ISM 50 - Business Information Systems

Lecture 17

Instructor: John Musacchio

UC Santa Cruz

November 25, 2008
Class announcements

- For Tuesday 12/2
  - Folio Article 3 Due
  - Read: Messerschmitt Ch 9 (273-289)

- Upcoming Student Presentations
Student Presentation

Ryan Okrant (Home Depot)
Aside: DB’s in different software stacks

- Which companies are competitors?
- Which are complimenters?
- Which are both!?
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></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></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>

How should mySQL grow in order to meet it’s 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**

<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></td>
<td>Microsoft</td>
<td></td>
<td>Oracle IBM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IBM</td>
</tr>
<tr>
<td>Web Sites 10%</td>
<td>My SQL Cost</td>
<td>Reliability</td>
<td>Scalability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Support</td>
<td>Longevity</td>
</tr>
</tbody>
</table>

- 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>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>Microsoft</td>
<td>Oracle</td>
<td>IBM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reliability</td>
<td>Scalability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Support</td>
<td>Longevity</td>
</tr>
<tr>
<td>My SQL Cost</td>
<td>Stay Put?</td>
<td></td>
<td></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>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>
<tr>
<td>Web Sites 10%</td>
<td>My SQL Cost</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Many of these customers already using MySQL with websites
- Less emphasis on global organization
- 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></th>
<th>Small 20%</th>
<th>Medium 30%</th>
<th>Large 50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise wide data</td>
<td>Microsoft</td>
<td></td>
<td>Oracle</td>
</tr>
<tr>
<td></td>
<td>90%</td>
<td></td>
<td>IBM</td>
</tr>
<tr>
<td>Web Sites</td>
<td>My SQL</td>
<td>Cost</td>
<td>Maybe?</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td></td>
<td></td>
</tr>
</tbody>
</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>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
</tr>
<tr>
<td>Physical</td>
</tr>
</tbody>
</table>
**Physical Layer:** Convey bits over a wire

Bits: 010110...

![Diagram showing voltage over time with noise](image)
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
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 the end of data so the number of ones is always odd
    - 10011 or 10000
  - If receiver counts an even number of ones, then it knows the data was corrupted.
More Link Layer.. -- Ethernet

Want to allow multiple hosts to *share a link*

<table>
<thead>
<tr>
<th>Host A</th>
<th>Host B</th>
<th>Host C</th>
<th>Host D</th>
</tr>
</thead>
</table>

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.
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)
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 Santa Rosa

Truck to Santa Cruz

Post Office Analogy
Network Layer

Destination Address: 128.114.60.202

Host A
128.114.60.200
(IP Address)

Host B

Host C
128.114.60.201

Host D
128.114.60.203

Link 1

Link 2

Link 3
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

- 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

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"

Host A

Host B

Host C

114.211.1.1

114.211.1.2

114.211.1.3

114.211.1.4

Wild Card
Routing Table has Wild Cards

<table>
<thead>
<tr>
<th>Host</th>
<th>IP Address</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host A</td>
<td>114.211.1.1</td>
<td></td>
</tr>
<tr>
<td>Host B</td>
<td>114.211.1.2</td>
<td></td>
</tr>
<tr>
<td>Host C</td>
<td>114.211.1.3</td>
<td></td>
</tr>
<tr>
<td>Host C</td>
<td>114.211.1.4</td>
<td></td>
</tr>
</tbody>
</table>

**Routing Table**

<table>
<thead>
<tr>
<th>Network</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>114.211.1.X</td>
<td>Link 1</td>
</tr>
<tr>
<td>200.261.19.X</td>
<td>Link 2</td>
</tr>
</tbody>
</table>
Internet Routing is Hierarchical

Backbone or NSP: (MCI, ATT)

Autonomous System (AS)

ISP or IAP (CRUZIO, AOL)

AS

Customer AS
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**