DHT Overlay Schemes for Scalable p-Range Resource Discovery

Liping Chen *
Department of Computer Science
Univ of Illinois at Urbana Champaign
lchen2@uiuc.edu

K. Selçuk Candan,Junichi Tatemura
Divyakant Agrawal, Dirceu Cavendish
NEC Laboratories America, Inc
{ksc,tatemura,agrawal}@sv.nec-labs.com

1. Introduction

The *information service* is a critical component of a Grid infrastructure for resource discovery. Although P2P computing paradigm could address some of the scalability issues that plagued Grid resource discovery, most existing Distributed Hash Table (DHT) based P2P overlays have difficulty in treating attribute range queries that are common in resource discovery lookups due to the inherent randomness of hash functions.

Recently, there have been various attempts to solve the range search problem over DHT networks [1, 2, 3, 5]. Central to all of these is a mapping scheme which maps the tree-structured logical index space to some DHT-based physical node space. In this paper, we propose a general framework to put all these under the same umbrella based on how mapping of the tree-structured index that identifies a physical node responsible of a particular range is done through *replication* . We identify three schemes which should cover the spectrum of all meaningful replication schemes: the tree replication scheme (TRS) replicates the entire tree; the path caching scheme (PCS) replicates paths from root to leaves; and the node replication scheme (NRS) replicates individual logical nodes.

Instead of proposing a new tree structure or a new DHT overlay, we focus on investigating the properties of these three general mapping schemes, usable with different tree structures and DHT overlays. We analytically and experimentally compare and contrast these three schemes in the context of p-range queries (point-in-range queries, which are common in Grid resource discovery), in terms of *lookup cost*, *update cost*, and *redirection workload* per peer.

* This work was mainly done during the first author's internship with NEC Labs America.

2. Index Replication Schemes

The logical space are usually tree-structures to support efficient range searches such as B-tree for 1-D data and k-d tree, R-tree, etc. for multidimensional data. These differ in the ways they split the space for efficient access and the ways of maintaining the corresponding data structure. The physical space is a DHT overlay. In the rest of this section, we present three protocols for mapping tree-structured logical data onto a physical space. These three schemes differ in the granularity with which they use *replication*.

Tree Replication Scheme TRS replicates the search hierarchy in its *entirety*. In this respect, it is similar to the scheme use in RST [2]. Unfortunately, a simple analysis shows that to achieve load scalability, the number of directory replicas should be $O(N)$, where N is the total number of nodes in the network.

Path Caching Scheme Most prefix-based range search structures (such as P-grid [1] or [3]) are based on PCS, which construct a single logical tree and perform replication at the physical level. Each physical node has only a carefully selected partial view (the path from the root) of the logical tree.

The Node Replication Scheme NRS replicates each internal node explicitly. Thus, unlike the PCS, replication is done at the logical level itself: for each node the number of logical replicas is proportional to the number of its leaf descendants. Hence each time a node splits, we create one more replica for each one of the nodes along the path from the node to the root.

In TRS, lookup cost is $O(logN)$. For PCS, [1] shows that their randomization scheme can ensure that the *expected* number of hops along the logical path needed is bounded by $log(N)$. Thus the search cost is $O(log^2 N)$. In general, for both PCS and NRS, in a height-balanced tree, the search cost is $O(log^2 N)$. In a non-height-balanced tree, the worst case search cost is $min\{O(h \times logN), O(N)\}$, where $O(N)$ is the cost of a simple broadcast and h is the height of the tree.

In TRS, each update requires $min\{O(KlogN), O(N)\}$

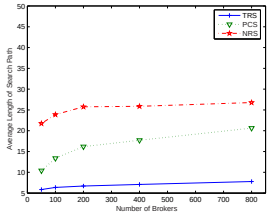

Figure 1. Avg Lookup cost as N increases

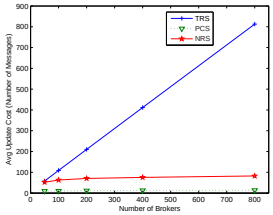

Figure 2. Avg update cost as N increases

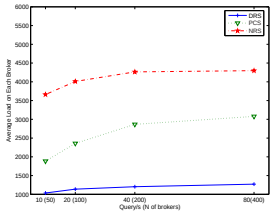

Figure 3. Avg workload as N increases

messages to update replicas. For PCS and NRS, if the height does not need to be balanced, update cost is $O(logN)$ for PCS and $O(log^2N)$ for NRS. If the height needs to be balanced, many times a simple broadcast might do better. In PCS, the update cost is $O(N)$ messages. In NRS, in addition to the $O(log^2N)$ replication cost, the overall cost is $O(log^2N + N)$.

In TRS, the directory lookup load is shared equally among the K replicas, while the DHT redirection load is shared equally among the N peers. In PCS, since the peers are symmetric to each other both in terms of directory-lookup as well as re-direction behavior, both workloads are balanced. NRS is designed for explicit balancing of the load in the system. Each node maintains information about only four other nodes: two of its parent's replicas, left child and right child. Both the replica selection in the update phase and upward traversal in lookup phase are randomized to achieve balanced workload.

3. Evaluation of the Three Schemes

In this section, we evaluate and compare the three mapping schemes using simulation. We used a k-d tree, which does not force height-balancing, as the logical space and Chord [4] as the physical space.

Lookups: Figure 1 shows the average hop counts for the three schemes as the number of brokers increases. TRS is clearly the best. PCS performs better than NRS as internal index nodes are cached at the leaves. However, all three schemes show good lookup scalability.

Updates: Figure 2 shows the average update cost in terms of messages exchanged as the number of brokers increases. TRS shows a linear update cost. PCS provides the best update performance since change effects are only local. The update cost of NRS is slightly higher as the update requires replication of the nodes along the path to the root.

Workload: A good workload scalability would require that as the network (brokers and user queries) grows, the workload on each of the brokers stays stable. Figure 3 shows the average workload on each broker (the number of messages) as the network grows. All three schemes show good scalability.

4. Conclusion

We have presented a general framework that maps tree-structured logical search space for range indexing onto a DHT-based physical node space, with aim of answering range queries, based on three replication schemes. As shown by analysis as well as experiments, these three schemes have different performance characteristics in query cost, update cost, and workload distribution. In general, the TRS scheme may not be applicable to update-heavy environments, like Grid resource brokerage services. PCS and NRS are both scalable and present comparable performance under different resource brokerage scenarios.

References

[1] K. Aberer, P. Cudré-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, R. Schmidt, and J. Wu. Advanced peer-to-peer networking: The P-Grid System and its Applications. *PIK Journal, Special Issue on P2P Systems*, 2003.

[2] J. Gao and P. Steenkiste. An adaptive protocol for efficient support of range queries in DHT-based systems. In *ICNP'04*, 2004.

[3] S. Ratnasamy, J. Hellerstein, and S. Shenker. Range queries over DHTs. In *Technical Report IRB-TR-03-009, Intel Corp.*, 2003.

[4] 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 the 2001 conference on applications, technologies, architectures, and protocols for computer communications*, pages 149–160. ACM Press, 2001.

[5] E. Tanin, A. Harwood, and H. Samet. A distributed quadtree index for peer-to-peer settings. In *ICDE*, 2005.