Internet Services

- Internet brings ubiquitous connection
  - Internet-based applications/services accessible to online users through Internet.
  - Examples?

- Trends:
  - Centralization of applications/services
  - Scalability requirements: many simultaneous user accesses; large amount of hosted data, ...

Search Engine as An Example: Step 1 - Crawling

- Crawling - get all these Web pages out there:
  - First retrieve some root pages;
  - Parse their content and follow hyperlinks to retrieve more pages;
  - Depth-first search or breadth-first search?

Performance Analysis for Crawling

- What are the resources involved?
  - CPU processing for TCP/HTTP protocol handling and the parsing of page content
  - writing to disk storage
  - network bandwidth to remote web sites
  - Assume average page size 10KB
    - raw processing power of a single CPU
      - 1000 pages/sec
    - I/O to a single disk
      - 100 seeks/sec ⇒ up to 100 pages/sec
    - network bandwidth from/to the Internet
      - T1 link (1.5Mbit/s) ⇒ 12 pages/sec
      - T3 link (45Mbit/s) ⇒ 360 pages/sec
Parallel Crawling

- Challenge
  - Avoid redundant crawling

Search Engine as An Example: Step 2 - Indexing

- Indexing
  - Crawled raw web pages are not easy to search.
  - We index them to formats that are easy to search.
- As part of indexing, we need to give each page an ID
  - Using a hash function.

<table>
<thead>
<tr>
<th>Computer:</th>
<th>Page #123</th>
<th>Page #357</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Networks:</td>
<td>Page #124</td>
<td>Page #468</td>
<td>...</td>
</tr>
</tbody>
</table>

- Fast intersection?

Search Engine as An Example: Step 3 - Online Search

- Scalability, reliability

Partitioning and Replication

Index servers

Page servers

Local-area network

Web server/Query handler

Firewall

Internet

Index servers

Page servers

Local-area network

Web server/Query handler

Firewall/Router

Internet
Load Balancing over Internet Servers

- Popular sites like Google or Facebook receive tens or hundreds of millions of hits per day.
- A large number of replicated servers are used at these sites.
- **Key question:** how to balance client requests over these servers?
- **Goals:**
  - Performance
  - Ease of deployment

Discussions on DNS Rotation

- **Advantages**
  - Require almost no change on the existing Internet architecture
- **Problems**
  - DNS Caching
  - Rigid load balancing policy
    - can't balance based on runtime load changes
    - slow or no adjustment in response to failures

Load Balancing on Internet Servers

Technique 1 - DNS Rotation

- IP address of Facebook.com
- Web servers for Facebook.com
- DNS server for Facebook.com

Technique 2 – Cooperative Offloading

- IP address of Facebook.com
- Firewall/Router
- Web servers for Facebook.com
Discussions on Cooperative Offloading

- Can be combined with the DNS rotation.
- Advantages:
  - More flexible policy is possible
  - Be more responsive to runtime workload and server failures (to a certain degree)
- Problems:
  - Need software changes on servers
  - Longer delay

Cooperative Offloading vs. TCP Handoff

- Software changes on the servers
- Delays
More About Load Balancing Router

Load balancing policies in LB routers (Goal: transparency, plug-and-play)
- Simple rotation
- Least number of active requests
- Shortest response time

How deep do we look into the network protocol stack?
- Network layer (IP)?
- Transport layer (TCP/UDP)?
- Application layer?

Summary

- Scalable Internet servers
  - partitioning
  - replication
- Load balancing for Internet servers
  - DNS rotation
  - Cooperative offloading (w. TCP handoff)
  - Load balancing router
- Changes required on the components:
  - DNS server??
  - Web server??
  - Client??
  - Router??