Naming & DNS

CS 475, Fall 2019
Concurrent & Distributed Systems
Review: Recurring Problem: Replication

- Replication solves some problems, but creates a huge new one: consistency

Set $A=5$

"OK"!

Read $A$

"6"!

OK, we obviously need to actually do something here to replicate the data… but what?
Review: CAP Theorem

• Pick two of three:
  • Consistency: All nodes see the same data at the same time (sequential consistency)
  • Availability: Individual node failures do not prevent survivors from continuing to operate
  • Partition tolerance: The system continues to operate despite message loss (from network and/or node failure)
  • You can not have all three, ever
Review: CAP Theorem

- C+A: Provide strong consistency and availability, assuming there are no network partitions
- C+P: Provide strong consistency in the presence of network partitions; minority partition is unavailable
- A+P: Provide availability even in presence of partitions; no sequential consistency guarantee, *maybe can guarantee something else*
Review: Relaxing Consistency

- We can relax two design principles:
  - How stale reads can be
  - The ordering of writes across the replicas
Review: Choosing a consistency model

• Sequential consistency
  • All over - it’s the most intuitive
• Causal consistency
  • “Increasingly useful” but not really widely used - still pay coordination cost, unclear what the performance benefits are
• Eventual consistency
  • Very popular in industry and academia
  • File synchronizers, Amazon’s Bayou and more
Example: Facebook

- Problem: >1 billion active users
- Solutions: Thousands of servers across the world
- Need 100% availability!
Example: Facebook

- Problem: >1 billion active users
- Solutions: Thousands of servers across the world
- What kind of consistency guarantees are reasonable? Need 100% availability!
- If I post a story on my news feed, is it OK if it doesn’t immediately show up on yours?
- Two users might not see the same data at the same time
- Now this is “solved” anyway because there is no “sort by most recent first” option anyway
Example: Airline Reservations

- Reservations and flight inventory are managed by a GDS (Global Distribution System), who acts as a middle broker between airlines, ticket agencies and consumers [Except for Southwest and Air New Zealand and other oddballs]
- GDS needs to sell as many seats as possible within given constraints
Example: Airline Reservations

- If I have 100 seats for sale on a flight, does it matter if reservations for flights are reconciled immediately?
- If I have 5 seats for sale on a flight, does it matter if reservations are reconciled immediately?
- Result: Reservations can be made using either a strong consistency model or a weak, eventual one
- Most reservations are made under the normal strong model (reservation is confirmed immediately)
- GDS also supports “Long Sell” - issue a reservation without confirmed availability, need to eventually reconcile it
- Long sells require the seller to make clear to the customer that even though there’s a confirmation number it’s not confirmed!
HW3 Graded

HW3 Grades, as of Sun Nov 03 2019
• Each KVStore client will have the entire dataset cached locally
• When updating values, the update will be propagated to each replica
Each KVStore client will have the entire dataset cached locally
When updating values, the update will be propagated to each replica
Ivy vs HW4

- Ivy never copies the actual values until a replica reads them (unlike HW4)
  - Invalidate messages are probably smaller than the actual data!
- Ivy only sends update (invalidate) messages to replicas who have a copy of the data (unlike HW4)
  - Maybe most data is not actively shared
- Ivy requires the lock server to keep track of a few more bits of information (which replica has which data)
  - With near certainty Ivy is a lot faster :)

14
Today

• This week - case studies in replication
• Today: DNS and naming (partially explaining how the internet works)
• Reminder:
  • HW4 is due 11/18!
How do we find data?

- DNS - Domain Name System - responsible for mapping IP address to human-readable domain names
- DNS is a distributed system
- Not immediately obvious how to scale: how do we maintain replication, some measure of consistency?
Partitioning + Replication

How do clients know where to ask for data?
Partitioning + Replication

• If input is structured, can possibly leverage that structure to build these buckets (**name spaces**)
• Example: File system
  • Map from: /home/bellj/teaching/swe622 to file contents
  • Could have different machines responsible for each tier?
  • We will look at file systems on Wednesday
• Example: DNS system
  • Maps from: www.jonbell.net TO: 104.24.122.171
  • Different machines for each tier?
Domain Name System

• Obvious solution: Local file
  • Keep local copy of mapping from all hosts to all IPs (e.g., /etc/hosts)
  • Hosts change IPs regularly: Download file frequently
• IPv4 space is now full
  • 32-bits: 4,294,967,296 addresses
  • At 1 byte per address, file would be 4GB
  • Not a lot of disk space (now, DNS introduced in the late 80s)
Domain Name System

