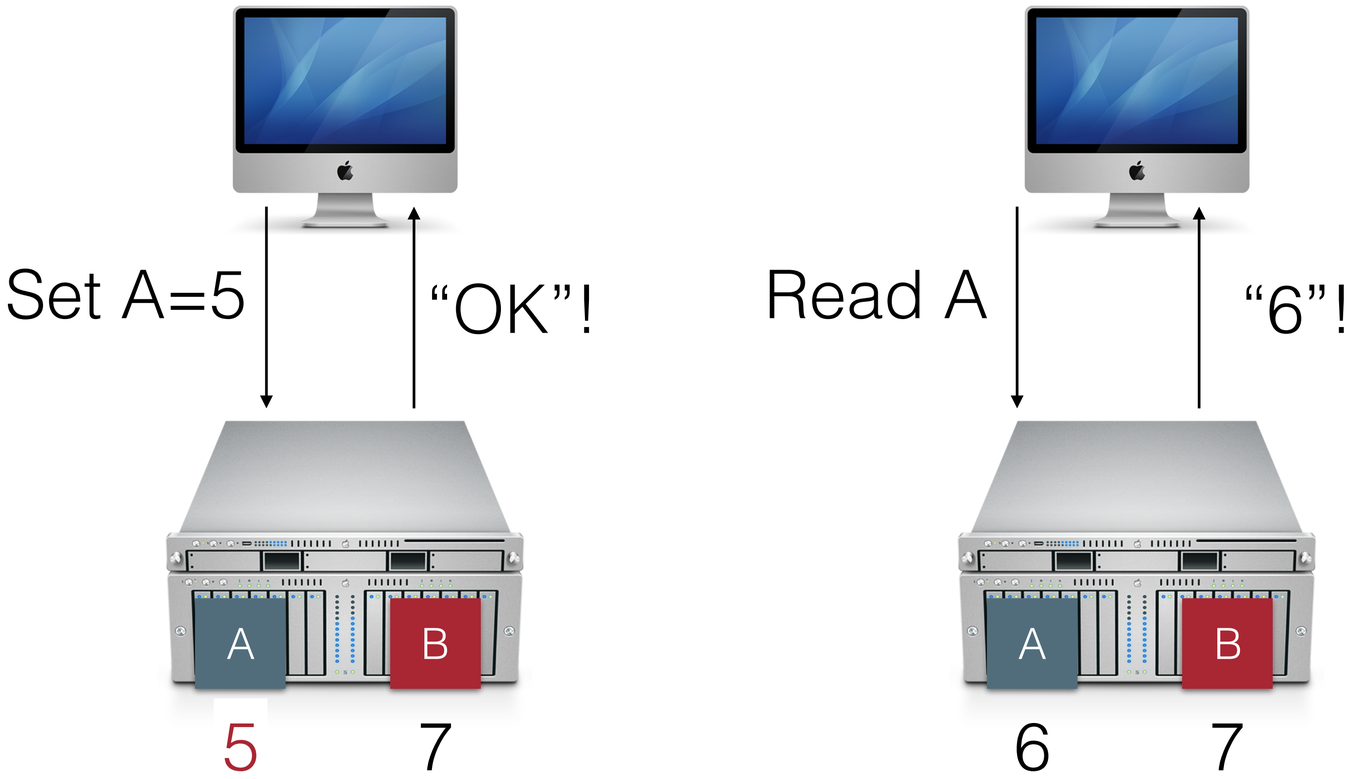
Naming & DNS

CS 475, Fall 2019
Concurrent & Distributed Systems

GEORGE MASON UNIVERSITY

Review: Recurring Problem: Replication

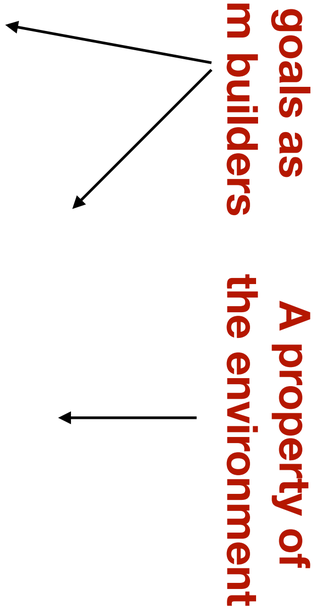
- Replication solves some problems, but creates a huge new one: consistency



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

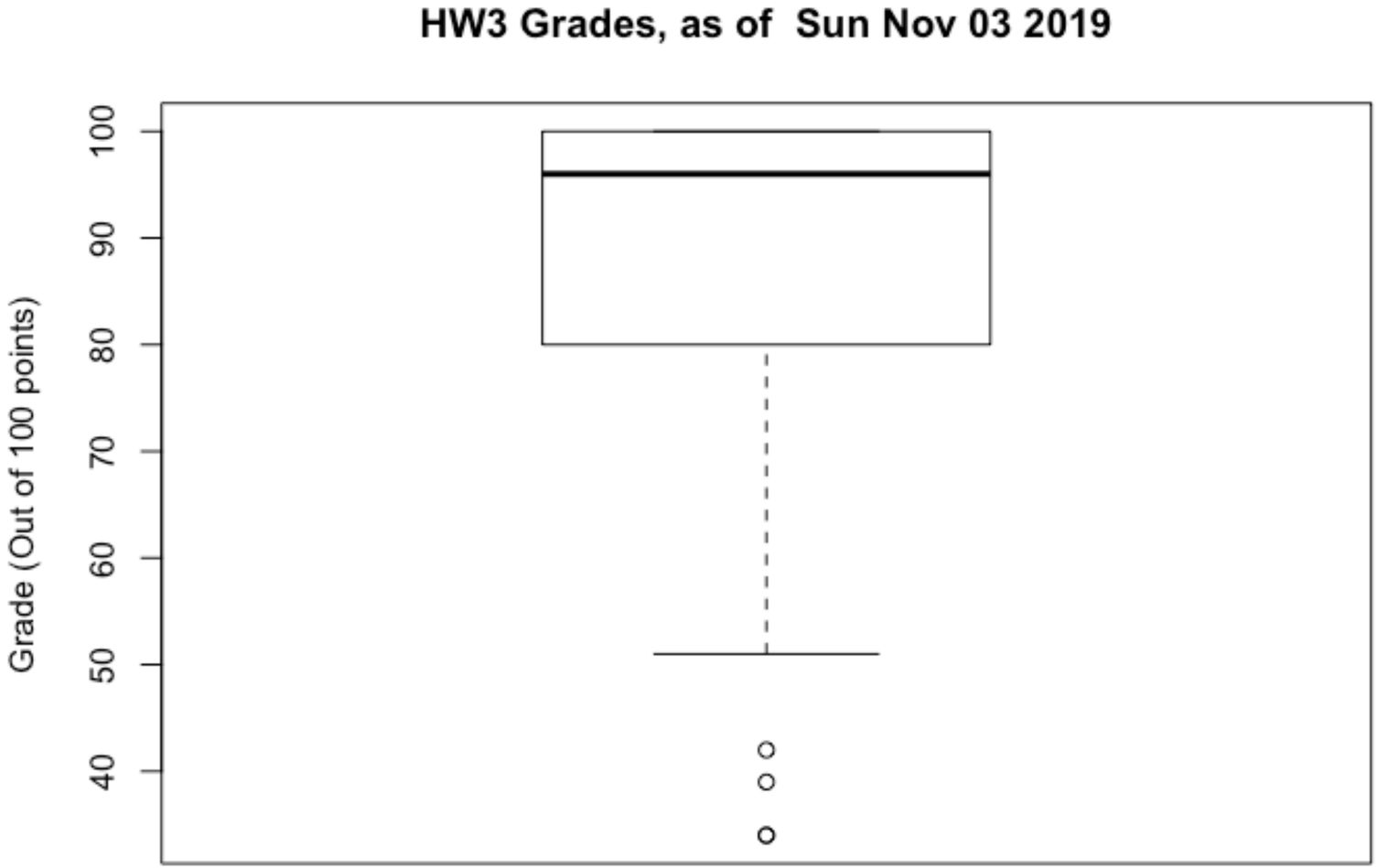
Example: Airline Reservations

- 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

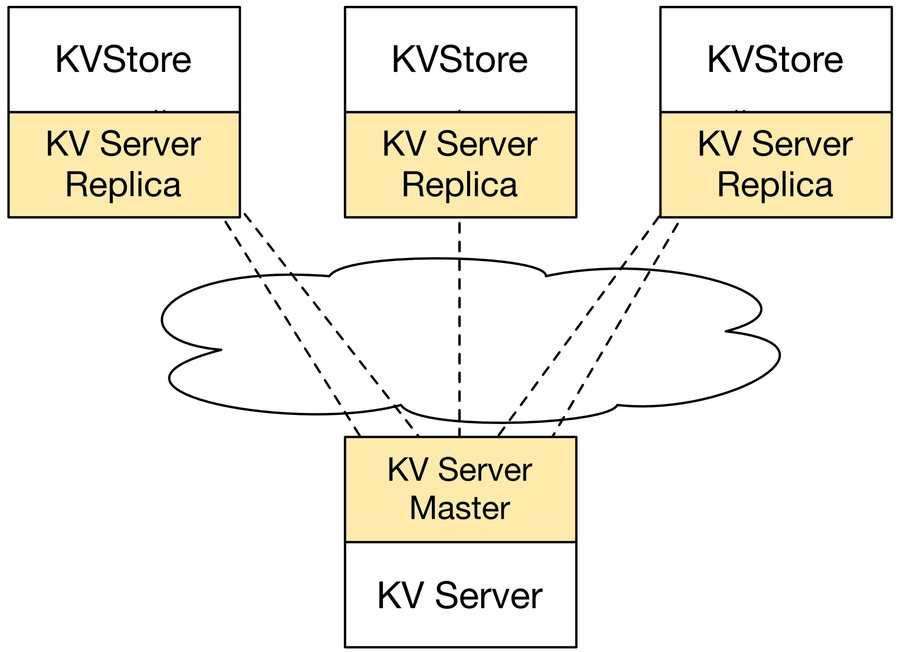
- If I have 100 seats for sale on a flight, does it matter if reservations for flights are reconciled immediately?
- If I have 5 seats for sale on a flight, does it matter if reservations are reconciled immediately?
- Result: Reservations can be made using either a strong consistency model or a weak, 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
Grade (Out of 100 points)
100
90
80
70
60
50
40

