

DNS and Modern Network Services

Amin Vahdat
CSE 123b
April 27, 2006

Announcements

- Midterm: May 9
- Second assignment due May 15

Domain Name System

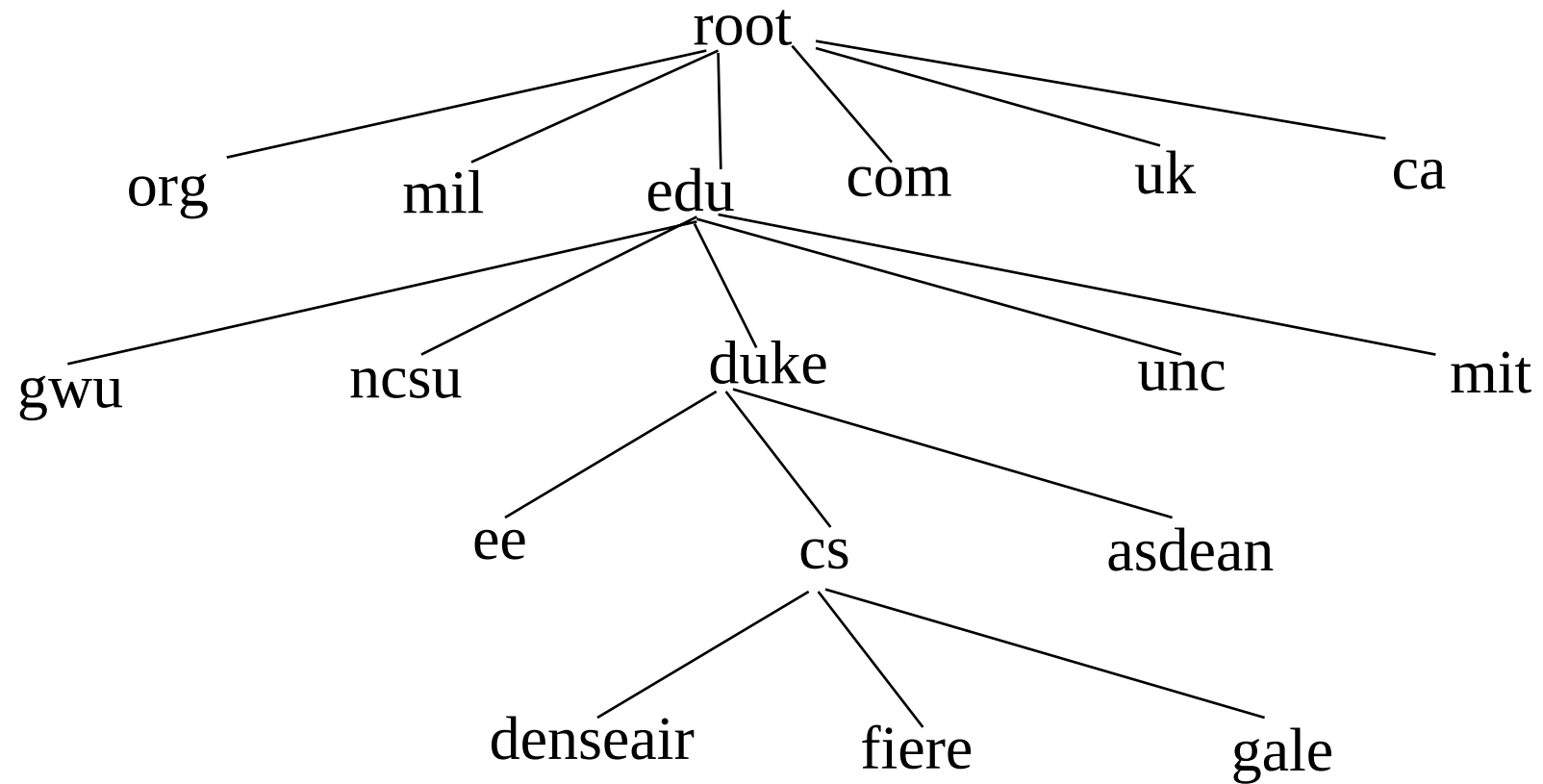
Motivation

- 1982: single hosts.txt file stored and distributed from a central site
- Contained all hostname to IP address mappings
- Centralized control did not fit with distributed management
- Number of hosts changed from number of timesharing systems to number of workstations
 - Organizations to users
 - Exponential resource usage for distributing the file

