VHash: An Optimized Voronoi-Based Distributed Hash Table

Double Blind

***Abstract*—Distributed Hash Tables (DHT) provide a fast and robust decentralized means of key-value storage and retrieval and are typically used in Peer-to-Peer applications. DHTs assign nodes a single identifier derived from the hash of their IP address and port, which results in a random overlay network. A random overlay network is not explicitly optimized for certain metrics, such as latency, trust, energy, or hops on an overlay network. It is often desirable to generate an overlay network that is optimized for one or more specific metrics.**

This paper presents VHash, a DHT protocol to construct an overlay optimized for such metrics. VHash exploits a fast and efficient Delaunay Triangulation heuristic in a geometric space. We used VHash to generate an overlay with edges that minimize latency. While we focused on latency in this paper, VHash optimizes on any defined metrics. This overlay outperformed an overlay generated by the Chord DHT protocol in terms of lookup time. VHash provides a robust, scalable, and efficient distributed lookup service.

I. Introduction

A Distributed Hash Table is used provide an overlay network for many P2P applications. Each node in the overlay network maintains a routing table of a subset of other nodes in the overlay. The configuration and rules of the routing table vary from one protocol and another; the entries of the table might be separated by powers of 2 [1], be determined by shared prefixes [2], or be chosen according to a probabilistic distribution [3], but the goal is to minimize the number of overlay hops the distributed lookup needs to make.

A routing table created to minimize overlay hops does exactly that; it does not necessarily create routes with minimized latency. However, what if more information about the nodes could be encoded as part of the overlay, such as the node's latency or energy? We wanted to design an overlay where network metrics, such as inter-node latency, node energy levels, or non-euclidean metrics like trust could be embedded in the network. The most straightforward way is to assign each network metric to a node as a coordinate in a space with the hashed key treated as an additional coordinate. If each node is defined as object in this space, we can then use a distance function to choose the shortest path over these metrics.

This opens up two problems: how do we generate an overlay incorporates the and how do we generate the coordinates that correspond to the We present VHash as our solution to these problems

a DHT designed to take inter-node latency information into account when generating an overlay on a massive scale. VHash creates an approximation of a Voronoi network to define the routing tables and dictate where content is stored in the network. We accomplish this by assigning each node d coordinates, rather than than a single key. The naive method of doing so is to assign coordinates to servers based on the geographic location of nodes. More complex approaches approximate a minimum latency space based on internode latency. Our paper presents the following:

- We describe the VHash protocol and the underlying approximation algorithm for quick and efficient Voronoi region approximations. Our approximation is simple, distributed, and greedy and accurately approximates the Voronoi region of objects in a space with an arbitrary number of dimensions, something than would require a great deal of computation.
- We use VHash to provide us with an overlay for embedding network metrics, our other innovation. We present our basic spring model for embedding nodes in the overlay with latency information and discuss other network metrics that can be used with VHash.
- We created a simulation to prove that the overlays created by VHash are accurate enough for routing messages from arbitrary source nodes to random destination locations. We also show that by embedding the underlay
- We compare the VHash protocol to the other protocols that are based off of Voronoi region approximation. We also contrast the properties of VHash with well known extant protocols.
- We present the related work and discuss what plans we have for embedding different problems in VHash.

II. VHash

The process of generating the keys/coordinates is covered in the next section.

III. Network Metric Embedding

IV. Simulations

Scale free networks are internet shaped[**?**]

V. Related Work

VI. Future Work

References

[1] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *ACM SIGCOMM Computer Communication Review*, vol. 31, pp. 149–160, ACM, 2001.

[2] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Middleware 2001*, pp. 329–350, Springer, 2001.

[3] J. M. Kleinberg, "Navigation in a small world," *Nature*, vol. 406, no. 6798, pp. 845–845, 2000.