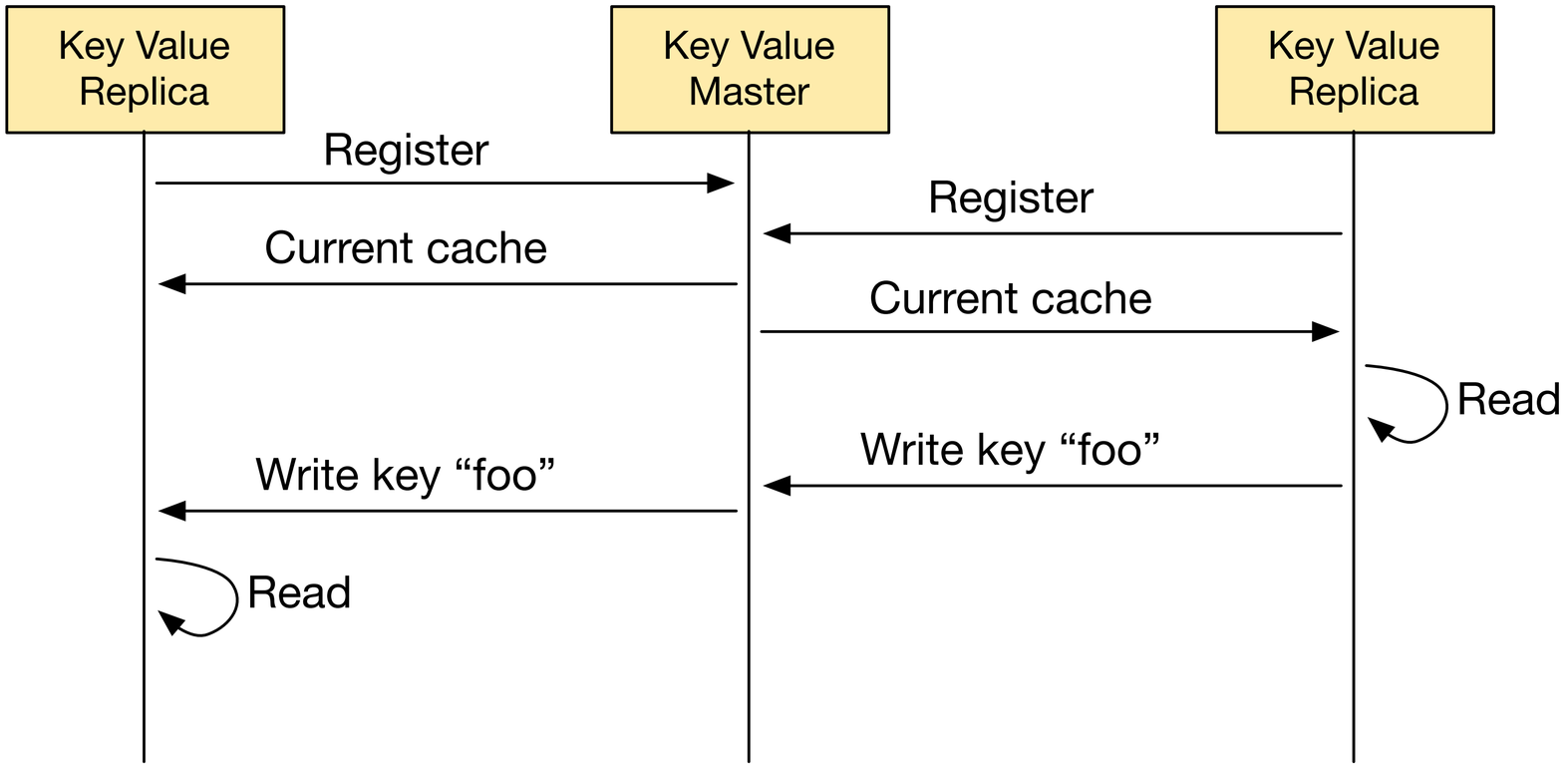
HW4 - Push-based replication

- Each KVStore client will have the entire dataset cached locally
- When updating values, the update will be propagated to each replica



HW4 - Push-based replication

- Each KVStore client will have the entire dataset cached locally
- When updating values, the update will be propagated to each replica



Ivy vs HW4

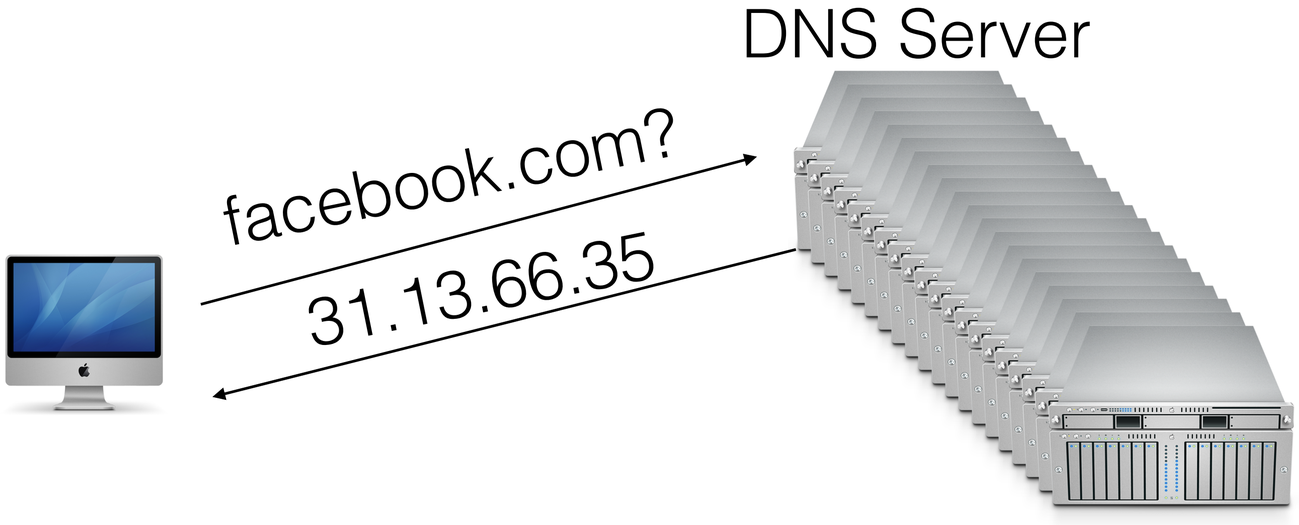
- Ivy never copies the actual values until a replica reads them (unlike HW4)
  - Invalidate messages are probably smaller than the actual data!
- Ivy only sends update (invalidate) messages to replicas who have a copy of the data (unlike HW4)
  - Maybe most data is not actively shared
- Ivy requires 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 :)

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

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+


your Tandy 1000s/3000/4000 or PC Compatible on a user-installable card. Includes manual and software for easy installation. The easy way to get hard disk power for your computer system. **25-4059** **799.00**

Add an External Hard Disk Drive

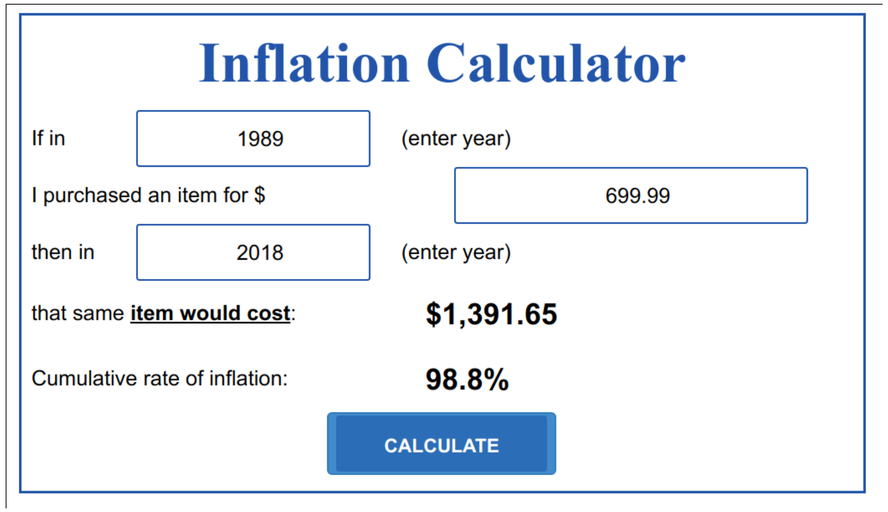
699.00 **Low As $40 Per Month***

- **Alternative Expansion Option**
- **Allows Greater Data Storage**

20-Megabyte External Hard Disk Drive. (Cable Kit and installation required for secondary unit.) Requires Hard Disk Controller Board (25-1007).
25-1041 . . . . . **699.00**

Hard Disk Controller Board. For Tandy 1000 SX/SL and original Tandy 1000 only. Allows you to add hard disk drives for up to 40 million characters of storage. Includes cable for use with 10 or 20-megabyte hard disks.
25-1007 . . . . . **299.95**