Domain Name System

- Hierarchical namespace with typed data
- Control delegated in hierarchical fashion
 - Convince node above you to delegate control
- Designed to be extensible w/support for new data types
- 1985: some hosts solely utilize DNS

Hierarchical Design



Domain Name System (DNS)

- Translate human understandable names to machine understandable names

- E.g., `www.cs.ucsd.edu` → `132.239.51.20`

- Hierarchical structure

- Every DNS server knows where the “root” is

- The root can tell you how to get to `.edu`

- `.edu` server can tell you how to find `ucsd.edu`

- `ucsd.edu` tells you about `cs.ucsd.edu`

- `cs.ucsd.edu` translates `www.cs.ucsd.edu` → `132.239.51.20`

- Caching along the way to improve performance

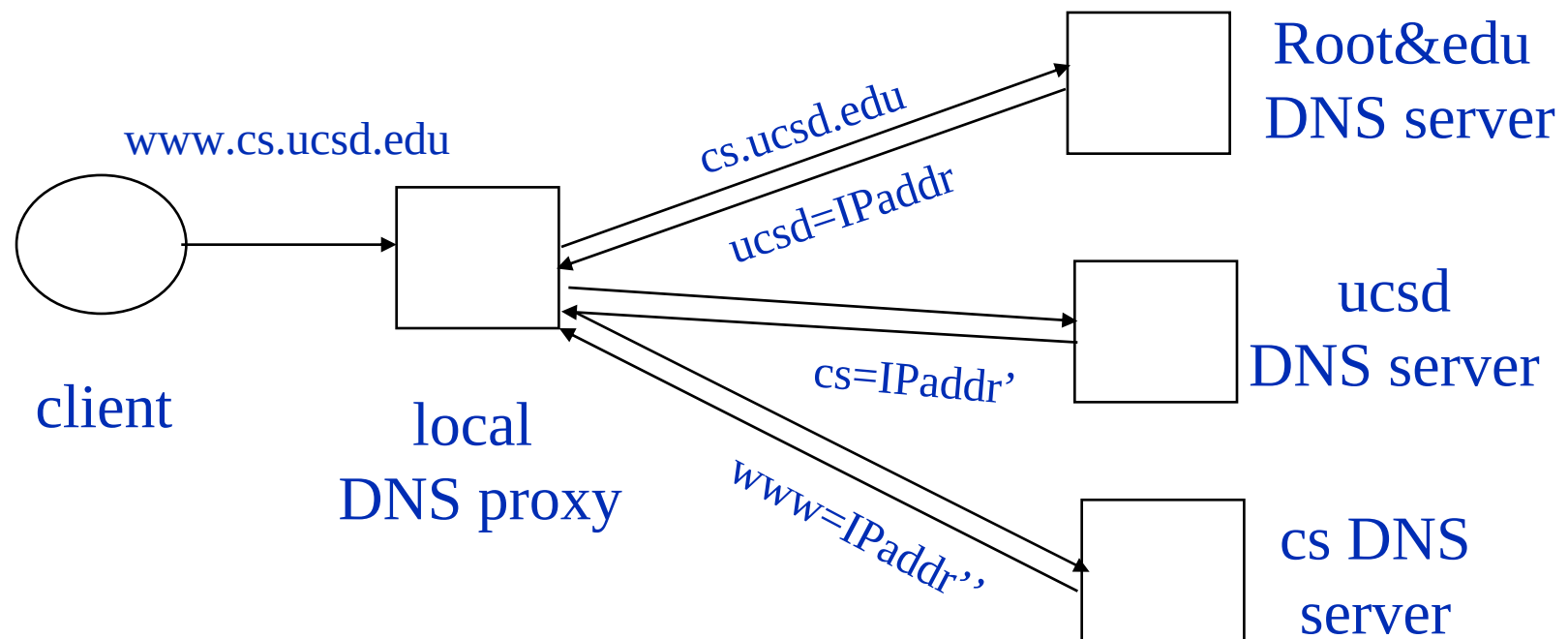
Query Processing

- Query local name server
 - Authoritative/cached answers
- Support both recursive and iterative queries
- If not cached locally, locate server lowest in the hierarchy with entry in local DB
 - In the worst case, contact root (.)
 - Cache locally with TTL

Zones and Caching

- Mechanisms for data distribution
- Zones
 - Provide local autonomy
 - Any contiguous set of nodes in the tree
 - Can be grown to arbitrary size
 - Each domain should provide redundant servers
- Caching
 - Time to live (TTL) associated with each R
 - low value => higher consistency
