Cerco: Supporting Range Queries with a Hierarchically Structured Peer-to-Peer System

Simon Rieche, Klaus Wehrle
Distributed Systems Group
RWTH Aachen University, Germany
Simon.Rieche@rwth-aachen.de
Klaus.Wehrle@rwth-aachen.de

Leo Petrak, Clemens Wrzodek
Wilhelm-Schickard-Institute for Computer Science
University of Tübingen, Germany
Leo.Petrak@uni-tuebingen.de
Clemens.Wrzodek@student.uni-tuebingen.de

Abstract

Structured Peer-to-Peer systems are designed for a highly scalable, self organizing, and efficient lookup for data. The key space of the so-called Distributed Hash Tables (DHTs) is partitioned and each partition with its keys and values is assigned to a node in the DHT. For data retrieval however, the very nature of hash tables allows only exact pattern matches.

We propose Cerco, a simple solution for the problem of range queries by employing a hierarchically structured P2P approach based on the principles of Distributed Hash Tables. We show that a dynamic hierarchy of DHTs with on-demand classification of items can positively influence the response time of queries while maintaining lookup correctness.

1. Introduction

Within the wide area of Peer-to-Peer-based (P2P) [9] systems, Distributed Hash Tables (DHTs), due to their features, are more and more frequently used for data lookup in distributed applications [1]. Their simple, efficient, scalable, and self-organizing algorithms for data management and retrieval offer crucial advantages compared e.g. to unstructured P2P approaches [5] and client-server solutions.

But a major weakness of DHTs is the search for stored content. Because the assignment of data to nodes is based on hash functions, only exact pattern matching can be used. For example, a file "linux-debian.iso" which is stored under its filename in the DHT can be found by a search with the exact name only. The search for "linux.iso" would not find the debian linux distribution.

We propose a simple solution for the problem of range queries using a hierarchically structured P2P approach based on the principles of Distributed Hash Tables.

2. Related Work

A tree-based method for range queries are the so-called **range search trees** (RSTs) [8, 7]. Every layer of a tree covers the whole search space which is distributed among all nodes in that layer, so every node covers a specific range. Since this leads to heavy load of data on the root node and lower data load on the leaves, separate load balancing matrices (LBMs) [7, 6] are necessary. **Prefix hash trees** (PHTs) [10, 4] use a binary representation of items. Items with the same prefix get stored in the same leaf node. But using this system for long strings or for a large number of elements is not recommended. The **LSH Forest** approach [2] is using tree structures for similarity searches, but searching for a specific prefix, suffix or range is not possible. **Chord#** [11] basically eliminated the hash function to reach a sorted distribution of elements over the key space. By using this method, one may get some problems with the average load per node, since no separated load balancing is implemented.

Probably the closest related solution is **Mercury** [3]. It takes key-value-pairs and builds a separate Chord-like ring for every key. Every ring gets a key-order preserving function to store its values. A node in this system can be responsible for multiple areas in multiple rings. However, Mercury is only recommended for use with multi key data.

None of the proposed solutions considers all of the most desired features. Some of them do not process exact range queries [2], some are slow in processing queries, some have problems with scalability, most show load balancing problems [11], etc. In our solution, we try to overcome these disadvantages as described below.

3. A Hierarchically Structured P2P System

We propose to use structures like rings of key-value-pairs for the organization of the content system. It is similar to a classical DHT technique, however without involvement of a regular hash function. Instead, a simple unicode representation of items, grouped and ordered, e.g. by name, will be

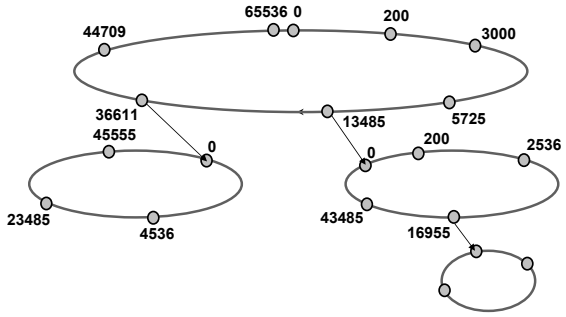

Figure 1. Architecture of the Hierarchically Structured P2P System.

applied. In high load regions, e.g. for some letter ranges, the system decides on demand to generate hierarchical subrings of a whole region with the same arrangement as in the above layer. If necessary, the system creates even subrings of subrings to reach an acceptable load balancing in the whole system. This leads to structures like multicast distribution trees and short response times for range queries because of a simple access mechanism to involved items. Figure 1 shows an example of such an hierarchically structured P2P system. Node 13,485 is overloaded and creates a subring for keys beginning with 13,485. Node 13,485 delegates queries to the nodes in the subring. For redundancy issues, the successor and predecessor of 13,485 at the top level ring know the successor and predecessor of the node in the subring and the other way around.

