Scalable Internet Servers and Load Balancing

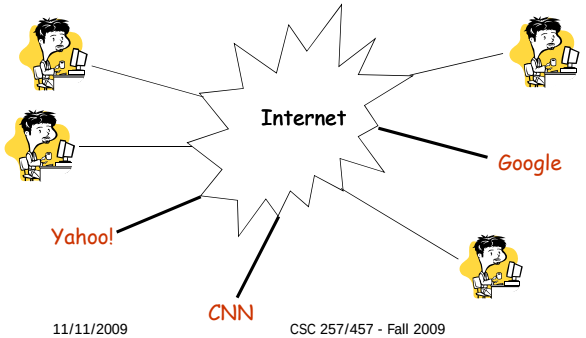
Kai Shen

Internet Online Applications

- Internet online applications
  - Applications accessible to online users through Internet.
- Examples
  - Online keyword search engine: Google.
  - Web email: Gmail.
  - News: CNN, NBC news.
  - Web directory: Yahoo!, MSN.
- Scalability requirements
  - Many simultaneous user accesses; large amount of hosted data, …
- Internet servers
  - Computer systems that host these online applications

Internet Servers are at the Application Layer

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



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? Remove duplicates.

Performance Analysis for Crawling

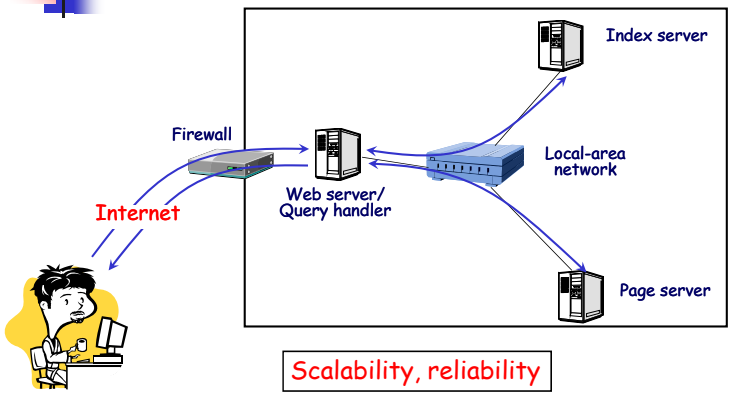
- What are the resources involved?
  - CPU processing for TCP/HTTP protocol handling and the parsing of page content
  - writing to disk storage
  - network bandwidth to remote web sites
- Assume average page size 10KB
  - raw processing power of a single CPU
    - 1000 requests/sec
  - I/O to a single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

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.

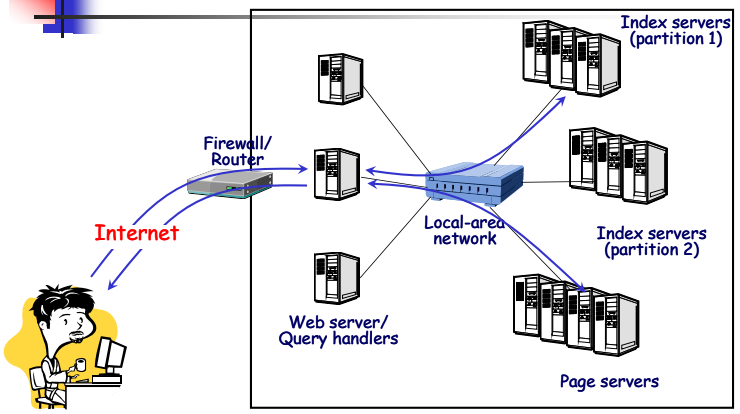
Computer:	Page #123	Page #357	… …
Networks:	Page #124	Page #468	… …

