Chapter 6: Distributed Systems: The Web

Fall 2013
Jussi Kangasharju
Chapter Outline

- Web as a distributed system
- Basic web architecture
- Content delivery networks
- Replication of web applications
Web: Distributed or Not?

- Is the web a distributed system?
- Recall our definition:
  - Collection of independent computers → OK
  - Appears as single coherent system → ?!??
- Single coherent system = transparencies fulfilled?
- Sharing of resources → OK
Traditional Web-Based Systems

1. Get document request (HTTP)

2. Server fetches document from local file

3. Response
Multitiered Architectures

1. Get request

HTTP request handler

Web server

3. Start process to fetch document

CGI program

5. HTML document created

4. Database interaction

Database server

6. Return result
Important Elements

- Browser
- Servers and server farms
- Proxies
- Caching proxies
Processes – Clients (1)

Figure 12-5. The logical components of a Web browser.

- User interface
- Browser engine
- Rendering engine
  - Network comm.
  - Client-side script interpreter
  - HTML/XML parser
Processes – Clients With a Proxy

Terminology:
Proxy = simply proxying of requests and responses
Caching proxy = proxy with a cache

Commonly “proxy” = “caching proxy”
Web Server Clusters (1)

Redirection independent of requested content
Web Server Clusters (2)

Redirection as function of requested content
Content-aware vs. Content-agnostic

- **Content-aware server selection:**
  - Allows fine-grained selection and allocation of resources
  - Higher overhead at redirection point
  - No need to replicate all content on all servers

- **Content-agnostic server selection:**
  - Typically DNS load balancing
  - All servers must have identical content
  - Very high traffic → Even load distribution
Web Proxy Caching

1. Look in local cache

2. Ask neighboring proxy caches

3. Forward request to Web server

HTTP Get request
Refresher: Names in the Web

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Host name</th>
<th>Pathname</th>
</tr>
</thead>
<tbody>
<tr>
<td>http://</td>
<td><a href="http://www.cs.vu.nl">www.cs.vu.nl</a></td>
<td>/home/steen/mbox</td>
</tr>
<tr>
<td>(a)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Host name</th>
<th>Port</th>
<th>Pathname</th>
</tr>
</thead>
<tbody>
<tr>
<td>http://</td>
<td><a href="http://www.cs.vu.nl">www.cs.vu.nl</a> : 80</td>
<td>/home/steen/mbox</td>
<td></td>
</tr>
<tr>
<td>(b)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Host name</th>
<th>Port</th>
<th>Pathname</th>
</tr>
</thead>
<tbody>
<tr>
<td>http://</td>
<td>130.37.24.11 : 80</td>
<td>/home/steen/mbox</td>
<td></td>
</tr>
<tr>
<td>(c)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Why Names Are Important?

- URLs identify content on the web

- URL typically identifies origin server by name → DNS

- Can do many tricks with DNS

- DNS load balancing for server farms

- DNS redirection for content delivery networks
The principal working of the Akamai CDN.
Total Redirection

- Any request for origin server is redirected to CDN
- CDN takes control of content provider’s DNS zone
- Benefit: All requests are automatically redirected
- Disadvantage: May send lots of traffic to CDN, hence expensive for the content provider
Selective Redirection

- Content provider marks which objects are to be served from CDN
  - Typically, larger objects like images are selected
- Refer to images as: `<img src="http://cdn.com/foo/bar/img.gif">
- When client wants to retrieve image, DNS request for cdn.com gets resolved by CDN and image is fetched from the selected content server

- Pro: Fine-grained control over what gets delivered
- Con: Have to (manually) mark content for CDN
Replication of Applications

- Previously only replication of static content
- Can also replicate applications
- Recall: Application = Server + Database
- Replication of applications = Replication of database
- Full or partial replication of database?
  - Amount of data? Updates? Query containment?
- Warehouse-scale computing (see next chapter)
Replication of Applications

Figure 12-21. Alternatives for caching and replication with Web applications.
Chapter Summary

- Web as a distributed system
- Basic web architecture
- Content delivery networks
- Replication of web applications