

Computer Systems & Network Administration

Lecture 07. Web & LNMP stack

Outline

- Web Hosting
- Proxy
- Web Server
- SQL
- Programming Language
- Case Study

Web Hosting

WWW

- World Wide Web
 - Sir Timothy Berners-Lee, CERN, 1989
- Based on 3 components
 - HTML
 - HTTP
 - URL

HTML

- Hyper-Text Markup Language
 - Describe the structure of **text-based** information in a document
 - HTML 2.0 defined by [RFC 1866, IETF](#)
- HTML specification have been maintained by [World Wide Web Consortium \(W3C\)](#)

HTML - HTML5

- Published in **Oct.2014** by W3C
- Many new syntactic features are included
 - **article, aside, footer, header, nav, section...**
- Include and handle **multimedia** and **graphical** content
- **video, canvas, audio**

§ 1.9. A quick introduction to HTML

This section is non-normative.

A basic HTML document looks like this:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Sample page</title>
  </head>
  <body>
    <h1>Sample page</h1>
    <p>This is a <a href="demo.html">simple</a> sample.</p>
    <!-- this is a comment -->
  </body>
</html>
```

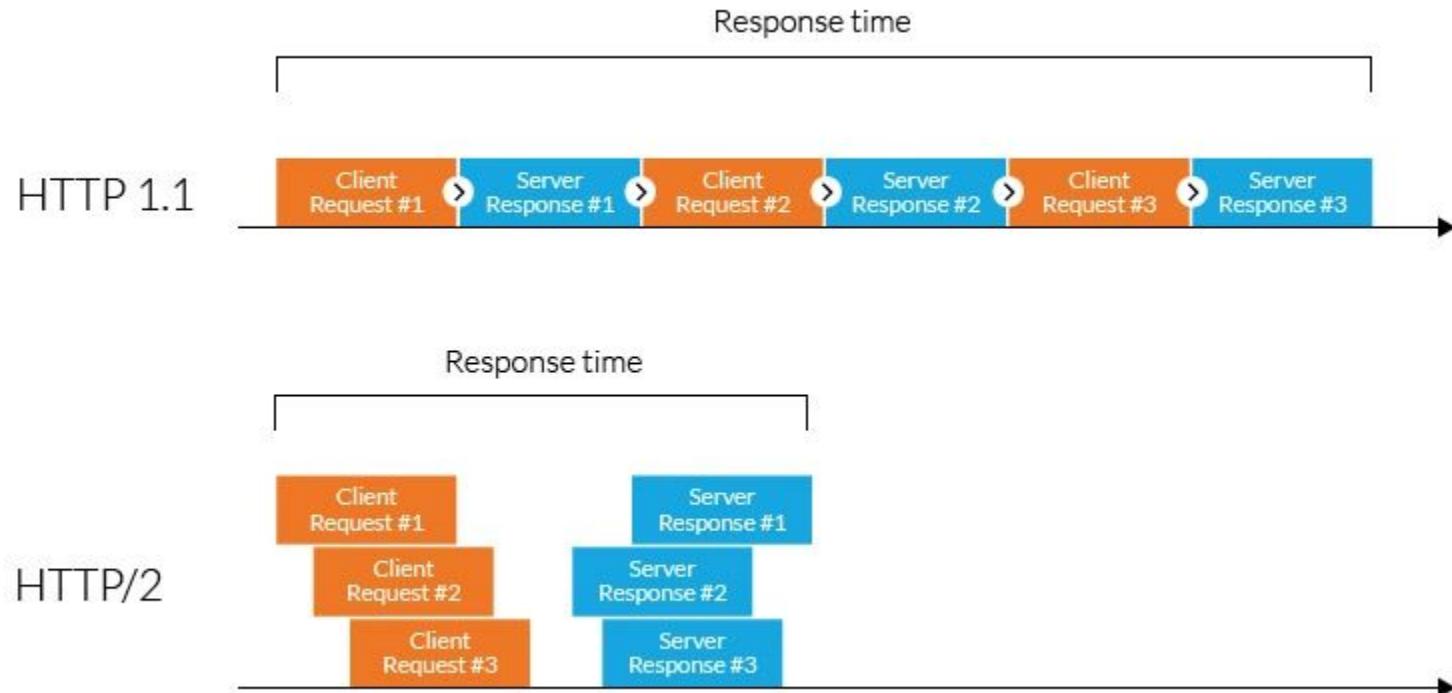
HTTP

- Hyper-Text Transfer Protocol
 - TCP-based protocol
 - Stateless
 - Communication method between client and server
All browsers have to follow this standard
 - Originally designed to transmit HTML content
Now is being used to transmit lots kind of content
 - Pictures, Files, Videos...etc
 - Mobile App API
 - HTTPS: Secure HTTP
 - We will cover security in future topic

HTTP - HTTP/2

- Derived from project SPDY, developed by Google
- RFC 7540, IETF
- Decrease latency by
 - HTTP Header data compression
 - HTTP/2 Server Push
 - Pipelining of requests
 - Fixing the head-of-line blocking problem in HTTP 1.x
 - Multiplexing multiple requests over a single TCP connection
 - Binary protocol

HTTP - HTTP 1.x vs. HTTP 2



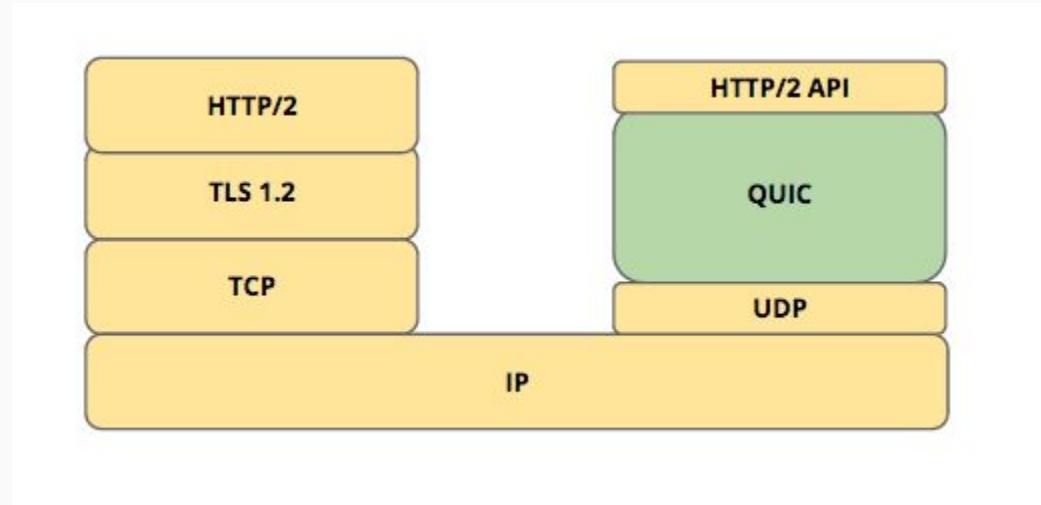
HTTP - HTTP3 / QUIC

- QUIC
 - Quick UDP Internet Connections
 - General-purpose transport layer network protocol
 - Designed by Jim Roskind at Google
- Perform better than HTTP2
 - Greatly reduce overhead during connection setup
 - Use UDP instead of TCP
 - If there's data loss, only that connection will be affected, other connection can still move on

HTTP - HTTP3 / QUIC

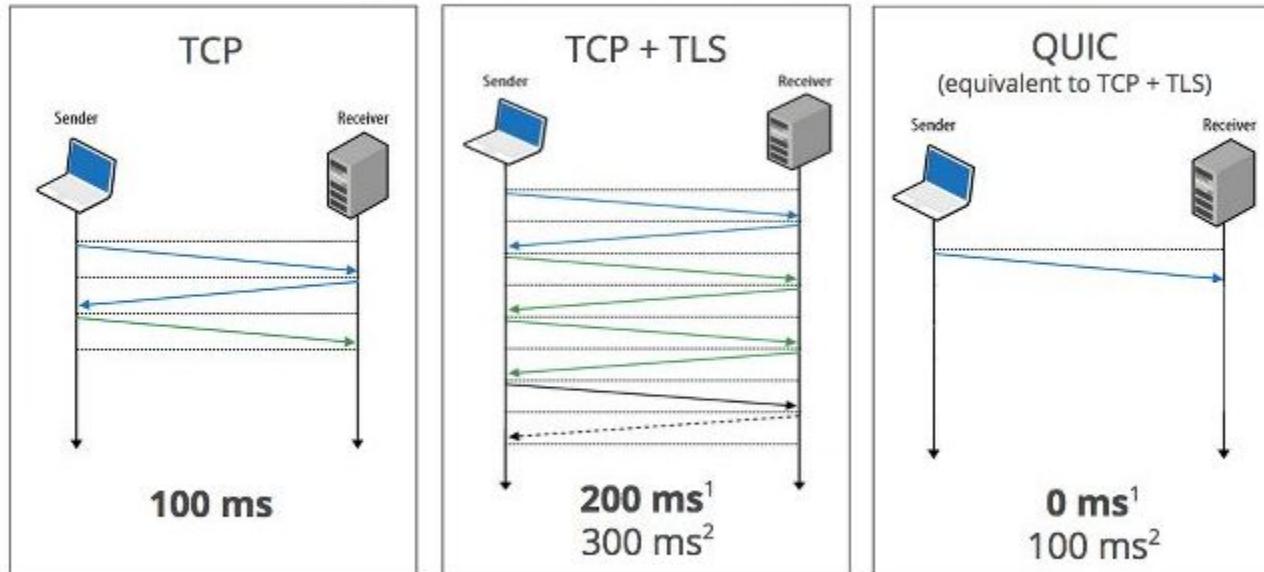
- Being used by more than half of all connections from Chrome to Google's servers
 - Not enabled by default on Microsoft Edge, Firefox and Safari
- Still a [draft](#) in IETF, a special workgroup is created for QUIC
- HTTP3 is based on QUIC

HTTP - HTTP3 / QUIC



<https://peering.google.com/#/learn-more/quic>

Zero RTT Connection Establishment



URL

- Uniform Resource Locator
 - [RFC 1738, IETF](#)
 - Describe how to access an object shared on the Internet
 - Format
 - <protocol>://[[username[:password]@]hostname[:port]][/directory][[/filename]
 - Example
 - `https://www.google.com/`
 - `ftp://ftp:ftp@ftp.speed.hinet.net:21/test_400m.zip`
 - `wss://ws.ptt.cc/bbs`

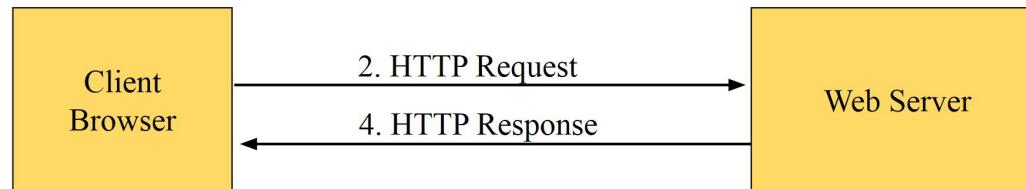
URL - URL Protocols

Protocol	Purpose	Example
http	Access a file using HTTP	http://clients1.google.com/generate_204
https	Access a file using HTTPS	https://www.google.com
ftp	Access a file using FTP	ftp://ftp:ftp@ftp.speed.hinet.net:21/test_400m.zip
sftp	Access a file using SFTP	sftp://user:password@someserver/foo
file	Access a local file	file:///home/user/bar
mailto	Send mail	mailto:nasa@imslab.org
wss	Access remote content using WebSocket	wss://ws.ptt.cc/bbs
ssh	Access remote machines using SSH	ssh://user:password@someserver

Web - Client-Server Architecture

- Client-server architecture
 - Web Server: Return HTTP request
 - Web Client: Request certain page using URL

1. Send the request to server which URL point to
3. Respond the HTML resource pointed by URL



5. Show the data which HTML resource describes.

Web - Client-Server Architecture