- Obvious solution: Local file
  - Keep local copy of mapping from all hosts to all IPs (e.g., `/etc/hosts`)
  - Hosts change IPs regularly: Download file frequently
  - IPv4 space is now full
    - 32-bits: 4,294,967,296 addresses
    - At 1 byte per address, file would be 4GB
    - Not a lot of disk space (now, DNS introduced in the late 80s)

We need 200x of these to hold 4GB: $270K+
Domain Name System

- Obvious solution: Local file
  - Keep local copy of mapping from all hosts to all IPs (e.g., /etc/hosts)
  - Hosts change IPs regularly: Download file frequently
  - IPv4 space is now full
    - 32-bits: 4,294,967,296 addresses
    - At 1 byte per address, file would be 4GB
    - Not a lot of disk space (now, DNS introduced in the late 80s)
    - But a lot of constant internet bandwidth
- More names than IPs
  - Aliases
- Not scalable!
Domain Name System

- **Obvious solution: Local file**
  - Keep local copy of mapping from all hosts to all IPs (e.g., `/etc/hosts`)
  - Hosts change IPs regularly: Download file frequently
  - IPv4 space is now full
    - 32-bits: 4,294,967,296 addresses
    - At 1 byte per address, file would be 4GB
    - Not a lot of disk space (now, DNS introduced in the late 80s)
    - But a lot of constant internet bandwidth
  - More names than IPs
    - Aliases
  - **Not scalable!**

- **Obvious solution: Well known centralized server**
  - Single point of failure
  - Traffic volume
  - Access time
  - **Not scalable!**

---

http://a.root-servers.org/static/index.html
Domain Name System

• Goals
  • Scalable
  • Robust
    • High availability
  • Decentralized maintenance
  • Global scope
    • Names mean the same thing everywhere

• Non-goals
  • Atomicity
  • Strong consistency
Idea: break apart responsibility for each part of a domain name (**zone**) to a different group of servers

Each zone is a continuous section of name space
Each zone has an associate set of name servers
DNS

• Can have more/less servers replicating each zone based on popularity
• DNS responses are cached at clients
  • Caches periodically time out; bigger zones tend to have longer timeouts
  • Quick response for the same request, also for similar requests
DNS: Example
DNS: Example

Local DNS Server

Local server has cached the server for *.cs.gmu.edu!

hydra.cs.gmu.edu
129.174.115.88

129.174.115.88

129.174.115.88
cs.gmu.edu
Domain Name System

Root Servers

net
org
edu
com
gov
uk
root-servers

www
gmu

www
www

www
Domain Name System

Global Layer
- net
- org
- edu
- com
- gov
- uk

Administrational Layer
- root-servers
- gmu

Managerial Layer
- www
- www
- cs
- www
• 13 root servers
  • [a-m].root-servers.org
  • E.g., d.root-servers.org
• Handled by 12 entities
• How many physical servers?
  • a) Less than 13
  • b) 13
  • c) Tens
  • d) Hundreds
  • e) Thousands
  • f) Millions
### Domain Name System - Root Servers

- **13 root servers**
  - [a-m].root-servers.org
  - E.g., d.root-servers.org
- **Handled by 12 entities**
- **How many physical servers?**
  - a) Less than 13
  - b) 13
  - c) Tens
  - d) Hundreds
    - 980
  - e) Thousands
  - f) Millions

<table>
<thead>
<tr>
<th>Entity</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verisign, Inc.</td>
<td>a</td>
</tr>
<tr>
<td>Information Sciences Institute</td>
<td>b</td>
</tr>
<tr>
<td>Cogent Communications</td>
<td>c</td>
</tr>
<tr>
<td>University of Maryland</td>
<td>d</td>
</tr>
<tr>
<td>NASA Ames Research Center</td>
<td>e</td>
</tr>
<tr>
<td>Internet Systems Consortium, Inc.</td>
<td>f</td>
</tr>
<tr>
<td>U.S. DOD Network Information Center</td>
<td>g</td>
</tr>
<tr>
<td>U.S. Army Research Lab</td>
<td>h</td>
</tr>
<tr>
<td>Netnod</td>
<td>i</td>
</tr>
<tr>
<td>Verisign, Inc.</td>
<td>j</td>
</tr>
<tr>
<td>RIPE NCC</td>
<td>k</td>
</tr>
<tr>
<td>ICANN</td>
<td>l</td>
</tr>
<tr>
<td>WIDE Project</td>
<td>m</td>
</tr>
</tbody>
</table>
Domain Name System - Root servers

www.root-servers.net
Domain Name System - Root servers

www.root-servers.net
Domain Name System - Root servers
### Domain Name System - Root servers