180 **PRICES APPLY AT PARTICIPATING RA**

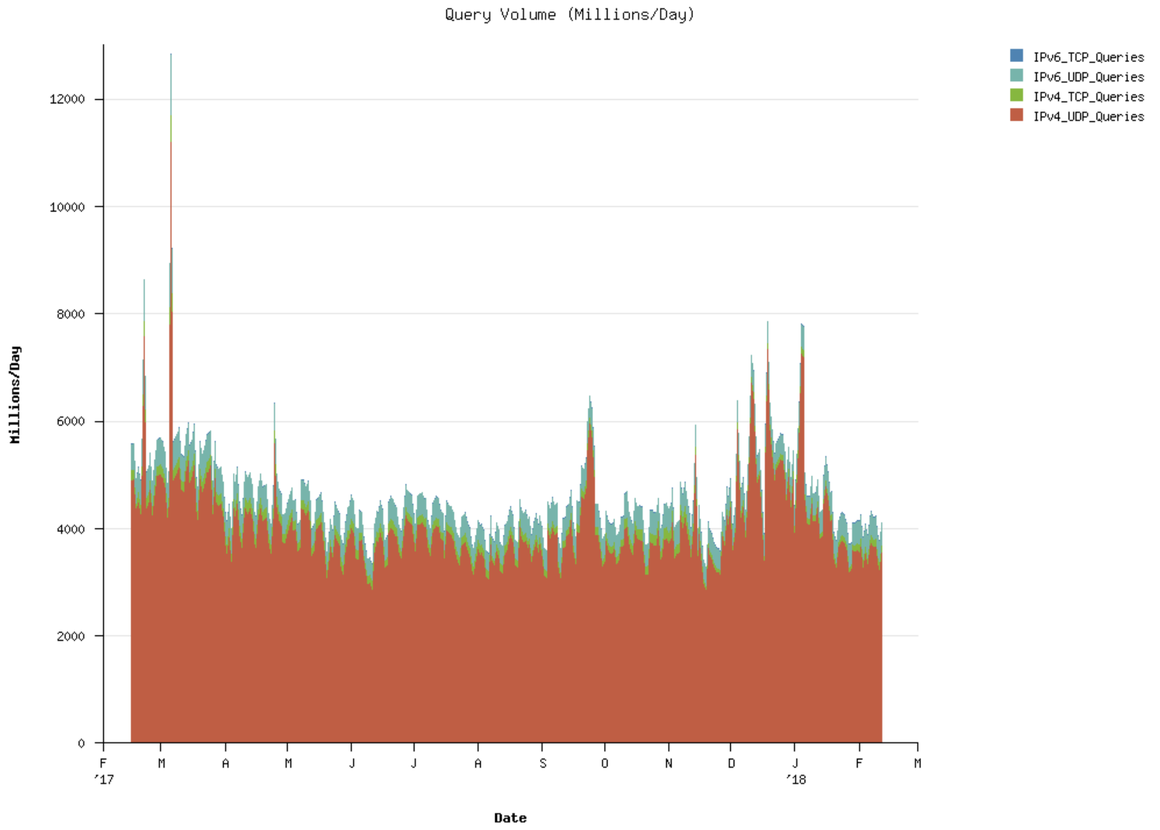


Domain Name System

- 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!**



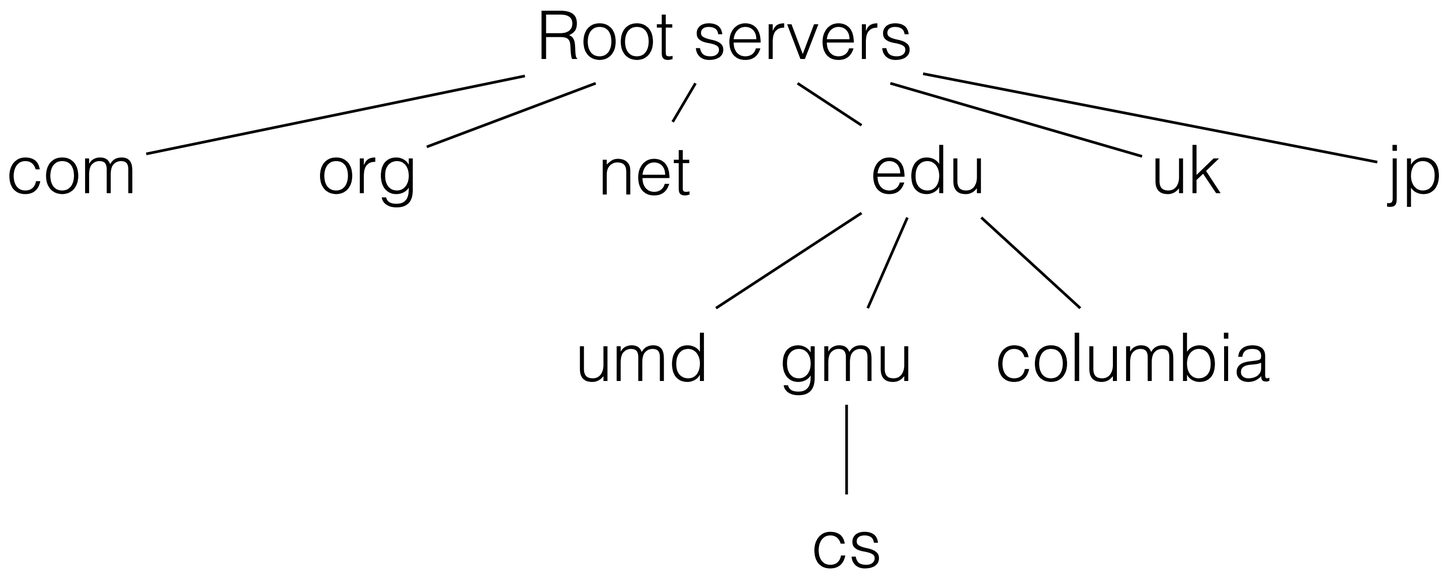
http://a.root-servers.org/static/index.html

Domain Name System

- Goals
  - Scalable
  - Robust
    - High availability
  - Decentralized maintenance
  - Global scope
    - Names mean the same thing everywhere
- Non-goals
  - Atomicity
  - Strong consistency

DNS

Idea: break apart responsibility for each part of a domain name (**zone**) to a different group of servers

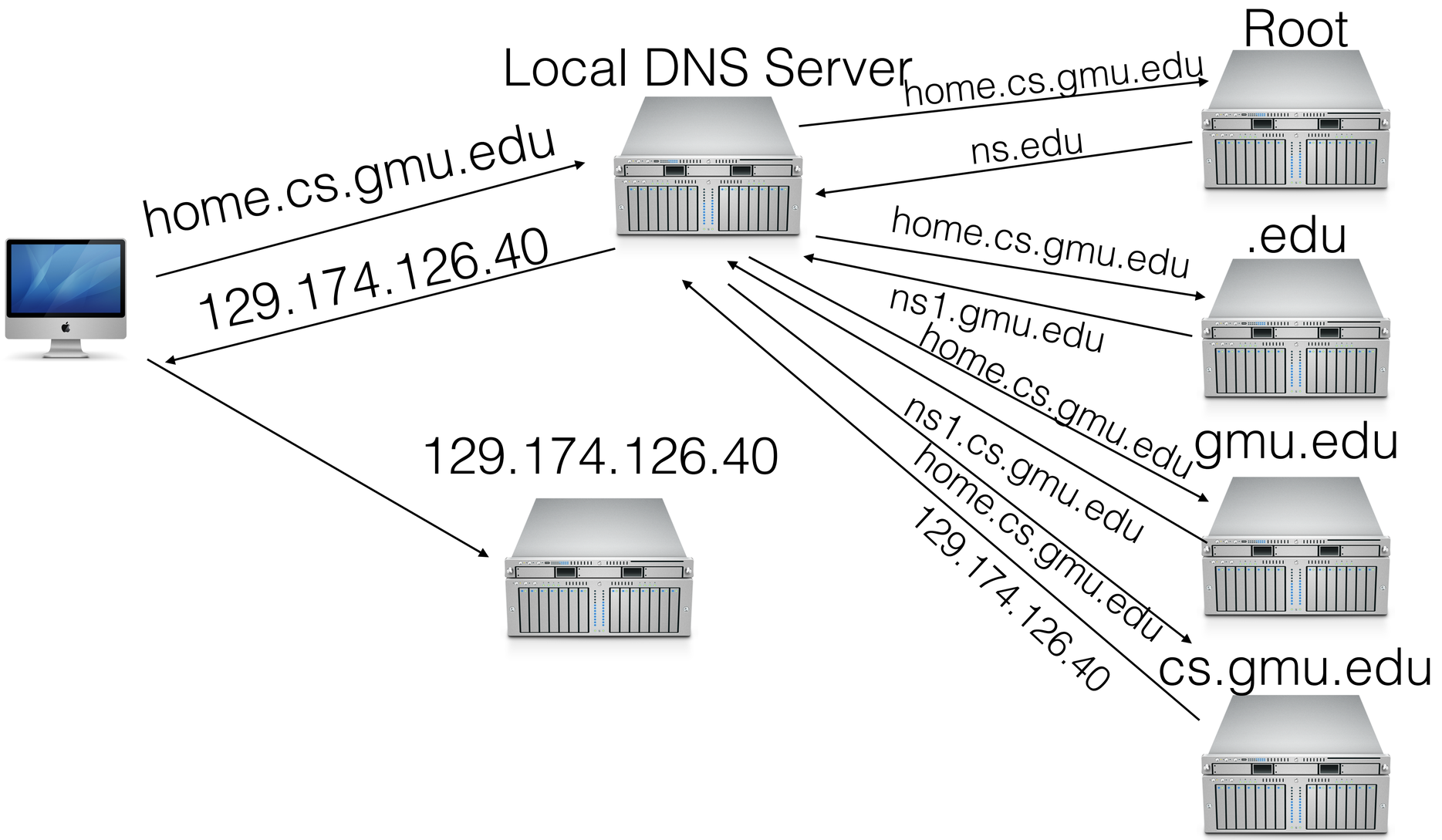


Each zone is a continuous section of name space
Each zone has an associate set of name servers

