Day 5
CSD 1380
Cast Class
RPC Semantics & Mechanism

at-least-once
at-most-once
exactly-once

request retry
provide
duplicate suppress
response replay
different combinations

request + retry
⇒
"at least once"

request + suppress
⇒
suppress duplicates

suppress duplicates + response
⇒
restart
<table>
<thead>
<tr>
<th></th>
<th>at least once</th>
<th>at most once</th>
<th>exactly once</th>
</tr>
</thead>
<tbody>
<tr>
<td>at least</td>
<td>retry</td>
<td>reply</td>
<td></td>
</tr>
<tr>
<td>once</td>
<td>suppress</td>
<td>response</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>reply</td>
<td></td>
</tr>
</tbody>
</table>
Data filtering

map/reduce

Data

map

reduce

1. distance (network distance)
2. uneven data partition
3. background mappers speeds (server speeds)
4. server failures/lost jobs

select count(*) from brown where name = "theo"

reduce

filter/map

[CPU monitoring] monitoring
1/0 throughput
H/W specification
Load balancing
Heartbeats
n/i topology
Problem detection

- Heartbeats (servers that keep going down)
- Progress reports (helps track fast vs slow)
- Monitoring (CPU/IO/queue length)

Solution (Designs)

- Quarantine servers that frequently go down
- Inspect requests

How to define “thresholds” = instead of static threshold; use percentile

Productivity rating = send job that are proportional in size to ratings

- Increase / decrease the # of processes on that server

Cloning request – where two servers do the same thing if you pick fastest?

- Reduce total time
  - Original server might be slow now but that can change
  - Just care about who finished first
monitoring \[\Rightarrow\] weights for either load balancing
or placement of map/reduce workers/tasks

Summary of today

- Map reduce (structure)
- * signals for detecting issues (heartbeats, progress, monitoring)
- * ways to address fix issues (retry, cloning, quarantining)

Next week: Load Balancing