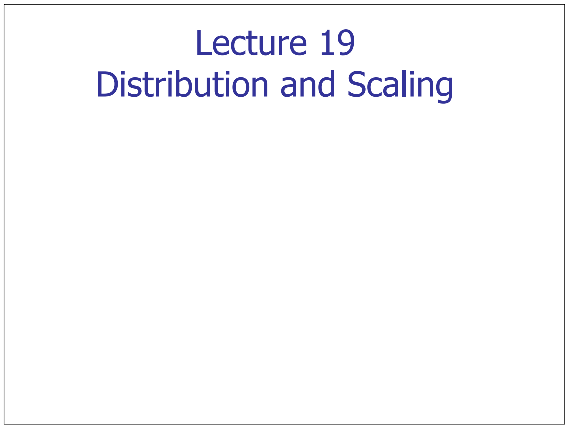
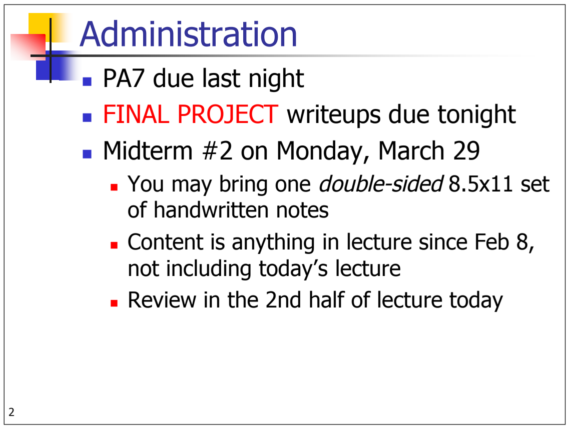
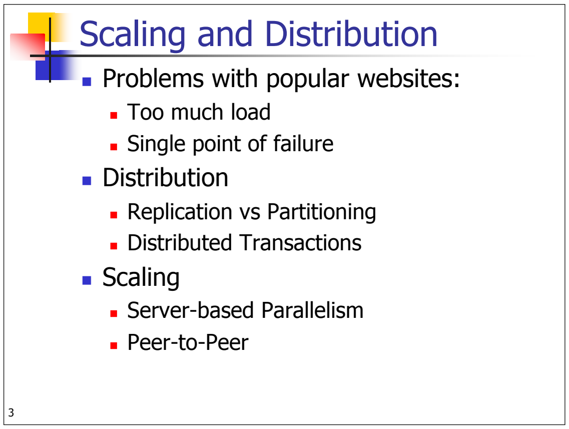


Lecture 19
Distribution and Scaling



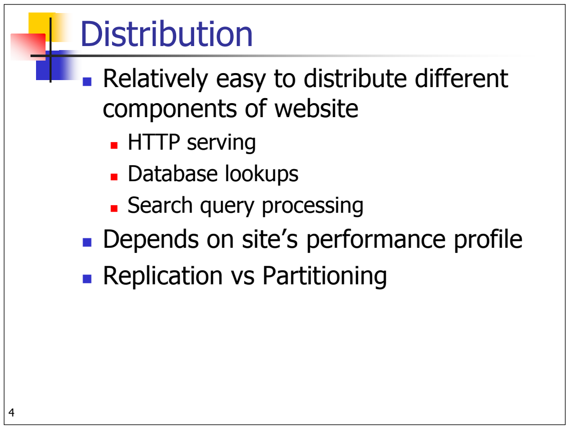
Administration

- PA7 due last night
- FINAL PROJECT writeups due tonight
- Midterm #2 on Monday, March 29
  - You may bring one *double-sided* 8.5x11 set of handwritten notes
  - Content is anything in lecture since Feb 8, not including today's lecture
  - Review in the 2nd half of lecture today



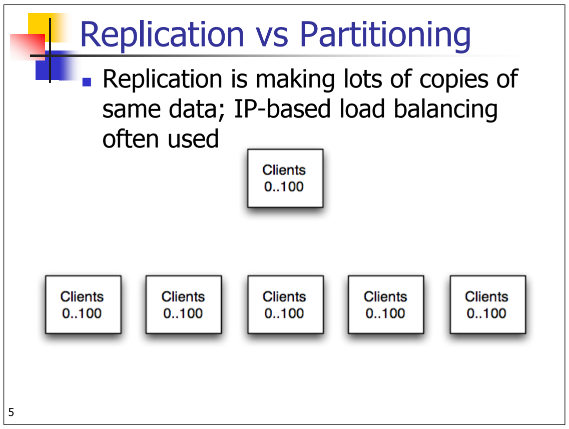
Scaling and Distribution

- Problems with popular websites:
  - Too much load
  - Single point of failure
- Distribution
  - Replication vs Partitioning
  - Distributed Transactions
- Scaling
  - Server-based Parallelism
  - Peer-to-Peer



Distribution

- Relatively easy to distribute different components of website
  - HTTP serving
  - Database lookups
  - Search query processing
- Depends on site's performance profile
- Replication vs Partitioning



Replication vs Partitioning

- Replication is making lots of copies of same data; IP-based load balancing often used