```
% telnet www.google.com 80 [487/1092]
Trying 172.217.160.68...
Connected to www.google.com.
Escape character is '^]'.
GET /
HTTP/1.0 200 OK
Date: Mon, 29 Mar 2021 02:27:28 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Set-Cookie: 1P_JAR=2021-03-29-02; expires=Wed, 28-Apr-2021 02:27:28 GMT; path=/; domain=.google.com; Secure
Set-Cookie: NID=212=iU_i8R9IK1WFL_9YUENQgYu8gkCeWh0iFBL6WUMWkQnt3g0kNkAUplr8isBVINSmNHSoLiAdj7PEGe9imBdDZz6Ax4j
R8nUrxF1fSiI0jKapAem34pDbG0UYeHbwLS6KL8xuoWThXa0UgCUiUazNIPlQJtT0sRAW8AVEYVx2J0; expires=Tue, 28-Sep-2021 02:2
7:28 GMT; path=/; domain=.google.com; HttpOnly
Accept-Ranges: none
Vary: Accept-Encoding

<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="zh-TW"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="qJH8o/vpdCfSeW7L1ZKag==">(function() {window.google={kEI:'EDthYPSuCIq0mAW6_IG4Cw', kEXPI:'0,1302440,59,56910,954,5105,206,4804,926,1390,383,246,5,1354,1301,39
50,1122515,1233,1196554,465,31,328984,51224,16114,28684,9188,8384,4858,1362,9290,3026,4742,12841,4998,6242,6986
,2054,920,873,4192,6430,7432,7095,4517,2778,919,2277,8,2796,887,706,1279,2214,530,147,1103,840,518,1465,4,52,42
58,108,1341,2,2063,606,2025,1775,520,4269,328,1284,2004,6785,3227,2845,9,5597,6755,5096,7876,4929,108,3407,908,
2,940,2615,2397,7470,3275,3,576,1835,4625,148,5990,7985,4,1528,2301,1239,1145,6449,2893,459,1555,3791,276,12555
,1753,2658,3906,1,336,518,912,564,299,165,656,30,3854,4275,3286,2213,1593,712,638,3584,3496,9206,587,11,731,665
0 > 0* > [tmux] > ↑ 1h 29m 20s < 0.1 0.1 0.1 < 2021-03-29 < 10:28 < 🔒 Tsundere-XPS
```

Web - Client-Server Architecture

- Client
 - Send requests to servers
 - Action “path or URL” Protocol
 - Actions: GET / POST / HEAD
 - e.g.: GET /index.html
 - HTTP1.1
 - Headers
 - Header_name: value
 - e.g.: Host: www.google.com
 - (blank line)
 - Data ...
- Server
 - Respond to the clients
 - Status
 - 200
 - 403
 - 404
 - 502
 - [HTTP 418](#)
 - [HTTP Status Codes](#)
 - Header
 - Same as client
 - e.g.: Content-Type: text/html
 - (blank line)
 - Data ...

Web - HTTP Protocol - GET vs. POST

- GET vs. POST
 - GET
 - Parameter in URL
 - GET /get.php?a=1&b=2 HTTP/1.1
 - No data content
 - Example
 - URL: <https://www.google.com>
 - Using Form: <form method="GET" action="get.php"> ... </form>
 - POST
 - Parameters in Data Content
 - POST /post.php HTTP/1.1
 - Corresponding in HTML files
 - Using Form: <form method="POST" action="post.php">...</form>

Web - HTTP Protocol - GET vs. POST

	GET	POST
BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser history	Parameters are not saved in browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

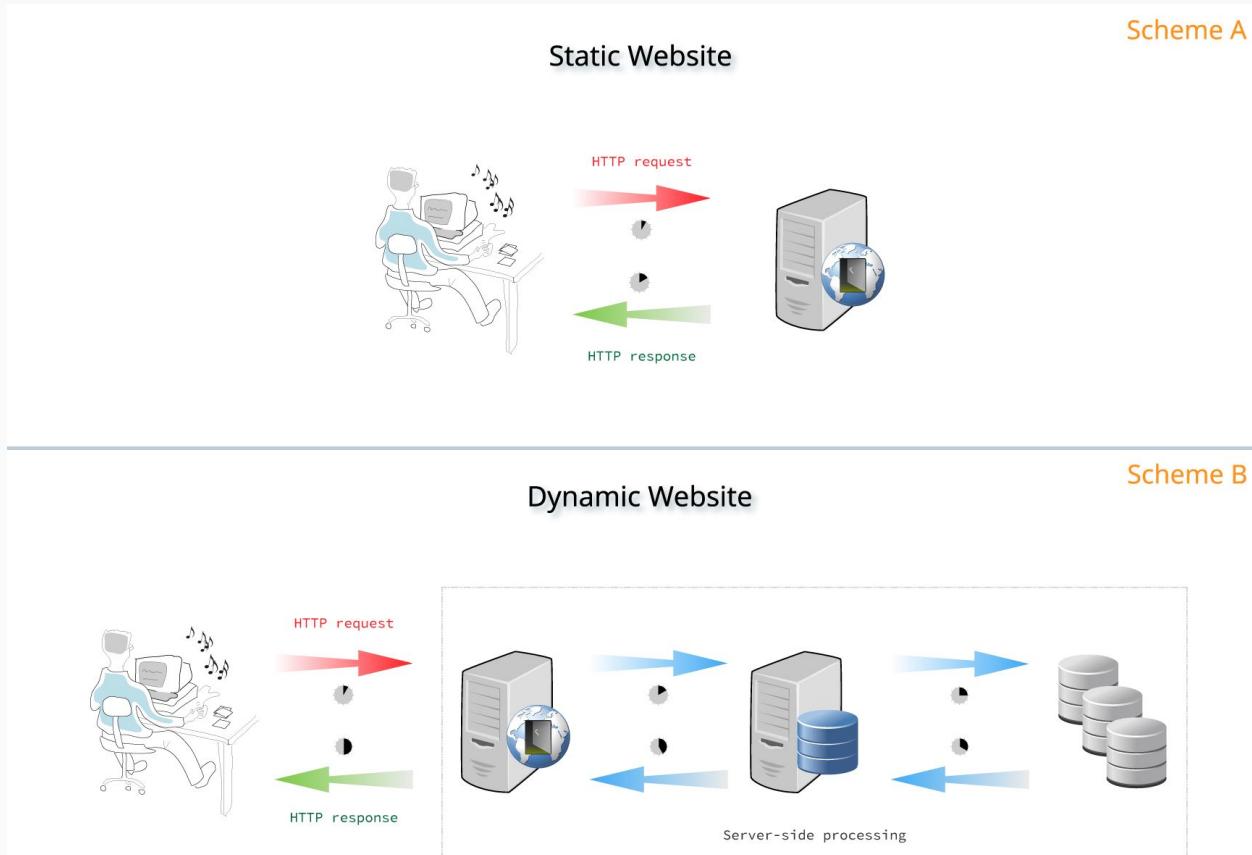
Web - HTTP Protocol - HTTP Headers

- HTTP Headers
 - Content Information
 - Cache Control
 - Authentication
 - URL Redirection
 - Transmitting Cookies
 - Knowing where client come from
 - Knowing what software client uses
 - ...

Web - Static vs. Dynamic Content

- **Static Content**
 - Media / JavaScript Library / CSS Framework / Favicon / Personal Homepage...etc
- **Dynamic Content**
 - Client Script Language
 - JavaScript / ECMAScript ...etc
 - Client Interactive Technology
 - AJAX / PWA ...etc
 - Server Side Technologies
 - CGI
 - Perl / ASP.NET / JSP / PHP / NodeJS / Ruby on Rails / Golang...etc

Web - Static vs. Dynamic Content



<https://about.gitlab.com/blog/2016/06/03/ssg-overview-gitlab-pages-part-1-dynamic-x-static/>

Web - Virtual Hosting

- Providing more than one domain-name in one web server instance
- IP-based virtual hosting vs. Hostname-based virtual hosting
- IP-based
 - Several IP in one instance, use IP to distinguish between requests
- Hostname-based
 - One IP in one instance, use hostname to distinguish between requests

Web - Virtual Hosting(with nginx)

```
server {  
    listen 80;  
    server_name 1.1.1.1;  
    root /www/one;  
}  
  
server {  
    listen 80;  
    server_name 1.1.1.2;  
    root /www/two;  
}
```

```
server {  
    listen 80;  
    server_name one.example.com;  
    root /www/one;  
}  
  