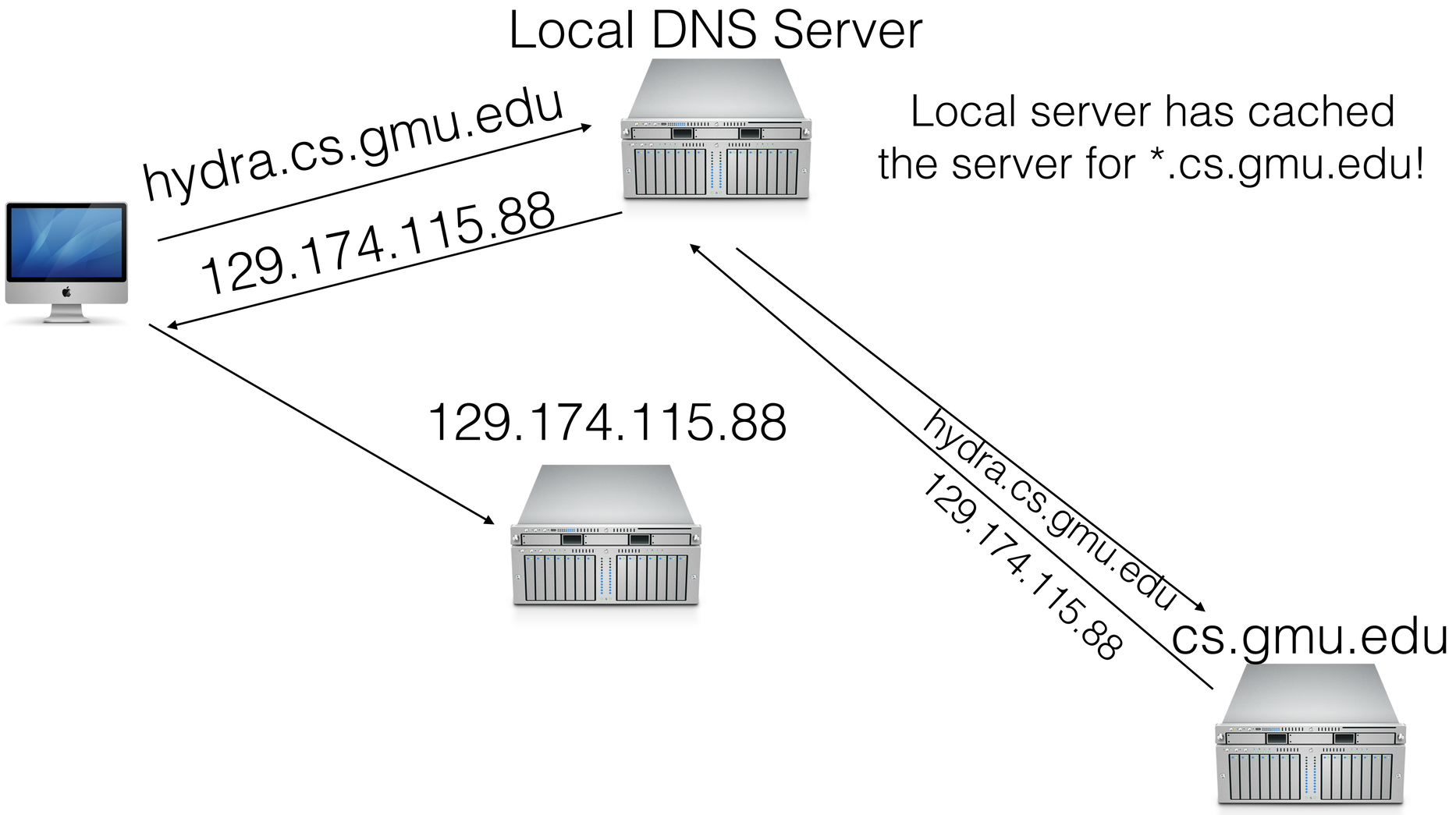
DNS

- 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

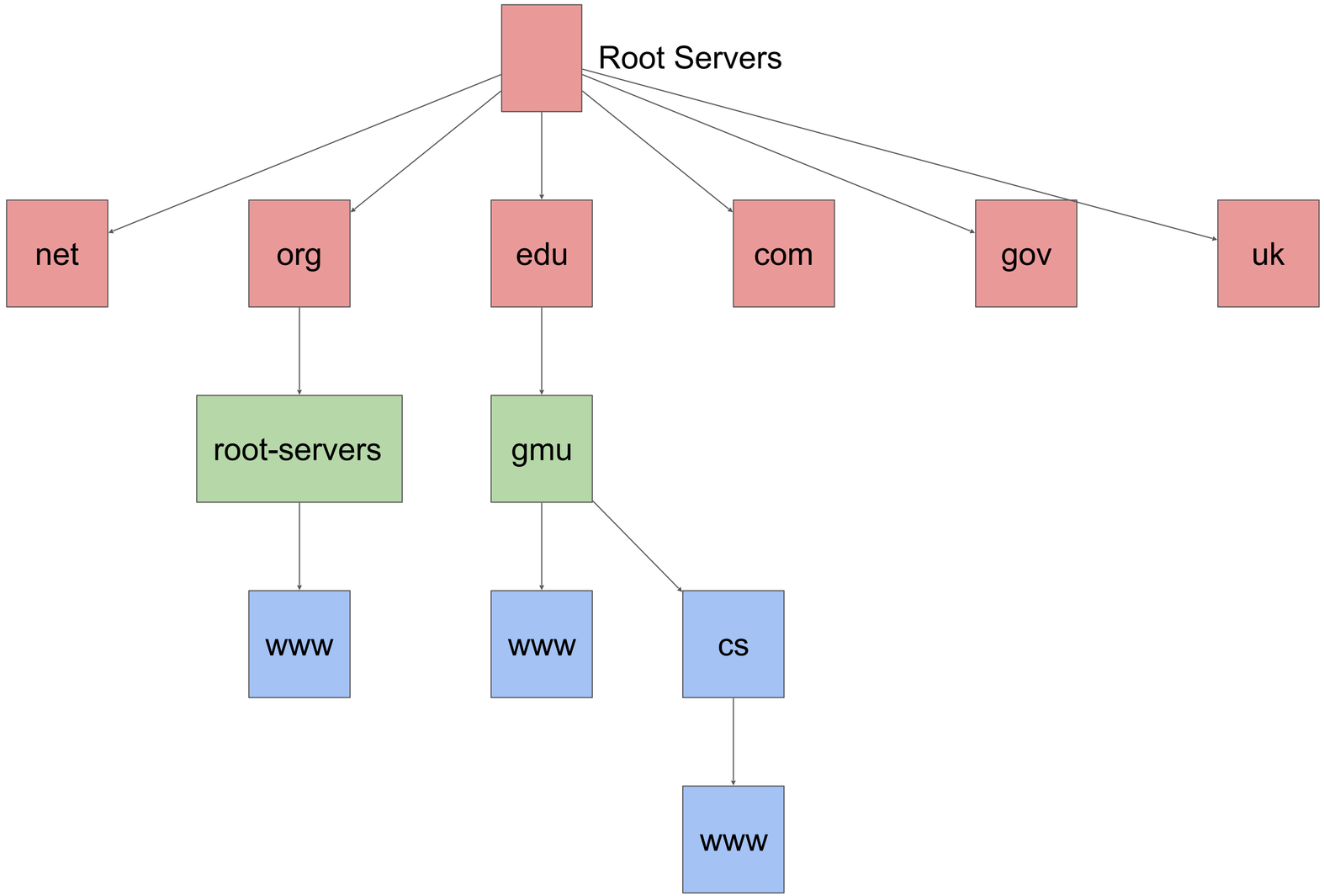
DNS: Example


Root
Local DNS Server
home.cs.gmu.edu
ns.edu
home.cs.gmu.edu
129.174.126.40
.edu
home.cs.gmu.edu
ns1.gmu.edu
home.cs.gmu.edu
gmu.edu
ns1.cs.gmu.edu
home.cs.gmu.edu
129.174.126.40
cs.gmu.edu
129.174.126.40

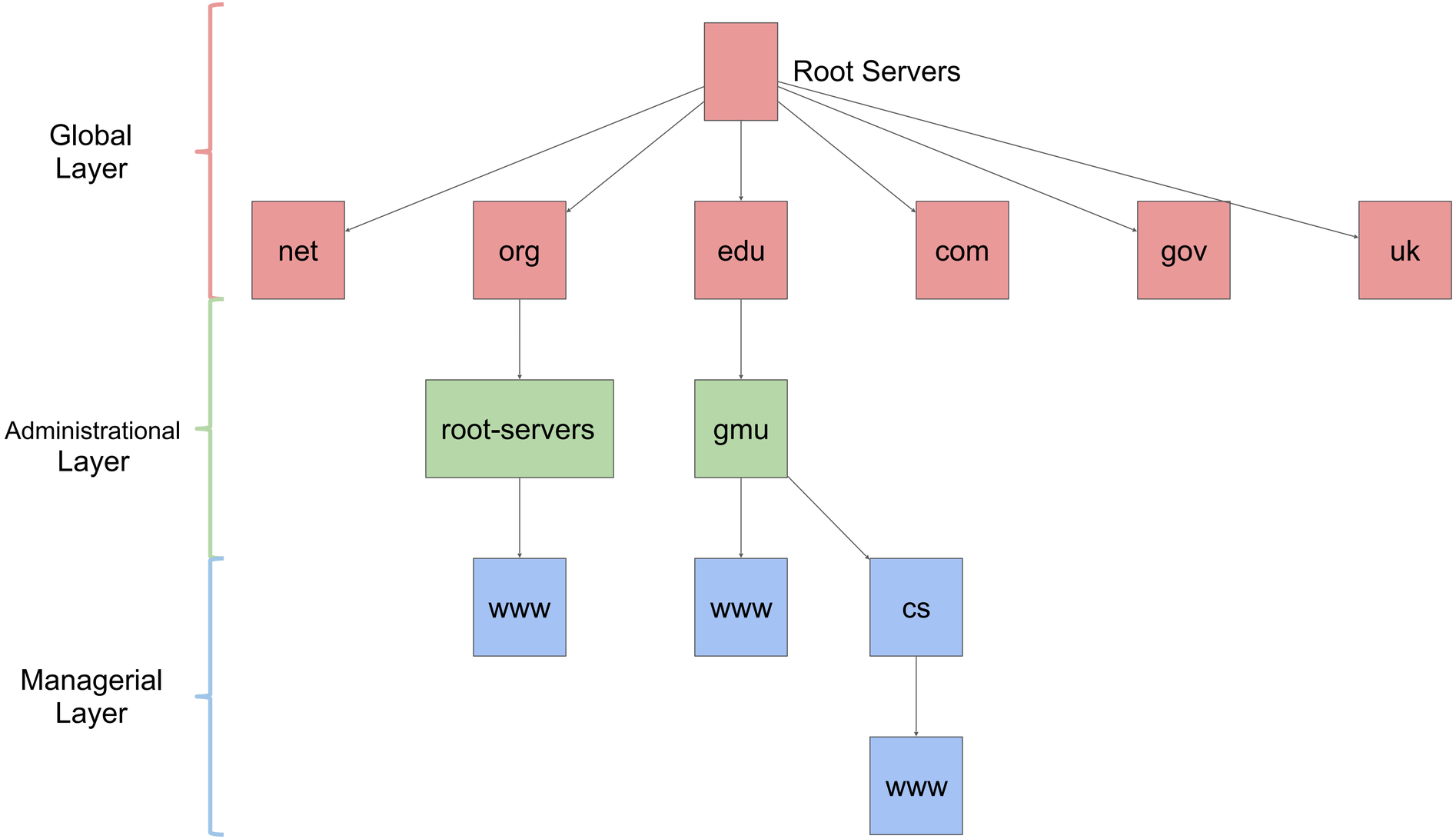
DNS: Example


Local DNS Server
hydra.cs.gmu.edu
129.174.115.88
Local server has cached
the server for *.cs.gmu.edu!
129.174.115.88
hydra.cs.gmu.edu
129.174.115.88
cs.gmu.edu

Domain Name System



Domain Name System



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
  - e) Thousands
  - f) Millions

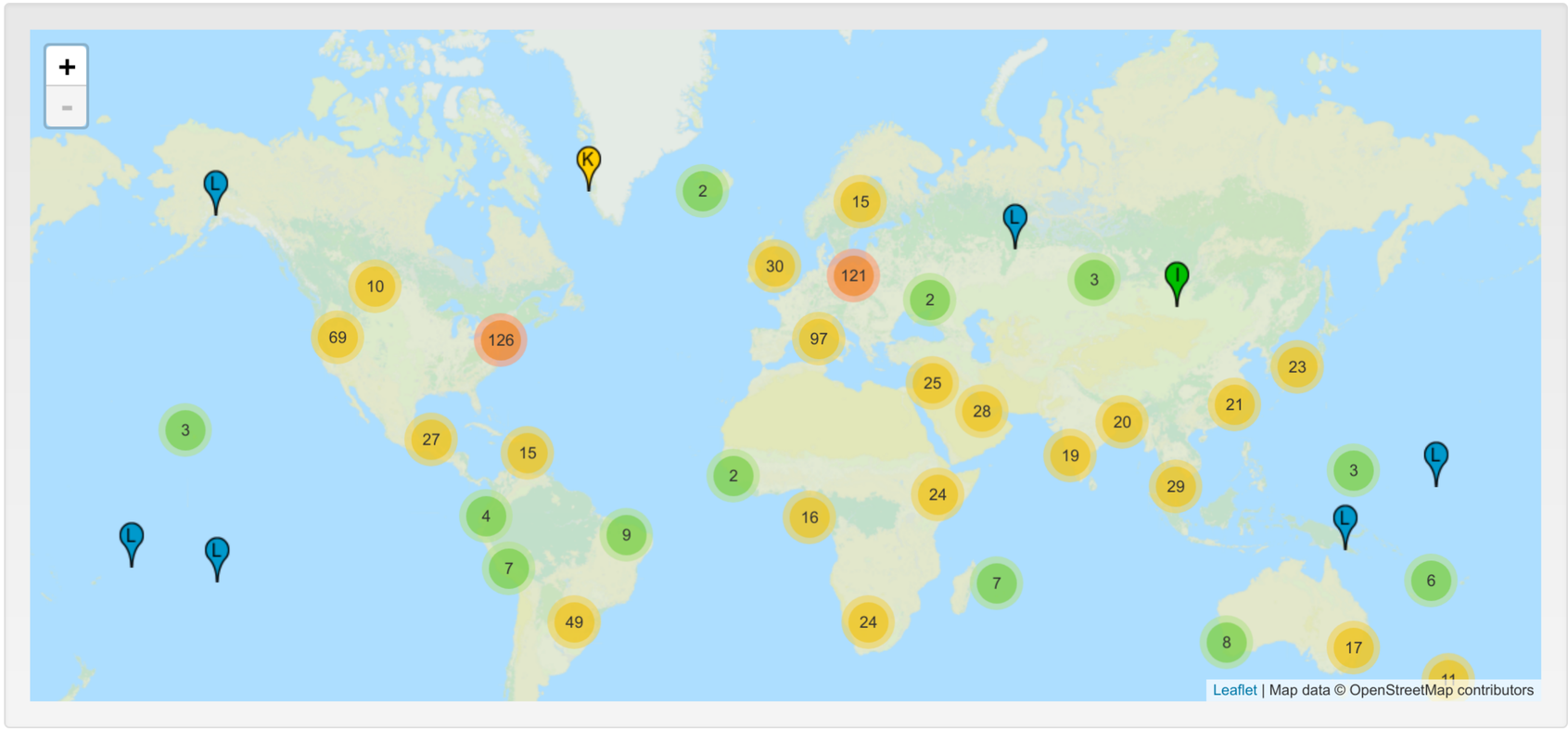
Verisign, Inc.	`a`
Information Sciences Institute	`b`
Cogent Communications	`c`
University of Maryland	`d`
NASA Ames Research Center	`e`
Internet Systems Consortium, Inc.	`f`
U.S. DOD Network Information Center	`g`
U.S. Army Research Lab	`h`
Netnod	`i`
Verisign, Inc.	`j`
RIPE NCC	`k`
ICANN	`l`
WIDE Project	`m`

