Student Presentation
Open vs. Proprietary Standards

- Open standard – a standard that is well documented, unencumbered by intellectual property rights and restrictions, and available to any vendor.

- What are the advantages?

- What are the disadvantages?
Databases

by

David G. Messerschmitt
Databases

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.
Two capabilities

- **Aggregation**: accessing multiple databases
- **Sharing**: two or more applications accessing the same databases

Applications I and II are connected to databases.
Relational table

<table>
<thead>
<tr>
<th>Table</th>
<th>Employee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Address</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Record</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field
SQL interface

- SQL (Structured Query Language)
- Presents single abstract interface to the application logic
  - For manipulating, and extracting data from database
- Standardized, not vendor specific

- Encapsulates various internal details
  - Data partitioning and replication
  - Host mapping
  - File representation
  - etc.
Database operations

Each operation results in a new table

“PROJECT”

“SELECT”
Database Operations

<table>
<thead>
<tr>
<th>Passengers</th>
<th>Departments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Dept ID</td>
</tr>
<tr>
<td>Alice</td>
<td>1</td>
</tr>
<tr>
<td>Bob</td>
<td>1</td>
</tr>
<tr>
<td>Chris</td>
<td>2</td>
</tr>
<tr>
<td>Dept Name</td>
<td>Dept ID</td>
</tr>
<tr>
<td>Engineering</td>
<td>1</td>
</tr>
<tr>
<td>Sales</td>
<td>2</td>
</tr>
</tbody>
</table>

JOIN

<table>
<thead>
<tr>
<th>Name</th>
<th>Dept ID</th>
<th>Dept Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>Engineering</td>
</tr>
<tr>
<td>Bob</td>
<td>1</td>
<td>Engineering</td>
</tr>
<tr>
<td>Chris</td>
<td>2</td>
<td>Sales</td>
</tr>
<tr>
<td>Year</td>
<td>City</td>
<td>Accommodation</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>---------------------</td>
</tr>
<tr>
<td>2002</td>
<td>Oakley</td>
<td>Bed&amp;Breakfast</td>
</tr>
<tr>
<td>2002</td>
<td>Oakley</td>
<td>Resort</td>
</tr>
<tr>
<td>2002</td>
<td>Oakland</td>
<td>Bed&amp;Breakfast</td>
</tr>
<tr>
<td>2002</td>
<td>Oakland</td>
<td>Resort</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Camping</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Bed&amp;Breakfast</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Resort</td>
</tr>
<tr>
<td>2002</td>
<td>Albany</td>
<td>Camping</td>
</tr>
<tr>
<td>2002</td>
<td>Albany</td>
<td>Bed&amp;Breakfast</td>
</tr>
<tr>
<td>2003</td>
<td>Oakley</td>
<td>Bed&amp;Breakfast</td>
</tr>
<tr>
<td>2003</td>
<td>Oakley</td>
<td>Resort</td>
</tr>
<tr>
<td>2003</td>
<td>Oakland</td>
<td>Bed&amp;Breakfast</td>
</tr>
<tr>
<td>2003</td>
<td>Oakland</td>
<td>Resort</td>
</tr>
<tr>
<td>2003</td>
<td>Berkeley</td>
<td>Camping</td>
</tr>
<tr>
<td>2003</td>
<td>Berkeley</td>
<td>Bed&amp;Breakfast</td>
</tr>
<tr>
<td>2003</td>
<td>Berkeley</td>
<td>Resort</td>
</tr>
<tr>
<td>2003</td>
<td>Albany</td>
<td>Camping</td>
</tr>
<tr>
<td>2003</td>
<td>Albany</td>
<td>Bed&amp;Breakfast</td>
</tr>
</tbody>
</table>

- Entries are simple data types or compositions of those types
  - Integer, string, etc.
mySQL

What does mySQL make?

How Successful is mySQL?
- Visibility: Fortune magazine, more mentions on www
- Reaction from giants
- Revenue growth 2001 700k, 2002 6.2m, 2003 10m
- Good performance reviews
- Recent SAP alliance
- But Market share tiny:
  - $10 million out of $10 billion market!

Why Success?
- Good Technology
- Large DBMS bloated with features most don’t need
- Innovative OSS model
**MySQL**

How does OSS work?

**Two Types of License:**

- **GPL**
  - Free
  - No Support
  - Any software that uses MySQL as a module must itself be made GPL

- **Commercial License**
  - Support
  - Could be distributed with non-open source software
  - Not Free:
    - MySQL: Classic $250, Pro $495 (for ~ 50 users)
    - Compare to:
      - MSFT $3150 single proc for 50 users
      - IBM $33000 single proc for 50 users
      - Oracle $40000 single proc for 50 users
Aside: DB’s in different software stacks

- Which companies are competitors?
- Which are complimenters?
- Which are both!?
mySQL

- Which segments of market is mySQL strong in?
  - Large Companies or Small Companies?
  - Web applications or Critical Enterprise data?

