Squirrel: A peer-to-peer web cache

Sitaram Iyer (Rice University)

Joint work with
Ant Rowstron (MSR Cambridge)
Peter Druschel (Rice University)

PODC 2002 / Sitaram Iyer / Tuesday July 23 / Monterey, CA
Web Caching

1. Latency,
2. External traffic,
3. Load on web servers and routers.

Deployed at: Corporate network boundaries, ISPs, Web Servers, etc.
Web Cache

Centralized Web Cache

Browser

Cache

Client

Corporate LAN

Internet

Web Server
Cooperative Web Cache
Decentralized Web Cache

Internet

Squirrel

Browser

Cache

Corporation LAN

Web Server

Client

Browser

Cache

Client
Distributed Hash Table

Peer-to-peer location service: Pastry

- Completely decentralized and self-organizing
- Fault-tolerant, scalable, efficient

Operations:
- Insert(k,v)
- Lookup(k)

Peer-to-peer routing and location substrate

(nodes)

\(<key,value>\)
Why peer-to-peer?

1. Cost of dedicated web cache
   No additional hardware

2. Administrative effort
   Self-organizing network

3. Scaling implies upgrading
   Resources grow with clients
Setting

- Corporate LAN
- 100 - 100,000 desktop machines
- Located in a single building or campus
- Each node runs an instance of Squirrel
- Sets it as the browser’s proxy
Mapping Squirrel onto Pastry

Two approaches:

• Home-store
• Directory
Home-store model

client

URL
hash

home

LAN Internet
Home-store model

…that’s how it works!
Directory model

Client nodes always cache objects locally.

Home-store: home node also stores objects.

Directory: the home node only stores pointers to recent clients, and forwards requests.
Directory model

client

Internet

LAN

home
Directory model

Randomly choose entry from table
Directory: Advantages

Avoids storing unnecessary copies of objects.

Rapidly changing directory for popular objects seems to improve load balancing.

Home-store scheme can incur hotspots.
Directory: Disadvantages

Cache insertion only happens at clients, so:

• active clients store all the popular objects,
• inactive clients waste most of their storage.

Implications:
1. Reduced cache size.
2. Load imbalance.
Directory: Load spike example

- Web page with many embedded images, or
- Periods of heavy browsing.

Many home nodes point to such clients!

Evaluate …
## Trace characteristics

<table>
<thead>
<tr>
<th></th>
<th>Redmond</th>
<th>Cambridge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total duration</td>
<td>1 day</td>
<td>31 days</td>
</tr>
<tr>
<td>Number of clients</td>
<td>36,782</td>
<td>105</td>
</tr>
<tr>
<td>Number of HTTP requests</td>
<td>16.41 million</td>
<td>0.971 million</td>
</tr>
<tr>
<td>Peak request rate</td>
<td>606 req/sec</td>
<td>186 req/sec</td>
</tr>
<tr>
<td>Number of objects</td>
<td>5.13 million</td>
<td>0.469 million</td>
</tr>
<tr>
<td>Number of cacheable objects</td>
<td>2.56 million</td>
<td>0.226 million</td>
</tr>
<tr>
<td>Mean cacheable object reuse</td>
<td>5.4 times</td>
<td>3.22 times</td>
</tr>
</tbody>
</table>
Total external traffic

- Centralized cache
- Directory
- Home-store
- No web cache

Per-node cache size (in MB) vs. Total external traffic (GB) [lower is better]
Total external traffic

- No web cache
- Directory
- Home-store
- Centralized cache

Per-node cache size (in MB)

Total external traffic (GB) [lower is better]

Cambridge
LAN Hops

Redmond

% of cacheable requests

0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

Total hops within the LAN

0 1 2 3 4 5 6

Centralized  Home-store  Directory

% of cacheable requests
LAN Hops

Cambridge

LAN Hops

% of cacheable requests

Total hops within the LAN

Centralized  Home-store  Directory
Load in requests per sec

Max objects served per-node / second

Number of times observed

Home-store
Directory
Load in requests per sec

Cambridge

Max objects served per-node / second

Number of times observed

1e+07
1e+06
100000
10000
1000
100
10
1

Home-store
Directory
Load in requests per min

Redmond

Max objects served per-node / minute

Number of times observed

Home-store
Directory
Load in requests per min

Max objects served per-node / minute

Cambridge
Fault tolerance

Sudden node failures result in partial loss of cached content.

Home-store: Proportional to failed nodes.
If 1% of Squirrel nodes abruptly crash, the fraction of lost cached content is:

<table>
<thead>
<tr>
<th></th>
<th>Home-store</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Redmond</strong></td>
<td>Mean 1%</td>
<td>Mean 1.71%</td>
</tr>
<tr>
<td></td>
<td>Max 1.77%</td>
<td>Max 19.3%</td>
</tr>
<tr>
<td><strong>Cambridge</strong></td>
<td>Mean 1%</td>
<td>Mean 1.65%</td>
</tr>
<tr>
<td></td>
<td>Max 3.52%</td>
<td>Max 9.8%</td>
</tr>
</tbody>
</table>
Conclusions

• Possible to decentralize web caching.
• Performance comparable to a centralized web cache,
• Is better in terms of cost, scalability, and administration effort, and
• Under our assumptions, the home-store scheme is superior to the directory scheme.
Other aspects of Squirrel

• Adaptive replication
  – Hotspot avoidance
  – Improved robustness

• Route caching
  – Fewer LAN hops
Thanks.