Search Engine as An Example: Step 3 – Online Search



Scalability, reliability

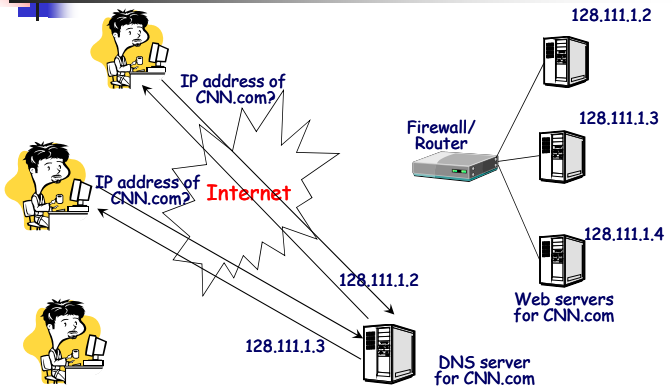
Partitioning and Replication



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?

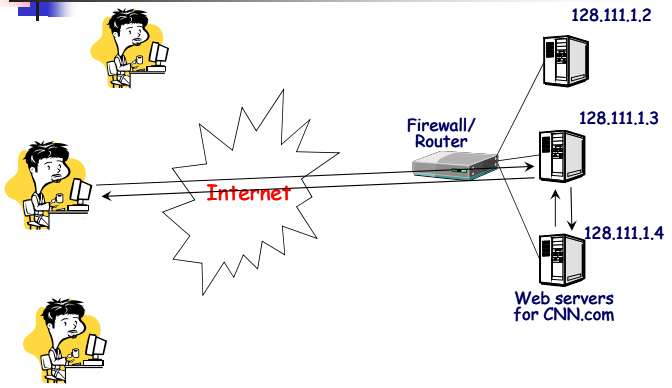
Load Balancing on Internet Servers Technique 1 - DNS Rotation



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

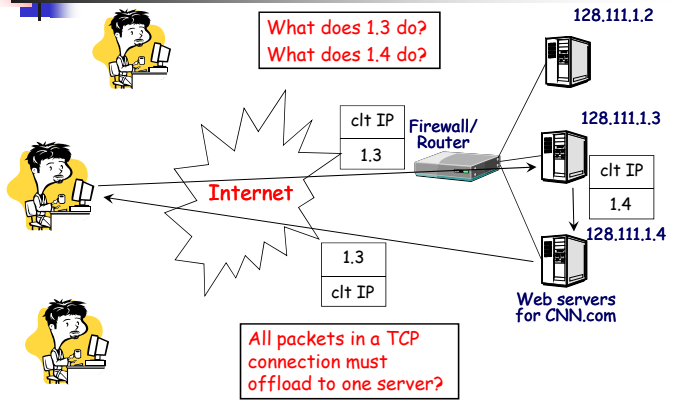
Load Balancing on Internet Servers Technique 2 – Cooperative Offloading



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

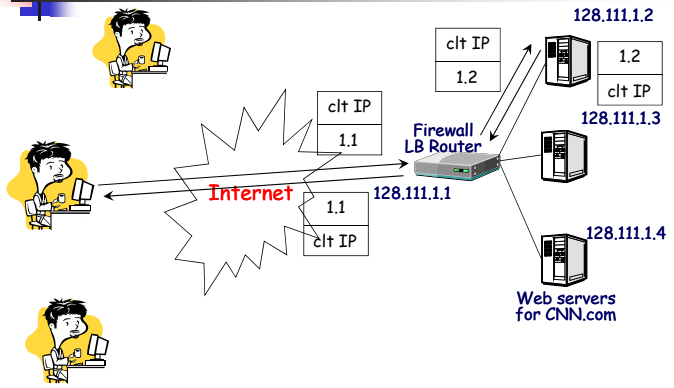
Cooperative Offloading with TCP Handoff [Pai et al. ASPLOS1998]



Cooperative Offloading vs. TCP Handoff

- Software changes on the servers
- Delays

Load Balancing on Internet Servers Technique 3 – Load Balancing Router



More About Load Balancing Router

How deep do we look into the network protocol stack?

- Network layer (IP)?
- Transport layer (TCP/UDP)?
- Application layer?

Load balancing policies in LB routers (Goal: transparency, plug-and-play)

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

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??