server {  
    listen 80;  
    server_name two.example.com;  
    root /www/two;  
}
```

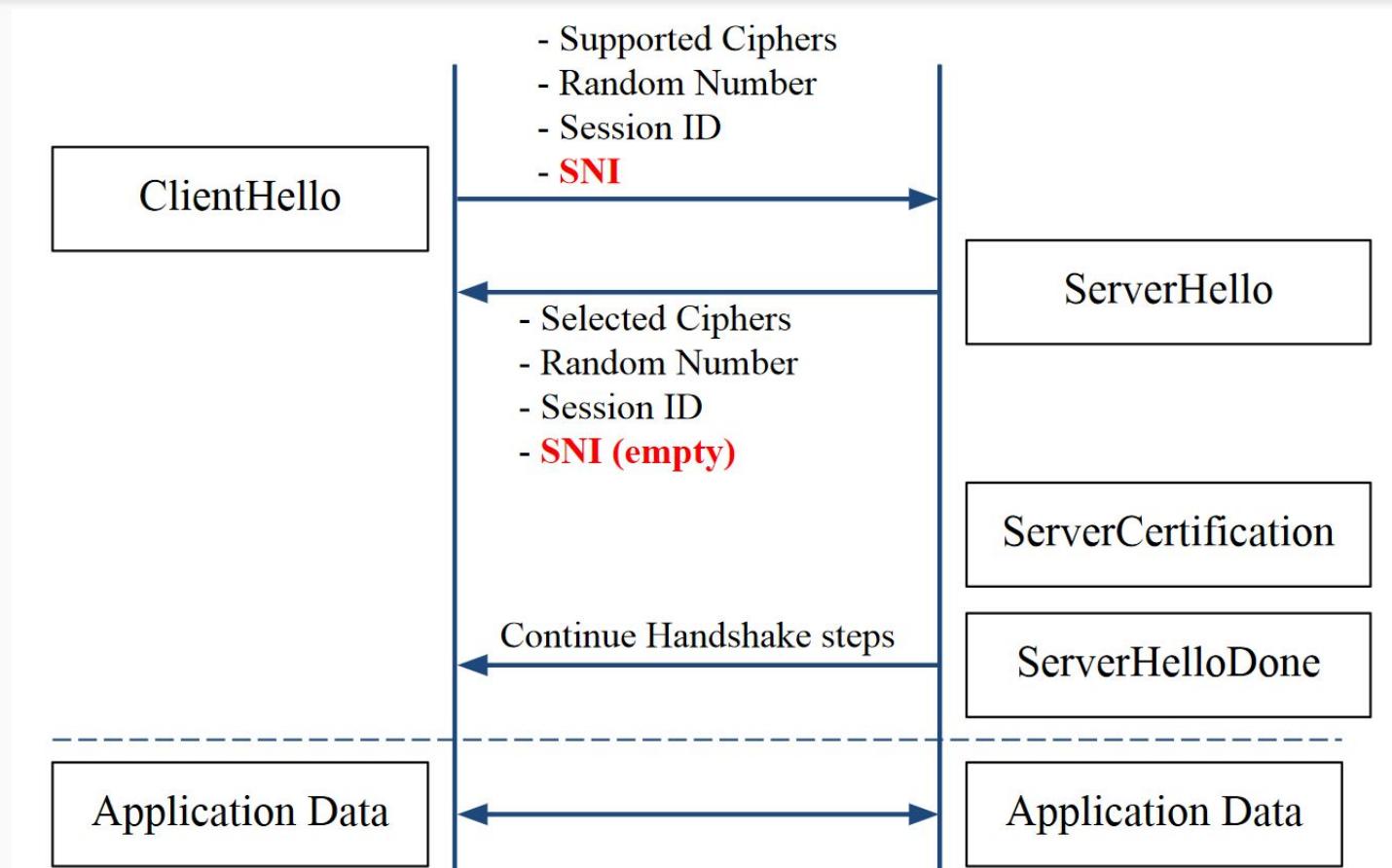
Web - Virtual Hosting

- It's extremely inefficient to use one IP to host one web service
 - Hostname-based virtual hosting is very useful
- You don't actually need that much IPv4 addresses

Web - Server Name Indication (SNI)

- Extension to the Transport Layer Security (TLS) by which a client indicates which hostname it is attempting to connect to at the start of the handshake process
 - Use single IP address for multiple SSL sites
- No encryption of hostname
 - Possible information leak

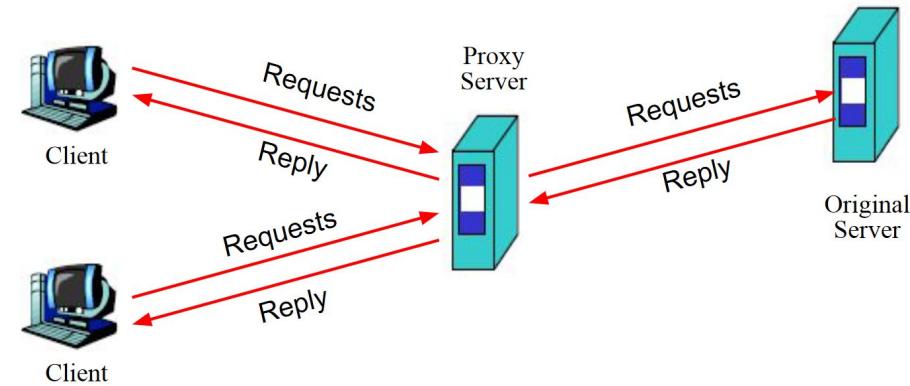
Web - Server Name Indication (SNI)



Proxy

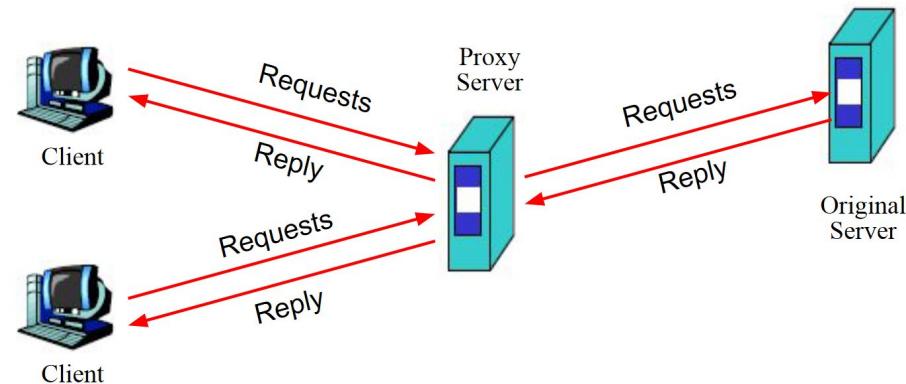
Proxy

- Services the requests of its clients by
 - Making requests to other servers
 - Caching some results for further same requests
- Goals
 - Performance
 - Stability
 - Central Control
 - ...etc



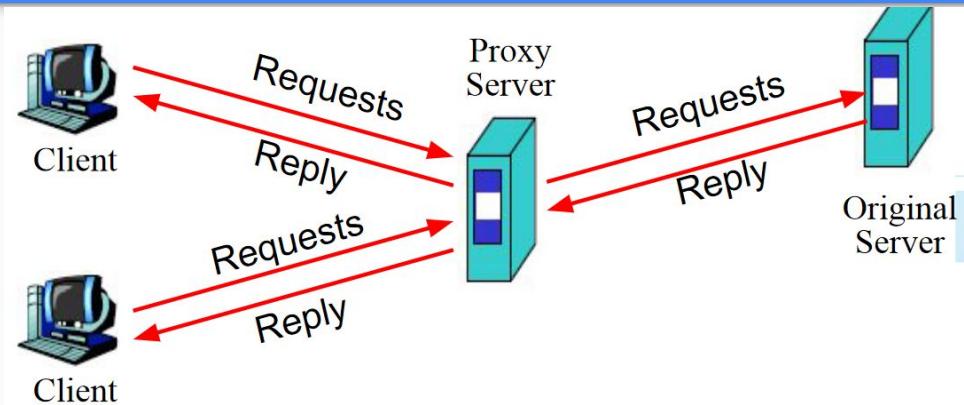
Proxy

- Types
 - Forward Proxy
 - Reverse Proxy
- Targets
 - Web pages / FTP files
 - TCP/IP connections
 - ...etc



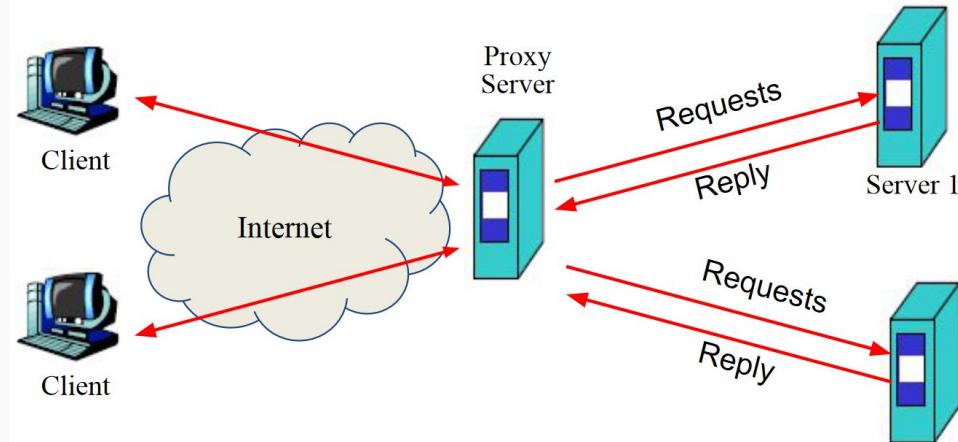
Proxy - Forward Proxy

- Proxy the outgoing requests
 - Bandwidth saving
 - Performance
 - Central Control
- When requests are
 - in cache: return cached objects
 - Otherwise, proxy server requests objects from the originate server, then cache the response, and send it back to the client



Proxy - Reverse Proxy

- Proxy the incoming request for
 - Reducing server load
 - Load Balance
 - Fault Tolerance
- Reverse Proxy act as **originate server**, accepts **incoming requests**, hands the request to target server, get the reply, then send it back to the client



Proxy - Reverse Proxy

- Modern Reverse Proxy Usage
 - Application Layer Load Balancing (L7)
 - Application Layer Service Health Check
 - Global Server Load Balancing
 - SSL Off-load
 - Data Acceleration
 - gzip, brotli
 - Caching
 - Programmable Server Load Balancing
 - [F5 Network](#)
 - [A10 Networks](#)

