Notes for Harvard CS75 Web Development Lecture 9 Scalability by David Malan

Jan 16, 2016 | System Design | 3423 Hits 1 Comment

Lecture 9 video

Web Hosts

Features:

- Is IP address blocked in some countries/regions?
- SFTP vs. FTP. SFTP is secure and all the traffic is encrypted, which is important for user names and passwords
- Some hosting companies may offer you some unbelievable features, like unlimited storage spac, at a very low price. It's very likely that you and another hundreds users are sharing the same machine and contending for resources. This is because sometime people actually don't need that many resources.
- Virtual private server. May still share one machine with other users, but you have your own copy of the operating system. Run multiple virtual machine on a physical machine. Only you and the system administrators have access to your files
- If you want more privacy, then probably you have to operate your own servers

AWS EC2

How to scale

Vertical Scaling

Get more RAM, processors, disks,…, for one machine, but you will exhaust the financial resources/state of arts technology.

Horizontal Scaling

Plural number of machines, use multiple servers to build the topology.

Load Balancer

Need to distriubte inbound HTTP requests

Return the public IP address of the load balancer, and let the load balancer determin how to actually route data to the backend server (private address).

Implementation

- Dedicated servers for gifs, jpegs, images, videos, etc for different host HTTP header
- Round robin. Or, the load balancer can be a DNS setup which returns the IP address of server 1 when the first time someone asks for a url, then return the IP address of server 2 when the second time someone asks for the same url, then server 3, server 4,…, eventually wrapping up. Downside: one server may get a really computational heavy user;
- Based on the load on a server
- Have a server specifically for storing sessions. But what if that machine breaks down. Lacks redundacy. Can add RAID (redundant array of independent disks), striping and redundancy

When we type the url in the browser and hit enter, the OS will send a packet to DNS server which will translate host names to IPs and vice versa. If we click a link on a website, there is a cache to store the IP address so the OS doesn't have to send the same DNS request again. Both OS and browser have a cache. Time to Live (TTL) values associated with an answer from a DNS server, 5 minutes, 1 hour, or 1 day. Global load balancing…

If the backend is PHP based and the session in PHP is broken. And if you were on Server 1, then by chance you are sent to Server 2 by a round-robin, you might have to log in again. Or think about shopping cart.

Stick sessions (when you visit a website multiple times your session is somehow preserve even if there are multiple backend servers)

Cookies:
Can store the address of the server so the next the user visit the website, he goes to the same back-end

server. Downside: the private IP of the back-end server may change; the private IP is now visible to the whole world

==>

Store a random number and let the load balancer remember which number belongs to which server

- Software
  - ELB
  - HAProxy
  - LVS
- Hardware
  - Barracuda
  - Cisco
  - Citrix
  - F5

PHP Acceleration
php.exe compiles php everytime but throws away the result. Some software can keep the result. Like .py vs. .pyc.

Caching

- .html vs. MySQL database/XML (avoid regenerating) more performance vs. more space. But requires a lot of work when want to update/redesign the page
- MySQL query cache: query_cache_type: 1
- memchached: store whatever you want in RAM (garbage collection: expire objects based on when they are put in)

Replication:

Master-Slave

Master: the main database that you write/read data to/from.
Slave: anytime a query is executed on the
master that same query is copied down to one or more slaves and they do the exact same thing

Advantages:

- If the master is down, promote one of the slaves and do some configuration. (redundacy)

- If there are a lot queries, you could just load balance across database servers
- For read heavy websites, any select can go to all four databases, while any insert/update/delete has to go to server master

Mastter-Master

you could write to either server one or two and if you happen to write to server1 that query gets replicated on server2 and vice versa so now you could keep it simple

Load balancing + Replication

active + active pair of load balancers
active + passive pair of load balancers, passive promote itself when receives no more packets from the active one.
and send packets to each other

Partitioning
A-M cluster and O-Z cluster

High Availability
One load balancer, two master replicating each other

Summary

```
Internet ==(TCP 443 80)==>
two load balancers ==(TCP 80)==>
web servers ==x==>
two load balancers ==x (TCP 3306)==>
two <==X==>  master databases


Fire wall on switch ports
```

System Design Web

Share

3D Array Size

326.Power of Three