GSCI 1380: Day 9 (Tapestry Continued)
Last class

* Consistent hash keep
* Scaling consistent hash
* Chord (Table size = \( \log_2(N) \))
* Tapestry (Table size = \( \log_b(N) \times B \))

This Class

* Tapestry In Detail
  1. Routing Table Creation (closest node)
  2. Key lookup
  3. Overview
  4. Deleting Nodes
  5. Adding Notes

\( \text{Node} = \text{Server} \)
Tapestry Key Structure

Key space: \( N = 64 \) \( \leq 0 \ldots 63 \)

Base: \( 4 \) \( \leq 0 \ldots 33 \)

\[ \log_4 64 = 3 \]

First row of table

000 - 333

Base ID space

Starts with 0

Key ID space = 0 ... 63

Pick a base \( \Rightarrow \) 000 - 333

New representation of key space with new "base"

Number of cols = Base

Table size:

\[ \text{base} \times \log_4 N \]

\# of cols \# of rows

\( \log_4 N \)

\# of rows

O 1 2 3

23 *

2 * * *

* * *

First digit wildcard

Route for 231

Wildcard

0
Node IDs: 011, 211, 222, 231, 201, 300, 333

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>* 011</td>
<td>* 211</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>* 201</td>
<td>211</td>
<td>222</td>
<td>231</td>
</tr>
<tr>
<td>2</td>
<td>211</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Options for closeness:
1. random
2. no hop (minimize latency)
3. $\min(|A - B|) = \text{absolute difference in IDs}$

Closeness:

$$a \leftarrow b$$

$$\min(|a - b|, |a - c|)$$

Points about table creation:
1. follow the rules (constraints placed by columns & rows)
2. if you have several options, pick based on closeness
3. there may be empty entries
4. always one entry per row
Context / Overview

Client ➔ LB ➔ FE ➔ FC ➔ FE ➔ Backend/Replica/Storage Shards

Clusters

Tapestry / Dynamo

Key-Value Store (distributed hash table)

Storage Servers (shards)

set(keyID, "email")

get(keyID)

Pic ➔ Move

abstraction of a large key-value store

e.g. Google's MapReduce (GFS)
P2P (Peer to Peer)

Boston

Japan

So. Africa

0. H/W < different resources
1. Notes come and go at will
2. Geo distributed
always publish all your data (every 5 minutes) → ensures that your data stays in tapestry even after failures

Publishing frequency is predefined
Lookups (in Tapestry) (Node) Server in charge of some set of IDs

Goal: find "Root"

Root = lookup (231)

Pick a random Node

Node 231

232

23*

$O(\log B N)$
Prefix Routing

* every node
* tries to find an entree in its table
  that has the largest overlap
* then forwards you to this table
Node IDs: 011, 211, 222, 231, 201, 300, 333

<table>
<thead>
<tr>
<th>KeyID</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>011</td>
<td>0</td>
</tr>
<tr>
<td>222</td>
<td>1</td>
</tr>
<tr>
<td>333</td>
<td>2</td>
</tr>
</tbody>
</table>

**Lookup** (KeyID, step) →

- **row** = step
- **col** = KeyID at (step)
- **nextNode** = Table[**col**, row]
- step++
- **nextNode**.lookup(KeyID, step)

start @ 0

```
lookup (333, 0) → 211
lookup (333, 1) → 300
lookup (333, 2) → 333
```
Node IDs: 011, 211, 222, 231, 201, 300, 333

<table>
<thead>
<tr>
<th>Node ID</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>011</td>
<td>-</td>
<td>211</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>211</td>
<td>211</td>
<td>222</td>
<td>222</td>
<td></td>
</tr>
<tr>
<td>21*</td>
<td>211</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

nextNode = Table[col, row]
step++
nextNode, lookup(keyID, step)

lookup(keyID, step)
row = step
col = keyID % (step)

Surrogate routing? routing when you have an empty entry
Node failure/removal

Graceful
* Say Good bye to nodes & hand over your data

1. Who to give your data to?
   * New root

2. How to update broken routing tables
   * need to figure out which tables are impacted by exit
   * Use backpointers to determine nodes to notify

not Graceful
* Just disconnect
* Failure is detected w/ heartbeats
* Data persists because clients republish the data
im adding you to my routing table

put node into backpoint list if it adds you to its routing table
Summary

1. Overview/Context for Tapestry
2. Building a route table
3. Hosting in Tapestry (lookups)
4. Exiting the Tapestry P2P network (graceful & ungraceful)
5. How to add nodes into the network
Node IDs: 011, 211, 222, 231, 201, 300, 333
Node IDs: 011, 211, 222, 231, 201, 300, 333