Web Server

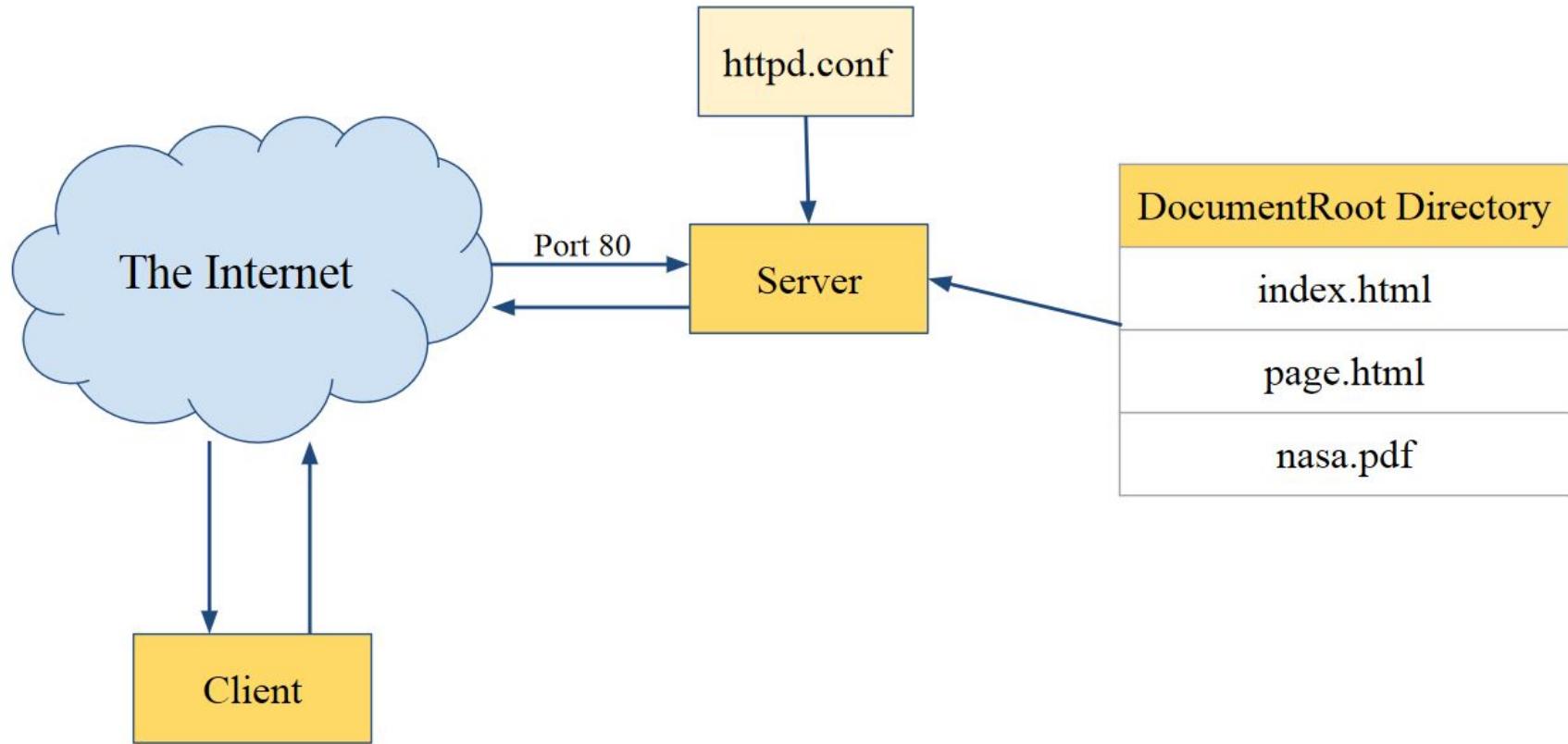
Web Server

- Handling incoming request and return requested data to client
- Software
 - Apache HTTP Server
 - **Nginx**
 - Caddy
 - lighttpd
 - Microsoft IIS (for Microsoft Windows)
 - Apache Tomcat (for Java application)

Apache HTTP Server

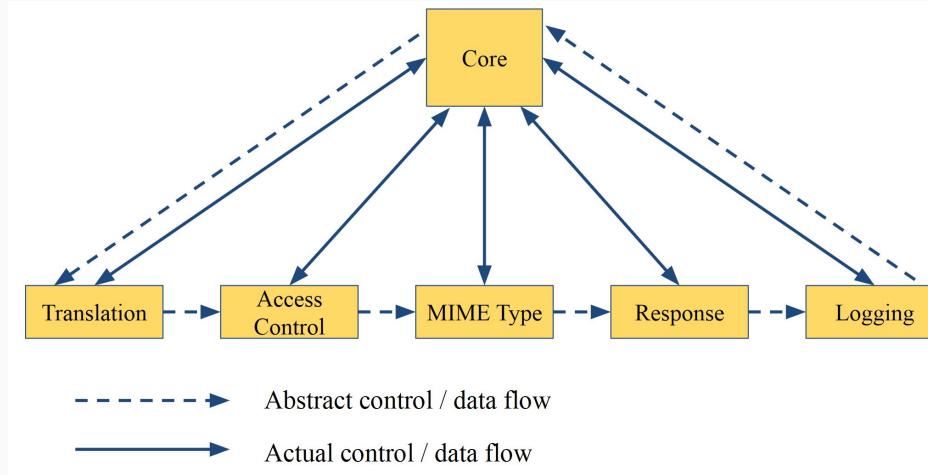
- Apache Software Foundation
- Apache HTTP Server Project
- Web httpd server that supports
 - HTTP/2
 - Modular design
 - Can be customized by writing modules using Apache module API
 - Freely available across many platforms
- Two main parts
 - Core
 - Implements basic functions and provide the interface for Apache modules
 - Modules
 - Extends / Overrides the function of Core

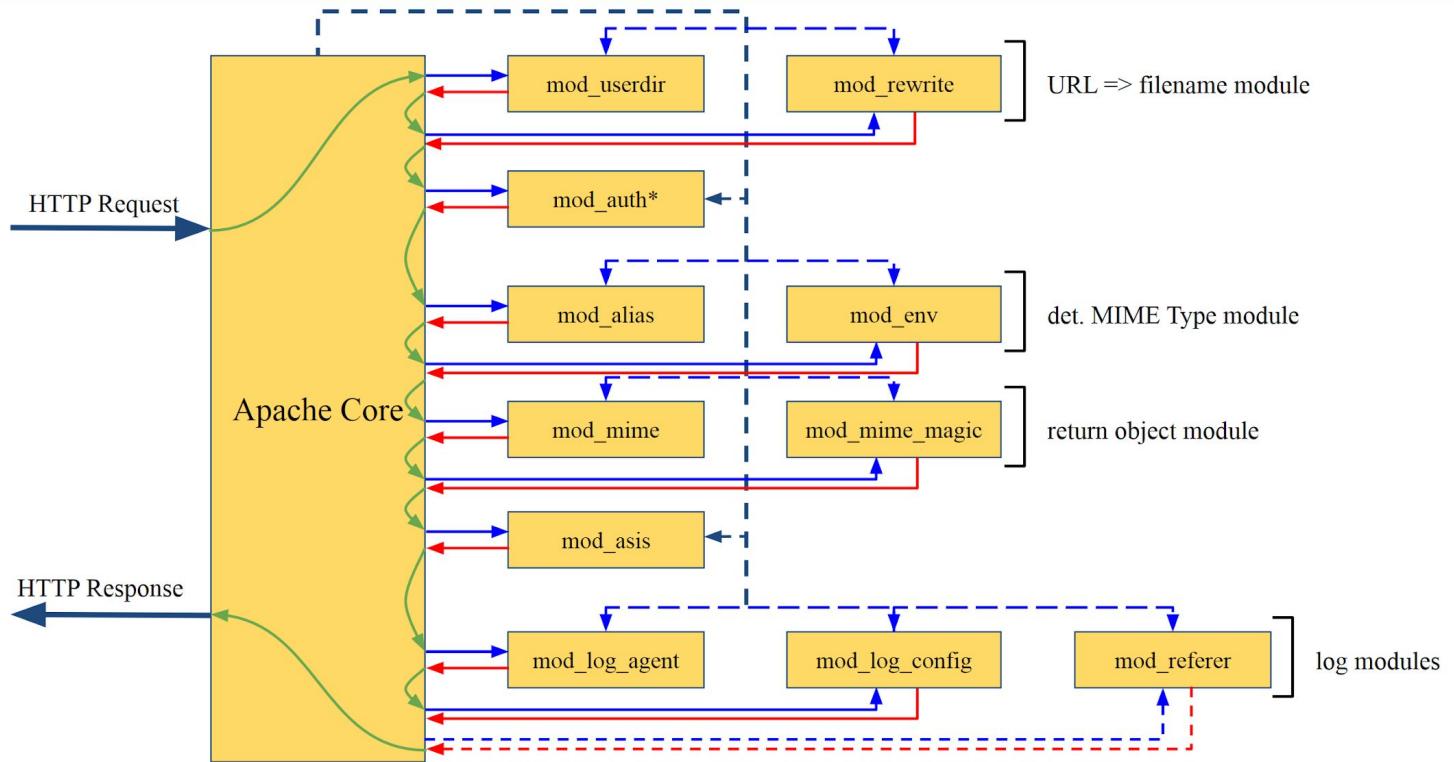
How Apache HTTP Server works - request & response



How Apache HTTP Server works - each request-response

- Apache breaks client request into several steps which are implemented as modules





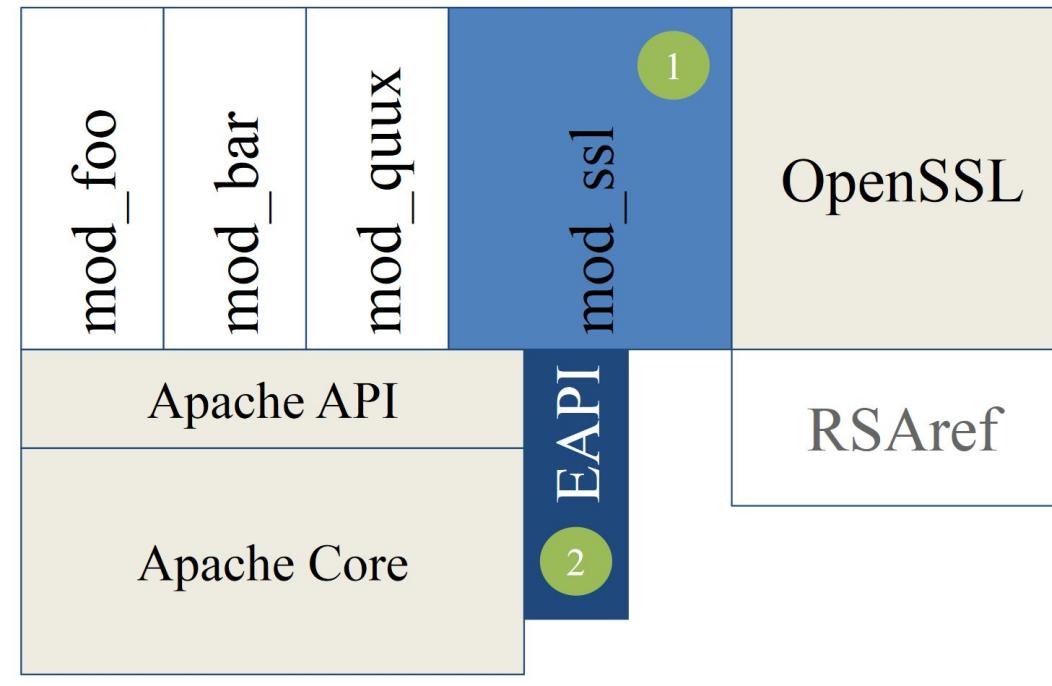
→ Calls/Uses
 ← Returns control / data
 → Initialize call

Execution Flow(for a Request)

Component / Module

HTTP Req / Resp

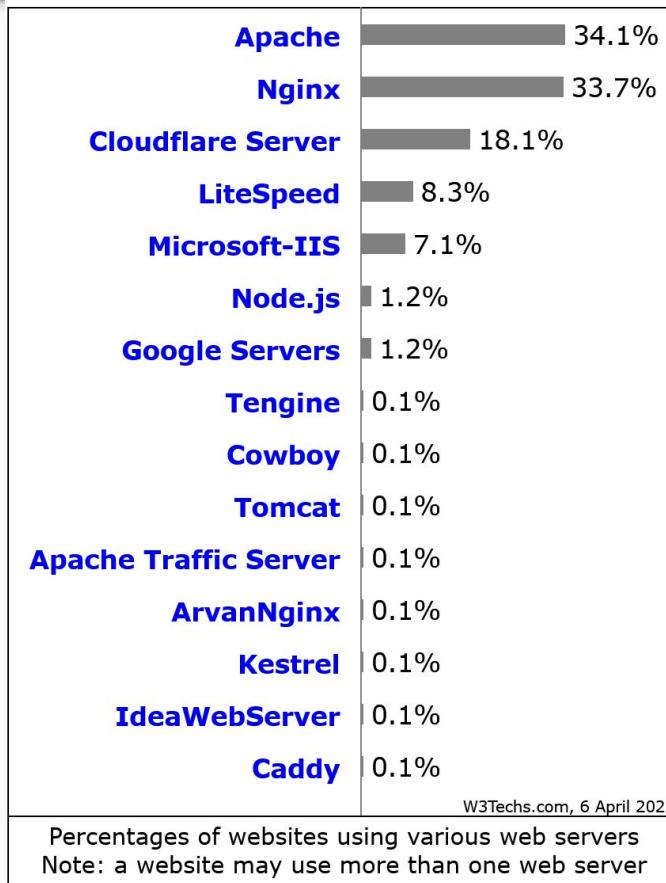
Apache with mod_ssl



nginx

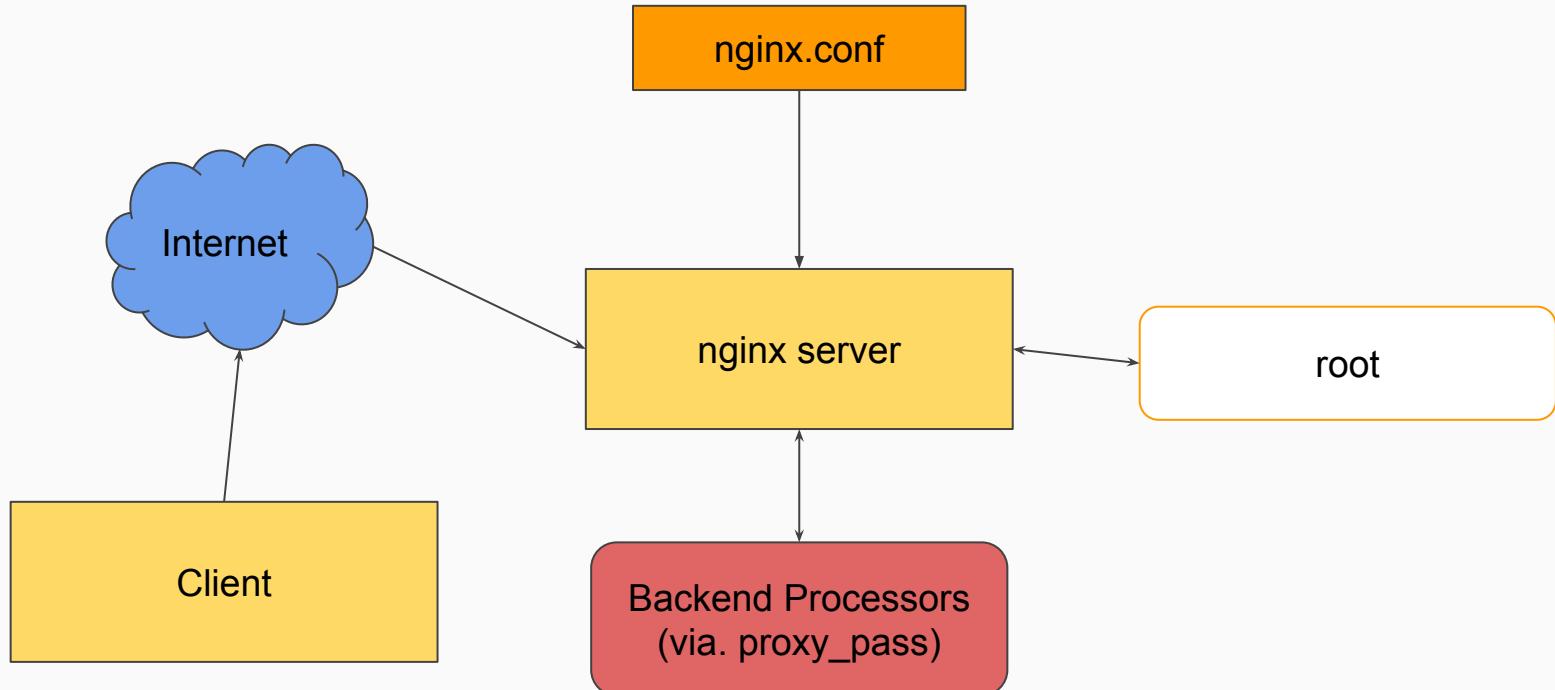
- Developed by [Igor Sysoev](#) in 2002, has commercial solution [Nginx Plus](#)
Nginx Inc. was acquired by [F5 Networks](#)
- Spotlight
 - Ability to handle 10k simultaneous connections with low memory footprint
 - Reverse proxy with caching
 - Load balancing with health check function
 - TLS/SSL support with SNI and OCSP stapling
 - gRPC support
 - WebSocket support
 - HTTP/2 protocol support

Usage statistics of web servers

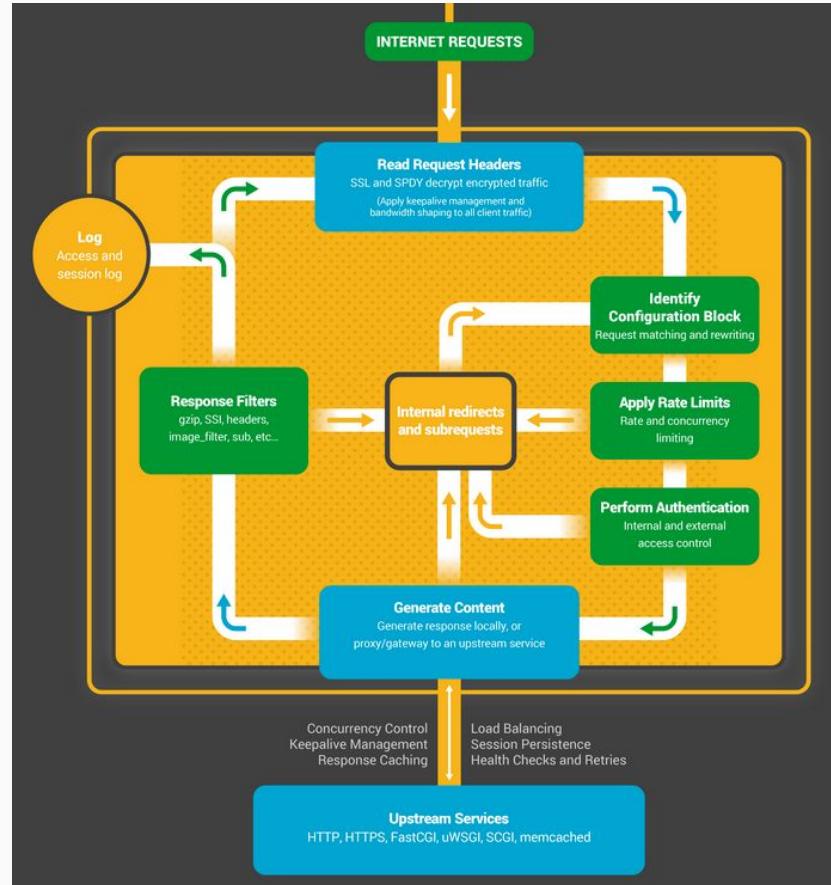


https://w3techs.com/technologies/overview/web_server

How nginx works - request & response



Inside NGINX: How We Designed for Performance & Scale



Full Example Configuration | NGINX

[PRODUCTS](#)[SOLUTIONS](#)[RESOURCES](#)[SUPPORT](#)[PRICING](#)[BLOG](#)[FREE TRIAL](#)[CONTACT US](#)

Full Example Configuration

nginx.conf