 - high value => better performance (less traffic)

DNS Lookup Example



1988 Status

- 20k hosts available through DNS (!)
- 30 top level domain names
- SRI managed all non-country top levels
- 7 Root servers
 - 1 query per second, driven by tuning of parameters
 - 50% of 1988 traffic could be eliminated with further tuning
- Query breakdown
 - All info (25-40%)
 - Hostname to address (30-40%)
 - Address to hostname (10-15%)
 - Mail MX record (<10%)

Performance

- Performance worse than designed for (distributed system)
- Clients see 500 ms to 5 second response time for root servers
- Delegated domain performance much worse
 - 3 to 10 seconds, with 30 to 60 seconds not unreasonable
- Negative caching
 - Initially 20-60% of requests were for bad data (old style mail addr)
 - Still 10-50% (typically 25%) for bad data
 - Programs produce steady stream of bad names
 - Negative caching initially an optional feature

Discussion

- Distributed debugging
- Write once, run anywhere? (debug everywhere)
- Mechanisms for pushing code updates
- Self-tuning system
 - Caching responses even if unreasonable (reverse data and TTL)
 - Developers do not want to tune system
 - Especially if they are getting reasonable performance
 - Globally vs. locally optimal
- Security?

Lessons from Giant-Scale Services

Giant-Scale Services

- Challenges for network services:

