Web hosting basics

CIS 2235 Linux System Administration

Why web hosting?

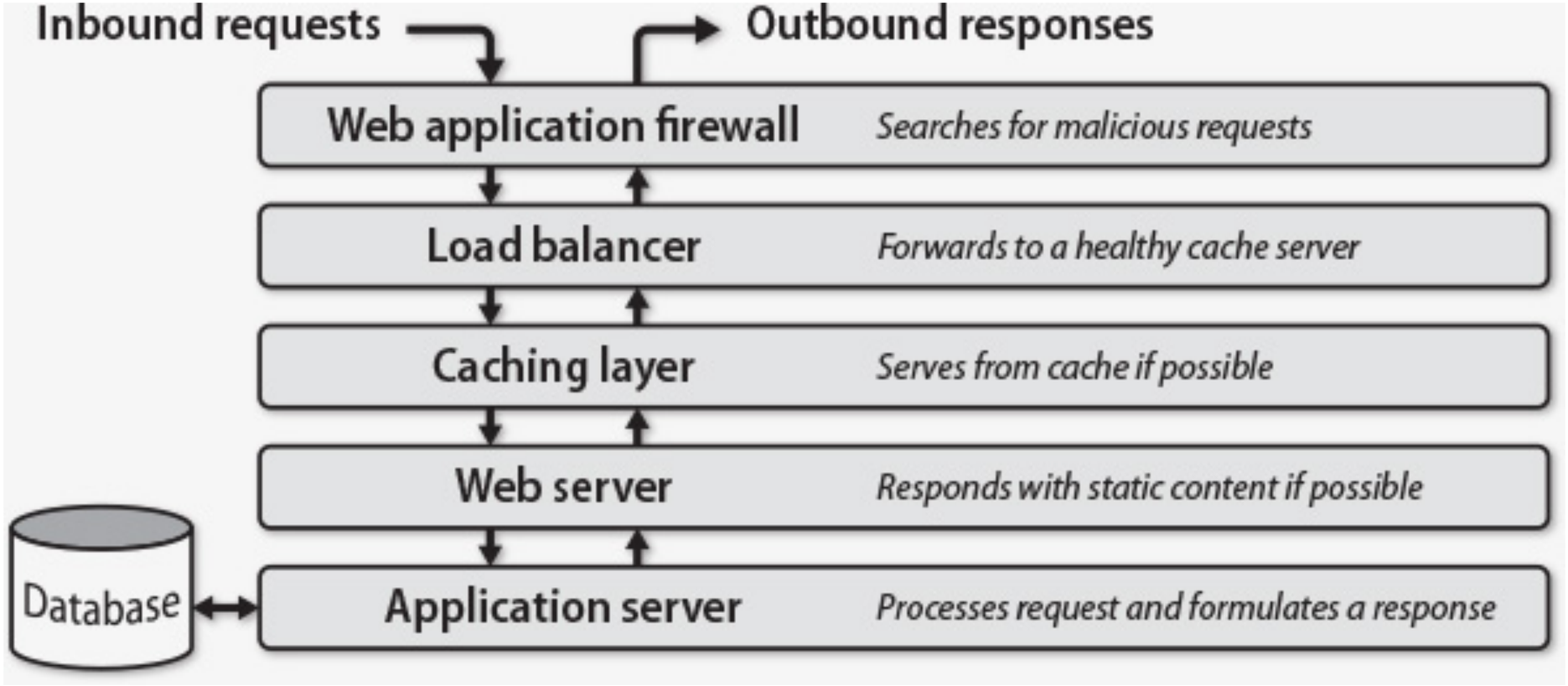
- One of the most common uses for Linux machines is to serve web applications
- ~ 2/3 of the top one million web sites are served by Linux or FreeBSD (per w3techs.com)
- 80% of web server software (not counting the OS) is open source software
- Configuring and maintaining this stack is a significant part of an IT administrator's job

Complications

Web applications are not a single piece of software on a single system.