```
user          www www;    ## Default: nobody
worker_processes 5;    ## Default: 1
error_log    logs/error.log;
pid          logs/nginx.pid;
worker_rlimit_nofile 8192;

events {
    worker_connections 4096;    ## Default: 1024
}

http {
    include      conf/mime.types;
    include      /etc/nginx/proxy.conf;
    include      /etc/nginx/fastcgi.conf;
    index       index.html index.htm index.php;

    default_type application/octet-stream;
    log_format  main '$remote_addr - $remote_user [$time_local]   $status '
                  '"$request" $body_bytes_sent "$http_referer" '
                  '"$http_user_agent" "$http_x_forwarded_for"';
    access_log  logs/access.log main;
    sendfile    on;
    tcp_nopush  on;
```

 Search Wiki

Wiki home

Getting Started

- Dynamic SSI Example
- Full Example Configuration
 - nginx.conf
 - proxy.conf
 - fastcgi.conf
 - mime.types
- Another Full Example
- Simple Load Balancing
- Reverse Proxy with Caching
- SSL-Offloader
- Log Rotation
- Server Block Examples
- PHP FastCGI Example
- PHP-FastCGI on Windows

nginx - installation

```
F74076310@F74076310:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream libtiff5 libwebp6 libxpm4 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream libtiff5 libwebp6 libxpm4 nginx nginx-common nginx-core
0 upgraded, 17 newly installed, 0 to remove and 4 not upgraded.
Need to get 2431 kB of archives.
After this operation, 7891 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

nginx - Check server status

```
F74076310@F74076310:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
F74076310@F74076310:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2021-04-06 05:59:56 UTC; 20s ago
    Docs: man:nginx(8)
 Main PID: 28926 (nginx)
   Tasks: 2 (limit: 1136)
  Memory: 4.9M
 CGroup: /system.slice/nginx.service
         └─28926 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             ├─28927 nginx: worker process

Apr 06 05:59:56 F74076310 systemd[1]: Starting A high performance web server and a reverse proxy server...
Apr 06 05:59:56 F74076310 systemd[1]: Started A high performance web server and a reverse proxy server.
```

nginx - Get default server page

```
F74076310@F74076310:~$ curl localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

nginx configuration

- Usually we divide nginx configuration into two parts
 - Global settings
 - Website settings
- **Global settings**
 - Worker amount
 - Basic settings
 - Virtual Host settings
- **Website settings**
 - Website information
 - SSL / TLS
 - Logging
 - Reverse Proxy
 - ...etc

nginx configuration (cont.)

- Paths
 - /etc/nginx
 - Configuration folder
 - /var/log/nginx
 - Log folder
 - /var/lib/nginx
 - Temporary folder (For proxy files)
 - /usr/lib/nginx/modules
 - nginx modules folder

nginx configuration (cont.)

- Global settings
 - /etc/nginx/nginx.conf
- Website settings
 - /etc/nginx/sites-enabled/*
 - /etc/nginx/conf.d/*.conf

nginx - nginx.conf

```
File: /etc/nginx/nginx.conf

1 user www-data;
2 worker_processes auto;
3 pid /run/nginx.pid;
4 include /etc/nginx/modules-enabled/*.conf;
5
6 events {
7     worker_connections 768;
8     # multi_accept on;
9 }
10
11 http {
12
13     ## 
14     # Basic Settings
15     ##
16
17     sendfile on;
18     tcp_nopush on;
19     tcp_nodelay on;
20     keepalive_timeout 65;
21     types_hash_max_size 2048;
22     # server_tokens off;
23
24     # server_names_hash_bucket_size 64;
25     # server_name_in_redirect off;
26
27     include /etc/nginx/mime.types;
28     default_type application/octet-stream;
29
30     ##
31     # SSL Settings
32     ##
33
34     ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
35     ssl_prefer_server_ciphers on;
36
37     ##
38     # Logging Settings
39     ##
40
41     access_log /var/log/nginx/access.log;
42     error_log /var/log/nginx/error.log;
43
44     ##
```

nginx - nginx.conf (cont.)

```
43
44    ## 
45    # Gzip Settings
46    ##
47
48    gzip on;
49
50    # gzip_vary on;
51    # gzip_proxied any;
52    # gzip_comp_level 6;
53    # gzip_buffers 16 8k;
54    # gzip_http_version 1.1;
55    # gzip_types text/plain text/css application/json application/javascript text/xml application/xml+rss text/javascript;
56
57    ##
58    # Virtual Host Configs
59    ##
60
61    include /etc/nginx/conf.d/*.conf;
62    include /etc/nginx/sites-enabled/*;
63
64
65
66 #mail {
67 #   # See sample authentication script at:
68 #   # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
69 #
70 #   # auth_http localhost/auth.php;
71 #   # pop3_capabilities "TOP" "USER";
72 #   # imap_capabilities "IMAP4rev1" "UIDPLUS";
73 #
74 #   server {
75 #       listen      localhost:110;
76 #       protocol   pop3;
77 #       proxy      on;
78 #   }
79 #
80 #   server {
81 #       listen      localhost:143;
82 #       protocol   imap;
83 #       proxy      on;
84 #   }
85 #}
```

nginx - /etc/nginx/sites-enabled/default

```
File: /etc/nginx/sites-enabled/default

1  ##
2  # You should look at the following URL's in order to grasp a solid understanding
3  # of Nginx configuration files in order to fully unleash the power of Nginx.
4  # https://www.nginx.com/resources/wiki/start/
5  # https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
6  # https://wiki.debian.org/Nginx/DirectoryStructure
7  #
8  # In most cases, administrators will remove this file from sites-enabled/ and
9  # leave it as reference inside of sites-available where it will continue to be
10 # updated by the nginx packaging team.
11 #
12 # This file will automatically load configuration files provided by other
13 # applications, such as Drupal or Wordpress. These applications will be made
14 # available underneath a path with that package name, such as /drupal8.
15 #
16 # Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
17 ##

18 # Default server configuration
19 #
20 server {
21     listen 80 default_server;
22     listen [::]:80 default_server;
23
24     # SSL configuration
25     #
26     # listen 443 ssl default_server;
27     # listen [::]:443 ssl default_server;
28     #
29     # Note: You should disable gzip for SSL traffic.
30     # See: https://bugs.debian.org/773332
31     #
32     # Read up on ssl_ciphers to ensure a secure configuration.
33     # See: https://bugs.debian.org/765782
34     #
35     # Self signed certs generated by the ssl-cert package
36     # Don't use them in a production server!
37     #
38     # include snippets/snakeoil.conf;
39
40     root /var/www/html;
41
42     # Add index.php to the list if you are using PHP
43     index index.php index.html index.htm index.nginx-debian.html;
```

nginx - /etc/nginx/sites-enabled/default (cont.)