- Why would a major enterprise want to pay so much more for an Oracle or IBM DB?
### My SQL: market

<table>
<thead>
<tr>
<th>Enterprise wide data 90%</th>
<th>Small 20%</th>
<th>Medium 30%</th>
<th>Large 50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft</td>
<td></td>
<td></td>
<td>Oracle IBM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IBM</td>
</tr>
<tr>
<td>Reliability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scalability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Longevity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Sites 10%</td>
<td>My SQL Cost</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How should mySQL grow in order to meet its stated goal of getting to $100 million in revenue?

Figure Adapted from “Teaching Note for MySQL Open Source Database,” 6/1/04, Stanford GSB.
My SQL: Growth Strategy

- Lack of Brand identity in this segment
- MySQL lacks the organization to offer support
- Large enterprises have high switching costs

Figure Adapted from “Teaching Note for MySQL Open Source Database,” 6/1/04, Stanford GSB.
# My SQL: Growth Strategy

<table>
<thead>
<tr>
<th></th>
<th>Small 20%</th>
<th>Medium 30%</th>
<th>Large 50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise</td>
<td>Microsoft</td>
<td></td>
<td>Oracle</td>
</tr>
<tr>
<td>wide data</td>
<td></td>
<td></td>
<td>IBM</td>
</tr>
<tr>
<td>90%</td>
<td></td>
<td></td>
<td>Reliability</td>
</tr>
<tr>
<td>Web Sites</td>
<td>My SQL</td>
<td>Stay Put?</td>
<td>Scalability</td>
</tr>
<tr>
<td>10%</td>
<td>Cost</td>
<td></td>
<td>Support</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Longevity</td>
</tr>
</tbody>
</table>

- Not a big enough market to reach stated $100 million goal.

Figure Adapted from “Teaching Note for MySQL Open Source Database,” 6/1/04, Stanford GSB.
My SQL: Growth Strategy

<table>
<thead>
<tr>
<th>Enterprise wide data 90%</th>
<th>Small 20%</th>
<th>Medium 30%</th>
<th>Large 50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Sites 10%</td>
<td>My SQL</td>
<td>Microsoft</td>
<td>Oracle IBM</td>
</tr>
</tbody>
</table>

- Many of these customers already using MySQL with websites
- Less emphasis on global organization
- Leverage SAP alliance
- Leverage SAP alliance
- Up against Microsoft.

Figure Adapted from “Teaching Note for MySQL Open Source Database,” 6/1/04, Stanford GSB.
# My SQL: Growth Strategy

<table>
<thead>
<tr>
<th>Enterprise wide data 90%</th>
<th>Small 20%</th>
<th>Medium 30%</th>
<th>Large 50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft</td>
<td></td>
<td></td>
<td>Oracle IBM</td>
</tr>
<tr>
<td>Reliability</td>
<td></td>
<td></td>
<td>Reliability</td>
</tr>
<tr>
<td>Scalability</td>
<td></td>
<td></td>
<td>Scalability</td>
</tr>
<tr>
<td>Support</td>
<td></td>
<td></td>
<td>Support</td>
</tr>
<tr>
<td>Longevity</td>
<td></td>
<td></td>
<td>Longevity</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Web Sites 10%</th>
<th>My SQL Cost</th>
<th>Maybe?</th>
</tr>
</thead>
</table>

- + builds on existing brand and strengths
- - Market not so big

Figure Adapted from “Teaching Note for MySQL Open Source Database,” 6/1/04, Stanford GSB.
Networks
What are some examples of communications networks?

- Public Telephone Network
- Internet
- LANs (Local Area Networks)
What does a network do?

1) Transport data from one host to another.
Network Architecture

- Network architectures are layered
- Each layer
  - uses the services of the layers below
  - To offer more advanced services to layer above
- Allows layers to be designed independently
- We will talk about 3 layers next...

<table>
<thead>
<tr>
<th>Network</th>
<th>Link</th>
<th>Physical</th>
</tr>
</thead>
</table>
Physical Layer: Convey bits over a wire

Bits: 010110...

Voltage

1 1
0 0

Time

Voltage

0 0 0

Time

Sender

Wire

Noise

Receiver
Physical layer

- Other schemes for mapping a bit sequence to a physical sequence are possible.
  - These are called *modulation schemes*
Link Layer

- **Make a Frame link out of a bit link**
  - Instead of endless sequence of 1s and 0s, we want distinct “packages” of data that are separate from each other

- **Say we want to send 2 Frames with data**
  - 01010101010111010 and 101010101011010
  - Concatenate them and send them as a sequence?