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~~

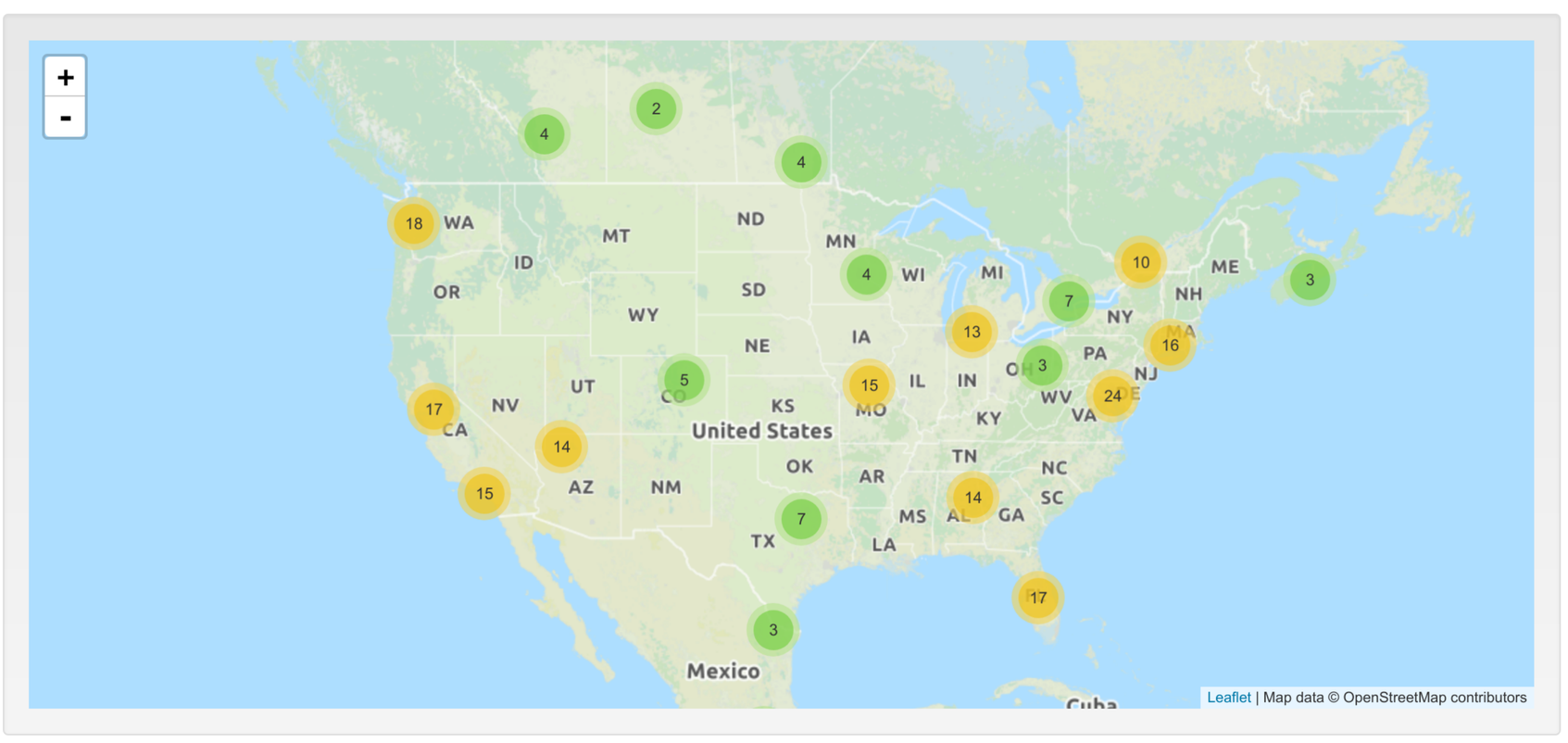
Verisign, Inc.	a	5
Information Sciences Institute	b	2
Cogent Communications	c	10
University of Maryland	d	128
NASA Ames Research Center	e	191
Internet Systems Consortium, Inc.	f	193
U.S. DOD Network Information Center	g	6
U.S. Army Research Lab	h	2
Netnod	i	60
Verisign, Inc.	j	159
RIPE NCC	k	58
ICANN	l	157
WIDE Project	m	9

