Class announcements

- Reading for next Wednesday (May 27):
  - Messerschmitt Chapter 8 p. 233-258 (reading instead of Ch 10)
- Reading for Monday (June 1):
  - Akamai Case
- Student Presentations Wednesday (May 27):
  - Any one looking for extra credit? You can make a presentation about your paper or a news story for X points....
- Student presentation for June 1
  - Tracie Kemmerle

Today's Class

- Student presentations
  - Alma Y. Montes
  - John Kuan
  - Wrap up mySQL
  - Oracle/Sun discussion
  - Protocols and Algorithms
  - Networks

mySQL

How does OSS work?

Two Types of License:
- GPL (General Public License)
  - 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

<table>
<thead>
<tr>
<th>General Software Stack</th>
<th>ERP Software Stack</th>
<th>Web Application Software Stack</th>
<th>Banking Software Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middleware</td>
<td>Oracle or MySQL, IBM, etc</td>
<td>Apache Web Server</td>
<td></td>
</tr>
<tr>
<td>Operating System</td>
<td>MS Windows or other OS</td>
<td>Linux or other OS</td>
<td>IBM z/OS or other OS</td>
</tr>
</tbody>
</table>

- Which companies are competitors?
- Which are complimenters?
- Which are both?
My SQL: market

- Small 20%
- Medium 30%
- Large 50%

Enterprise wide data 90%
Web Sites 10%

My SQL Cost

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

My SQL: Growth Strategy

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

My SQL: Growth Strategy

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

My SQL: Growth Strategy

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

My SQL: Growth Strategy

- Builds on existing brand and strengths
- Market not so big

Oracle Buys Sun! (Announced 4/20)

- Oracle announced it would buy Sun Microsystems for $7.4 B cash
  - Earlier, IBM made a cheaper offer for Sun, but talks fell through
  - Oracle in 2009: strong proprietary DBMS business; gradually gaining market share with its ERP products
  - Recall: by buying Sun, Oracle gets?
    - Sun HW (workstations, servers)
    - Solaris OS (UNIX)
    - Java language leadership, OpenOffice suite
    - MySQL open source DBMS (ISM 50: case study 5/18)
Oracle/Sun Discussion

- “More Oracle databases run on the Solaris Sparc than any other system,” said Oracle CEO Larry Ellison, noting Linux was second. “We’ll engineer the Oracle database and Solaris operating system together. With Sun we can make all components of the IT stack integrated and work well.”

- Oracle buys Sun Microsystems: trying to become the Apple of the enterprise...

- Oracle’s stack of IT stuff now includes:
  - Java;
  - Solaris;
  - Enterprise applications ranging from CRM to ERP to business intelligence;
  - The database (Oracle and MySQL);
  - The middleware;
  - The storage hardware;
  - Cloud computing services;
  - And servers.

Oracle/Sun Discussion

- “It’s clear that Oracle is targeting the next generation data center—as is the rest of the industry. The list of tech titans looking to remake the data center:”
  - IBM;
  - HP;
  - Cisco;
  - Dell;
  - Oracle;
  - Juniper
  - EMC
  - VMware
  - others...

Break into groups of 3 or 4

Discuss

A) Is the acquisition of Sun a basic change in strategy for Oracle - why or why not?

B) What made Sun an attractive acquisition target for Oracle?

C) Has Sun been an Oracle rival, a substitute, or supplier?

D) What’s your best guess at the long term role of Sun in the industry? (What’s the industry?)

Write your ideas down.

What is Microsoft’s fate?

- Have we arrived in the post-Windows era? | Between the Lines | ZDNet.com http://blogs.zdnet.com/BTL/?p=16590

Preparation for networking topics:

Algorithms and protocols

Markup languages

Adapted from
David G. Messerschmitt
Chapter 11, Chapter 15
Algorithm

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

Start turn
Throw dice
Move token number of squares indicated on dice

Move to “jail” square

Yes
Land on “go to jail”?

No
Do not move; follow policies for square (like “pay rent”)

Finish turn

Algorithm building blocks

Start
Action
Action
Action
Finish
Sequence

Start
Start
Start
Action
Decision
Action
Action
Test
Action
Finish
Finish
Finish
Selection
Loop

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

Player 1
Player 2

One-turn algorithm

This is a protocol interaction diagram

Layered Protocols Example

HHC

HHC Application

Send Packet

Palm OS

Networking Infrastructure

Windows OS

Break Messages into Packets

Send Pass. Data As Message

HHC Server Application

HHC Server

Request Pass. Data

Application Level Protocol

HHC

Networking Infrastructure

Link Level Protocol

Windows OS

Send Packet

Palm OS

Networking Infrastructure

Break Messages into Packets

Send Pass. Data As Message

HHC Server Application

HHC Server

Request Pass. Data

Application Level Protocol

HHC

Three simple protocols (text, p. 331)

One-way message: send-receive
Two-way interaction: request-response; publish-subscribe

Example: HTTP (Hyper Text Transfer Protocol)

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

Client Server

Time

Client

Server

UDP (audio, video)

TCP (email, http)

HTTP client (browser)

RSS, ATOM

HTML documents

HTTP server

HTML documents

Example: HTML (Hyper Text Transfer Protocol)

Example: HTTP (Hyper Text Transfer Protocol)

<xml>
  <product>
    <model> Super Widget </model>
    <make> Widgets Incorporated </make>
    <sku> 123456789 </sku>
    <price> $300 </price>
  </product>
</xml>

Markup Languages

A markup language describes the structure of a document
- Based on tags
- Tags denote structural elements like sections, subsections, figures, etc
Internationally standardized, so application independent

Example: HTML

<html>
  <h1> Super Widget </h1>
  <h2> Widgets Incorporated </h2>
  <em> 123456789 </em>
  <p> $300 </p>
</html>

Trend: XML has become the information format of choice for networked applications

Tags Emphasize what the things mean rather than how to format their Presentation.

