Advanced Topics in Computer Systems, CS262B
Prof. Eric A. Brewer

DHT Geometry and Proximity

April 1, 2004

I. Background

Lots of DHTs, very hard to compare

- what really matters?
- idea: compare the component ideas, not the artifacts
- generalize the DHTs to make them more comparable
- better model of latency: "view from here"

Key results:

- geometry matters: but mostly as a way to ensure flexibility
- flexibility = more choice => better fault tolerance, better proximity

II. Geometry

Static resilience = abilty to route well without repair

Geometry implies how many possible choices you have for neighbors (neighbor selection)

May also imply how many choices you have among neighbors at a given hop

Example: can take Chord steps out of order => more choice at a given point

Flexilibility: what is not prevented by the geometry? (e.g. proximity, multiple choices?)

- neighbor selection: how many potential neighbors do you have? (choose O(lg n) of those)
- route selection: number of next possible hops?

Sequential neighbors add flexibility, but lead to slow (O(n)) progress; mostly for correctness

Basics:

- Tree: lots of nodes that can make progress, but then only one is actually used for routing
- Hypercube (CAN): like a tree but can route in any order, but only one possible neighbor in each direction!
- Ring: can be flexible in neighbors (original Chord was not), and in order. Can also take longer paths
- Hybrid: tree + sequential (Pastry)

Static resilience:

- depends mostly on flexibility of routes, since neighbors don’t change!
- implies that ring and hypercube do well.. hybrid OK, and tree not so good
- seq neighbors really help with resilience, at some cost of stretch
- seq neighbors are better than reg neighbors for resilience, but worse for stretch

III. Proximity and Latency

PNS = choose proximate neighbors, requires neighbor flexibility

- PNS(k) = PNS based on sampling k nodes (an approximation)

PRS = choose proximate routes, requires route flexibility

Latency model: trace-based view of the latency distribution from a single point

- different and more accurate that previous assumptions

hypothesis: resilience depends on routing choices, performance depends on neighbor choices

PNS much more effective, mostly because there are many more neighbor choices than route choices!

- ring: 2^i choices for PNS vs i choices for PRS

Geometry doesn’t matter much other than its flexibility for PNS and PRS

- compare three pairs: PNS+PRS: ring vs XOR, PNS only: ring vs tree, PRS only: ring vs. hypercube
- actual latencies do go down significantly

IV. Local Convergence

Idea: want meeting points on the way to the destination

- enables caching, multicast, aggregation

Test: cluster of m nodes (the domain) reaching out to rest of network

- how many exit points are there?
- PNS limits exit points well
- but need really good sampling to find local neighbors...
- geometry matters indirectly via PNS support (assuming sampling)

V. Other thoughts

Does ring win?

- enables both PNS and PRS
- may win due to minimal constraint! (implies more flexibility)

Can you do better with constant state?

- need many possible neighbors for PNS, but need log n of them fundamentally
- need multiple paths (PRS) for robustness, but again may only need k choices (not lg n)
- butterfly failure in this paper is not very convincing
- does constant state matter? (is $k < \lg n$ for real networks?)

Better measure of resilience (during repair)?

- a resilient network enables slow repair and high availability
- likely related to expansion/conductance