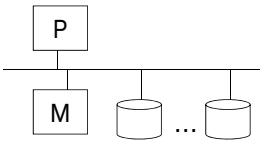
CS 347:
Distributed Databases and
Transaction Processing

Notes 01: Introduction

Hector Garcia-Molina

In CS245: Centralized DB system



Software:

Application
SQL Front End
Query Processor
Transaction Proc.
File Access

- Simplifications:
  - single front end
  - one place to keep locks
  - if processor fails, system fails, ...

In CS347

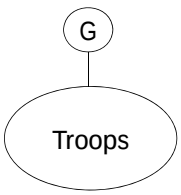
- Multiple processors (+ memories)
- Heterogeneity and autonomy of "components"

Multiple processors

- Opportunity for parallelism
- Opportunity for reliability
- Synchronization issues

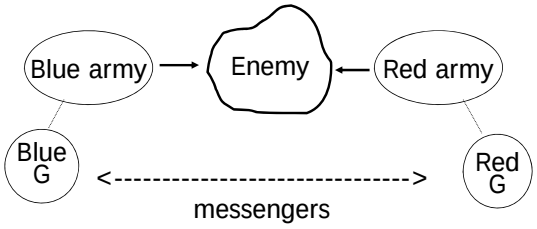
⇔ To illustrate synchronization problems:
Two Generals Problem

The one general problem (Trivial!)



➔ Battlefield

The two general problem:

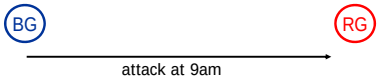


Rules:

- Blue and red army must attack at same time
- Blue and red generals synchronize through messengers
- Messengers can be lost

How Many Messages Do We Need?

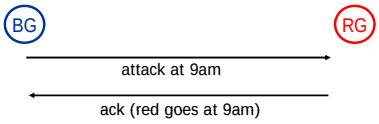
assume blue starts...



Is this enough??

How Many Messages Do We Need?

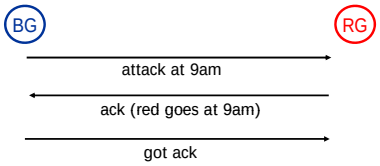
assume blue starts...



Is this enough??

How Many Messages Do We Need?

assume blue starts...



Is this enough??

Stated problem is Impossible!

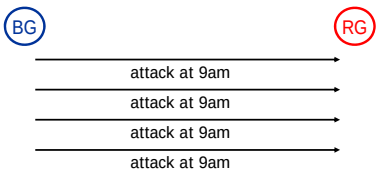
- **Theorem:** There is no protocol that uses a finite number of messages that solves the two-generals problem (as stated here)

Alternatives??

Probabilistic Approach?

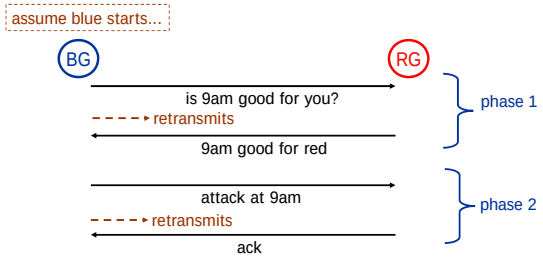
- Send as many messages as possible, hope one gets through...

assume blue starts...



Eventual Commit

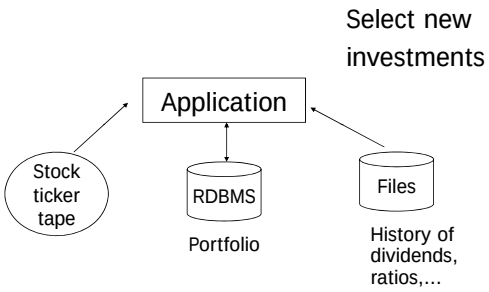
- Eventually both sides attack...



Commit Protocols

- Will study commit protocols like these...

Heterogeneity



Autonomy

Example: unable to get statistics for query optimization

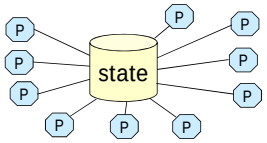
Example: blue general may have mind of his (or her) own!

- So, in CS347 we study data management with multiple processors and possible autonomy, heterogeneity
  - Impact on:
    - Data organization
    - Query processing
    - Access structures
    - Concurrency control
    - Recovery

- Renewed Interest in Distributed/Parallel Data Processing!
  - Massive web data, manage with many computers
  - How to crawl and search the web?
  - Peer-to-peer systems manage huge amounts of data
  - Data from many sources (e.g., comparison shopping): how to integrate?
  - Sensor Networks: data generated an many sensors/devices, need to analyze
  - Multi-player games (e.g., Second Life): tons of distributed data

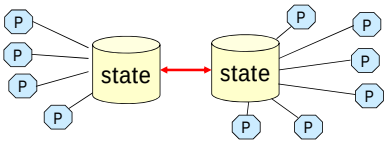
It's the ~~Economy~~ Data, Stupid!