<table>
<thead>
<tr>
<th>Operator</th>
<th>U.S. Army Research Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locations:</td>
<td>Sites: 2</td>
</tr>
<tr>
<td></td>
<td>Aberdeen Proving Ground, US, San Diego, US</td>
</tr>
</tbody>
</table>

[www.root-servers.net](http://www.root-servers.net)
Domain Name System - Root servers

Operator:
NASA Ames Research Center

Locations:
Sites: 191

Accra, GH
Amsterdam, NL
Amsterdam, NL
Arusha, TZ
Ashburn, US
Atlanta, US
Atlanta, US
Auckland, NZ
Auckland, NZ
Bangkok, TH
Barcelona, ES
Beaverton, US
Beirut, LB
Belgrade, RS
Berlin, DE
Berlin, DE
Boston, US
Boston, US
Brisbane, AU
Brisbane, AU
Bucharest, RO
Bucharest, RO
Budapest, HU
Buenos Aires, AR
Buenos Aires, AR
Burj,-30
 Cairo, EG
Calgary, CA
Cape Town, ZA
Cape Town, ZA
Chelmsford, US
Chennai, IN
Chicago, US
Chicago, US
Colombo, LW
Copenhagen, DK
Cordoba, AR
Dallas, US
Dar es Salaam, TZ
Dusseldorf, DE
Denver, US
Denver, US
Detroit, US
Detroit, US
Dhaka, BD
Djibouti, DJ
Doha, QA
Dubai, AE
Dublin, IE
Dublin, IE
Durban, ZA
Frankfurt, DE
Halifax, CA
Hamburg, DE
Helsinki, FI
Hong Kong, HK
Istanbul, TR
Jacksonville, US
Jakarta, ID
Johannesburg, ZA
Johannesburg, ZA
Kansas City, US
Kathmandu, NP
Kiev, UA
Klagenfurt, AT
Kuala Lumpur, MY
Kuwait City, KW
Las Vegas, US
Leeds, UK
Lima, PE
Lisbon, PT
London, UK
Los Angeles, US
Luanda, AO
Lyon, FR
Madrid, ES
Manchester, UK
Manchester, UK
Manila, PH
Maputo, MZ
Marseille, FR
McAllen, US
Melbourne, AU
Mexico City, MX
Miami, US
Milan, IT
Minneapolis, US
Mombasa, KE
Montgomery, US
Montreal, CA
Montreal, CA
Mountain View, US
Mumbai, IN
Munich, DE
Muscat, OM
Nairobi, KE
Nashville, US
Neuen, AR
New Delhi, IN
New York, US
Newark, US
Newark, US
Omaha, US
Osaka, JP
Oslo, NO
Ottawa, CA
Palo Alto, US
Panama City, PA
Paris, FR
Perth, AU
Perth, AU
Philadelphia, US
Phoenix, US
Phoenix, US
Port Louis, MU
Port of Spain, TT
Portland, US
Prague, CZ
Prague, CZ
Quito, EC
Reno, US
Richmond, US
Rio de Janeiro, BR
Rome, IT
Rouseau, DO
Salt Lake City, UT
San Diego, US
San Francisco, US
San Jose, US
Santa Ana, US
Santiago, CL
Saskatoon, CA
Sao Paulo, BR
Seattle, US
Seattle, US
Seoul, KR
Seoul, KR
Singapore, SG
Singapore, SG
Sochi, BG
Sochi, BG
St. George, US
St. George’s, GD
St. Louis, US
St. Louis, US
Stockholm, SE
Sydney, AU
Taipei, TW
Tallinn, EE
Tampa, US
Tampa, US
Tampere, FI
Tokyo, JP
Tokyo, JP
Toronto, CA
Turin, IT
Valparaiso, CL
Vancouver, CA
Vienna, AT
Vienna, AT
Warsaw, PL
Washington, US
Wellington, NZ
Willemsvat, CW
Winnipeg, CA
Yerevan, AM
Zagreb, HR
Zurich, CH
Domain Name System - Scale