Domain Name System - Root servers



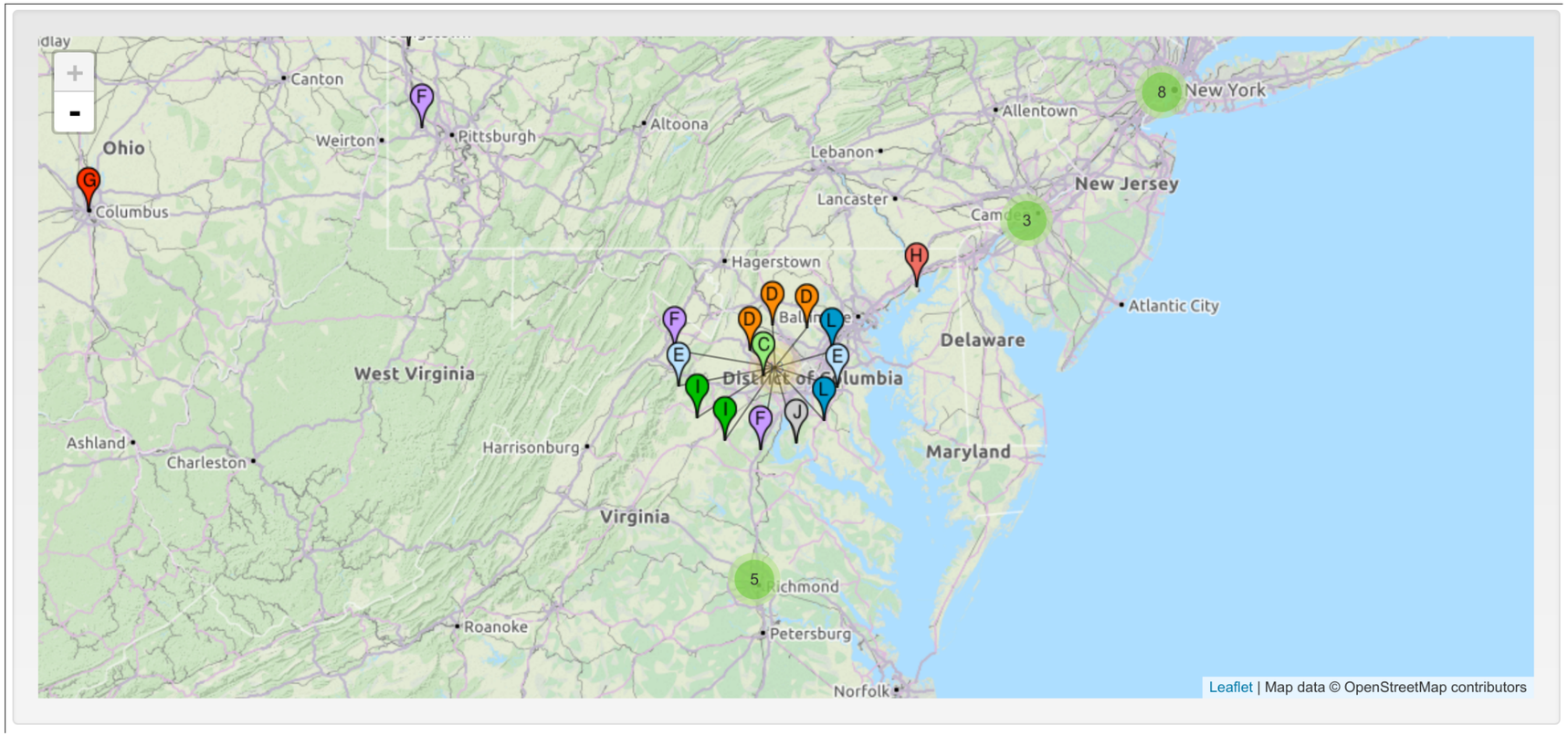
www.root-servers.net

Domain Name System - Root servers



www.root-servers.net

Domain Name System - Root servers

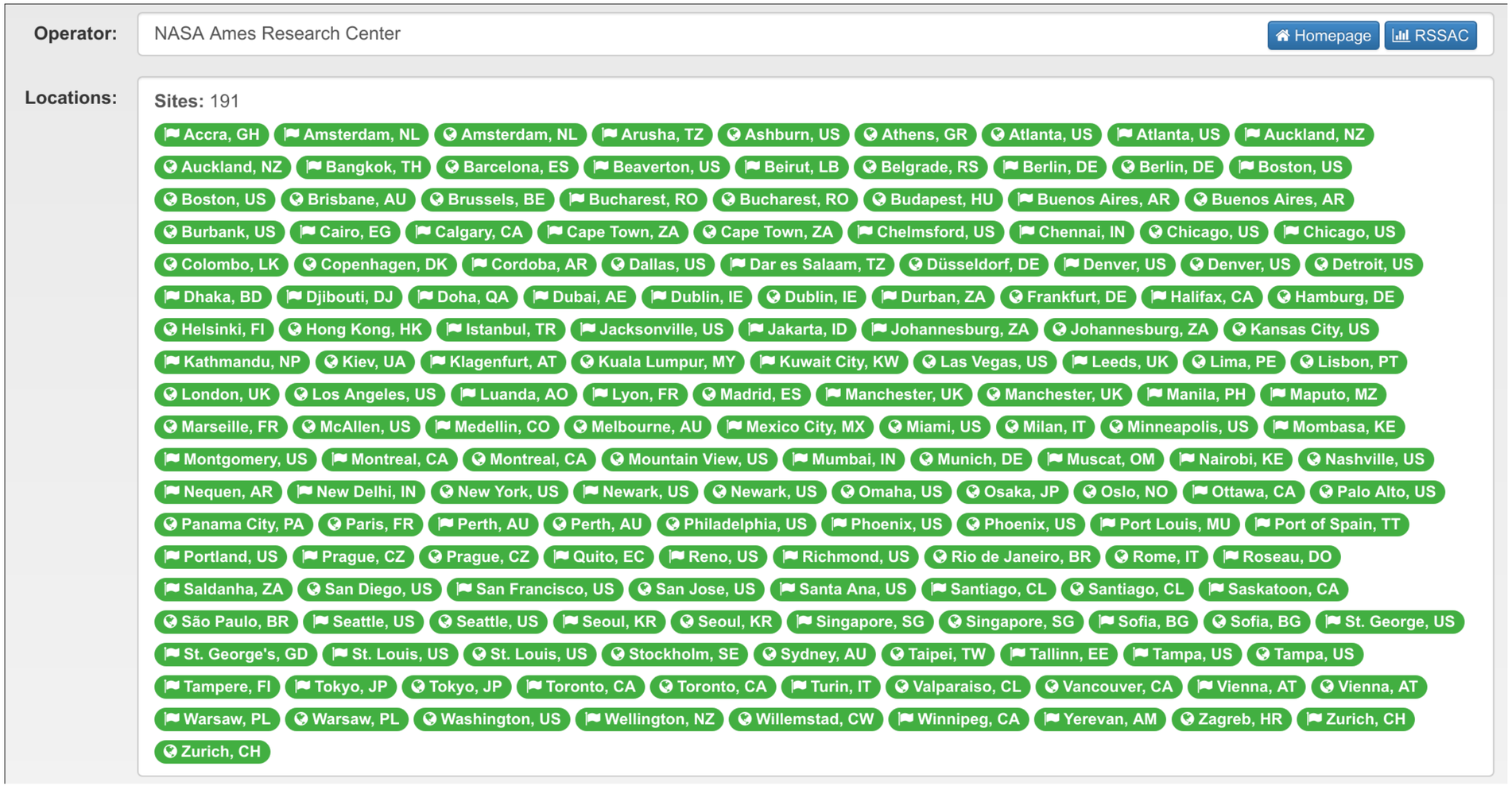


www.root-servers.net

Domain Name System - Root servers

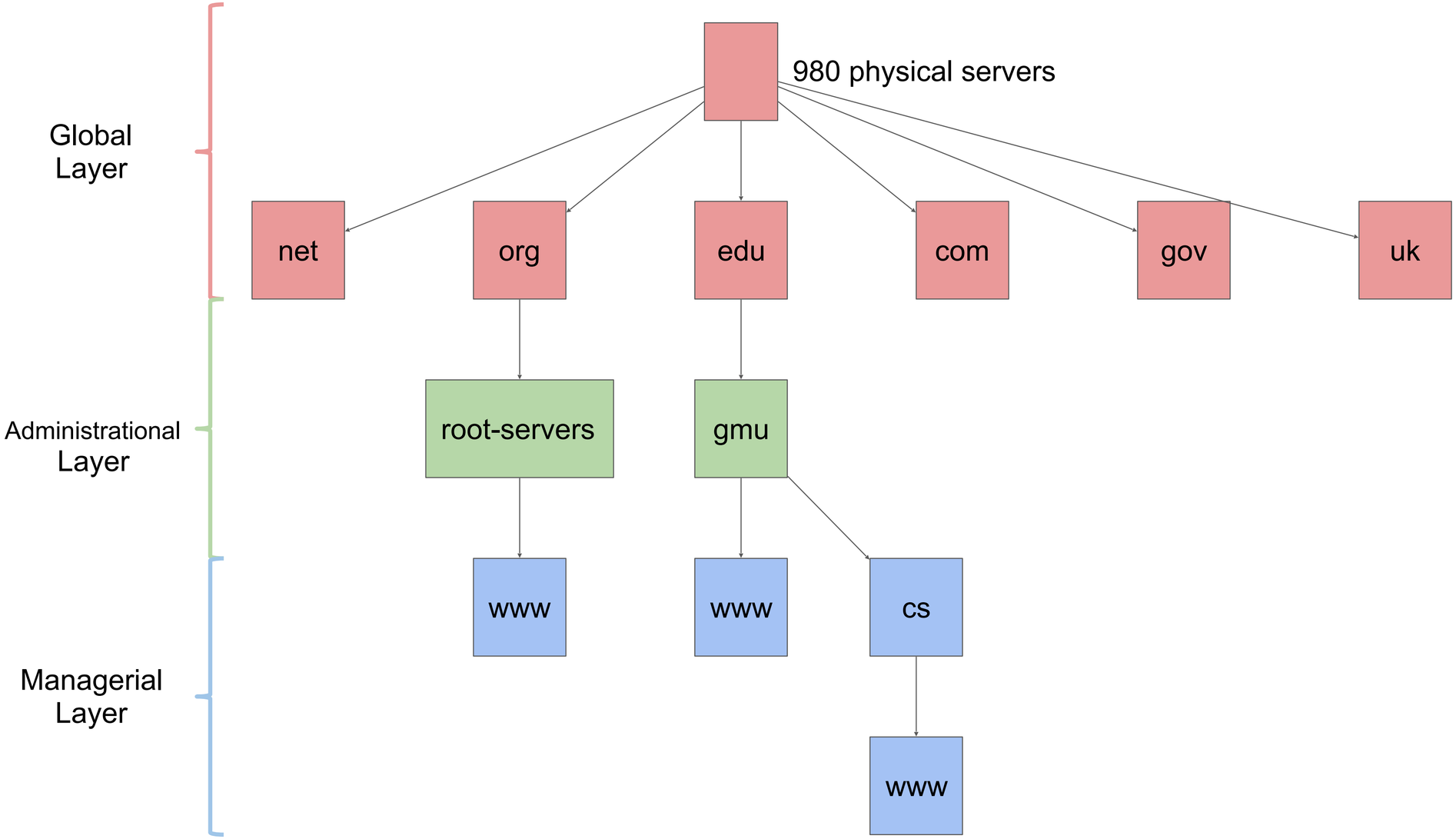
www.root-servers.net

Domain Name System - Root servers

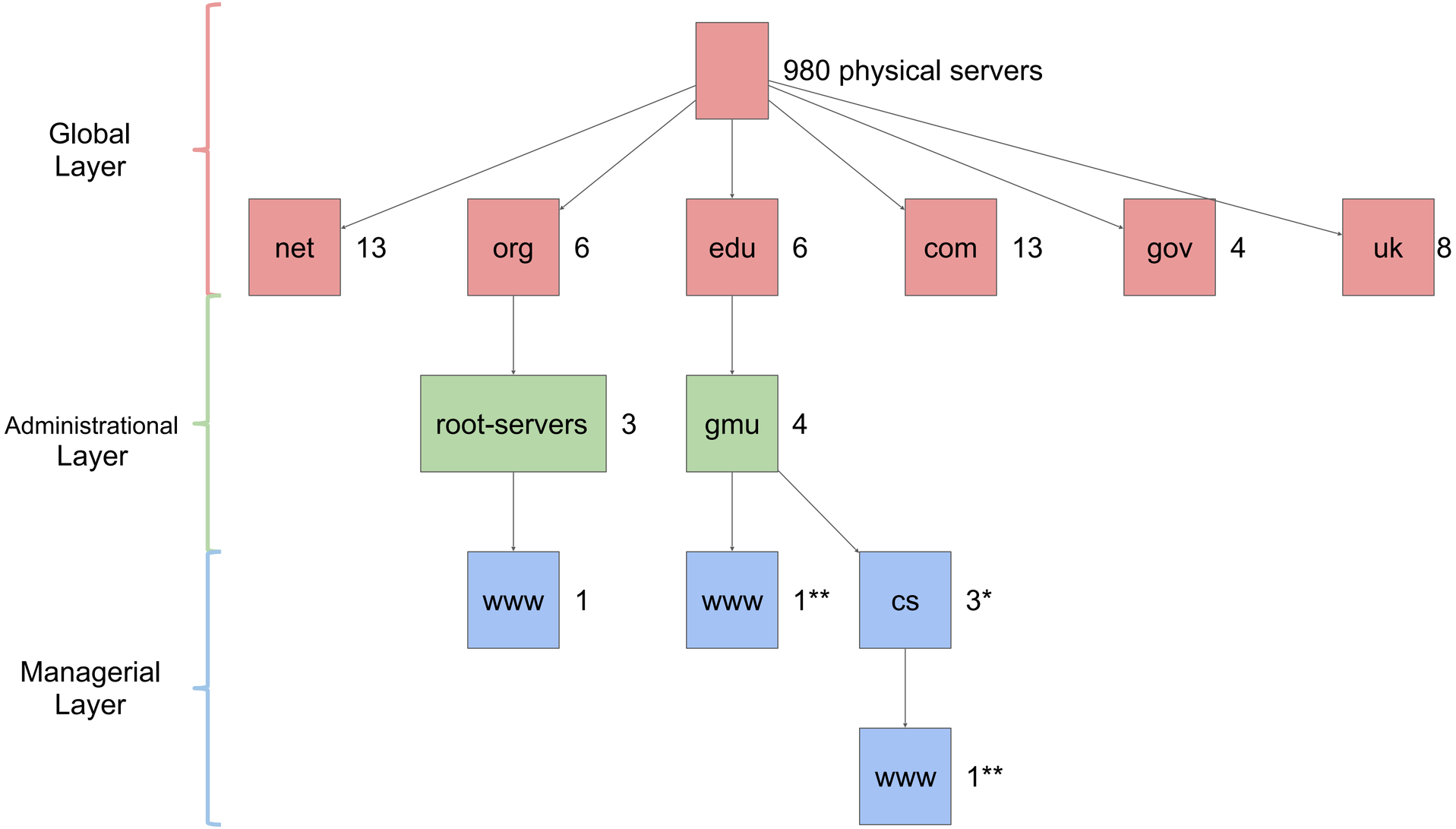


www.root-servers.net

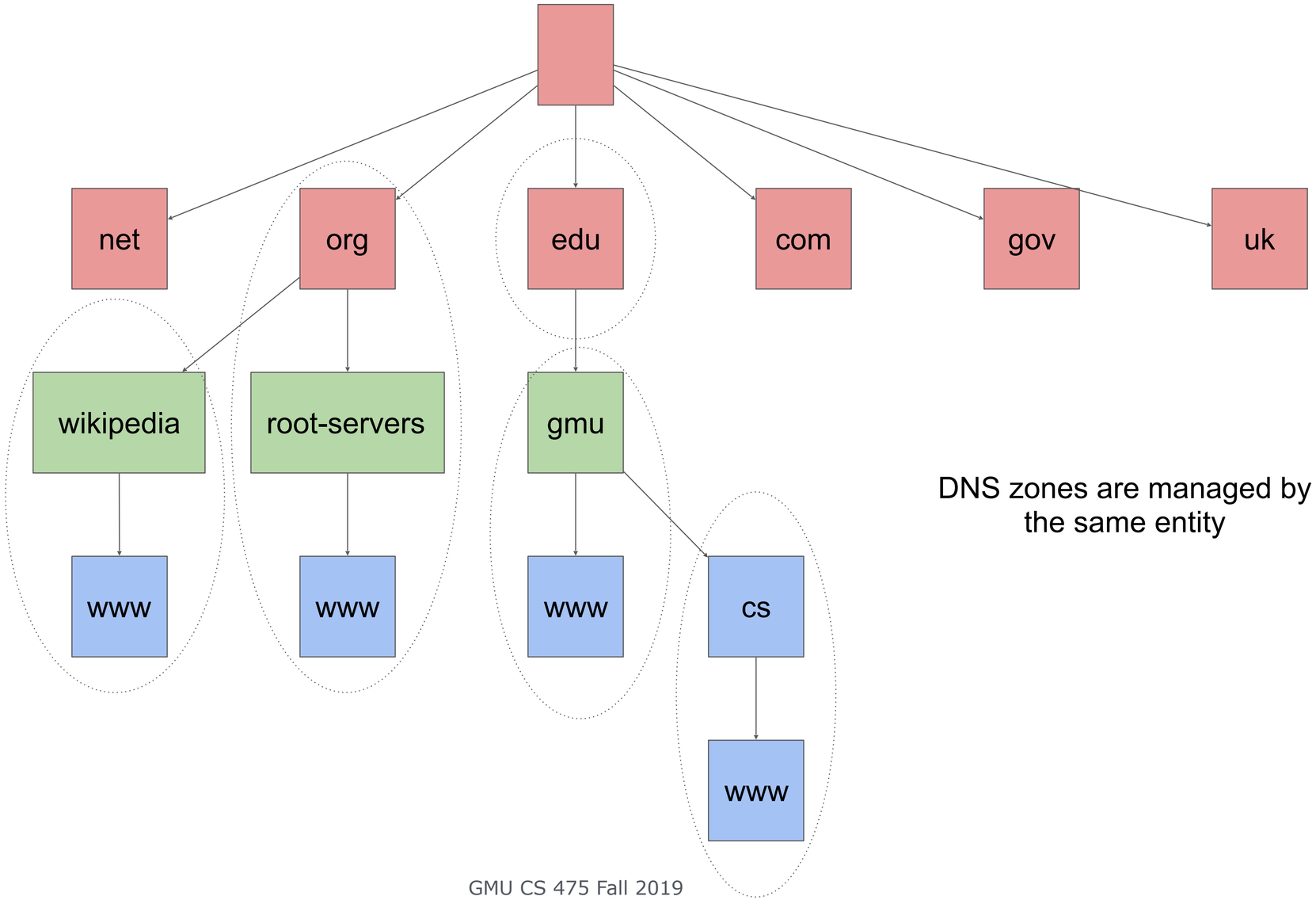
Domain Name System - Scale



Domain Name System - Scale



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...]

