A High-performance In-memory Web Proxy Cache

Zhengxiang Pan

April 30, 2003
Road Map

- Background
- Design & Implementation
- Experiments
- Conclusion
The Simultaneous Proxy Evaluation (SPE) architecture

The SPE architecture (Courtesy [2])
Motivation

- Some objects are uncachable
  - cache settings in HTTP
  - CGIs
  - Forms & URLs contain “?”

- The difference of the response time of a cache hit and a cache miss for a same object

- Related works
  - Modified squid
Implementation

- Language: Java
- Platform: JVM 1.3.1.02 or later
- Request Header Rewriting
  - Connection: Keep-Alive $\rightarrow$ Connection: Close
  - Request-URI: absoluteURI $\rightarrow$ abs_path + Host
- Multi-thread
  - Socket reading and writing are blocking
  - Using java Thread mechanism
Implementation

- Identifying request
  - GET and HEAD:
    - URI
    - If-Modified-Since,
    - If-Unmodified-Since,
    - If-Match,
    - If-None-Match,
    - and If-Ranges
  - POST: plus content’s MD5 digest
Implementation

- Hash Table
  - Key.hashCode(): compute hash code for bytes
  - Key.equals(): if two byte arrays are identical

- Reproducing response time
  - Record time stamp as base point when receive a request
  - Miss: Wrap a chunk of data into an Packet object with the relative time stamp
  - Hit: Unwrap an Packet and send it at the recorded time

- Memory management:
  - Evict oldest object
Experiments

- Conducted in WUME lab (Thanks!)
- CAProxy V.S. Modified-Squid
- Httperf as client sending a list of URLs
- Two runs for each proxy
- Data set: 300 URLs from NLANR log
Results

- **Cacheability**
  - Modified-Squid: 6% uncacheable
  - CAProxy: Cache everything

- **Difference in response time of miss/hit pairs**

<table>
<thead>
<tr>
<th></th>
<th>CAPROXY</th>
<th>MODIFIED-SQUID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>10.6</td>
<td>46.6</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>63</td>
<td>173.5</td>
</tr>
<tr>
<td>Median</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Percentage of zeros</td>
<td>95.70%</td>
<td>2.90%</td>
</tr>
</tbody>
</table>
Conclusion

- Can cache nearly all the HTTP responses regardless of their cacheability settings

- Significantly reduce the difference in response time of a cache miss and a cache hit of a same object
Reference