4. Supporting Range Queries

Nodes, that are responsible for the first letters of a query are located in the ring-based on the unicode number of the corresponding character (e.g. 1,114,112 regions for unicode [12] representation). Afterwards, each node decides, whether and how his character range should be split in more subrings, e.g. based on histograms of typical items. Please note, that the resulting structure is not obligatorily balanced, i.e. there are branches with different depth of hierarchy level.

The search process within our system is similar in principle to standard search mechanisms of DHTs, however with extended capabilities relating to range queries. The node, which is responsible for the first character of a query, coordinates the search process by forwarding the query to participating subrings and other first-letter-nodes of our range query on demand. The "handover" to subrings continues the same way for further first-letters of our range query. So the search is performed in $O(k \cdot \log N)$. Please note, that $\log N$ is the mean search time in the Chord system. However, in our approach the number of nodes N is considerably smaller as within usual Chord settings. The term k describes the mean depth of our hierarchy level.

Example: For range queries like "[AG-CY]" the node, which is responsible for the first letter (here node A) should be found first. This node coordinates the entire search process. It contacts all nodes responsible for the other starting letters of the parts of the query (here nodes B and C). These nodes now have to ask the nodes responsible for the next character within their ranges ($A - G$ for A subring, $A - Z$ for B subring, $A - Y$ for C subring) in the search string, and so on.

Wildcard searches, like "*abc", could be realized in a similar way: The first node aligns the entire process by initiation of a number of parallel search queries by all nodes responsible for the next (e.g. second sign within a query) character and assembling of parcial results by merging.

5. Conclusions

Range queries are important for structured P2P Systems in order to increase the possible application fields. Because the assignment of data to nodes is based on hash functions, only exact pattern matching can be used in a search for content. We propose Cerco, a solution for the problem of range queries using a hierarchically structured Peer-to-Peer approach based on the principles of Distributed Hash Tables. Our future activities include the implementation task and investigation by simulation.

References

[1] H. Balakrishnan et al. Looking Up Data in P2P Systems. *Communications of the ACM*, 46(2), 2003.

[2] M. Bawa et al. LSH Forest: Self-Tuning Indexes for Similarity Search. In *Proc. of WWW'05*, 2005.

[3] A. R. Bharambe et al. Mercury: Supporting Scalable Multi-Attribute Range Queries. In *Proc. of SIGCOMM*, 2004.

[4] Y. Chawathe et al. A Case Study in Building Layered DHT Applications. In *Proc. of SIGCOMM*, 2005.

[5] J. Eberspächer et al. *First and Second Generation of Peer-to-Peer-Systems*. In Peer-to-Peer Systems and Applications, LNCS 3485, Springer, 2005.

[6] J. Gao. *A Distributed and Scalable Peer-to-Peer Content Discovery System Supporting Complex Queries*. PhD thesis, Carnegie Mellon University, 2004.

[7] J. Gao et al. An Adaptive Protocol for Efficient Support of Range Queries in DHT-based Systems. In *Proc. of ICNP'04*.

[8] J. Gao et al. Rendezvous Points-Based Scalable Content Discovery with Load Balancing. In *Proc. of NGC'02*, 2002.

[9] D. S. Milojicic et al. Peer-to-Peer Computing. Technical Report HPL-2002-57, HP Laboratories, 2002.

[10] S. Ramabhadran et al. Prefix Hash Tree - An Indexing Data Structure over DHTs. In *Proc. of PODC'04*, 2004.

[11] T. Schuett et al. Chord#: Structured Overlay Network for Non-Uniform Load-Distribution. ZIB-Report 05-40, 2005.

[12] The Unicode Consortium. *The Unicode Standard, Version 4.0*. Addison-Wesley, 2003.