Load Balancing and Cluster-based Internet Services

Kai Shen
Dept. of Computer Science, University of Rochester

Scalable Internet Services
- Throughput: Serving a large number of requests per minute.
- Response time: Nobody wants to wait for more than two seconds.
- Reliability: With 1,000 servers and 4,000 disks in your machine room, something is gonna break every day!!

Server Replication
- Multiple servers to provide the same service
  - More throughput: if each Web server can answer 10 requests/second, then ten Web servers can answer 100 requests/second (Well, at least in theory)
  - Better reliability
  - Faster response??
- Challenges:
  - Figure out who is least loaded
  - Figure out who is dying

Load Balancing on Internet Servers
Technique 1 - DNS Rotation
Discussions on DNS Rotation

- Problems
  - DNS Caching
  - Rigid load balancing policy
    - no adjustment in response to runtime load changes
    - slow or no adjustment in response to failures
- Is there anything good about it?
  - Require almost no change on the existing Internet/server architecture

Discussions on Cooperative Offloading

- Advantages:
  - More flexible policy is possible
  - Be more responsive to runtime workload and server failures
- Problems
  - Need a lot more software changes on the Web servers
  - Longer delay
Discussions on Load Balancing Switch

- Different with cooperative offloading
  - We are messing around with TCP/IP kernel
  - better performance - no extra application-level processing
- Changes required on the party involved
  - DNS server??
  - Web server??
  - client??
  - switch??????

Load Balancing Policies in LB Switches

- Simple rotation
- Least number of active requests
- Shortest response time

Modern LB Switch Construction

Much like an Ethernet switch or a hardware IP router:
switching at wire speed

Cluster-based Internet Services

- Functional partitioning
- Server replication
- Data partitioning
  - Scalability: hosting a large amount of persistent data.
Data Partitioning for an Index Search Engine

The index database is split into many partitions ⇒ better scalability

<table>
<thead>
<tr>
<th>partition 1</th>
<th>partition 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java:</td>
<td>Mexico:</td>
</tr>
<tr>
<td>Page #123</td>
<td>Page #124</td>
</tr>
<tr>
<td>Page #357</td>
<td>Page #124</td>
</tr>
<tr>
<td>...</td>
<td>Page #377</td>
</tr>
</tbody>
</table>

Food: Page #124 | Page #468 | ... 

Medicine: Page #12 | Page #468 | ... 

A Cluster-based Index Search Engine

Index servers

Document servers

Local-area network

Firewall

Switch

Internet

Web server/Query handlers

Index servers (partition 2)

Index servers (partition 1)

Summary

- Load balancing on Internet servers
  - DNS rotation
  - cooperative offloading
  - LB switches
- Cluster-based Internet servers
  - functional partitioning
  - data partitioning
  - replication
- Is there anything missing?