- They are a collection of software components that cooperate
- Each piece must be resilient to:
  - system failure
  - load spikes
  - attacks
  - network failures

Components of a typical web application stack



Examples

Type	Purpose	Examples
Application server	Runs web app code, interfaces to web servers	Unicorn, Tomcat
Cache	Speeds access to frequently requested content	Varnish, Squid
Load balancer	Relays requests to downstream systems	Pound, HAProxy
Web app firewall [a]	Inspects HTTP traffic for common attacks	ModSecurity
Web server	Serves static content, couples to other servers	Apache, NGINX

a. Often abbreviated WAF

Not all applications have all these pieces, but many do — or have even more complicated architectures.

Some software packages can fill more than one role, too.

Firewall

- Inspects HTTP traffic for common attacks
- Only allows specified traffic through
- Hides machines behind it to reduce risk
- Generally hardware, but can be software (iptables)

load balancer

A highly available web server should run on multiple servers

- allows performing system maintenance, like patching
- crashing won't take down the entire site
- harder to overload

Load balancers distribute incoming requests to a group of servers, based on pre-determined rules.

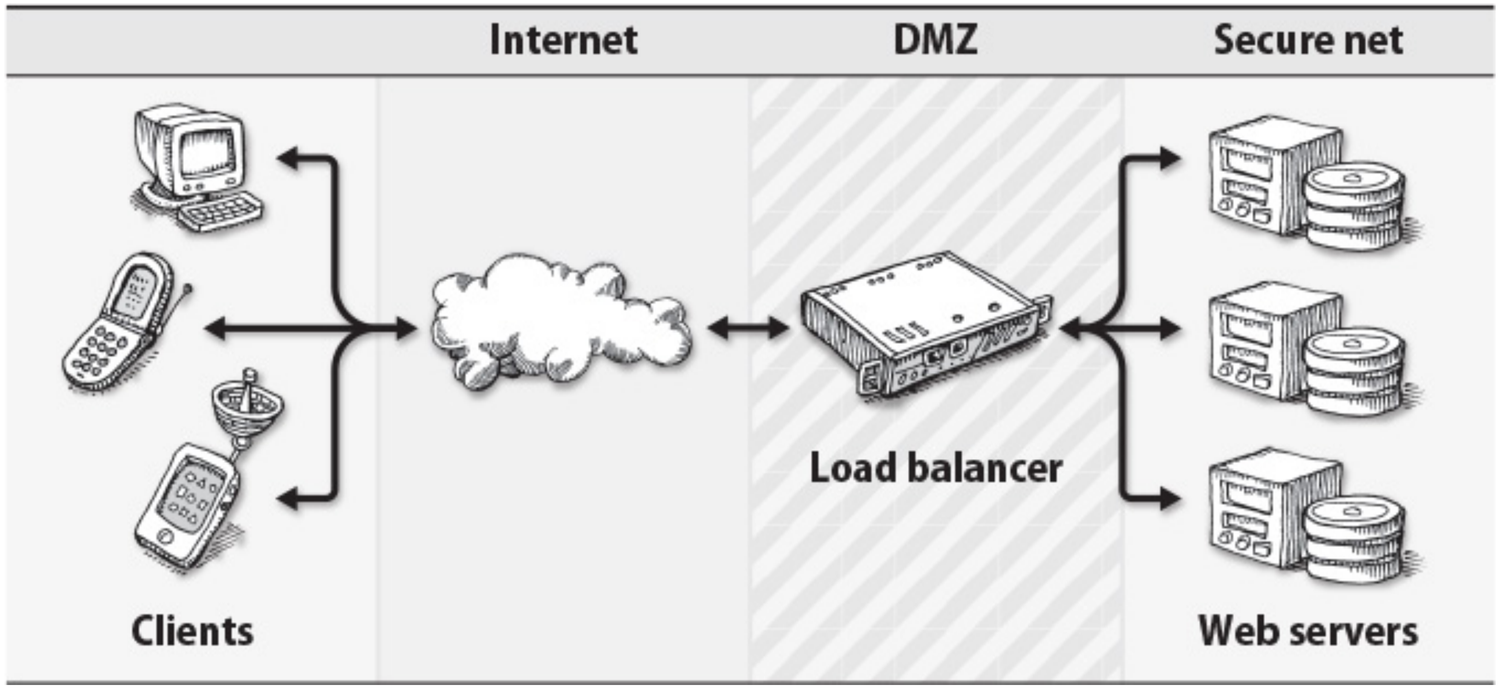
Also monitor the status of the servers, and will route traffic away from servers that aren't responding.

