VHASH: Spatial DHT based on Voronoi Tessellation

Abstract

Distributed Hash Tables are used as a tool to generate overlay networks for P2P networks. Current DHT techniques are not designed to take the nature of the underlying network into account when organizing the overlay network; they assign nodes locations in a ring or tree, limiting the ability of these networks to be more efficient. We present VHASH as a spatial DHT based on approximate Delunay Triangulation to integrate location information of nodes into overlay network topology. VHASH allows for the creation of P2P networks with faster record look-up time, storage, and maintenance with a geographically diverse set of nodes.

VHash VHash was created to allow for spacial representations to be mapped to hash locations guided by previous work into object location embedding (Beaumont et al., 2007). Rather than focus on minimizing the amount of hops required to travel from point to point we wish to minimize the time required for a message to reach its recipient. VHash actually has a worse worst case hop distance ($O(\sqrt[d]{n})$) than other comparable distributed hash tables ($O(lg(n))$) (Stoica et al., 2001). However, VHash can route messages as quickly as possible rather than traveling over a grand tour that an overlay network may describe in the real world. In addition, Vhash offers routing with constant size memory, as average degree in a metric space is defined as a function of the number of dimensions (8 for a 2 dimensions).
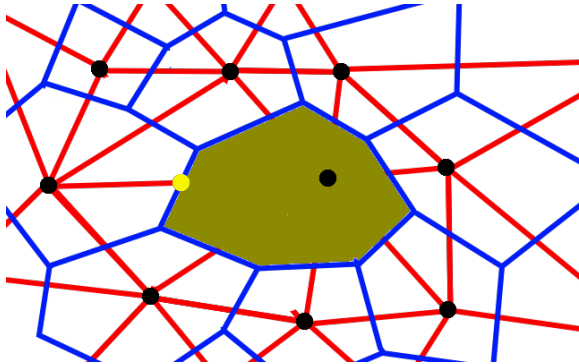

Figure 1: Here, a new node is joining the networks and has established that his position falls in the the yellow shaded Voronoi region.
Triangulation.

Algorithm 1 Vhash Routing

1: P_0 is this node's set of peers
2: N is this node
3: m is a message addressed for L
4: $Forwards$ is the set $P_0 \cup N$
5: find C: member of $Forwards$ which has the shortest distance to L
6: **if** C is N **then**
7:     N is the responsible party.
8:     Handle m
9: **else**
10:     Forward m to C for handling or further routing
11: **end if**   1

Mechanism VHash maps nodes to a d dimension toroidal unit space overlay. This is essentially a hypercube with wrapping edges. The toroidal property makes visualization difficult but allows for a space without a sparse edge, as all nodes can translate the space such that they are at the center of the space. This is a helpful property in ensuring efficient routing to all other nodes.

VHash nodes are responsible for the address space defined by their Voronoi region. This region is defined by a list of peers maintained by the node. A minimum list of peers is maintained such that the node's Voronoi region is well defined. The links connecting the node to its peers correspond to the links of a Delaunay

Message Routing When routing a message to an arbitrary location, a node calculates who's Voronoi region the message's destination is within among the itself and its peers. If the destination falls within its own region, then it is responsible and handles the message accordingly. Otherwise, it forwards the message to the

responsible peer and that peer attempts to route the message. This process mimics a pre-computed A* (Hart et al., 1968) routing algorithm over the network.

In terms of metric length, the routing distance in Vhash converges to length $O(C^d)$. A standard DHT overlay is necessarily $O(lg(n))$ as the distance it takes for the DHT to route 1 hop is equivalent to the distance a Vhash message requires to reach the destination.

Joining and Maintenance Joining the network is a straightforward process. A new node first learns the location of at least one member of the network to join. The joining node then chooses a location in the hash space either at random or based on a problem formulation (such as geographic location or latency).

After choosing a location, the joining node sends a "join" message to its own location via the known node. The message is forwarded to the current owner of that location, the "parent" node. The parent node replies with a maintenance message containing its full peer list. The joining node uses this to begin defining the space for which it is responsible. The parent adds the new node to its own peer list and removes all his peers occluded by the new node. Regular maintenance propagates the new node's information and repairs the overlay topology.

Each node in the network performs maintenance periodically by a maintenance message to its peers. The maintenance message consists of the node's information and the information on that node's peer list. When a maintenance message is received, the receiving node considers the listed nodes as candidates for its own peer list and removes any occluded nodes.

Algorithm 2 VHash Greedy Peer Selection

1: $Candiates$ is the set of candidate peers
2: $Peers$ is the set of this node's peers
3: $Canidates$ is sorted by each node's closeness to this node
4: The closest member of $Canidates$ is popped and added to $Peers$
5: **for all** n in $Canidates$ **do**
6:    c is the midpoint between this node and n
7:    **if** Any node in $Peers$ is closer to c than this node **then**
8:       reject n as a peer
9:    **else**
10:       Add n to $Peers$
11:    **end if**
12: **end for**

Data Storage Resources in the network, be it raw data or assigned tasks, are assigned hash locations. The node responsible for a given hash location is responsible for the maintenance of that resource. When a node fails, its peers take responsibility of its space. Thus it is important to provide peers with frequent backups of a node's assigned resources. That way, when a node fails, its peers can immediately assume its responsibilities.

A resource is accessed by contacting the node responsible for the resource. However, the requester generally has no idea which node is responsible for any particular resource. The data request message is addressed to the location corresponding to the resource, rather than the node responsible for that location. The message is forwarded over the overlay network, each hop bringing the node closer until it reaches the responsible node, who sends the resource or an error if the resource does not exist.

References

Olivier Beaumont, A-M Kermarrec, Loris Marchal, and Etienne Rivière. Voronet: A scalable object network based on voronoi tessellations. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pages 1–10. IEEE, 2007.

P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100–107, 1968. ISSN 0536-1567. doi: 10.1109/TSSC.1968.300136.

Ion Stoica, Robert Morris, David Karger, M Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *ACM SIGCOMM Computer Communication Review*, volume 31, pages 149–160. ACM, 2001.