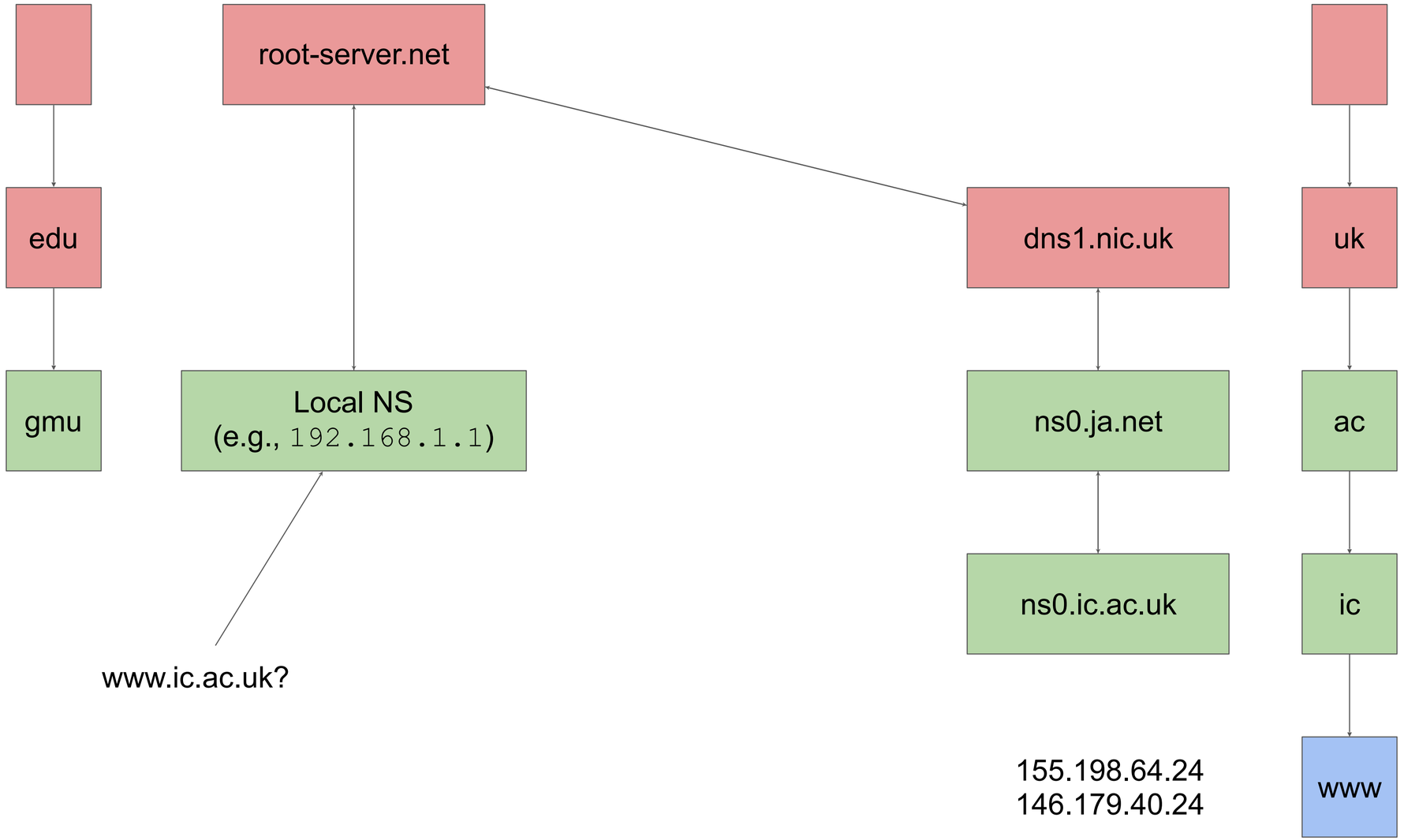
Domain Name System - Example Query

```
> 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:
www.gmu.edu.            77455   IN      CNAME   jiju3.gmu.edu.
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
```

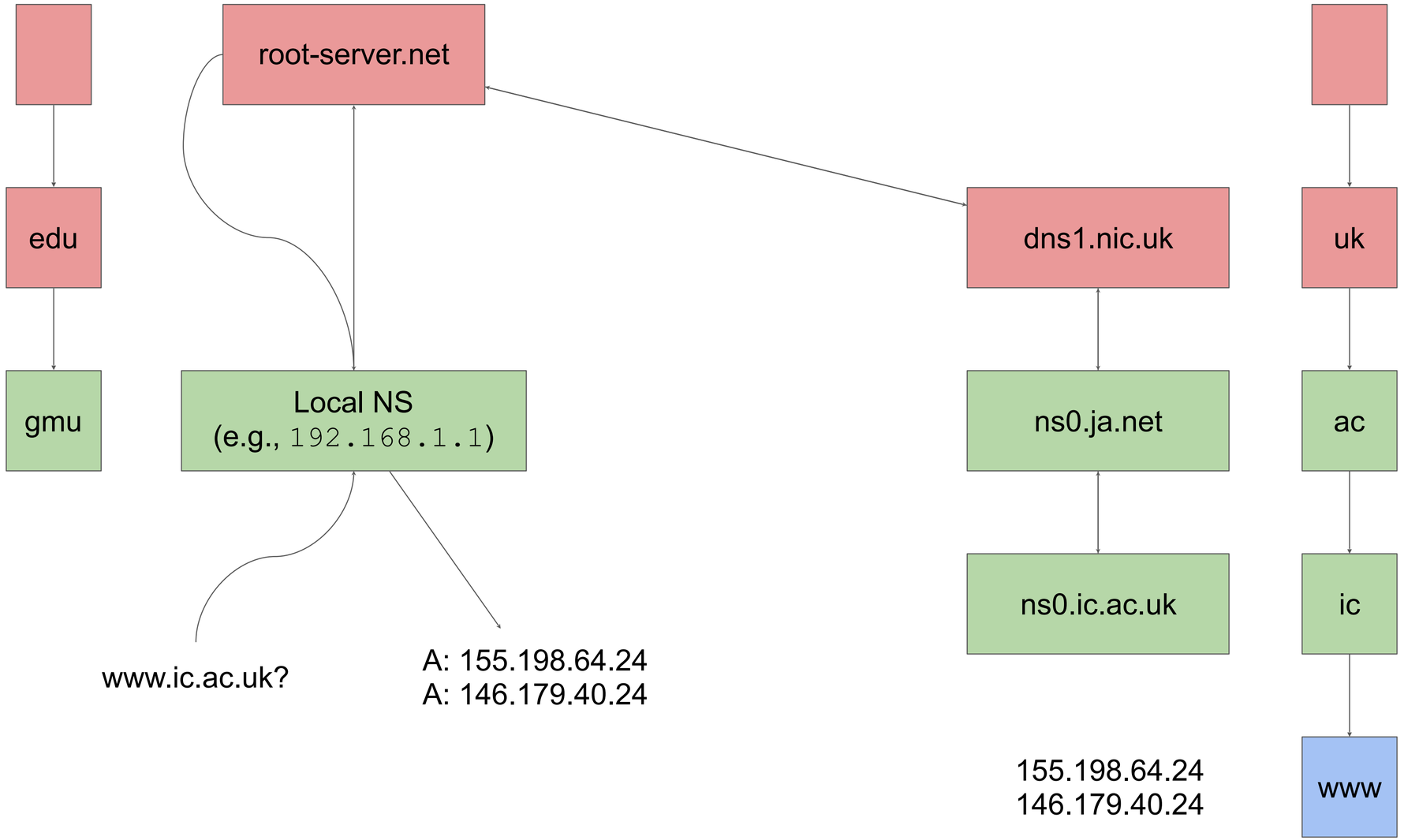
Domain Name System - Example Query

```
> 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
```