Global Layer

- net
- org
- edu
- com
- gov
- uk

Administrational Layer

- root-servers
- gmu

Managerial Layer

- www
- www
- www
- cs
Domain Name System - Zones

DNS zones are managed by the same entity
Domain Name System - Questions/Answers

• DNS message format is the same for questions and answers
  • Header
    • ID
      • Question/Answer have the same ID
    • Flags
      • 1 bit for question/answer, etc.
    • Number of questions
    • Number of answers
    • Number of authority RRs (Resource Records)
    • Number of additional RRs
  • Questions
  • Answers
  • Authority
  • Additional Info
Domain Name System - Resource Records (RRs)

- RR format: (class, name, value, type, ttl)
  - Class: Internet (IN)
  - Type
    - A (AAAA for IPv6)
      - name is hostname
      - value is IP address
    - NS
      - name is domain
      - value is authoritative server for domain
    - CNAME
      - name is alias for some canonical (real) name
      - value is canonical name
    - And more...
Domain Name System - Example Query

dig(1) - Linux man page

Name
dig - DNS lookup utility

Synopsis
dig [@server] [-b address] [-c class] [-f filename] [-k filename] [-m] [-p port#] [-q name] [-t type] [-x addr] [-y hmac:]name:key] [-4] [-6] [name] [type] [class] [queryopt...]
dig [-h]
dig [global-queryopt...] [query...]
Domain Name System - Example Query

> dig www.gmu.edu a
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20159
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 3
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.gmu.edu. IN A
;; ANSWER SECTION:
jiju3.gmu.edu. 46534 IN A 129.174.1.59
;; AUTHORITY SECTION:
gmu.edu. 86013 IN NS eve.gmu.edu.
gmu.edu. 86013 IN NS uvaarpa.virginia.edu.
gmu.edu. 86013 IN NS magda.gmu.edu.
;; ADDITIONAL SECTION:
eve.gmu.edu. 3219 IN A 129.174.253.66
magda.gmu.edu. 1993 IN A 129.174.18.18
uvaarpa.virginia.edu. 84640 IN A 128.143.2.7
;; Query time: 2 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Thu Feb 15 10:19:24 EST 2018
;; MSG SIZE  rcvd: 212
Domain Name System - Example Query

> dig www.ic.ac.uk a

;; Got answer:

;; ->>>HEADER<<<< opcode: QUERY, status: NOERROR, id: 28622

;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:

;; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;www.ic.ac.uk.                  IN      A

;; ANSWER SECTION:

www.ic.ac.uk.           271     IN      CNAME   wrp.cc.gslb.ic.ac.uk.
wrp.cc.gslb.ic.ac.uk.   1       IN      A       146.179.40.24

;; Query time: 1 msec

;; SERVER: 192.168.1.1#53(192.168.1.1)

;; WHEN: Thu Feb 15 10:23:52 EST 2018

;; MSG SIZE  rcvd: 83
Domain Name System - Resolution

- **root-server.net**
- **edu**
- **gmu**
- **Local NS (e.g., 192.168.1.1)**
- **www.ic.ac.uk?**

- **dns1.nic.uk**
- **ns0.ja.net**
- **ns0.ic.ac.uk**
- **uk**
- **ac**
- **ic**

- **155.198.64.24**
- **146.179.40.24**
- **www**
Domain Name System - Resolution

- **edu**
  - **gmu**
    - *Local NS (e.g., 192.168.1.1)*
      - **www.ic.ac.uk?**
        - A: 155.198.64.24
        - A: 146.179.40.24

- **root-server.net**
  - **dns1.nic.uk**
    - **ns0.ja.net**
    - **ns0.ic.ac.uk**

- **uk**
  - **ac**
  - **ic**
  - **www**

- **155.198.64.24
146.179.40.24**
Domain Name System - (Recursive) Resolution
Domain Name System - Iterative Resolution

- **edu**
  - **gmu**
  - root-server.net
    - A: -
    - NS: dns1.nic.uk
  - Local NS (e.g., 192.168.1.1)
  - dns1.nic.uk
    - A: -
    - NS: ns0.ja.net
    - A: -
    - NS: ns0.ic.ac.uk
  - ns0.ja.net
    - A: 155.198.64.24
    - A: 146.179.40.24
  - ns0.ic.ac.uk
    - A: 155.198.64.24
    - A: 146.179.40.24
- **uk**
  - ac
  - ic
  - www
  - www.ic.ac.uk?
Domain Name System - Iterative Resolution

- root-server.net
  - A: -
  - NS: dns1.nic.uk

- gmu.edu
  - A: 155.198.64.24
  - A: 146.179.40.24
  - www.ic.ac.uk?