- **How can the receiver tell where the new frame begins?**

- **Solution**: insert a special sequence at the start of frame: for example: 01111110
Link Layer (cont’d)

- Also does error detection/correction
  - Insert extra information that helps the receiver to determine if the data has been corrupted.
  - Example: parity bit
    - Sender adds a 1 or zero to end of data so number of ones is always odd
      - 10011 or 10000
    - If receiver counts an even number of ones, than it knows the data was corrupted.
More Link Layer.. -- Ethernet

Want to allow multiple hosts to *share a link*

Host A  Host B  Host C  Host D

How do they avoid talking at the same time?

- Don’t transmit if you hear another host transmitting
- If there is a collision, stop wait a random amount of time, and try again
- This is a **Medium Access Control (MAC)** Protocol
How do the hosts on this Ethernet identify each other?

- Each host (actually each interface)
  - has a globally unique *MAC address*
  - *Cannot be changed*
Ethernet Hub

- Hub broadcasts packets on a link to all others
- As if all hosts connected to single link
  - We say it is a Single collision domain
- Only one host can talk at a time
If switch knows where the destination is
- Switch forwards an incoming frame to destination only.
- Otherwise, it broadcasts it to everyone.

Thus, parallel conversations possible.
Network Layer

Host A

- A wants to send some data to C
  - Suppose A knows C's address

- A sends a packet towards C
  - A marks his packet with C's Address (an IP Address)
Post Office Analogy

Bob in New York

Alice Smith
1156 High St
Santa Cruz 95064

NY Post Office

Plane to London

Plane to SFO

SFO Post Office

Look at beginning of zip code. Make forwarding decision

Look at address. Make forwarding decision

Santa Cruz Post Office

Truck to High St

Truck to 41st Ave

Truck to Santa Rosa

Truck to Santa Cruz
Network Layer

Host A
128.114.60.200 (IP Address)

Host B
128.114.60.201

Link 1

Link 2

Link 3

Host C
128.114.60.202

Host D
128.114.60.203

Destination Address: 128.114.60.202

Header  Payload Data
**Network Layer**

- **Host A**: 128.114.60.200
- **Host B**: 128.114.60.201
- **Host C**: 128.114.60.202
- **Host D**: 128.114.60.203

- **Link 1**: 128.114.60.202
- **Link 2**: 128.114.60.203

**Routing Table**

<table>
<thead>
<tr>
<th>Address</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.114.60.202</td>
<td>Link 2</td>
</tr>
<tr>
<td>128.114.60.203</td>
<td>Link 3</td>
</tr>
</tbody>
</table>

- **A uses Link 1 to send to B**
- **B looks at**
  - Packet Header
  - Routing Table

```
<table>
<thead>
<tr>
<th>Address</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.114.60.202</td>
<td>Link 2</td>
</tr>
<tr>
<td>128.114.60.203</td>
<td>Link 3</td>
</tr>
</tbody>
</table>
```
Routing in the Internet

Many feasible paths from source to destination.
Routing

- Updating the routing table
- Objective: each packet gets closer to destination

Packet forwarding

- Transmitting each packet on the appropriate output link
- Based on routing table
Routing Algorithms

Routers talk to each other to build their routing tables.

"I am accepting Traffic to 114.211.1.X"
Routing Table has Wild Cards

<table>
<thead>
<tr>
<th>Host</th>
<th>Address 1</th>
<th>Address 2</th>
<th>Link 1</th>
<th>Link 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host A</td>
<td>114.211.1.X</td>
<td>200.261.19.X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Host B</td>
<td>114.211.1.X</td>
<td>200.261.19.X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Host C</td>
<td>114.211.1.1</td>
<td>114.211.1.2</td>
<td>114.211.1.3</td>
<td>114.211.1.4</td>
</tr>
</tbody>
</table>
Routing Concerns

- Long routes
- Circular routes
- Hijacking routes
- Route flapping
IP Addresses vs Mac Addresses

- **Hierarchical**
  - The beginning bits tell you which network the host is on
  - Ex: UCSC addresses start with 128.114.X.X
  - The last bits tell you which host of the network

- **Changeable**
  - Changes with location of Host
  - 4 bytes
  - Only 4.2 billion

- **Not Hierarchical**
  - Beginning bits tell nothing useful

- **Not Changeable**
  - 6 bytes
  - 281 Trillion
Link and Network Layer Interaction

MAC Header

Ethernet Frame Payload

IP Header IP Payload

Router

Host A

Ethernet Hub

Router MAC address 00-A4-B7-34-57-23

Strip MAC header off frame. Forward IP packet based on Routing table.

