Outline

- Announcements
- Databases (cont’d)
- Algorithms and Protocols
- Student Presentations
- Akamai
Announcements

- Database Assignment due today (submit to eCommons – till 11 p.m.)

- Business paper – due next Tuesday (3/8/11)

- Homework 4 out – due next Tuesday (submit to eCommons)
Announcements II

• Forthcoming Student Presentations (3/8/2011)
  • Kelton Noel Gregory Case: American Airlines
  • Sze Wing Ng Bus Proj: Sport Chalet Mar 08

• Reading:
  • Chapter 10 of Messerschmitt (Reader 1)
  • American Airline Case Study (Reader 2)
  • Chapter 1 on Networking (see course webpage)
Student Presentations

- Miles Carvalho Dreszer Bus Proj: Jamba Juice
- Vidya Menon Kulavil Bus Proj: Walmart Mar 03
- Owen Mathys Bus Proj: FedEx
Treat data as a separate asset

- May be shared by multiple applications

Provide protection and integrity features appropriate to mission-critical data

- Access control
- Integrity constraints
- Persistence
- etc.
# The Relational Model

### EMPLOYERS

<table>
<thead>
<tr>
<th>EMPL_ID</th>
<th>EMPL_NAME</th>
<th>EMPL_POSITION</th>
<th>DEPT_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Alice</td>
<td>Manager</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>Bob</td>
<td>Programmer</td>
<td>1</td>
</tr>
<tr>
<td>102</td>
<td>Chris</td>
<td>Manager</td>
<td>2</td>
</tr>
<tr>
<td>103</td>
<td>David</td>
<td>Accountant</td>
<td>2</td>
</tr>
</tbody>
</table>

### DEPARTMENTS

<table>
<thead>
<tr>
<th>DEPT. ID</th>
<th>DEPARTMENT</th>
<th>DEPT. ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IT</td>
<td>San Jose</td>
</tr>
<tr>
<td>2</td>
<td>Finance</td>
<td>New York</td>
</tr>
</tbody>
</table>
Database operations - SQL

“PROJECT”

“SELECT”

Departments

Employees

Each operation results in a new table

Database operations - SQL

“PROJECT”

“SELECT”

Departments

Employees

Each operation results in a new table
# Database Operations

## Employers

<table>
<thead>
<tr>
<th>EMPL_ID</th>
<th>EMPL_NAME</th>
<th>EMPL_POSITION</th>
<th>DEPT. ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Alice</td>
<td>Manager</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>Bob</td>
<td>Programmer</td>
<td>1</td>
</tr>
<tr>
<td>102</td>
<td>Chris</td>
<td>Manager</td>
<td>2</td>
</tr>
<tr>
<td>103</td>
<td>David</td>
<td>Accountant</td>
<td>2</td>
</tr>
</tbody>
</table>

## Departments

<table>
<thead>
<tr>
<th>DEPT. ID</th>
<th>DEPARTMENT</th>
<th>DEPT. ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IT</td>
<td>San Jose</td>
</tr>
<tr>
<td>2</td>
<td>Finance</td>
<td>New York</td>
</tr>
</tbody>
</table>

## JOIN

### EMPLOYERS

<table>
<thead>
<tr>
<th>EMPL_ID</th>
<th>EMPL_NAME</th>
<th>EMPL_POSITION</th>
<th>DEPT. ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Alice</td>
<td>Manager</td>
<td>IT</td>
</tr>
<tr>
<td>101</td>
<td>Bob</td>
<td>Programmer</td>
<td>IT</td>
</tr>
<tr>
<td>102</td>
<td>Chris</td>
<td>Manager</td>
<td>Finance</td>
</tr>
<tr>
<td>103</td>
<td>David</td>
<td>Accountant</td>
<td>Finance</td>
</tr>
</tbody>
</table>

### DEPARTMENTS

<table>
<thead>
<tr>
<th>DEPT. ID</th>
<th>DEPARTMENT</th>
<th>DEPT. ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IT</td>
<td>San Jose</td>
</tr>
<tr>
<td>2</td>
<td>Finance</td>
<td>New York</td>
</tr>
</tbody>
</table>
Application Logic and Tables

Application logic (programming language)

SQL

Database
Databases & OLTP
Recall - Two capabilities

**Aggregation**: accessing multiple databases

**Sharing**: two or more applications accessing the same databases
Example - Travel Agency

Travelocity.com

CheapTickets.com

What can go wrong?
Example - Travel Agency

Travelocity.com

CheapTickets.com

Hotels
Cars
Airtickets

A resource might be unavailable
Example - Travel Agency

Two applications might try to access & update the same resource concurrently.
An application or a host might crash before the completion of the transaction.
Example - Travel Agency