```
45      server_name _;
46
47      location / {
48          # First attempt to serve request as file, then
49          # as directory, then fall back to displaying a 404.
50          #try_files $uri $uri/ =404;
51      }
52
53
54      # pass PHP scripts to FastCGI server
55      #
56      #location ~ \.php$ {
57      #    include snippets/fastcgi-php.conf;
58      #
59      #    # With php-fpm (or other unix sockets):
60      #    fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
61      #    # With php-cgi (or other tcp sockets):
62      #    fastcgi_pass 127.0.0.1:9000;
63      #}
64
65      # deny access to .htaccess files, if Apache's document root
66      # concurs with nginx's one
67      #
68      #location ~ /\.ht {
69      #    deny all;
70      #}
71  }
72
73
74  # Virtual Host configuration for example.com
75  #
76  # You can move that to a different file under sites-available/ and symlink that
77  # to sites-enabled/ to enable it.
78  #
79  #server {
80  #    listen 80;
81  #    listen [::]:80;
82  #
83  #    server_name example.com;
84  #
85  #    root /var/www/example.com;
86  #    index index.html;
87  #
88  #    location / {
89  #        try_files $uri $uri/ =404;
90  #    }
91  #}
```

.htaccess

- Allow admin or users to control access to certain directories
- Usage
 - Create .htaccess file
 - Generate user entry
 - Modify nginx configuration
 - Reload nginx

.htpasswd - Set Up

```
root:~/ # apt install -y apache2-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2-utils is already the newest version (2.4.41-4ubuntu3.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root:~/ # htpasswd -c /etc/nginx/.htpasswd user█
```

nginx - .htaccess

```
location /jsonrpc {
    proxy_pass http://127.0.0.1:6800;
    auth_basic "Not available";
    auth_basic_user_file /etc/nginx/.htpasswd;
}
```

[PRODUCTS](#)[SOLUTIONS](#)[RESOURCES](#)[SUPPORT](#)[PRICING](#)[BLOG](#)[FREE TRIAL](#)[CONTACT US](#)

NGINX 3rd Party Modules

Below is a list of third-party modules for NGINX and NGINX Plus, created and maintained by members of the NGINX community. NGINX, Inc. does not provide support for these modules, so please reach out to each individual module developer for issues or help. For information on how to contribute a module to this list, see <https://github.com/nginxinc/nginx-wiki>.

For a list of officially supported modules from NGINX, Inc. and NGINX Plus Certified Modules from our partners, see <https://www.nginx.com/products/nginx/modules>.

For information on how to contribute a module to this list, see <https://github.com/nginxinc/nginx-wiki>.

Module	Description	Repository
Accept Language Module	Accept-Language header parser	https://github.com/giom/nginx_accept_language_module
AFCGI	Asynchronous/multiplexing FastCGI for NGINX	https://github.com/rsms/afcgi
Akamai G2O	Restricts access to content to Akamai edge servers using G2O headers	https://github.com/kaltura/nginx_mod_akamai_g2o
Array Var	Add support for array variables to NGINX config files	https://github.com/openresty/array-var-nginx-module

Wiki home

[Getting Started](#)[Community](#)

NGINX 3rd Party Modules

- [Accept Language Module](#)
- [Audio Track for HTTP Live Streaming](#)
- [HTTP Digest Authentication](#)
- [Circle Gif](#)
- [Clojure](#)
- [Concat](#)
- [Upstream Consistent Hash](#)
- [Upstream Domain Resolve](#)
- [HTTP Drizzle](#)
- [HTTP Echo Module](#)
- [Upstream Fair Balancer](#)

Do you need to compile NGINX yourself?

- The suggestion is no
 - From time to time, these web servers can have huge exploit in the wild
 - You want to install the security patch as soon as it's released
 - Manually compile these kind of services can be a hassle when you want to upgrade the software
- Only manually compile when
 - You got lots of server, they all have same spec
 - You want to tune the software specifically for your platform
 - ...etc

FastCGI & php-fpm

FastCGI

- Nginx web server can serve static files. But we want to execute some program to achieve more function.
- CGI: a protocol between web server and your program
- FastCGI
 - CGI create a new process to handle new requests.
 - FastCGI reuse a process to handle requests, so it only read config once.
 - Faster
 - But php slow

PHP-FPM

- A implementation of FastCGI
- How to use
 - apt install php7.4-fpm
 - systemctl enable php7.4-fpm --now
- Nginx connect php-fpm by unix socket

```
fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
```

- Then, we can execute php program

I don't want to write php

- Some back-end languages(or frameworks) can listen port directly
 - python django -> uwsgi
 - nodejs
- But we still use nginx to do load balance
 - nginx is faster than theese language

What we do in lab6 server

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
    # Add index.php to the list if you are using PHP  
  
    server_name _;  
  
    root /www;  
    location /lab6 {  
        root /www/lab6;  
        try_files /index.php $uri =404;  
        fastcgi_split_path_info ^(.+\.php)(/.+)$;  
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;  
        fastcgi_index index.php;  
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;  
        include fastcgi_params;  
    }  
}
```

nginx conf

```
<?php  
$now = time();  
if ($now >= strtotime("today 3:06:45") && $now <= strtotime("today 3:07:15")){  
    echo "NASA{WOW_YOU_GET_FLAG}";  
    die();  
}  
if ($now >= strtotime("today 4:57:45") && $now <= strtotime("today 4:58:15")){  
    echo "NASA{gotosleepplease}";  
    die();  
}  
if ($now >= strtotime("today 6:45:45") && $now <= strtotime("today 6:46:15")){  
    echo "NASA{don_t_lie_to_me_you_send_request_by_yourself}";  
    die();  
}  
if ($now >= strtotime("today 22:14:45") && $now <= strtotime("today 22:15:15")){  
    echo "NASA{I_m_not_a_robot}";  
    die();  
}  
echo "it's not time to rock!";
```

/www/lab6/index.php

RESTful

RESTful

- Representational State Transfer -> REST
- If a web is designed based on REST -> RESTful
- Uniform Interface
 - Based on resources
 - Every resources can be accessed by Uniform Resource Identifier(URI)
 - e.x. <https://google.com/search>
 - Client can change resource
 - Each message includes enough information to describe how to process the message.

RESTful

- Client–server architecture
- Statelessness
 - Server don't save client's state(e.x. session cookie)
 - Client should save all it's state
 - Client don't need to connect same server because client have **ALL** state
 - JWT
- Cacheability
 - Responses must define themselves either cacheable or non-cacheable
- Layered System
 - A client don't know whether it is connected directly to the end server or to an intermediary along the way.

SQL

Structured Query Language

- SQL
 - The most popular computer language used to create, modify, retrieve and manipulate data from **relational database** management systems
 - [Intro to SQL: Querying and managing data | Khan Academy](#)

MySQL

- A multithreaded, multi-user, SQL database management system
- Owned and sponsored by MySQL AB, acquired by Sun Microsystems in 2008.
 - Oracle acquired Sun in 2009
 - Forked version: [MariaDB](#), [Percona Server](#)



MySQL (cont.)

- Features
 - Writing in C/C++, tested by many compilers, **portable to many platforms**
- Providing APIs for C/C++, Java, Perl, PHP, Python, Ruby...etc
- **Multi-threaded** kernel, supporting system with multiple CPUs
- Optimized algorithm for **SQL** query
- Multi-language support
- Lots of connection method: TCP/IP, ODBC, JDBC, UNIX domain socket
- **Free software** (GNU GPL v2)

PostgreSQL

- Originally named POSTGRES, referring to its origins as a successor to [Ingres](#) database, developed in UC Berkeley
- Features
 - User-defined types
 - Table inheritance
 - Sophisticated locking mechanism
 - Foreign key referential integrity
 - Views, rules, subquery
 - Nested transactions
 - Multi-version concurrency control
 - Asynchronous replication
 - Native Windows Server support
 - Point-in-time recovery



CRUD

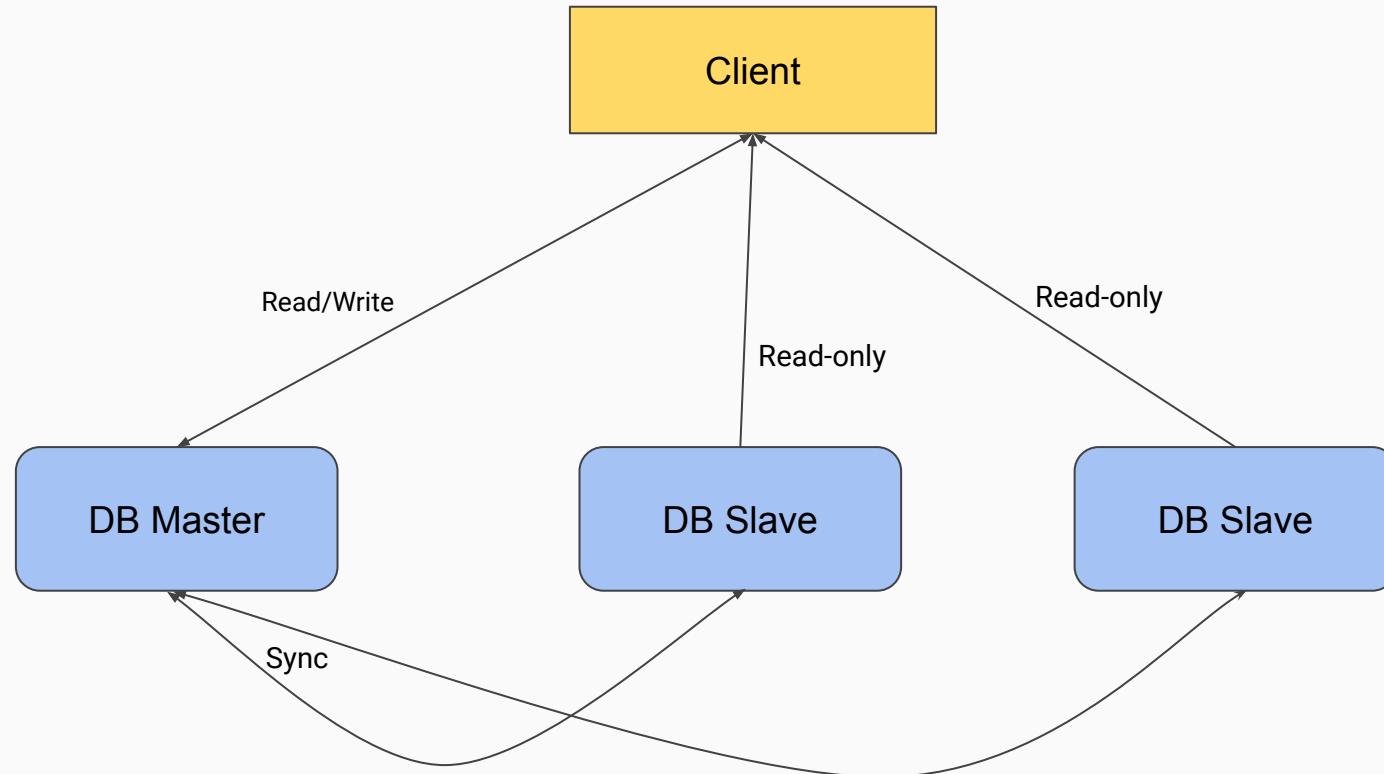
CRUD	SQL	REST	MongoDB
Create	INSERT	POST	Insert
Read	SELECT	GET	Find
Update	UPDATE	PUT	Update
Delete	DELETE	DELETE	Remove