- High availability

Critical in today's environment: \$1000/sec of lost revenue during downtime

- Evolution
- Growth

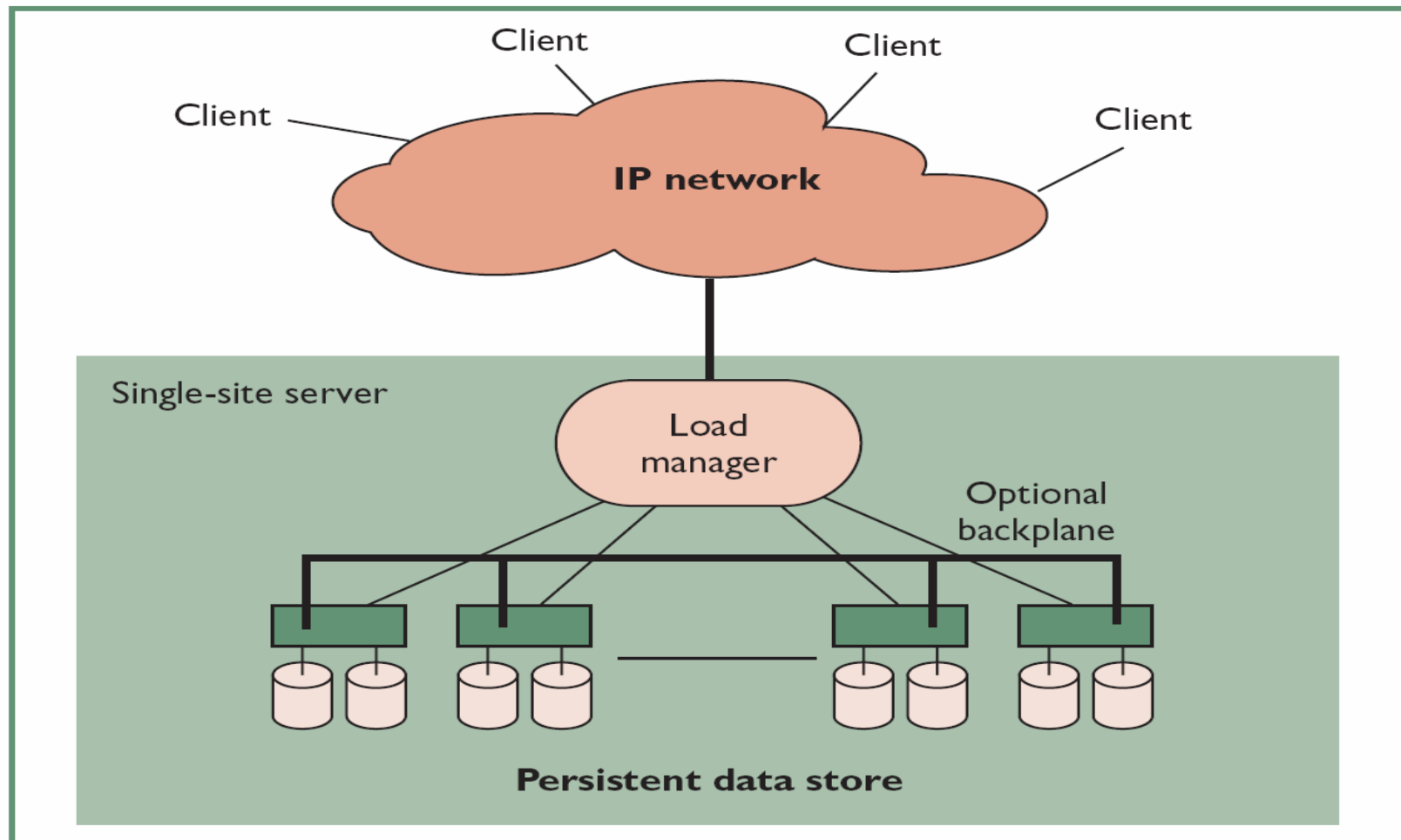
- This paper does not address

- Service monitoring, configuration, QoS, security, logging, and log analysis
- Wide-area replicated services
- Write intensive services

Benefits of Network Services

- Access anywhere, anytime
- Availability via multiple devices
- Groupware support
 - calendaring, teleconferencing, messaging, etc.
- Lower overall cost
 - Multiplex infrastructure over active users
 - Dedicated resources are typically 98% idle
 - Central administrative burden
- Simplified service updates
 - Update the service in one place, or 100 million?