A customer’s transaction should be completed in its entirety, or aborted.
Transaction Processing

“The coordination of multiple resources and the shared access to common resources in a systematic and consistent way”

Examples?
- Financial applications (stock market, ATMs)
- Reservations (travel, theatre)
- Manufacturing (inventory, purchasing, billing)
- Etc...
Online Transaction Processing (OLTP)

- Transaction Processing for networked applications

- 4 Important Properties of transactions: **ACID**
  - Atomicity
  - Consistency
  - Isolation
  - Durability
The ACID properties

- **Atomicity**
  - All transaction components should either complete together (commit) or abort
  - E.g. All reservations (airline, hotel, car) should be grouped as a single transaction that either commits, or aborts

- **Consistency**
  - A transaction must leave the system in a consistent state at the end of the transaction, or else abort
  - E.g. Either a consistent set of reservations has been made, or none

- **Isolation**
  - Concurrent transactions are allowed only if they don’t interfere with each other
  - Two travel agents can concurrently access the same database if the reservations are for different dates/places

- **Durability**
  - A transaction leaves the resources in a permanent state after it commits
Structure of a Transaction

Durable Starting State -> Actions to be performed

Successful completion -> Durable, Consistent End State

Abort

Rollback
OLTP

- Simplifies application development

- Enables protection and integrity of mission-critical data in a transparent way
  - for the end user
  - for the application developer
Algorithms and protocols

Adapted from
David G. Messerschmitt
Algorithm named after 9th century Persian mathematician Abu Jaffar Mohammed Ibn Musa Al-Khowarizmi

Specified sequence of steps that

- accomplish a designated task
- in a finite number of steps

Representation:

- simple algorithm: flowchart
- complicated algorithm: program
Example: one turn at monopoly

1. Start turn
2. Throw dice
3. Move token number of squares indicated on dice
4. Land on “go to jail”? Yes: Move to “jail” square Yes: Finish turn No: Do not move; follow policies for square (like “pay rent”)

Programming languages support these three building blocks.
 Protocol

- Distributed algorithm ...
- Realized by two or more modules to coordinate their actions or accomplish some shared task
- **Module interoperability requires a protocol**
  - Prescribed order of method invocations
  - Part of interface documentation
Monopoly players protocol

This is a protocol interaction diagram
Application and infrastructure

Internally, the network uses protocols to implement the services it provides.

The application defines its own application-level protocols.
Example:

- **HEADQUARTERS**
  - Airline Dataserver

- **Airline Intranet**

- **HHC Server**

- **Wireless Link**

- **HHC**
Layered Protocols Example

- **HHC Server**
  - HHC Server Application
    - Windows OS
      - Break Messages into Packets
    - Networking Infrastructure
  - Request Pass. Data
  - Send Pass. Data As Message
  - Send Packet
  - Acknowledge Packet
  - Link Level Protocol

- **HHC**
  - HHC Application
    - Palm OS
    - Networking Infrastructure
Three simple protocols

One-way message: send-receive
Two-way interaction: request-response
Push: publish-subscribe
Send - Receive

Client

send

Server

receive

Time
Request - Response
Send - Acknowledge
Example: HTTP (Hyper Text Transfer Protocol)

1. User activates URL
2. HTTP request
3. HTTP response (embedded document)
4. Browser displays document (if HTML) or invokes “helper application”
Three ways of locating things

Name
- “Joe Bloe”

Address
- “1299 Hearst St, Berkeley, CA”

Reference
- “Postmaster of Berkeley CA”
Name

- Symbolic (character string) representation
- Easy for people to remember or guess
- Identifies, but
  - Does not locate directly
    - Distinction important for mobile entities
- Not unique: entities can have more than one name (called aliases)
Hierarchical names

Hierarchy makes names easier to remember or guess

Host domain names:
- “info.sims.berkeley.edu”
- designates administrative hierarchy

File names:
- “c:\My Documents\Docs\Resume.doc”
- designates folder hierarchy
Address

- Route or path to entity
  - is directly specified, or
  - can be inferred

- Independent of who or what is locating entity

- Topological specification
Example

Path from to is \((R,D,D,D,R,R,R,R)\)

Is \((R,D,D,D,R,R,R,R)\) an address?
No! -- not an address, because it depends on starting point
Example

Address of (6,5) is (6,5)

Route from can be inferred
Abstract representation of an entity

Interaction is with representation
- infrastructure arranges redirection to actual entity
- especially appropriate for things that move

Example
- A Cell phone number is a reference.
- A Wired phone number is an address.
Name services

1. name
2. address or reference
3. interaction
Program

- Precise description of an algorithm in a formal language that is called programming language
- Actions are applied to data
Formulation in a language

- **Natural language**
  - No strict syntactic rules
  - Great density and semantic capability

- **Formal language**
  - Strict syntax and semantics

- **Programming language**
  - Formal language in which computations can be described
  - Executable by an electronic computer
Can we solve all problems?