*MongoDB is a kind of Document-oriented database

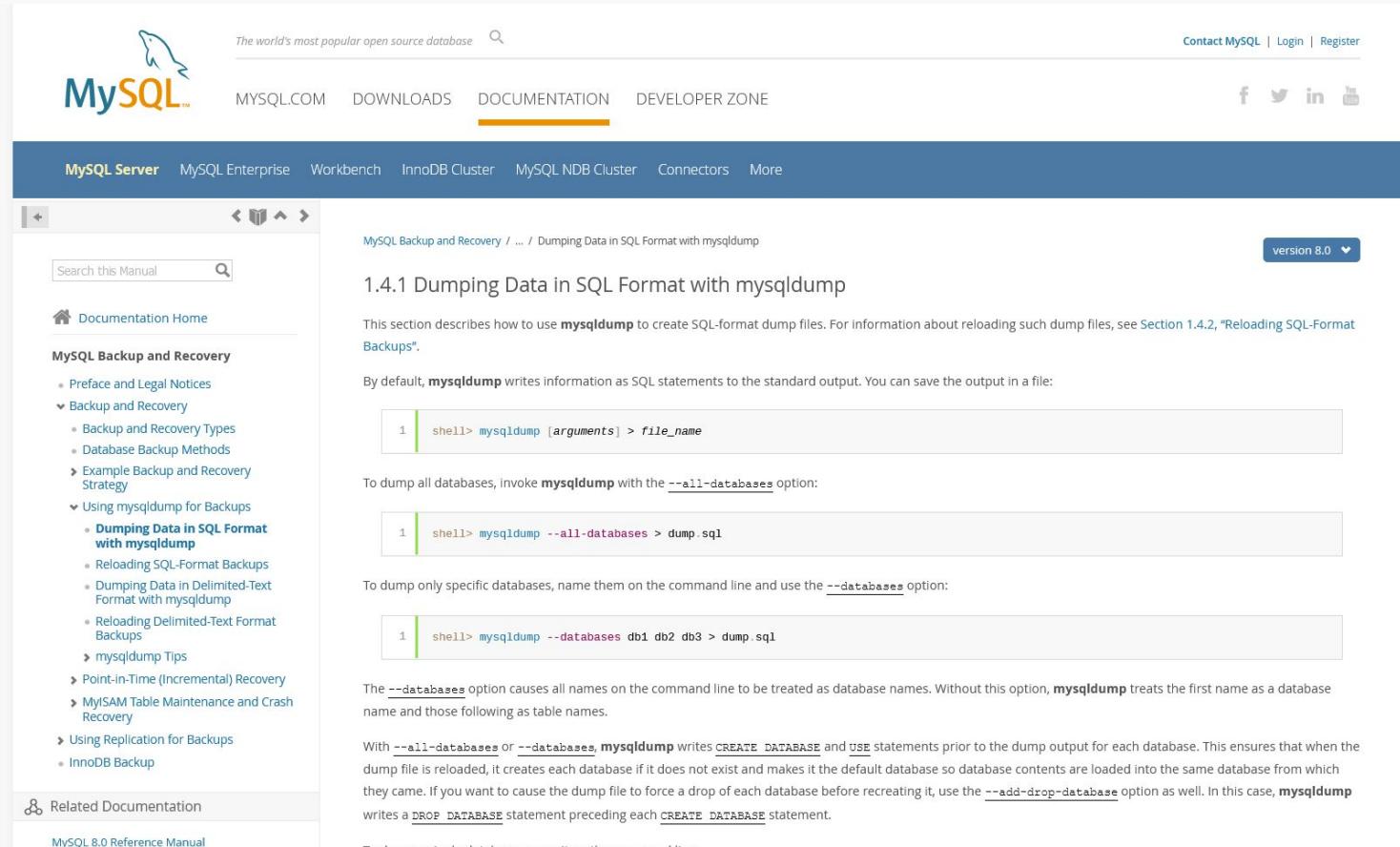
ACID

- **Atomicity**
 - guarantees that each transaction is treated as a single "unit", which either succeeds completely, or fails completely
- **Consistency**
 - during any time in transaction, database always valid
- **Isolation**
 - every transaction is isolated from any other transaction. effects of an incomplete transaction might not even be visible to other transactions
- **Durability**
 - after a transaction be committed, it will remain committed even in the case of a system failure

MySQL - Master-Slave



MySQL :: MySQL Backup and Recovery | Dumping Data in SQL Format with mysqldump



The screenshot shows the MySQL Documentation Home page. The top navigation bar includes links for MySQL.COM, DOWNLOADS, DOCUMENTATION (which is highlighted), and DEVELOPER ZONE. Below this is a secondary navigation bar with links for MySQL Server, MySQL Enterprise, Workbench, InnoDB Cluster, MySQL NDB Cluster, Connectors, and More. The main content area is titled "MySQL Backup and Recovery / ... / Dumping Data in SQL Format with mysqldump". A sidebar on the left contains a search bar, a documentation home link, and a detailed table of contents for MySQL Backup and Recovery, including sections on Preface and Legal Notices, Backup and Recovery (with sub-sections for Backup and Recovery Types, Database Backup Methods, Example Backup and Recovery Strategy, and Using mysqldump for Backups), Point-in-Time (Incremental) Recovery, MyISAM Table Maintenance and Crash Recovery, Using Replication for Backups, and InnoDB Backup. At the bottom of the sidebar is a "Related Documentation" section. The main content area features several code snippets in a terminal-like interface, illustrating how to use mysqldump. A "version 8.0" dropdown menu is visible in the top right corner.

1.4.1 Dumping Data in SQL Format with mysqldump

This section describes how to use **mysqldump** to create SQL-format dump files. For information about reloading such dump files, see Section 1.4.2, “Reloading SQL-Format Backups”.

By default, **mysqldump** writes information as SQL statements to the standard output. You can save the output in a file:

```
1 shell> mysqldump [arguments] > file_name
```

To dump all databases, invoke **mysqldump** with the --all-databases option:

```
1 shell> mysqldump --all-databases > dump.sql
```

To dump only specific databases, name them on the command line and use the --databases option:

```
1 shell> mysqldump --databases db1 db2 db3 > dump.sql
```

The --databases option causes all names on the command line to be treated as database names. Without this option, **mysqldump** treats the first name as a database name and those following as table names.

With --all-databases or --databases, **mysqldump** writes CREATE DATABASE and USE statements prior to the dump output for each database. This ensures that when the dump file is reloaded, it creates each database if it does not exist and makes it the default database so database contents are loaded into the same database from which they came. If you want to cause the dump file to force a drop of each database before recreating it, use the --add-drop-database option as well. In this case, **mysqldump** writes a DROP DATABASE statement preceding each CREATE DATABASE statement.

To dump a single database, name it on the command line.

Case Study

Load balance

- Nginx proxy

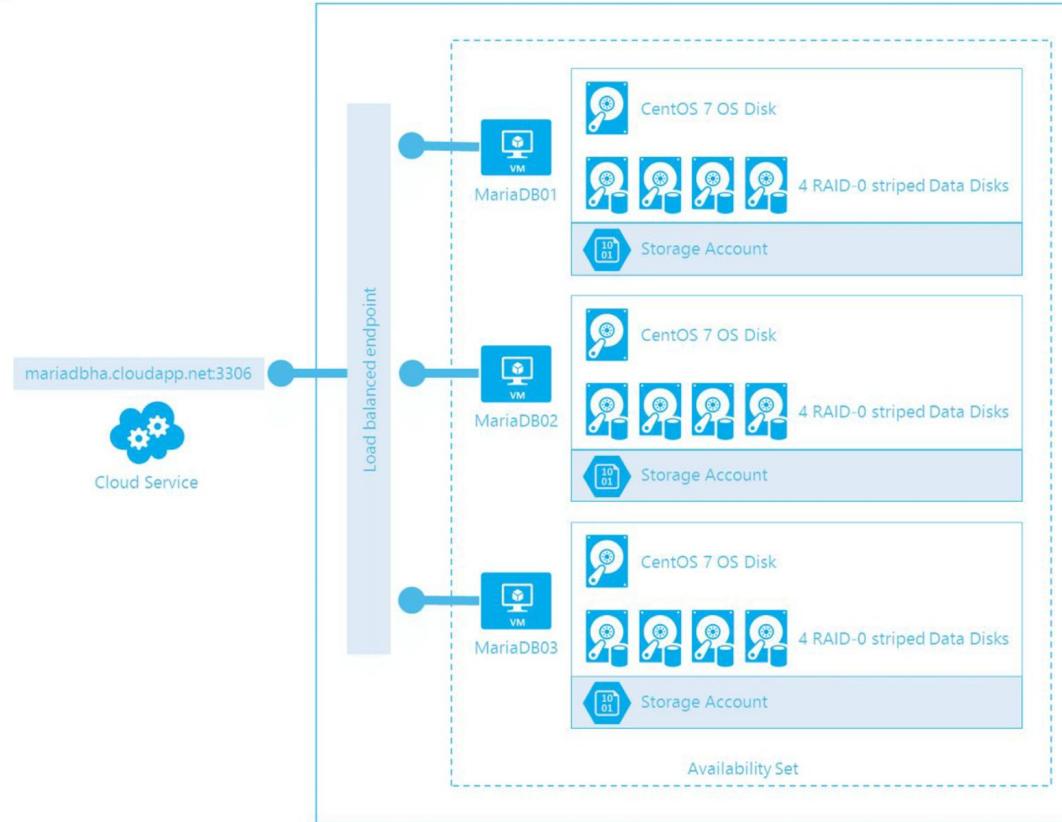
```
upstream backend {  
    server 172.16.1.1:3000;  
    server 172.16.1.2:3000;  
}  
  
server {  
    listen 80;  
    server_name www.example.com;  
    location / {  
        proxy_pass http://backend;  
    }  
}
```



Load balance

- Open Source
 - haproxy - <http://www.haproxy.org/>
 - envoy - <https://www.envoyproxy.io/>
- Commercial
 - F5
 - A10
 - AWS ELB
 - Google Cloud Load Balance

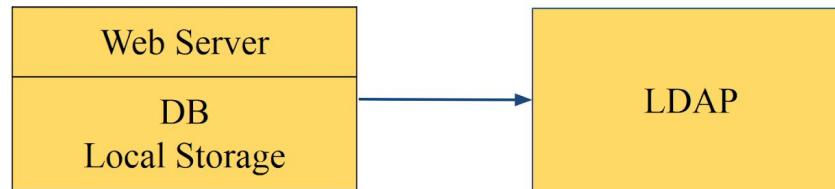
MySQL cluster



<https://sabbour.me/running-a-mariadb-cluster-on-azure/>

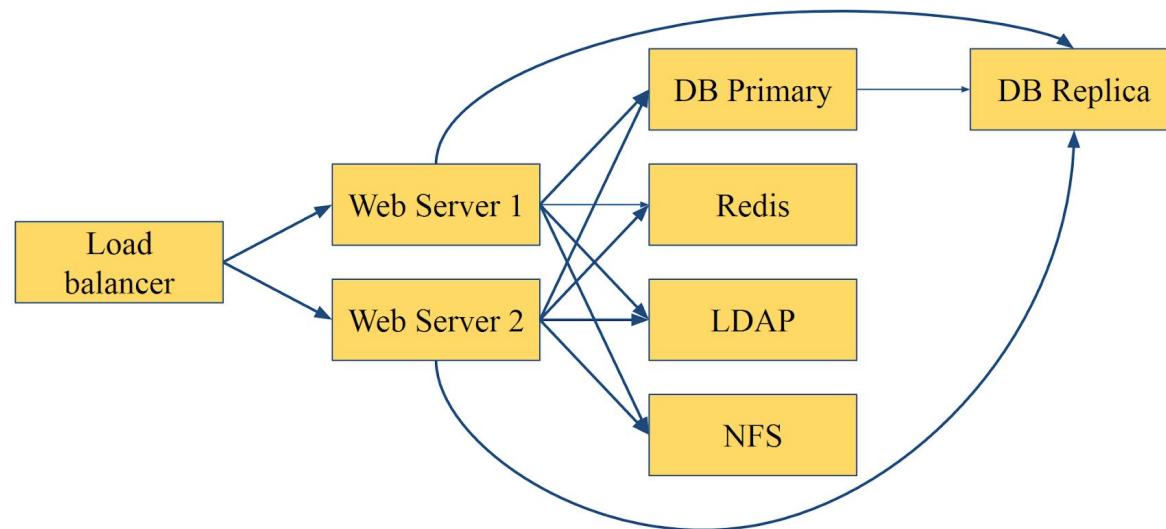
Up to 150 users

- One machine running the application server.
 - Web server
 - database server
 - local storage
- Authentication via an existing LDAP or Active Directory server.



