and write requests that are writes. Finally, *Intervals* is the number of inter-event times of the length of time indicated. Intervals affect assumptions relating to disconnections. If an interval is longer than 15 minutes and shorter than 1 hour, it is assumed that the client is experiencing an unexpected disconnection. If the interval is more than one hour, it is considered to be an expected disconnection. More detail on the trace collection, the post-processing that was done, and the meaning of individual fields is given in [1].

Table 1. Trace Characteristics

Trace		1	2	3
Active Hours		12.9	7.0	8.5
Total Requests		21013	53860	17690
Unique Files		383	1000	443
Unique Bytes (MB)		17	18	22
Read Transfers (MB)		4596	4718	3723
Write Transfers (MB)		4	35	2
Write Requests (%)		38.7	67.9	25.2
Intervals	15-60 (min)	10	6	9
	> 60 (min)	11	5	16

4.2 Results

For this brief presentation performance is assessed primarily through *time expansion* – the ratio of the total time required to process the entire trace in the simulated mobile environment to the time required when strongly connected. It reflects the additional cost to the user of mobile operation and is affected primarily by the additional time required to service read misses over the network and by any interference induced by updates.

The impact of adapting block size is shown in Figure 1. For each trace, adaptation results in reduced time expansion over non-adaptive block caching. In fact, the positive impact on the mobile user is actually even better than what Figure 1 suggests. When only the active hours are considered (i.e. idle time is removed), the time to service actual requests is reduced by 12.5%, 4%, and 5.5% for Trace 1, Trace 2, Trace 3, respectively, at a cache size of 1 MB. With less cache at the client, the reduction is even larger. Full results are available in [4].

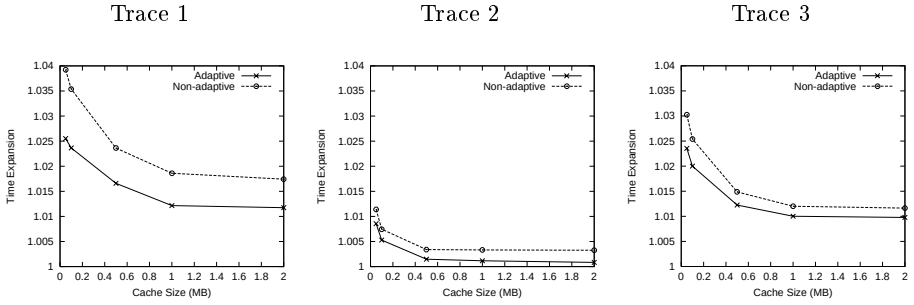

Fig. 1. Time Expansion

Figure 2 shows that adaptive block caching adds only a small amount of extra traffic to the network. This suggests that the benefits of adaptive block caching can be provided at little or no extra cost.

5 Conclusions

The ability to adapt dynamically to environmental changes has been identified as an important success factor for mobile computing. We are experimenting with adaptive approaches to file cache management, seeking to understand the extent to which adaptation can provide improved performance for mobile users.

Adaptation can be beneficial when appropriate adaptation strategies are applied in the constantly changing mobile environment. Adaptive block caching can improve performance over the non-adaptive approaches by dynamically adjusting the amount of data to be fetched as available bandwidth changes. The reduction in time expansion, particularly at small cache sizes, is important when

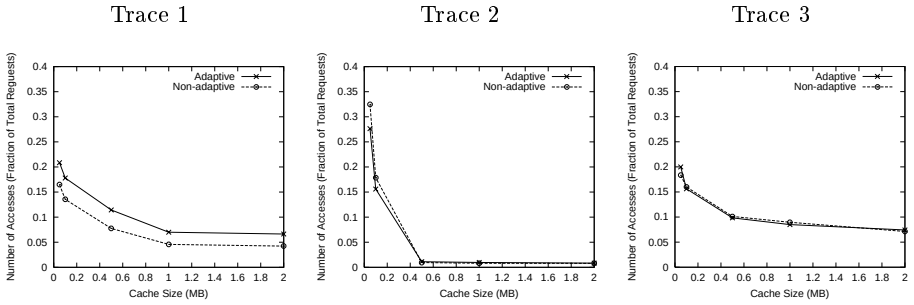

Fig. 2. Number of Network Accesses

the resources at the mobile client are very scarce – as they are for palmtops or PDAs. There may, of course, be negative aspects to adaptation, but our results suggest that these negative impacts remain at acceptable levels.

On balance our adaptive approaches offer performance benefits with very little extra cost. Although this particular study is limited in its scope it adds to the growing list of successful adaptive approaches and provides support for further work in this important area.

6 Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by TRLabs in Saskatoon.

References

1. K. Froese and R. Bunt. Cache management for mobile file service. *The Computer Journal*, 42(6):442-454, 1999.
2. L. Huston and P. Honeyman. Partially connected operation. In *Proceedings of the Second USENIX Symposium on Mobile and Location-Independent*, pages 91-97, Ann Arbor, MI, April 1995.
3. J. Kistler and M. Satyanarayanan. Disconnected operation in the Coda file system. *ACM Transactions on Computer Systems*, 10(1):3-23, January 1992.
4. J. Mei. Adaptive file cache management for mobile computing. M.Sc. thesis, Department of Computer Science, University of Saskatchewan, Saskatoon, SK, 2002.
5. L. Mummert, M. Ebling, and M. Satyanarayanan. Exploiting weak connectivity for mobile file access. In *Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles*, pages 143-155, Copper Mountain Resort, CO, December 1995.
6. M. Satyanarayanan. Fundamental challenges in mobile computing. In *Proceedings of the Fifteenth ACM Symposium on Principles of Distributed Computing*, pages 1-7, Philadelphia, PA, May 1996.