A Distributed, Searchable Web-Cache for Tighly Coupled Networks

Omar Bakr Matulya Bansal Fred Cheng R.J. Honicky

October 9, 2003

1 Introduction

In this project, we will aim to improve the usability and reduce the cost of Internet connectivity in locations for which connectivity is slow (in terms of latency and/or bandwidth), expensive and intermitent, most notably in developing countries. A primary objective of the project will be to design and at least partially implement a system which has the following properties:

1. The system will allow users to query a local database of documents found on the Internet, and retrieve many results very quickly.

2. The owner of the system which uses our software will not need to make any capital investment. This means that it will not require any additional hardware purchase, and also that the software will be freely available.

3. The software can function even if the system is entirely disconnected from the Internet. This may happen because of a network outage, or to save on phone and ISP charges.

We envision two primary customers of such as system. The first customer is an Internet cafe, the second is a library. Because we suspect that users of a library system will, in general, search and use documents of a much more persitent nature, including journal and news articles, online books, technical reference materials and online source code, we will focus on library systems.

2 Design and Architecture

We will design and at least partly implement a system which can be distributed over several different machines. Each machine will have two components. The first component is a search interface. This will be a standard search interface, like Google. When a query is made, the interface will retrieve results which are available on the local disks. On the results page(s), there will be a link which will allow the user to submit the search over the Internet to Google, using Google's web-services API. Any results retreived from the Internet will be cached locally.

The second component will be the searcable cache. Each machine in the system will store a subset of the results which have been previously returned (or pre-primed). Each query will be broadcast to the database on each machine, and the results will be merged by the search interface.

3 Questions We Would Like To Answer

1. Feasibility:
   (a) What is a typical library workload like in the US? In a developing country?
   (b) What sort of hardware is typical in a library?
   (c) Are machines in libraries in developing countries left on?
2. Architectural:
   (a) What algorithm will we use to distribute and look up data to each database?
   (b) What sort of data structures will the database use?
   (c) Should data be replicated, in case machines are not available?
   (d) What cache replacement policy will we use?
3. Statistical
   (a) What is the composition of a corpus of "library-type" documents?
   (b) What is the size of a corpus of "library-type" documents?
   (c) What is the distribution of access frequencies to documents in that corpus? How does that effect the cachability of the corpus.
   (d) What is a reasonable cache size, based on the corpus?

4 Some references...

We have begun to compile some references to related works.

1. **Peer to peer:** PAST[10], FARSITE[1], OceanStore[7],CFS[5]
2. **Distributed Hashing:** LH* [8], RUSH [6], SHARE [3]
3. **Adaptive Caching:** ACME [2], ARC [9]
4. **Search Engines:** Mining the Web [4]
5. **Similar Projects:** TEX [11]

References

[1] Atul Adya, William J. Bolosky, Miguel Castro, Ronnie Chaiken, Gerald Cermak, John R. Douceur, John Howell, Jacob R. Lorch, Marvin Theimer, and Roger Wattenhofer. FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI)*, Boston, MA, December 2002. USENIX.

[2] Ismail Ari, Ahmed Amer, Robert Gramacy, Ethan L. Miller, Scott A. Brandt, and Darrell D. E. Long. ACME: adaptive caching using multiple experts. In *Proceedings in Informatics*, volume 14, pages 143–158. Carleton Scientific, 2002.

[3] André Brinkmann, Kay Salzwedel, and Christian Scheideler. Compact, adaptive placement schemes for non-uniform capacities. In *Proceedings of the 14th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 53–62, Winnipeg, Manitoba, Canada, August 2002.

[4] Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers, San Francisco, CA, 2003.

[5] Frank Dabek, M. Frans Kaashoek, David Karger, Robert Morris, and Ion Stoica. Wide-area cooperative storage with CFS. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP '01)*, pages 202–215, Banff, Canada, October 2001. ACM.

[6] R. J. Honicky and Ethan L. Miller. A fast algorithm for online placement and reorganization of replicated data. In *Proceedings of the 17th International Parallel & Distributed Processing Symposium*, Nice, France, April 2003.

[7] John Kubiatowicz, David Bindel, Yan Chen, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon, Westly Weimer, Christopher Wells, and Ben Zhao. OceanStore: An architecture for global-scale persistent storage. In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Cambridge, MA, November 2000. ACM.

[8] Witold Litwin, Marie-Anna Neimat, and Donovan A. Schneider. LH*—a scalable, distributed data structure. *ACM Transactions on Database Systems*, 21(4):480–525, 1996.

[9] Nimrod Megiddo and Dharmendra S. Modha. ARC: A Self-Tuning, Low Overhead Replacement Cache. In *Proceedings of the 2003 Conference on File and Storage Technologies (FAST)*, pages 115–130, San Francisco, CA, March 2003.

[10] Antony Rowstron and Peter Druschel. Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP '01)*, pages 188–201, Banff, Canada, October 2001. ACM.

[11] W. Thies, J. Prevost, T. Mahtab, G. Cuevas, S. Shakhshir, A. Artola, B. Vo, Y. Litvak, S. Chan, S. Henderson, M. Halsey, L. Levison, and S. Amarasinghe. Searching the world wide web in low-connectivity communities. In *Proceedings of The 11th International World Wide Web Conference*, May 2002.