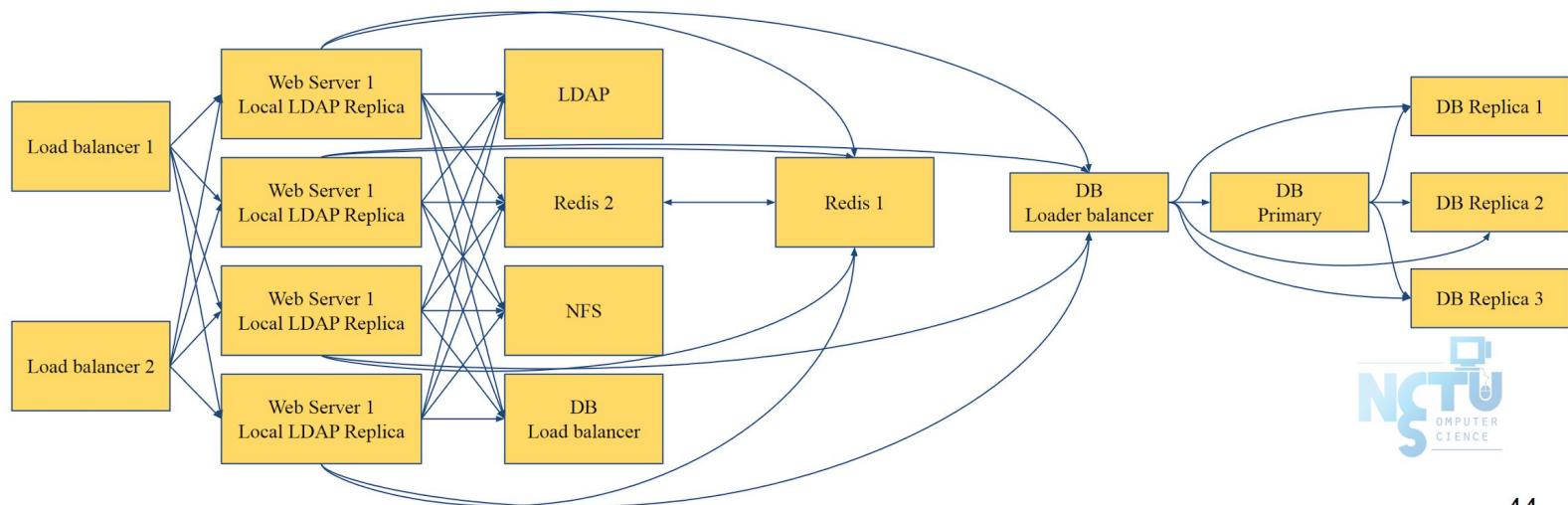
150 to 1,000 users

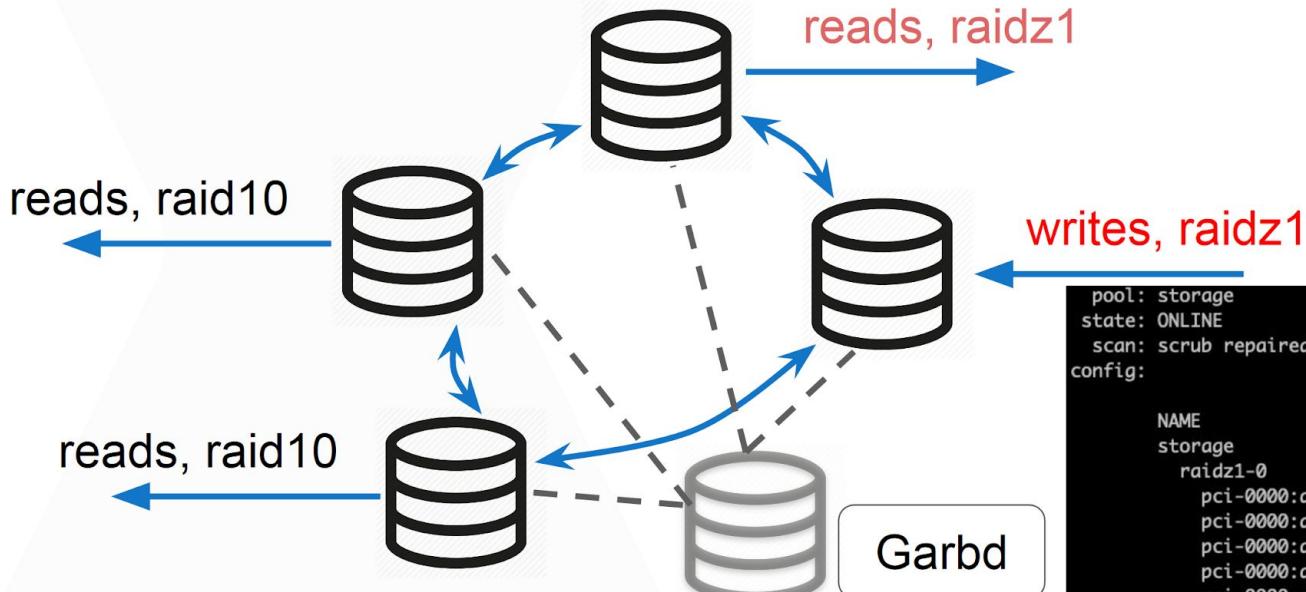
- High availability level
 - Every component is fully redundant and can fail without service interruption.
 - Backups without service interruption



5,000 to >100,000 users

- 4 to 20 application/Web servers.
- A cluster of two or more database servers
 - behind a load balancer to send all writes to the master and reads to the slaves.
- Storage is an NFS server, or an object store that is S3 compatible.





```
pool: storage
state: ONLINE
scan: scrub repaired 0B in 1h13m with 0 errors on Sun Nov 10 01:37:19 20
config:

  NAME
  storage
    raidz1-0
      pci-0000:d8:00.0-sas-exp0x500304801f1546ff-phy2-lun-0-part1
      pci-0000:d8:00.0-sas-exp0x500304801f1546ff-phy3-lun-0-part1
      pci-0000:d8:00.0-sas-exp0x500304801f1546ff-phy4-lun-0-part1
      pci-0000:d8:00.0-sas-exp0x500304801f1546ff-phy5-lun-0-part1
      pci-0000:d8:00.0-sas-exp0x500304801f1546ff-phy6-lun-0-part1
      pci-0000:d8:00.0-sas-exp0x500304801f1546ff-phy7-lun-0-part1
      pci-0000:d8:00.0-sas-exp0x500304801f1546ff-phy8-lun-0-part1
  logs
    mirror-1
      pci-0000:d8:00.0-sas-exp0x500304801f1546ff-phy38-lun-0-part3
      pci-0000:d8:00.0-sas-exp0x500304801f1546ff-phy39-lun-0-part3
  cache
    pci-0000:d8:00.0-sas-exp0x500304801f1546ff-phy0-lun-0-part1
    pci-0000:d8:00.0-sas-exp0x500304801f1546ff-phy1-lun-0-part1
  spares
    pci-0000:d8:00.0-sas-exp0x500304801f1546ff-phy9-lun-0-part1
```

注意：

本簡報大量引用網路討論，請斟酌服用



Source:

<https://www.facebook.com/707851765997333/photos/a.707865592662617.1073741827.707851765997333/707865582662618/>

為何每秒僅 401 個同時上線也會出包？

- 問題出於資料庫嗎？
- 某前輩指出，售票的事務邏輯很複雜，不要與一般的 QPS(每秒查詢處理量) 等同比較，之間相差的量級可能有百倍之有

資料庫使用的討論

- 以 MySQL 在 2010 年針對 5.1 版本進行的 TPS (每秒事務處理量) 效能測試為例
<http://www.percona.com/blog/2010/02/28/maximal-write-throughput-in-mysql/>
- 假設，
 - A. 以當初測試數據裡最嚴謹的數據來看，每秒至少在一萬上下 (10000 TPS)
 - B. 寬宏現在上線的資料庫處理能力與 4 年前的 MySQL 5.1 同等級。
(雖然實際上我認為他們用的是更快的資料庫)
 - C. 寬宏現有硬體不如測試環境上那般好。假設處理效能只達二分之一
 - D. 寬宏售票的事務比測試環境複雜十倍
- 所以， $10000 \div 2(C) \div 10(D) = 500$ TPS。即使在這麼艱難的條件下，單台資料庫也該能處理 500 TPS，高於先前推估的 401 同時在線人數

TonyQ 的解說

- 本質上你可以把所有演唱會位置視為一個格。交易這回事本質上就是把人分到他想要的格子。先不論金流來簡化問題，這個問題的關鍵因素在同時 (concurrent) 在線人數，這種大型演唱會搶票基本上都是同時萬人以上等級
- 我們會碰到的第一個問題是「我們要讓使用者知道哪個格有人、哪個格子沒人」，因為使用者要選位，這件事情就已經夠困難了

TonyQ 分析：Lock

- Lock 是千古難題，概念上不困難，但實作非常困難。race condition 一直是我們這個領域中最難處理的問題
- 很多人在談的機制都很合理，不論取號也好、各種作法也好。但重點是 concurrent 一萬人以上的情況，你一定要作「分散運算」，否則你在作業系統層面基本上很難處理這麼大量的 concurrent 操作，而且風險也很高（重點！）
- 但分散運算對 lock 來講，根本是先天的天險

學習電腦科學的你我，應該要能夠用專業看待這些經典議題

延伸閱讀：〈由 12306.cn 談談
網站性能技術〉

<http://coolshell.cn/articles/6470.html>

從線上售票系統看作業系統設計議題 (cont.)

真正幸福的人：不是搶到票，是可以像江蕙一樣選擇人生

撰文者 | 李柏鋒 | 李柏鋒的擴大機 | 瀏覽數 : 200+ | 2015-01-08

+A -A

392 2 0
f 賴 8+1 回應

收藏 列印 轉寄



來源：翻攝自江蕙臉書

放大顯示

Source: <http://www.businessweekly.com.tw/KBlogArticle.aspx?ID=10709>

Locking and MVCC in MySQL InnoDB

- Transaction Isolation Levels
- Locking
 - Pessimistic Locking
 - Optimistic Locking
- Multi-Version Concurrency Control (MVCC)
 - In InnoDB, no lock for read, lock for write, read/write are mutually exclusive
 - InnoDB multi versioning | MySQL

Locking and MVCC in MySQL InnoDB (cont.)

- Consistent Reads
 - Read operation is reading data from snapshot
- Lock
 - [Shared Locks](#)
 - [Exclusive Locks](#)

Questions?