load balancers, cont.

does not process requests, merely routes them

this allows higher volume

servers can be added/removed from rotation transparently to the end user



request distribution

Requests can be distributed different ways

- round robin - fixed rotation order
- load equalization - requests to go server that is "least" busy (connections or requests)
- partitioning - select server based on some criteria (like ip address) so the same client requests go to the same server

Common load balancers

Most common (software) load balancers are
- NGINX - also used as a web server
- HAProxy
- apache http, but it is very primitive

Also hardware load balancers
- F5
- Citrix

Amazon offers Elastic Load Balancer (ELB)
- completely managed service
- for use with EC2 machines

Cache servers

For performance reasons would like to be able to store the results for the most frequent requests

- reduces load on servers
- can be geographically closer to clients

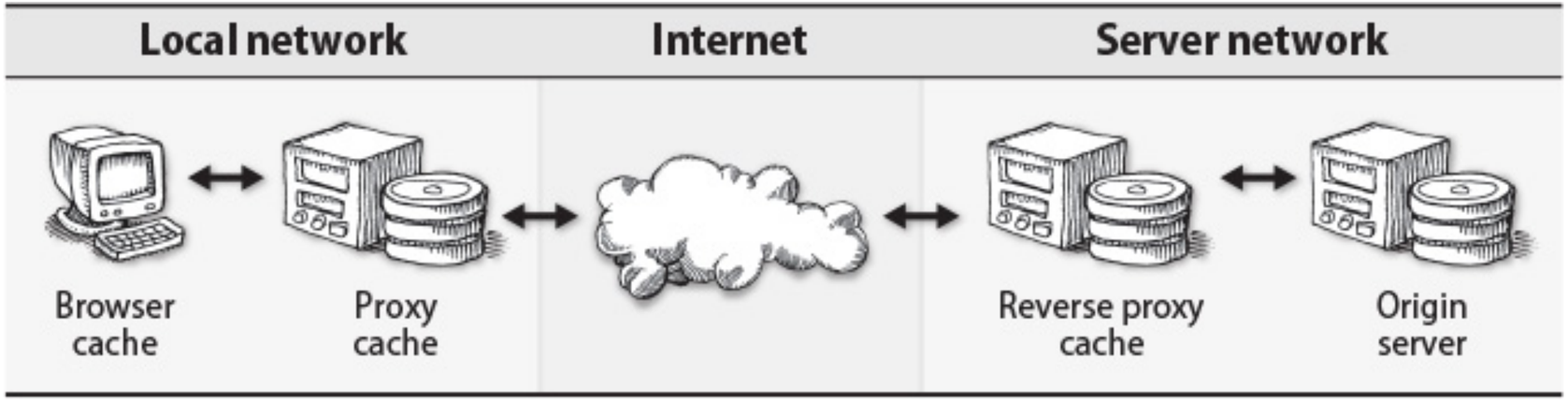
Caches get content from origin server

- origin server is just an upstream cache

Used for static content

- images, videos, css, static html files

Levels of cache



1: browser cache on local machine

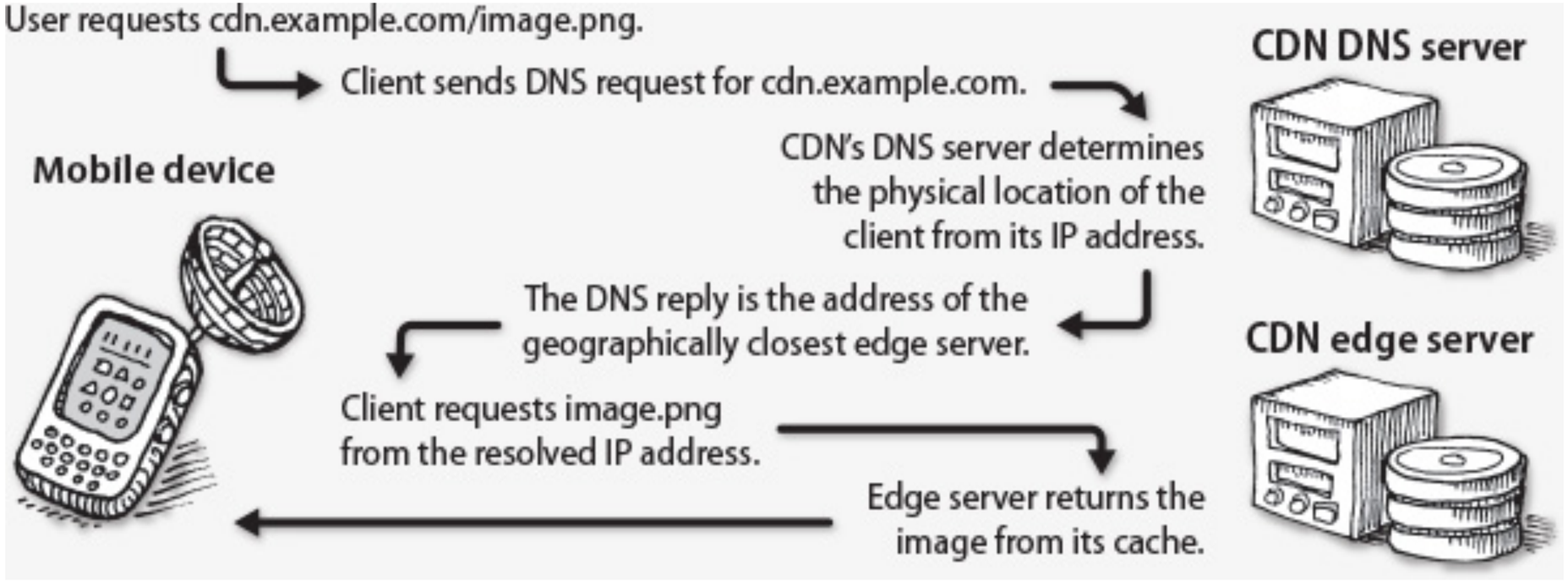
2: Organizations can have a proxy cache for their site

3: Web sites use reverse proxy cache (or edge servers)

Can be part of content delivery network

site content geographically closer to user

Content delivery network



CDNs

Commonly used for content rich sites

Netflix and YouTube use CDNs to serve their video files

CDN sites

- Akamai — large customers, most full featured