Collatz Conjecture (Ulam):

```java
while x!=1 do
    if (x is even) then x=x/2
    else x=3*x+1
```

Example:

7 → 22 → 11 → 34 → 17 → 52 → 26 → 13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1

Given a number x does the program hold?
Open problem!
Translation of programs

Source Code
(in a programming language)

Compiler

Input → Executable program → Output
(machine language)
Akamai Case
Internet Bottlenecks

- **First Mile (Server Capacity)** - 70% of website performance problems according to one study

- **Backbone** - Plentiful, but some shortage within metropolitan areas

- **Peering** - Exchange of traffic between NSPs

- **Last Mile to home**
  - 56 K modems are slow
  - Shared LAN limitations
Solutions

- **Expand Bandwidth**
  - Being done

- **Mirroring web cites**
  - Put exact copy of same web page to multiple servers
  - Tricky to duplicate content

- **Caching**
  - Problem: Stale Content
  - Problem: Hard to count “click throughs”

- **Content Distribution Networks...**
Akamai Freeflow

Local Office or ISP

Web Page

Text....

Large Company

Web Server

INTERNET

Akamai Server

NSP 1

NSP 2
Freeflow

- Deployed in 1999
- Akamai Infrastructure
  - 13000 servers in 954 networks by 2001
- Customers -
  - Large Commercial Websites
- Revenue model - $2000 per mbps served
  - (For comparison, normal Internet access cost 500 mbps at time)
2000 Financials

- **$196 Million Loss** (Before special charges)
- $90 million revenue
- %20 gross margin, after deducting
  - server depreciation
  - payments to network partners
  - Data center space
- But, most expenses of shouldn’t grow at same rate as number of customers, so margin should improve

- **$201.5 million SG&A**
  - (selling general and administrative)
  - (largely sales force cost)
  - Again, this might not grow at same rate as the number of customers.

- **$40 million R&D**
Competition

- Hosting firms (substitute)
  - Exodus

- Other CDNs
  - Sandpiper, Adero, Mirror Image

- Content Alliances
  - Akamai’s competitors banded together to share networks
2001 Market Changes

Bad

- Dot-coms bust
- Customers leave
  - “churn rate goes to 22% per quarter”

Good

- Hosting firms go bust (exodus)
- Some CDN competitors go bust.
- Competing CDN alliances mired in problems
EdgeSuite

- Assemble dynamic pages at edges rather than just serve heavy objects
- Value proposition
  - Performance improvement
  - Cost and complexity reduction
  - Scalability
  - Security
- Pricing – higher than old service
- Soon edge suite dominated revenue
Dynamic CDN technology: ESI (edge sides includes)

Develop as open standard why?

Akamai not big and credible enough to force a de-facto standard on market
Marketing

- Difference in selling old vs new products:
  - Old product
    - Geared toward speeding up websites
    - Revenues of their clients depended on speed
    - Easier to get sale
  - New Product
    - Simplify company IT function
    - Cost vs. revenue center
    - Harder sell. More data driven…
    - Consequently new product needs more professional sales force

- Channels?
  - Distribution Partners (IBM) credibility
  - Direct Sales Force too
Recent Performance

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(In thousands, except share data)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Consolidated Statements of Operations Data:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revenues</td>
<td>$210,015</td>
<td>$161,259</td>
<td>$144,976</td>
<td>$163,214</td>
<td>$89,766</td>
</tr>
<tr>
<td>Total cost and operating expenses</td>
<td>161,048</td>
<td>172,370</td>
<td>327,580</td>
<td>2,577,108</td>
<td>989,359</td>
</tr>
<tr>
<td>Net income (loss)</td>
<td>34,364</td>
<td>(29,281)</td>
<td>(204,437)</td>
<td>(2,435,512)</td>
<td>(885,785)</td>
</tr>
<tr>
<td>Net income (loss) attributable to common stockholders</td>
<td>34,364</td>
<td>(29,281)</td>
<td>(204,437)</td>
<td>(2,435,512)</td>
<td>(885,785)</td>
</tr>
</tbody>
</table>