Scalable Internet Servers and Load Balancing

CS 257/457
Dept. of Computer Science, University of Rochester

Internet Services and Servers

- Internet Services
  - Services hosted by computer systems, accessible to online users through Internet.
- Services on the Internet
  - Online keyword search engine: Google.
  - Web email service: hotmail.
  - News service: CNN.
  - Other portal services: Yahoo!, AOL, MSN.
- Internet Servers
  - Computer systems that host Internet services.

Internet Services are at the Application Layer

- Normally on the end hosts, involving no routers
- Work on transport-layer protocols TCP/UDP

An Example: How does Google work? (Part I)

- First, we need to get all these Web pages out there - crawling.
- Then we need to reformat them to make them easy to search - indexing.
- As part of indexing, we need to give each page a ID.

Question #1: how to support multi-keyword search?
Question #2: how to assign IDs?
An Example: How does Google work? (Part II)

So what is the big deal?

- **Scalability**: How about searching over 2 billion Web pages (with an index size of several Terabytes)?
- **Throughput**: How about serving 150 million search queries per day?
- **Response time**: Come on!! I have been waiting here for two seconds. Where is the result?
- **Reliability**: With 1,000 servers and 4,000 disks in your machine room, something is gonna break every day!!

Technique 1: Partitioning

The index database is split into many partitions ⇒ better scalability

- **Computer**: Page #123, Page #357, ... ...
- **Networks**: Page #124, Page #468, ... ...
- **Food**: Page #124, Page #377, ... ...
- **Medicine**: Page #12, Page #468, ... ...

Technique 2: 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
Revisit: How does Google work?

Load Balancing over Internet Servers

- Popular sites like Google or CNN 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? (hints: using DNS)

Load Balancing on Internet Servers

Technique 1 - DNS Rotation

Discussions on DNS Rotation

- Problems
  - DNS Caching
  - Rigid load balancing policy
    - can’t balance based on 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 architecture

In class discussion: how to use DNS for distributing requests to wide-area replicated sites?
Load Balancing on Internet Servers

Technique 2 – 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 a lot more software
- Longer delay

Technique 3 – Load Balancing Switch

Discussions on Load Balancing Switch

- Different with cooperative offloading
  - We are messing around with TCP/IP kernel
  - better performance – no extra application-level processing

Any changes required on the parties involved:
- DNS server??
- Web server??
- client??
- switch??????
Load Balancing Policies in LB Switches

Your idea??

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

Summary

- Scalable Internet servers
  - partitioning
  - replication
- Load balancing on Internet servers
  - DNS rotation
  - cooperative offloading
  - LB switches
- **What is missing:** how to get data from a popular site to many clients?