- Example: Multi-player games



It's the ~~Economy~~ Data, Stupid!

- Example: Multi-player games



Logistics

- LECTURES: Mondays and Wednesdays 12:50pm to 2:05pm, Skilling 193
- INSTRUCTOR: Hector Garcia-Molina; Office: Gates Hall 434 Email: hector@cs.stanford.edu; Office Hours: Mondays, Wednesdays 11am to 12noon.
- ADDITIONAL INSTRUCTOR: Zoltan Gyongyi; Email: zoltan@cs.stanford.edu.
- TEACHING ASSISTANT: Paul Heymann; Office: Gates 424; Email: cs347-spr0809-staff@lists.stanford.edu; News Group: su.class.cs347; Office Hours: TBD
- SECRETARY: Marianne Siroker; Office: Gates Hall 436; Email: siroker@cs.stanford.edu; Phone: (650) 723-0872

Logistics

- TEXTBOOK: No required textbook. Some material for the lectures will be drawn from the following book:
  - M. Tamer Ozsu and Patrick Valduriez, "Principles of Distributed Database Systems," Second Edition, Prentice Hall 1999.
- CLASS WEB PAGE: http://www.stanford.edu/class/cs347 Will contain homework assignments, course news, etc. Be sure to check it periodically.
- ASSIGNMENTS: about 5 homeworks
- GRADING: Homeworks: 20%, Midterm 30%, Final: 50%.

Tentative Syllabus (Part I)

DATE	TOPIC
Wednesday April 1	Introduction [N01]
Monday April 6	Data Fragmentation [N02] Z
Wednesday April 8	Query processing [N03] Z
Monday April 13	Query processing [N03] Z
Wednesday April 15	Query Optimization [N04] Z
Monday April 20	Concurrency Control, Failures [N06]
Wednesday April 22	Reliable Data Management [N07]
Monday April 27	Reliable Data Management [N07]
Wednesday April 29	Replicated Data Management [N08]

Tentative Syllabus (Part II)

DATE	TOPIC
Monday May 4	Midterm
Wednesday May 6	Network Partitions [N09]
Monday May 11	Peer to Peer Systems [N05]
Wednesday May 13	Peer to Peer Systems [N05]
Monday May 18	Map-Reduce
Wednesday May 20	Distributed IR Z
Wednesday May 27	Distributed Entity Resolution [new]
Monday June 1	Time [N10]
Wednesday June 3	Heterogeneous Systems, Source Capabilities [N11, 12, 13] Z

Concepts you should be familiar with:

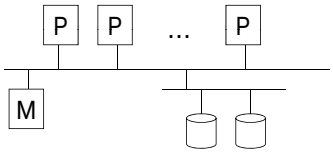
- CS245: query plan, cost estimation, join algorithms, recovery, logging,...
- Interconnection networks (bus, mesh, hypercube,...)
- Computer networks (LAN, WAN,...)

Introductory topics

- Database architectures
- Client-server systems
- Distributed vs. parallel DB systems
- Cloud Computing