Host B

Router

Ethernet Hub

Router

128.114.60.202

Host C
Transport Protocols

- The Internet is unreliable
  - It will make a “best effort” to get your packet to its destination

- Packets can be lost because of
  - Congestion
  - Link errors
  - Routing problems
Transmission Control Protocol (TCP)

- Retransmit mechanism for reliability
  - Receiver sends acknowledgements to sender
  - If a packet is lost, source fails to get ACK, and then retransmits.

- Congestion control
  - If congestion perceived (by lost packets)
  - Source reduces its send rate
    - When loss, sender reduces send rate by half
    - Otherwise slowly increases
TCP cont’d

<table>
<thead>
<tr>
<th>IP Header</th>
<th>TCP Header</th>
<th>Payload</th>
</tr>
</thead>
</table>

TCP port numbers
- TCP Header has a “port” number field
- Helps host sort out how to route packets to applications

Your Computer

Port 80

Port 143

Email Client

Packet
For some applications packet retransmissions are not worthwhile
  - Why?
For those applications, we use UDP
UDP is a transport protocol that
  - Does not do retransmissions
  - Does not do congestion control
Congestion Control

- When networks are congested, certain sessions (Source-destination pairs) should reduce offered rates.
  - Today all TCP sessions slow down when they detect packet losses.
  - UDP sessions do not slow down.

- What are some alternative strategies?
  - Have those whose applications aren’t as sensitive slow down more?
    - How would we know which are less sensitive
Pricing within the Internet

- **Customer pays an ISP**
  - Often Flat Rate per month

- **ISP pays a backbone AS**
  - Often just flat rate, dependent on access link speed.
  - Sometimes based on total usage

- **Backbone NSPs peer with each other**
  - Often for free if they exchange comparable amounts of traffic.

- **Overall...**
  - Internet billing today is much more course grained than telephone billing.
Domain Names

IP addresses are inconvenient for people
- 32 bits hard to remember
- 128 bits very hard to remember

Domain names
- e.g. argus.eecs.berkeley.edu
- Easier to remember than IP addresses
- However, we need some way of mapping domain names to IP addresses.
Domain Name System (DNS)

- Root Name Server
  - Berkeley Name Server
    - EECS Name Server
  - UCSC Name Server
    - SoE Name Server
Hierarchy in Addresses vs. Names

Addresses hierarchical in topology
- Maximize “wild cards” and distribute address administration

Names hierarchical in administration
- Single administered organizations often distributed topologically (e.g. ibm.com)
Transport Protocols

- The Internet is unreliable
  - It will make a “best effort” to get your packet to its destination
- Packets can be lost because of
  - Congestion
  - Link errors
  - Routing problems
OSI Layers

Application

Presentation

Session

Transport

Network

Link

Physical

Internet Explorer, Outlook Email, Real Player, ...

TCP, UDP

Internet Protocol (IP), ...

Ethernet, Wi-Fi, SONNET, ...

Modulation Schemes: QAM, OFDM, etc...
Some Typical Topologies

Home Network

- Ethernet Switch
- Router
- DSL Modem
- Telephone Line (to local Office)
Small/Medium Business

Web Site Server

T1 Line

T1 Modem

Router with Firewall

To Local Office
Network Service Provider

Network Access Point

Network Access Point
Large E-Business

Load Balancer

Incoming HTTP Requests

Interconnected with Gigabit Ethernet or other technology

Presentation Logic (Assembling Web page)

Web Servers

Application Servers

Logic Flow of Interaction

Databases

Customers

Merchandise

Orders
Web Caching

- Speed up web page loading by storing previously seen components locally

http://www.ucsc.edu
Akamai Case
Akamai Case

1) Akamai’s technology can be best classified as
   - A) Content Delivery Network
   - B) Database Management System
   - C) Thin Client

2) Akamai’s customers included
   - A) Home users wanting faster Internet access
   - B) Companies with content heavy web sites
   - C) Companies wanting a simple, inexpensive database

3) Which is **not** one way Akamai sold its product/service?
   - A) “Partner” firms like companies who do system integration
   - B) Retailers like Fry’s and Best Buy
   - C) A sales force employed by Akamai
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
- Akamai Server
- NSP 1
- NSP 2
- INTERNET
- Web Page
- Text...
- Large Company
- Web Server

Diagram showing the connection between local offices or ISPs, Akamai servers, and the Internet, with a Web Page and Large Company connected through a Web Server.
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
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

**Consolidated Statements of Operations Data:**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Revenues</strong></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>101,948</td>
<td>1/2,570</td>
<td>527,280</td>
<td>2,511,108</td>
<td>989,359</td>
</tr>
<tr>
<td><strong>Net income (loss)</strong></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>