Magnolia: A novel DHT Architecture for Keyword-based Searching

Ashish Gupta, Manan Sanghi, Peter Dinda, Fabian Bustamante
Department of Computer Science, Northwestern University
Email: {ashish,manan,pdinda,fabianb}@cs.northwestern.edu

The class of DHT-based P2P systems like Chord, Pastry, Tapestry, Kademlia greatly improve over unstructured P2P systems like Gnutella and Kazaa by providing (1) Scalable and efficient $O(\log(n))$ lookup and routing for any document (2) Good load balancing properties for very large number of keys or documents. However, to lookup a document, its complete initial identifier must be known to compute its unique hashed key and route to the correct node, which is a major disadvantage compared to unstructured systems.

Our goal in this ongoing project is to create a DHT-based P2P architecture that supports efficient partial keyword searches in a scalable manner. Some recent proposals for keyword search [2], [4], [1], [5] have suggested storing all documents pointers for a keyword on a node corresponding to keyID=$h(keyword)$. For example, all files which have "usenix" in their title are stored on a single node corresponding to $h(\text{"}usenix\text{"})$. Multiple keyword search can then be made possible by computing the hashes for each keyword and visiting corresponding nodes to fetch all results (which can be processed in the network for boolean operations before returning). Though correct, we argue that this approach does not align well with the goals of a DHT system for very large scale and transient networks. High amount of keyword heterogeneity in occurrence frequency as well as query frequency further aggravate the problem (These have been shown to follow Zipf distribution): (1) Millions of documents corresponding to a common keyword can end up on a single node. Overall, distribution of these document pointers can be heavily skewed over the nodes (2) When a node disappears, all document pointers corresponding to keyword(s) stored on this node are removed from the network, hampering future searches. This is especially problematic if the nodes storing pointers for popular keywords fail. (3) Nodes can be swamped with search traffic for these popular keywords creating routing hotspots (resulting from routing large number of messages to a single destination) as well as query hotspots.

We have designed a simple DHT architecture Magnolia which is not effected by the fore-mentioned problems while simultaneously providing $\log n$ hops for routing and lookup and low, bounded number of nodes visited and traffic generated. Our model scenario is a large scale P2P file sharing system with over 1 million nodes which show high transiency and is responsible for storing over 1 billion documents. Our architecture proposes novel node grouping and key distribution methods using a multi-hashing scheme and makes use of hash function properties to effectively distribute pointers corresponding to every keyword to a tunable number of nodes. Using Multi-hashing each keyword is balanced across a set of nodes in the system with little overlap between different set of nodes, which achieves both good load-balance in terms of traffic and key storage as well as making search highly robust to failures. We want to form these groups such that popular keywords have low probability of being assigned to the same group. We also propose a modified DHT routing architecture which can then store documents and lookup keyword queries in $\log(n)$ hops , though the keyword pointers are mapped to multiple nodes. The amount of traffic generated and number of nodes visited is also low and bounded.

Figure 1 shows the technique of multi-hashing. We have k hash functions $h_1(), ..., h_k()$ where $h_i()$ maps a keyword a m' bit key ($m' < m$, the total number of bits used in nodeID or documentID). For each keyword corresponding to every document instance (which we assume currently are derived from its title or meta information like

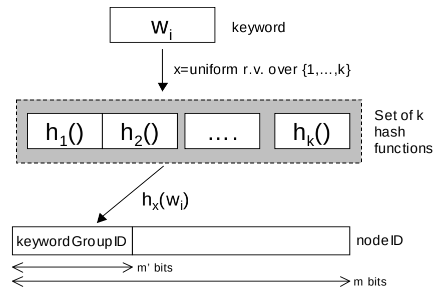

Fig. 1. The Multi-hashing process which maps each instance of a keyword to one of the k possible KeywordGroupIDs for a particular keyword

ID tags in mp3 files) we compute a m' bit key using $h_x(keyword)$ where x is a uniformly distributed random variable over the set $1, ..., k$. The intuition behind doing this is that the m' bit key corresponds to the first m' bits of the m-bit nodeID which is uniformly distributed over all the nodes. If $m' = 16$ and there are one million nodes, on average $\frac{2^{20}}{2^{16}} = 16$ nodes would have the same value for a particular m' bit value also called KeywordGroupID in our system. Since each instance of the keyword can map to any of the $h_i()$ hash functions, it can map to any of the k KeywordGroupIDs and be stored on any of these nodes belonging to these groups. The motivation behind this technique to distribute keyword instances is that using a novel DHT routing architecture these group of nodes belonging to a KeywordGroupID can be reached and searched with low and bounded number of hops, providing the same $\log(n)$ number of hops as the original proposals which route to a single node.

Current Status

We have currently worked out the design and details of the multi-hashing process and the corresponding DHT routing architecture which provides low and bounded response time for storage and lookup. Our Technical Report [3] gives more detail along with analytical treatment for important performance and scalability metrics: Load-balancing of keys (aggregate and per-keyword), routing and lookup performance, traffic generated and number of nodes visited and the routing state kept at each node. Our next step is to conduct a detailed evaluation of the system to measure important metrics using real world keyword and query distributions to provide a better understanding of advantages of Magnolia.

REFERENCES

[1] BAUER, D., HURLEY, P., PLETKA, R., AND WALDVOGEL, M. Bringing efficient advanced queries to distributed hash tables. In *Proceedings of IEEE LCN* (Nov. 2004).

[2] GARCES-ERICE, L., FELBER, P., BIERSACK, E. W., URVOY-KELLER, G., AND ROSS, K. W. Data indexing in peer-to-peer DHT networks. In *24th International Conference on Distributed Computing Systems (24th ICDCS'2004)* (Tokyo, Japan, Mar. 2004), IEEE Computer Society, pp. 200–208.

[3] GUPTA, A., SANGHI, M., DINDA, P., AND BUSTAMANTE, F. Magnolia: A novel dht architecture for keyword-based searching. *Technial Report, Northwestern University* (March 2005).

[4] HARREN, M., HELLERSTEIN, J. M., HUEBSCH, R., LOO, B. T., SHENKER, S., AND STOICA, I. Complex queries in DHT-based peer-to-peer networks. *Lecture Notes in Computer Science 2429* (2002), 242–??

[5] REYNOLDS, P., AND VAHDAT, A. Efficient peer-to-peer keyword searching. In *Middleware* (2003), pp. 21–40.