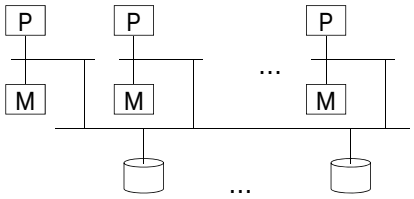
DB architectures

(1) Shared memory



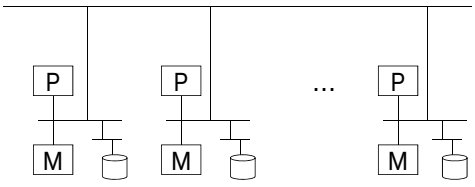
DB architectures

(2) Shared disk



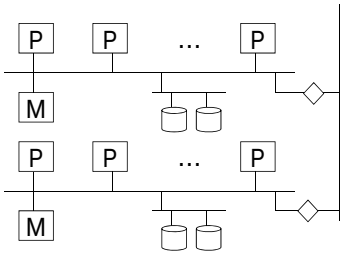
DB architectures

(3) Shared nothing



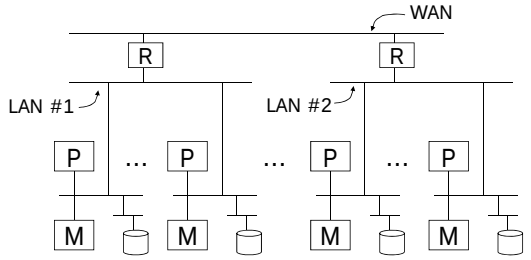
DB architectures

(4) Hybrid example



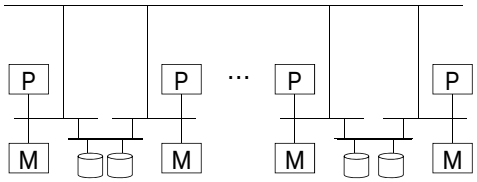
DB architectures

(4) Hybrid example 2



DB architectures

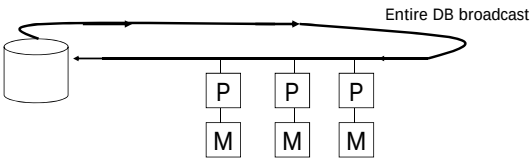
(4) Hybrid Tandem-like



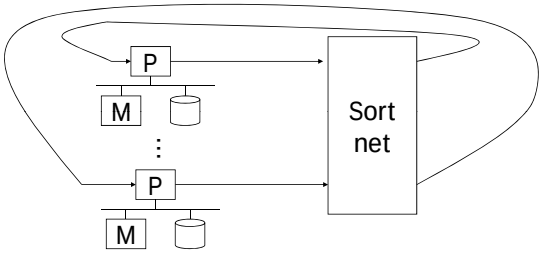
DB architectures

(5) Unusual?

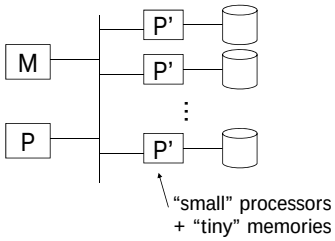
Datacycle (Broadcast disks)



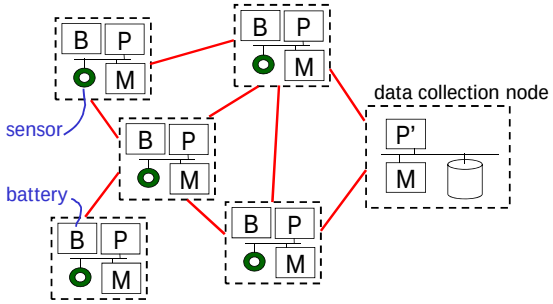
(5) Unusual Sorting network



(5) Unusual — processor per track or processor per disk



(6) Unusual — sensor networks

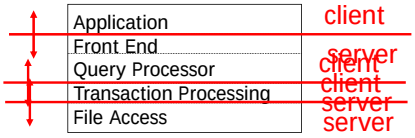


Issues for selecting architecture

- Reliability
- Scalability
- Geographic distribution of data
- Data "clusters"
- Performance
- Cost

Client-Server Systems

(or how to partition software)



Transaction Servers

- Clients ship transactions consisting of 1 or more SQL commands

E.g., Open DataBase Connectivity (ODBC)
(standard API)

Data Servers

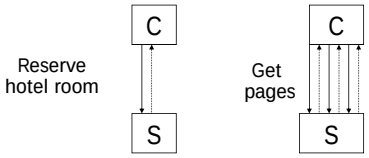
- Client requests pages or records
- Popular for OODB systems

Issues

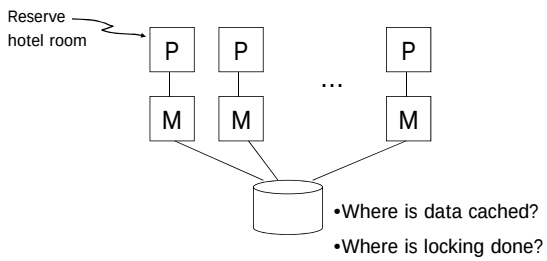
- Object granularity
- Where is data cached?
- Where is locking done?

Basic Tradeoff

- Offloading work to clients
- Data transmitted



Note: Similar issues arise when we partition software/functionality within server



Parallel or distributed DB system?

- More similarities than differences!

- Typically, parallel DBs:
  - Fast interconnect
  - Homogeneous software
  - High performance is goal
  - Transparency is goal

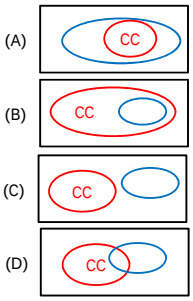
- Typically, distributed DBs:
  - Geographically distributed
  - Data sharing is goal (may run into heterogeneity, autonomy)
  - Disconnected operation possible

Cloud Computing

- Is CC just a marketing term??
  - utility (like power)
  - data or CPU cycles?
  - many processors, many storage units
  - business model

Is CC a subset, superset, disjoint from, or overlaps with:

- grid computing
- distributed computing
- Web 2.0
- Cluster Computing
- Peer-to-peer computing
- software as a service
- client-server computing
- data center as a computer
- massively parallel computing



CC Issues

- Customer lock-in
- Privacy
- Standards
- Software licensing

Next

- How to describe distributed data
- Query processing in parallel DBs
- Query processing in distributed DBs

Query processing in parallel DBs:

- Typically: we can distribute/ partition/ sort…. data to make certain DB operations (e.g., Join) fast

Query processing in distributed DBs:

- Typically: we are given data distribution; we need to find query processing strategy to minimize cost (e.g., communication cost)