A Distributed Greedy Heuristic for Computing Voronoi Tessellations With Applications Towards Peer-to-Peer Networks

Brendan Benshoof Andrew Rosen
Anu G. Bourgeois Robert W. Harrison
Department of Computer Science, Georgia State University

May 13, 2015

Outline

Distributed Hash Tables

- Abstractly, a DHT is a mechanism for maintaining a large state in a decentralized network.
- In practice, the state is a large number of key,value records.
- A Distributed hash table assigns those records to servers and routes request for those records to those servers.
- Current incarnations of Distributed hash tables assign servers and records locations in an arbitrary metric space.
- Servers are assigned responsibility for records that are “close” to them, and to peer with “nearby” servers.
- DHTs currently use a variety metric spaces.

Applications

- *P2P file sharing* is by far the most prominent use of DHTs. The most well-known application is BitTorrent [**?**].
- *Distributed Domain Name Systems* (DNS) have been built upon DHTs [**?**] [**?**]. Distributed DNSs are much more robust that DNS to orchestrated attacks, but otherwise require more overhead.
- Distributed *machine learning* [**?**].
- Many *botnets* are now P2P based and built using well established DHTs [**?**]. This is because the decentralized nature of P2P systems means there's no single vulnerable location in the botnet.

Extant Varieties of DHT

- Ring Based DHTs
  - Chord
  - Pastry
  - Tapestry
- Tree Based DHTs
  - CAN
  - Kademlia

How are DHTs and Vonroi Tesselation/Delunay Trianguation related?

A Server is responsible for records "close" to it (Voronoi Triangulation) A Server peers with other servers that bound it's Voronoi Region (Delunay Triangulation) DHTs often have peers in excess of the Delaunay triangulation to shorten lookups, however the peers that are the Delunay neighbors

- Ring Based DHTs
  - Chord: a unidirectional modulus ring metric
  - Pastry, Symphony: bidirectional modulus ring
- Tree Based DHTs
  - CAN: Euclidean distance
  - Kademlia: XOR distance

Why do we need DGVH?

- The different topologies DHTs utilize present optimization trade-offs (lookup latency, number of lookup hops, network robustness, availability, processing overhead)
- The primary effort in implementing a new metric space in a DHT is implementing Voronoi Regions/Delunay Triangulation algorithm in that metric.
- DGVH allows many metrics to be tested without requiring the design and development effort of generating an exact Voronoi Regions/Delunay Triangulation algorithm.

Distributed Greedy Voronoi Heuristic

- Geometrically intuitive method of approximating the one-hop delaunay peers of a Node
- Is guaranteed to form a connected mesh (unlike k-nearest heuristic)
- Can be utilized in any continuous metric space

DGVH Algorithim

1: Given node n and its list of *candidates*.
2: *peers* $\leftarrow$ empty set that will contain n's one-hop peers
3: Sort *candidates* in ascending order by each node's distance to n
4: Remove the first member of *candidates* and add it to *peers*
5: **for all** c in *candidates* **do**
6: m is the midpoint between n and c
7: **if** Any node in *peers* is closer to m than n **then**
8: Reject c as a peer
9: **else**
10: Remove c from *candidates*
11: Add c to *peers*
12: **end if**
13: **end for**

Visual Intutution

PUT PICTURES HERE!!!!

Realistic Canidate set Size

- In application, a single node will not need to calculate the triangulation for every peer in the network, rather it will only calculate it's own peers.
- A smart "join" process will prevent the Candidate Set from reaching $O(n)$
- Expected size of the candidate set is $O(degree^2)$ which compromises current peers and 2-hop peers
- The average degree of many metric spaces is $O(1)$, then therefore DGVH will practically run in $O(1)$ in those metric spaces.