Threshold-DHT: Optimizing Data Recovery for Wide Area Storage Systems

Hakim Weatherspoon
IRIS Student Workshop
November 7, 2004

Joint Work with Byung-Gon Chun, Dennis Geels, Chiu Wah Kelvin So, and John Kubiatwicz

Goal

- Redundancy
  - duplicate data to protect against data loss
- Place data throughout wide area
  - Data availability and durability
- Continuously repair loss redundancy as needed
  - Detect permanent failures and trigger data recovery

Challenge

- Permanent node failure
  - Data is lost
- Transient node failure - noise
  - Nodes return from failure with data
- Challenge - when to trigger data recovery?
  - Given noisy node failure signal

Solution - Extra Redundancy

- Given a minimum data availability threshold
  - Calculate number of required replicas, \( \theta = \frac{\log e}{\log (1 - a)} \)
    - Average node availability \( a \)
    - Target data availability \( 1 - \epsilon = 1 - (1 - a)^h \)
  - Add extra (expendable) replicas

- Trigger data recovery
  - Require extra number of replicas to simultaneously fail
  - Reduce false positives of transient failures
    - Delay recovery until nodes permanently removed

*Blake and Rodrigues. “High availability, scalable storage, dynamic peer networks: pick two*
Data Recovery in a DHT

- **Root**
  - Stores object on a put req
  - Returns object on a get req
- **Root set**
  - Redundant set of root nodes
- **Trigger Data Recovery**
  - After node failure signal
  - Shift new node into root set
  - Copy data to new node

Threshold-DHT

- Add extra replicas
- Trigger Data Recovery iff remaining < threshold

Evaluation Methodology

- **Trace-driven simulation**
- **Model maintaining data on PlanetLab**
- **Create trace using all-pairs ping**
  - Collected from February 16, 2003 to October 6, 2004
- **Measure**
  - Number of triggered data recovery v. time
  - Bandwidth per node v. time
  - Average bandwidth per node v. total replicas
    - total = threshold + extra

*Jeremy Stribling http://infospect.planet-lab.org/pings*
Simulation

Number of Available Nodes v. Time

<table>
<thead>
<tr>
<th>Time</th>
<th>0</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
<th>300</th>
<th>350</th>
<th>400</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb-07</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>May-07</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sep-07</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-07</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mar-08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jul-08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Evaluation

- $th = 9$ replicas (minimum data availability thresh)
  - Given target 6 9’s of data availability and
  - 0.8 average PlanetLab node availability
- Vary extra replicas 0 ... 52 (i.e. total 9 ... 61)
- Initial results use failure timeout $to = 1$ hr
- Also, vary failure timeout from 15 min ... 8 days

Evaluation

- Current DHTs do not use extra redundancy ($n = th = 9$)
  - Constantly trigger data recovery due to transient failures
- Threshold-DHT triggered data recovery due to combination of node failures and significant node joins

Evaluation

- Current DHTs use aggressive timeouts and no extra replica
  - E.g. $to = 15$ min and $n = th = 9$
- Extra redundancy uses significantly less bandwidth
- Moderate extra redundancy is sufficient
Conclusion

• Replace lost redundancy due to permanent failures
• Wide-area systems experience transient failures
• Extra redundancy and a minimum data availability threshold
  - Absorb noise due to transient failures
  - Reduce rate of triggering data recovery
• Reduce wide-area data maintenance bandwidth