- dns1.nic.uk
  - A: -
  - NS: ns0.ja.net
  - NS: ns0.ic.ac.uk

- uk
  - A: 155.198.64.24
  - A: 146.179.40.24

- ac

- ic
  - 155.198.64.24
  - 146.179.40.24

- www
 Domain Name System - Example Query

```sh
> cat /etc/resolv.conf #which DNS server am I using
# Generated by resolvconf
search fios-router.home
nameserver 192.168.1.1
> dig @192.168.1.1 www.ic.ac.uk a #recursive query
...
> dig +norecurse @192.168.1.1 www.ic.ac.uk a
# same answer, why?
# what if I had tried this first?
```

https://xkcd.com/908/
Domain Name System - Caching

Some gamers steamed over alleged Valve anti-cheat DNS spying - CSO
https://www.csoonline.com/.../some-gamers-steamed-over-alleged-valve-anti-cheat-dn... • Feb 16, 2014 - Goes through all your DNS Cache entries (ipconfig/displaydns); Hashes each one with MD5; Reports back to VAC Servers. Valve is not the only company that uses an anti-cheat system, but it is perhaps one of the most highly regarded companies as countless millions of gamers have Steam.
Various ...

Reddit user claims Valve Anti-Cheat scans your DNS cache - PC ...
https://gamefaqs.gamespot.com/boards/916573-pc/68593356 • For PC on the PC, a GameFAQs message board topic titled "Reddit user claims Valve Anti-Cheat scans your DNS cache".

Valve Anti-Cheat seems to scan your DNS cache, but probably doesn ...
https://www.neogaf.com - Discussions - Gaming Discussion • Feb 16, 2014 - Trust is a critical part of a multiplayer game community - trust in the developer, trust in the system, and trust in the other players. Cheats are a negative sum game, where a minority benefits less than the majority is harmed. There are a bunch of different ways to attack a trust-based system including writing a ...

Report: Valve anti-cheat scans your DNS history - Player Attack
https://www.playerattack.com/news/.../report-valve-anti-cheat-scans-your-dns-history/ • Feb 17, 2014 - Even if you've never actively visited a cheat website, there may be traces of them in your DNS, and that's what VAC is reportedly now looking for. The news was first posted to the Counter-Strike: Global Offensive Reddit, explaining that VAC now: Goes through all your DNS Cache entries (ipconfig ...

IS IT OK THAT VAC SCANS YOUR DNS CACHE? :: VAC Discussion - Steam...
steamcommunity.com › Steam Forums › VAC Discussion • Feb 15, 2014 - 16 posts - 7 authors
Valve ANSWER THIS! http://www.grocks.net/2014/02/16/steams-vac-protection-now-scans-ans-transfers-dns-cache/ What is going on with this DNS spying? We all know that various anti-cheat programs do check your DNS, some of them don't really collect data though. It has been proven that VAC collects ...
There are a number of **kernel-level paid cheats** that relate to [this Reddit thread](https://www.reddit.com/r/gaming/comments/1y70ej/valve_vac_and_trust/). Cheat developers have a problem in getting cheaters to actually pay them for all the obvious reasons, so they start creating DRM and anti-cheat code for their cheats. These cheats **phone home to a DRM server that confirms that a cheater has actually paid** to use the cheat.

VAC checked for the presence of these cheats. If they were detected VAC then checked to see which cheat DRM server was being contacted. This second check was done by **looking for a partial match to those (non-web) cheat DRM servers in the DNS cache**. If found, then hashes of the matching DNS entries were sent to the VAC servers. The match was double checked on our servers and then that client was marked for a future ban. Less than a tenth of one percent of clients triggered the second check. 570 cheaters are being banned as a result.

Gabe Newell, Valve’s CEO
Conclusion

- Resolving names requires a large scale distributed system
  - Domain Name System (DNS)
  - Distributed between several entities and among all continents
  - World-wide scale
  - High availability
This work is licensed under a Creative Commons Attribution-ShareAlike license

- This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/4.0/
- You are free to:
  - Share — copy and redistribute the material in any medium or format
  - Adapt — remix, transform, and build upon the material
  - for any purpose, even commercially.
- Under the following terms:
  - Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
  - ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
  - No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.