

Optimizing Search in Distributed Network Storage Using Compact Set Representations

Masters thesis for Andreas Diener

1 Introduction

Data is increasingly becoming one of the dominant resources in private and corporate life. More and more data needs to be available immediately to avoid disruption of many critical processes. This flood of data needs to be stored reliably while allowing for quick access. Peer-to-peer (P2P) systems with their deliberate avoidance of single-points-of-failure are gaining a reputation to provide resilient storage even under partial outages.

Many P2P systems support efficient retrieval of the document when its unique ID is known. Often this ID is not known, but a set of query items describing a set of documents. An open problem for widely distributed, heterogeneous P2P systems is how to search for stored documents efficiently. The goal of this thesis is to improve the knowledge in this field, resulting in more efficient search.

2 Current Search Strategies

A variety of search strategies have been tried since the term “peer-to-peer” was born only a few years ago. One path has been the evolution from brute-force flooding of the network [1], scouting associative graphs [2] to limited flooding in semantic structures [3]. Another avenue has been the use of a structured overlay network providing the distributed hash table (DHT [4, 5, 6, 7, 8]) abstraction and implementing indices on top of it, as exemplified by using the DHT to store tables for [9] or querying distributed inverted indices [10, 11]. The latter use Bloom filters [12, 13] to limit the amount of information transmitted in the query phase.

3 Task Description

To evaluate a query which refers to multiple Inverted indices, both sets traditionally must be present on the same machine when performing boolean operations on them. While this can be done in streaming mode [14] to reduce temporary storage requirements, the network bandwidth to transmit the data is still necessary and is expected to be the bottleneck in the typical case, i.e. the multi-site scenario. Despite the gains that have been achieved by using Bloom filters [12, 13], the amount of information to be transferred can still be many orders of magnitude larger than the size of the result set.

The goal of this thesis is to improve this query evaluation process in the following three steps.

3.1 Hierarchical Coding

To improve the worst case, hierarchical Bloom filters can be used. A naive implementation of the resulting multi-stage process will require many round-trip times and thus increase the latency until a result can be returned. The goal of this task is to find and evaluate methods which can be used to

- create efficient hierarchical codes which minimize the overhead introduced,
- define boolean set operations on top of this hierarchical code, and
- automatically determine an optimal strategy when and how this improvement should be used.

3.2 Improved Coding

Bloom filters provide a simple and easily understandable mechanism for lossy data compression with a defined error behavior (i.e., errors are limited to false positives). Investigate alternative, more efficient lossy compression schemes which maintain a reasonable subset of the set operations [11].

Are there alternative error behaviors that might lead to better compression while still resulting in useful set operation approximations?

3.3 Exchange of Relevance Information

Can compressed set representations be used to efficiently convey relevance information of members in the result set? Are they appropriate for set compression? Can the error of the final relevance score be bounded?

Consider the following sources of relevance information and evaluate which ones are feasible metrics in this scenario at all and which one would work best.

Static. Each document is assigned a fixed priority.

Google. The priority of a document is dependent of the priority of documents referring to it.

Teoma. The priority of a document is dependent of the priority of other documents referring to it and matching the same query.

4 General Comments

- Present a project time line after two weeks.
- There will be weekly meetings with the supervisor.
- Prepare a short intermediary report of 1–2 pages by mid-thesis.
- At the end of your thesis, the following has to be handed in: Source codes and a report (in English) including a one-page summary in both German and English.
- Source code is to be presented in machine-readable form (ASCII or UTF-8).
- The report should follow the rules of a scientific publication and include performance measures of the prototype. Two paper copies and a PDF document of the report have to be submitted. The summary is to be submitted in ASCII, UTF-8, or HTML as well.
- The final presentation of the project at both IBM and ETH is an integral part of the thesis.
- Development environment: Well-known system or language of your choice.
- Intellectual property and confidentiality rights and duties are handled in a separate contract.

5 Administrativa

Responsible Professor: Prof. Dr. Roger Wattenhofer, ETH Zürich

Supervisor: Dr. Marcel Waldvogel and Dr. Paul Hurley, IBM Research GmbH

Start: 7 September 2004

End: 6 February 2005

References

- [1] Jean Vaucher, Peter Kropf, Gilbert Babin, and Thierry Jouve. Experimenting with gnutella communities. In *Distributed Communities on the Web (DCW 2002)*, Lecture Notes in Computer Science, April 2002.
- [2] Edith Cohen, Amos Fiat, and Haim Kaplan. A case for associative peer to peer overlays. *SIGCOMM Computer Communications Review*, 33(1):95–100, 2003.
- [3] Chunqiang Tang, Zhichen Xu, and Mallik Mahalingam. psearch: information retrieval in structured overlays. *SIGCOMM Computer Communications Review*, 33(1):89–94, 2003.
- [4] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content-addressable network. In *Proceedings of ACM SIGCOMM*, September 2001.
- [5] Karl Aberer. P-Grid: A self-organizing access structure for P2P information systems. In *Sixth International Conference on Cooperative Information Systems (CoopIS)*, September 2002.
- [6] Anthony Rowstron and Peter Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, Heidelberg, Germany, November 2001.
- [7] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of ACM SIGCOMM 2001*, pages 149–160, San Diego, CA, USA, August 2001.
- [8] Marcel Waldvogel and Roberto Rinaldi. Efficient topology-aware overlay network. *ACM Computer Communications Review*, 33(1):101–106, January 2003. Proceedings of ACM HotNets-I (October 2002).
- [9] Ryan Huebsch, Joseph M. Hellerstein, Nick Lanham Boon, Thau Loo, Scott Shenker, and Ion Stoica. Querying the internet with PIER. In *Proceedings of 19th International Conference on Very Large Databases (VLDB)*, Berlin, Germany, September 2003.
- [10] Efficient peer-to-peer keyword searching. In *Proceedings of the ACM/IFIP/USENIX Middleware conference*, June 2003.
- [11] Daniel Bauer, Paul Hurley, Roman Pletka, and Marcel Waldvogel. Bringing efficient advanced queries to distributed hash tables. In *Proceedings of IEEE LCN*, November 2004.
- [12] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, July 1970.
- [13] Andrei Broder and Michael Mitzenmacher. Network applications of Bloom filters: A survey. In *Proceedings of the 40th Annual Allerton Conference on Communications, Control, and Computing*, pages 636–646, 2002.
- [14] Russell F. Haddleton. *Parallel Set Operations in Complex Object-Oriented Queries*. PhD thesis, University of Virginia, January 1998.