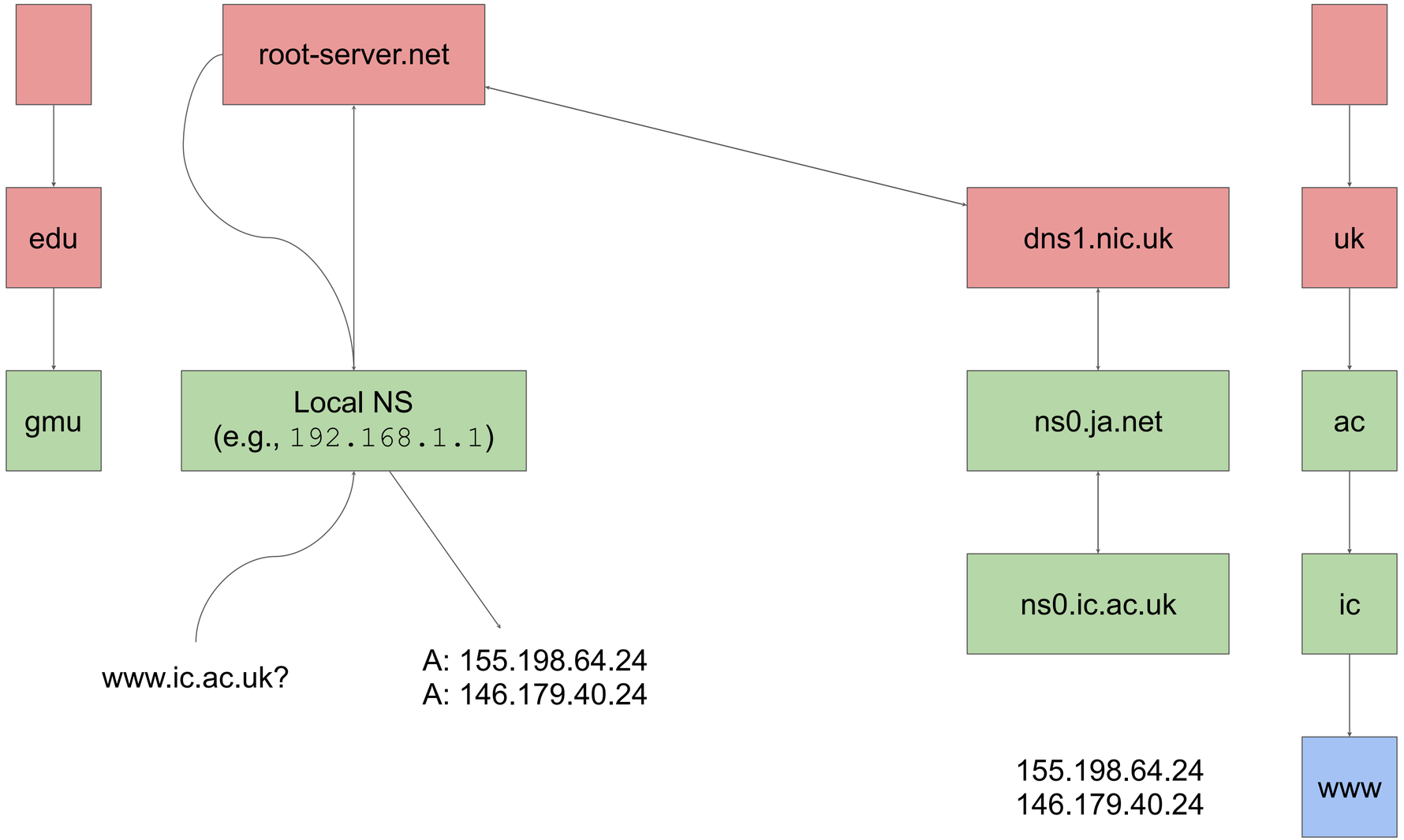
Domain Name System - Resolution



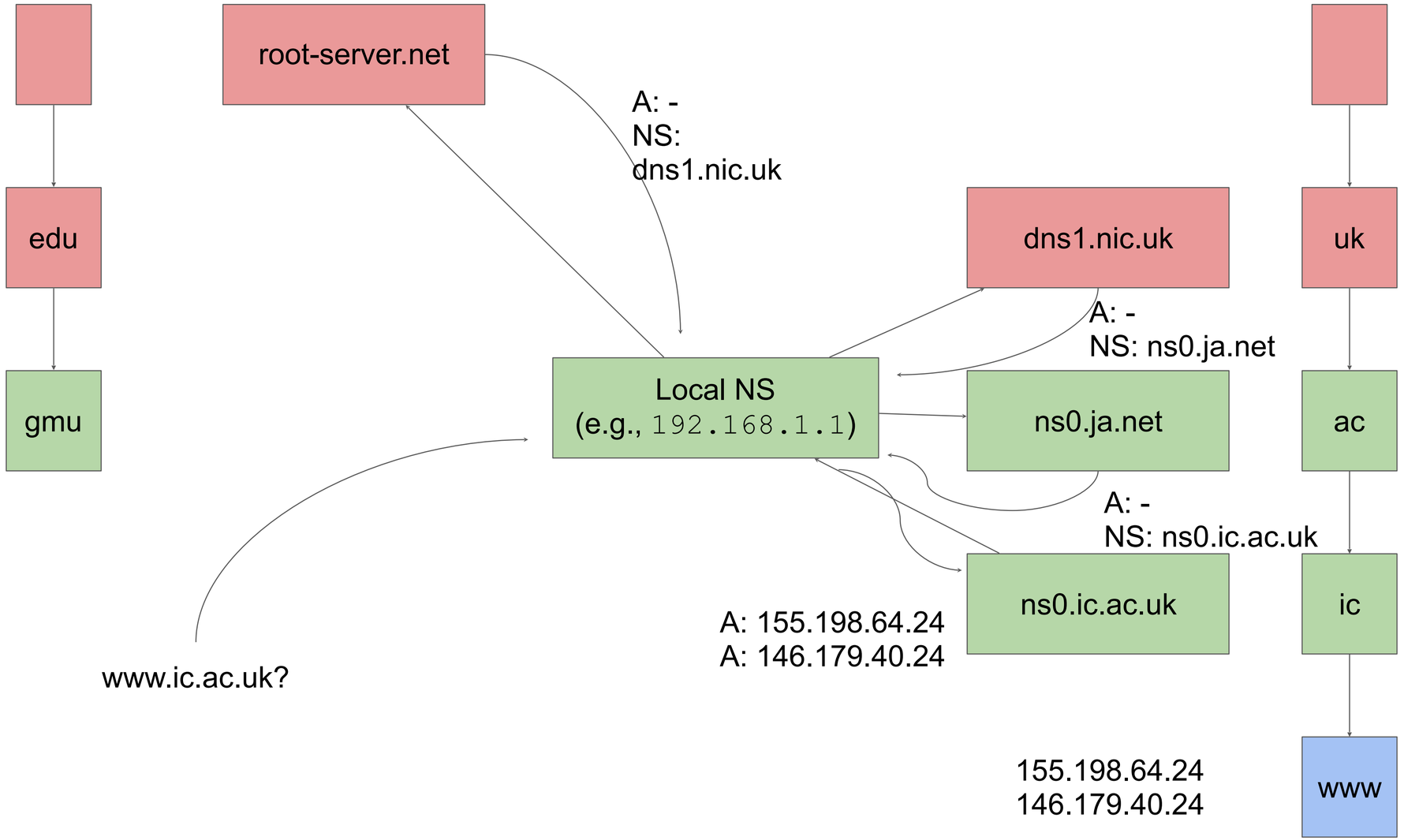
Domain Name System - Resolution



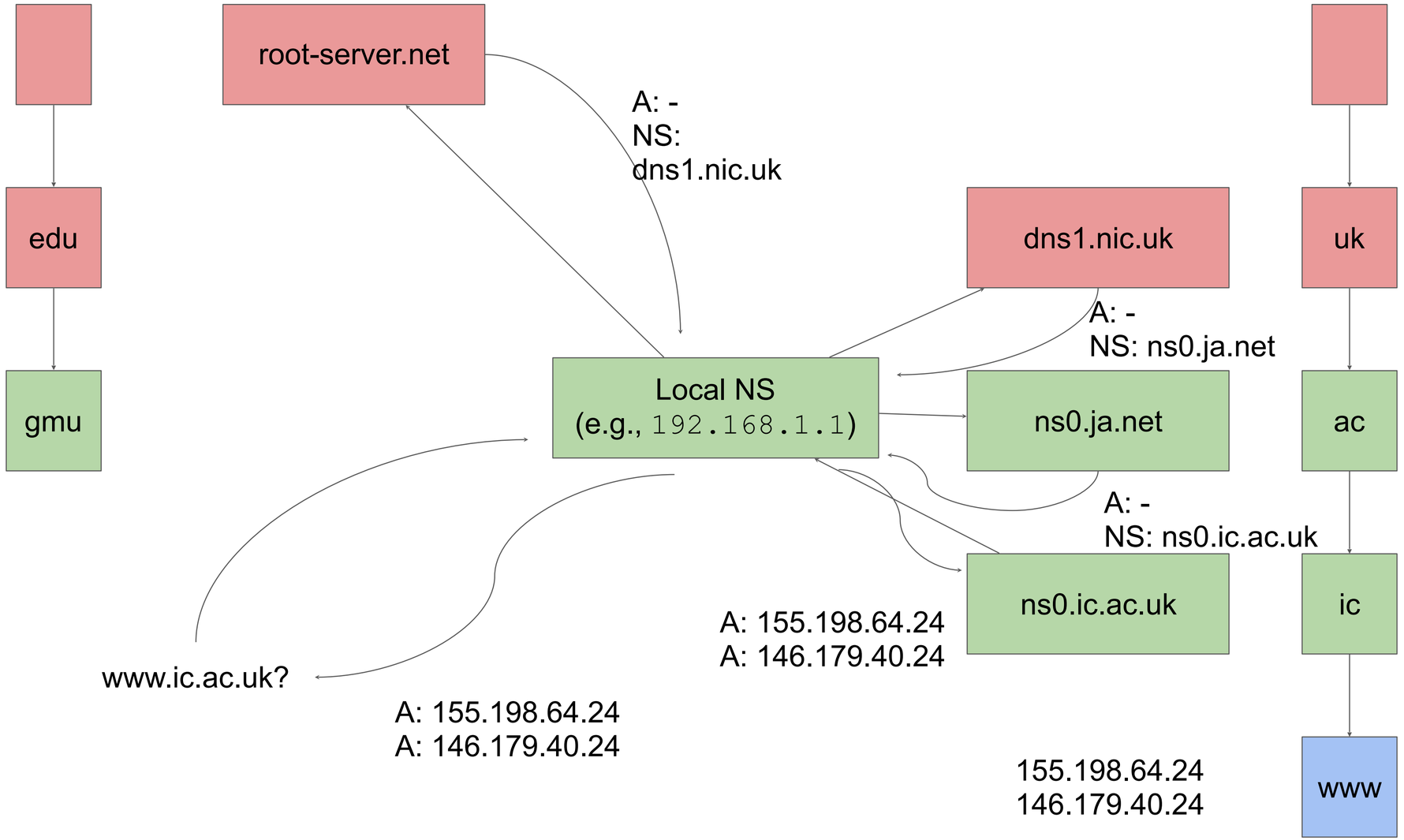
Domain Name System - (Recursive) Resolution



Domain Name System - Iterative Resolution

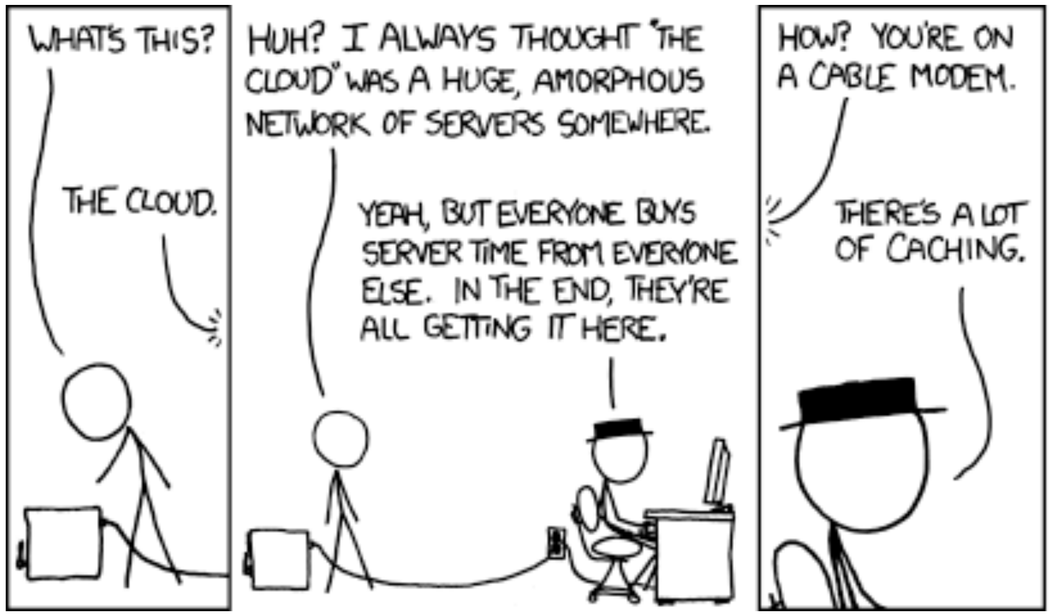


Domain Name System - Iterative Resolution



Domain Name System - Example Query

```
> 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/916373-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.ghacks.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 ...

Domain Name System - Caching

There are a number of **kernel-level paid cheats** that relate to this Reddit thread. 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

https://www.reddit.com/r/gaming/comments/1y70ej/valve_vac_and_trust/

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