Scalable Content-Addressable Network

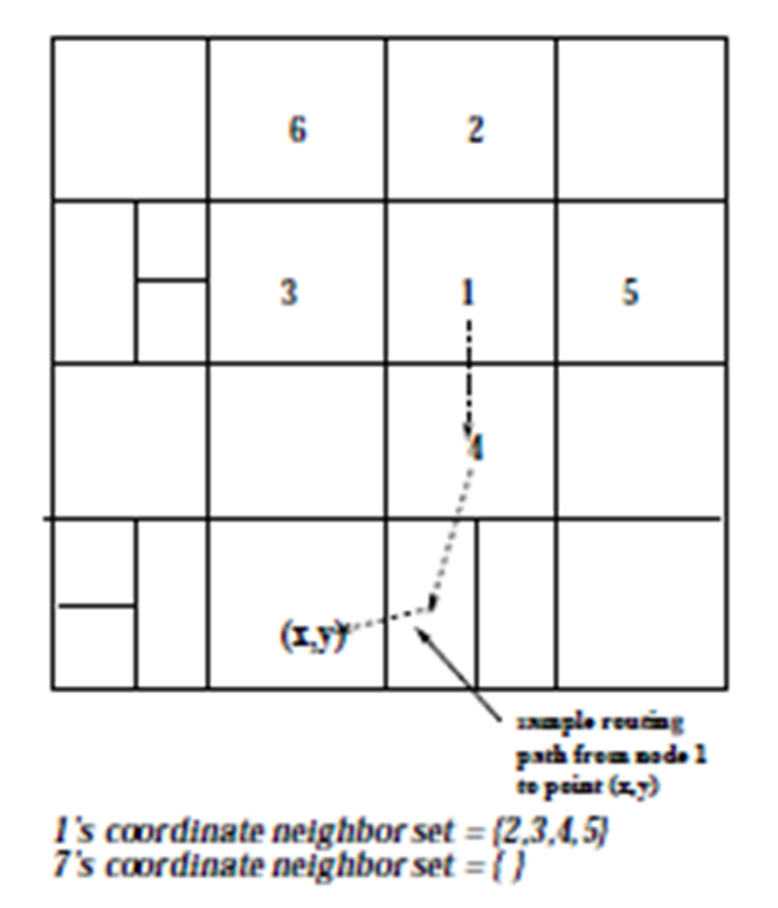
Eireann Leverett

How

- Torus
- Dimensions
- Nodes
- Hashing
- Realities
- Zone takeover
- Routing
- Overloading Zones

- We use a Torus because it is un-ending in each dimension. It is a circle where the last address neighbours the first, in every dimension.

Dimensions, nodes, & takeover



Hashing

- Critical to the success of the scheme
- Should distribute data uniformly across the space
- Choose your hash for other interesting properties (speed, uniqueness, timestamp)
- You can use multiple hashes, to distribute to multiple points (or the same hash transformed)

Overloading zones & Caching

- When Keys < Nodes
- Resists node failure
- Logical Rules Expansion
- Its distributed temporally and spatially
- Protect against byzantine failure
- When content is frequently requested give a copy to your neighbours
- Reduces latency and hops, and scales 2d
- Choosing your dimensions carefully for content helps

Realities

- It’s distributed logic- spatially
- You double the number of neighbours for each +1 to reality and increase the potential source of content by 1.
- With cacheing and routing this becomes large & beneficial

Routing

- Routing in co-ordinate spaces is fairly easy
- Modulo arithmetic means there is at least 2d naïve paths to data
- d space in n zones avg routing is (d/4)(n^1/d) hops
- Grow # of nodes while only growing path O(n^1/d)
- Only need to know your neighbours

Why?

- Content Availability
- Small routing tables
- Application level overlay
- Replication
- Node Failure
- Scalable
- Latency reduction
- Robust, reliable, distributed.

Latency

- Great reductions in latency through dimensionality and realities
- Caching handles load, but also reduces latency
- Measured in RTT not just hops

Summary & Criticisms

- Distributed
- Scalable
- Flexible
- Resistant to node failure/offline
- Low Latency
- Many parts simple to implement
- Content storage
- Overlay

- Choice of hash and design time decisions important
- Hash function bottle neck on size of storage
- Security an open question (bad nodes)
- Freshness of data?
- How is data found? Who?
- Properties are not dynamic