Network Service Components



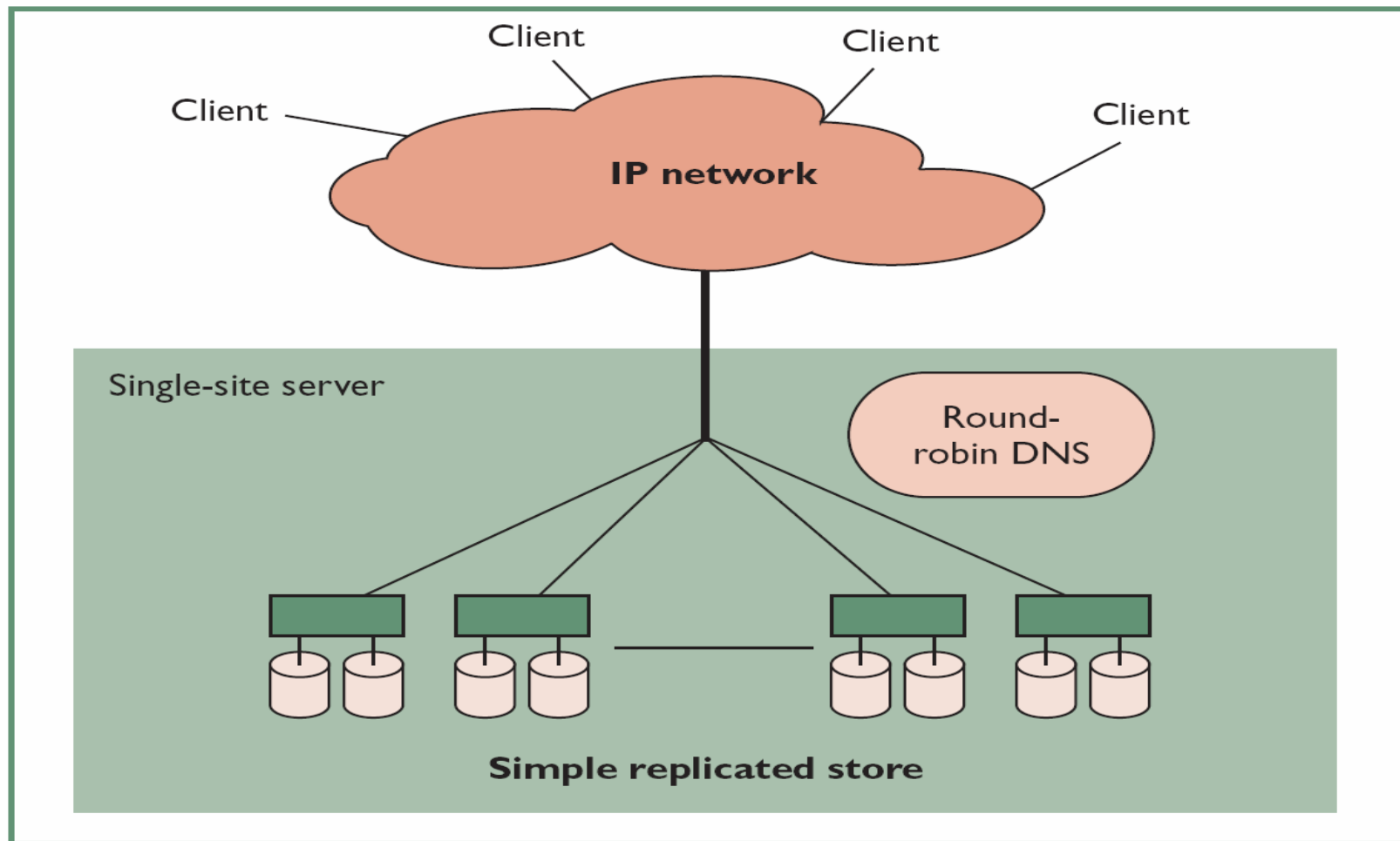
Clusters as Building Block

- No alternative to clusters for building network services that can scale to global use
- Key question: what is the lowest-level building block of a cluster?
 - Commodity pentium processor or higher-end SMP?
- Cluster benefits:
 - Incremental scalability
 - Adding one machine typically linearly improves performance
 - Independent components
 - Cost and performance

Load Management

- Started with “round-robin” DNS in 1995
 - Map hostname to multiple IP addresses, hand out particular mapping in a round robin fashion to clients
 - Does not hide failure or inactive servers
 - Exposes structure of underlying service
- Today, L-4 and L-7 switches can inspect TCP session state or HTTP session state (cookie sticky, etc.)
 - Perform mapping of requests to back end servers based on dynamically changing membership information

Service Replication



Service Partitioning