Clients 0..100

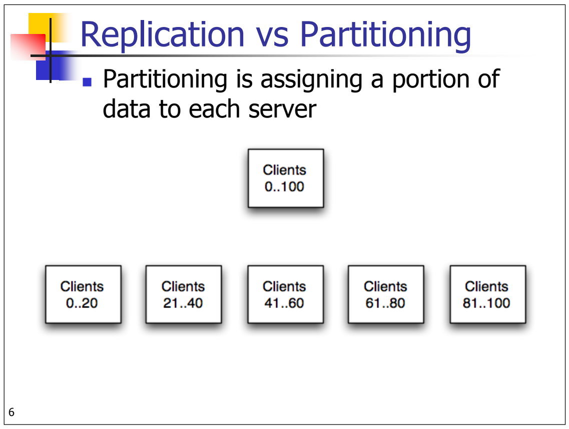
Clients 0..100

Clients 0..100

Clients 0..100

Clients 0..100

Clients 0..100



Replication vs Partitioning

- Partitioning is assigning a portion of data to each server

Clients 0..100

Clients 0..20

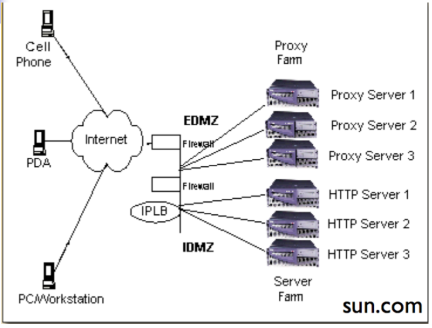
Clients 21..40

Clients 41..60

Clients 61..80

Clients 81..100

Distributing Load



sun.com

Distribution Algs

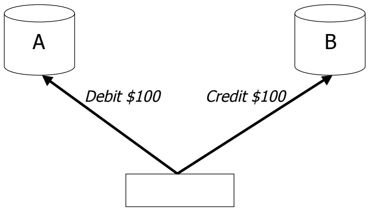
- **DNS Round-Robin** resolves a webserver domain name to a rotating set of target IP addresses
  - Geographically disparate servers
  - DNS caches give rough server-affinity
- **Request Distributor** is a single front-end machine that directs requests
  - Geographic distribution harder
  - Intelligent distribution possible

Choosing Rep vs Dist

- Both offer performance gains
- Replication:
  - Any replica can provide availability, disaster recovery
  - But heavy update & storage overhead
- Partitioning:
  - No storage & lower update overhead
  - If request involves stitching together results from many servers, there may be non-partitionable fixed costs per request
  - No improved availability

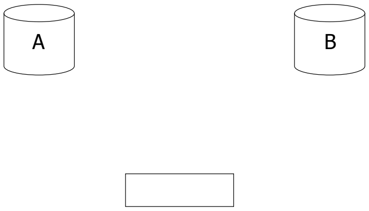
Distributed Writes

- What if req changes multiple servers?
  - Ship item & debit account
  - Cast spell & consume magic potion
- Want *both* or *neither; just-one* is bad



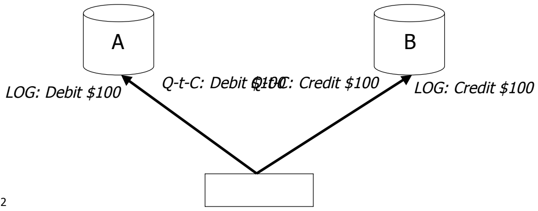
Distributed Writes

- 2-Phase Commit
  - Has a coordinator and several participants
  - Each participant has stable writeahead log



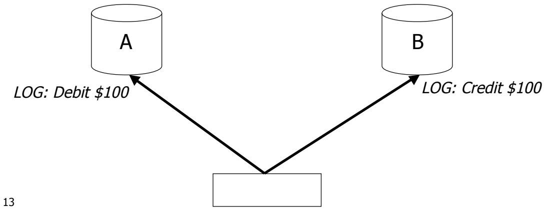
Distributed Writes

- Phase 1: Prepare
  - Coordinator sends *prepare* to each
  - Participants write to log, return *agree* or *abort*



Distributed Writes

- Phase 2: Commit
  - If all agreed, then send *commits* and receive *ACKs*
  - If any aborted, then send *rollbacks* and receive *ACKs*



Distributed Writes

- Why does it work?
  - Can only *commit* if all participants agree
- What about machine failure?
  - If participant dies, upon reboot check disk to see if commit msg has been received
    - If so, execute log and commit
    - If prepare-but-not-commit, wait for coordinator
  - If coordinator dies, upon reboot
    - Abort anything not-yet-committed
    - Resend commit messages if commit requested but not yet ACKed
- Weaknesses?

Midterm Review

- Information Retrieval Basics
  - Architecture
  - Fetcher issues
  - Inverted index, boolean query processing
- Relevance
  - Boolean retrieval model
  - Vector space model, tf-idf, and cosine distance, vector space similarity
  - How to integrate scoring w/inverted index

Midterm Review

- IR Evaluation
  - Data issues
  - Precision & Recall, PR curves, Kendall's Tau
- Link Analysis
  - Network analysis: prestige & importance
  - PageRank, Hubs & Authorities
- Other search stuff
  - Shingling for similarity measure
  - Inverted index construction, single-machine & distributed
  - Distributed search processing

Midterm Review

- Auctions
  - Different kinds of auction
  - Motives and analysis of auctions
  - Details about online ad auctions
- Recommender Systems
  - User-based vs item-based collab. filtering
  - Methods for measuring user-similarity
- Data Mining
  - Supervised vs unsupervised methods; x-validation
  - Apriori algorithm
- DNS
  - Structure, caching, namespace, Akamai