CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

P2P SYSTEMS

Lecture G

PASTRY

Pastry

- Designed by Antony Rowstron (Microsoft Research) and Peter Druschel (Rice University)
- Assigns ids to nodes, just like Chord (using a virtual ring)
- **Leaf Set** – Each node knows its successor(s) and predecessor(s)

Pastry Neighbors

- **Routing tables** based **prefix matching**
  - Think of a hypercube
- Routing is thus based on prefix matching and is thus log(N)
  - And hops are short (in the underlying network)

recall chord uses finger table / log(N) math

REFERS TO Brown tapestry project, explained much better !!!!!!

Pastry Routing

- Consider a peer with id 01110100101. It maintains a neighbor peer with an id matching each of the following prefixes:
  - 0*
  - 01*
  - 011*
  - … 0111010010*

* == end == starting from bit is different from this peer's corresponding bit

1st mismatching bit == 7th
first 6 bits == matching prefix

- When it needs to route to a peer, say 01110111001, it starts by forwarding to a neighbor with the largest matching prefix, i.e., 011101*

motivation: IP address similarity == closer network distance
neighbor w largest prefix == known server's IP that is closet to target to route to

Pastry Locality

- For each prefix, say 011*, among all potential neighbors with a matching prefix, the neighbor with the shortest round-trip time is selected
- Since shorter prefixes have many more candidates (spread out throughout the Internet), the neighbors for shorter prefixes are likely to be closer than the neighbors for longer prefixes
- Thus, in the prefix routing, early hops are short and later hops are longer
- Yet overall "stretch," compared to direct Internet path, stays short

Summary of Chord and Pastry

- Chord and Pastry protocols
  - More structured than Gnutella
  - Black box lookup algorithms
  - Churn handling can get complex
  - *O(log(N))* memory and lookup cost
    - *O(log(N))* lookup hops may be high
    - Can we reduce the number of hops?