- CloudFlare — smaller customers
- CloudFront (Amazon)

Open source caching software

Server	Notes
Squid	One of the first open source cache implementations Normally used as a proxy cache Includes important features like antivirus and TLS
Varnish	Exceptional configuration language Multithreaded Modular and extensible
Apache mod_cache	Good choice for sites already running **httpd**
NGINX	Good choice for sites already running NGINX Has a reputation for good performance
Apache Traffic Server	Runs at extremely high-traffic sites Supports HTTP/2 Donated to the Apache Foundation by Yahoo!

Web server

Web servers are used to

- server static content directly
- proxy HTTP connections to application servers

Common features

- virtual hosts — allowing multiple sites to coexist within a single server
- handling TLS (SSL) connections
- logging of incoming requests and outgoing responses
- basic HTTP authentication

Common web servers

Apache HTTP server aka httpd, which is the name of the process it runs as

- long time leader in space

NGINX (pronounced "engine-X")

- newer, still growing
- designed for speed and efficiency
- simpler configuration than Apache

IIS (internet information services) is Microsoft's web server

- used by MyWebGrocer, Inntopia

Current usage: w3Tech, netcraft survey

Application Servers

- Application servers handle the business logic of the site
  - somewhat arbitrary distinction, so don't hold too tightly to it
- also tend to include
  - connection pooling
  - transaction support
  - messaging services
- Common servers
  - Apache Tomcat
  - IBM Websphere
  - Microsoft IIS