Example: XML

<xml>
  <product>
    <model> Super Widget </model>
    <make> Widgets Incorporated </make>
    <sku> 123456789 </sku>
    <price> $300 </price>
  </product>
</xml>

Trend: XML has become the information format of choice for many applications.

- Kurt Cagle, O’Reilly Columnist, 12/23/08

- Document file structures
  - MS Office, OpenOffice, ...
- Modern server traffic: almost 100% XML.
  - 20% “SOAP” (runs on top of HTTP and XML)
  - 40% syndication (ATOM, RSS) using XML
  - 40% raw XML
- Enables “semantic web”. Why?
- All DBMS companies as of last year have come up with database products that store entries in XML format natively.
- Why did it take them so long?

XML in Ecommerce example

Supplier

Stuff4U

Product info

From each Supplier sent in XML

Retailer

Stuff4U

Consumer

Amazing Gadget

$500

Super Widget

$300

Trend: XML has become the information format of choice for many applications.

- Kurt Cagle, O’Reilly Columnist, 12/23/08

- Document file structures
  - MS Office, OpenOffice, ...
- Modern server traffic: almost 100% XML.
  - 20% “SOAP” (runs on top of HTTP and XML)
  - 40% syndication (ATOM, RSS) using XML
  - 40% raw XML
- Enables “semantic web”. Why?
- All DBMS companies as of last year have come up with database products that store entries in XML format natively.
- Why did it take them so long?
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...

Physical Layer: Convey bits over a wire

Bits: 010110...

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 the 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

- 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

**Ethernet Continued**

- 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

**Ethernet Switch**

- 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)

**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
- Not Changeable
  - Beginning bits tell nothing useful
  - 6 bytes
  - 281 Trillion

**Switches**
- Steer traffic within a closed network
- Larger networks have thousands of endpoints
- Devices identified by MAC address, assigned by device manufacturer
- Devices typically have the same physical interface type
- Networks relative easy to set up and administer

**Switches**
- Cisco’s first product, originally designed to bridge networks from different vendors
- Now almost all networks run TCP/IP, so router’s function has changed:
  - Core internet routing
  - Gateway to internet in enterprise networks
  - Residential internet connection
- Device IP address assigned by network administrator
- Can route between devices with different physical interface types
- Can support networks with billions of endpoints

**Routers: Network Layer**
- Non-scalable mechanism, like sending letters without a zip code
- Routers steer traffic based on devices’ assigned “geographic” network address
  - Logical address groupings, like ucsc.edu and att.net
  - Hierarchical network grouping allow distributed routing
- Routers can send traffic between devices with different physical interface types
- Summary: Switching is fast and simple, but not as scalable as routing

**Routers versus Switches**
- Switches steer traffic based on end point devices’ fixed address
  - Non-scalable mechanism, like sending letters without a zip code
- Routers steer traffic based on devices’ assigned “geographic” network address
  - Logical address groupings, like ucsc.edu and att.net
  - Hierarchical network grouping allow distributed routing
  - Routers can send traffic between devices with different physical interface types
- Summary: Switching is fast and simple, but not as scalable as routing

**Post Office Analogy**
- Look at beginning of zip code. Make forwarding decision
- Look at address. Make forwarding decision
**Network Layer**

![Diagram of network layer with hosts and links](image1)

**Packet forwarding**

![Diagram of packet forwarding](image2)

**Routing in the Internet**

![Diagram of internet routing](image3)

**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

![Diagram of routing algorithms](image4)
Routing Table has Wild Cards

Internet Routing is Hierarchical

Routing Concerns

- Long routes
- Circular routes
- Hijacking routes
- Route flapping

Link and Network Layer Interaction

OSI Layered Model of Networking

Issues In Networking

- Sharing of Limited Resources
  - How Should A and B share a link with limited bit rate?
**Issues In Networking**

- **Time Division Multiplexing**
  - Gives each connection the use of the link a fixed fraction of time.
  - Fixed fraction of resources reserved for each connection.
  - Technology called *circuit switching*.

  ![Diagram of Time Division Multiplexing](image)

- **Problem**
  - When A is silent, A’s fraction of link goes unused.

**Statistical Multiplexing**

- Because resources aren’t reserved, it’s possible offered load too high.
- Packets are put into a queue.
- If offered load remains too high, queue will fill up and overflow.

![Diagram of Statistical Multiplexing](image)

**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 port numbers**

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

Traffic can overload links
- Failure of statistical multiplexing
- Congestion must be limited in some fashion.

Network congestion

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?

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
- UCSC Name Server
- EECS 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)

OSI Layers

- Application: Internet Explorer, Outlook Email, Real Player, ...
- Presentation:
- Session:
- Transport: TCP, UDP
- Network: Internet Protocol (IP), ...
- Link: Ethernet, Wi-Fi, SONNET, ...
- Physical: Modulation Schemes: QAM, OFDM, etc...

Some Typical Topologies

**Home Network**
- Ethernet Switch
- Router
- DSL Modem
- Telephone Line

**Small/Medium Business**
- Ethernet Switch
- Router with Firewall
- T1 Modem
- Web Site Server
- T1 Line to Local Office

**ISP Topology**
- Telephone Company
- Local Office
- T1 Line
- Telephone Switch
- ISP Point of Presence
- DSL Modem
- Leased Lines to NAP
**Large E-Business**

- **Load Balancer**
- **Incoming HTTP Requests**
- **Presentation Logic (Assembling Web page)**
- **Application Servers**
- **Interconnected with Gigabit Ethernet or other technology**
- **Databases**
- **Customers**
- **Merchandise**
- **Orders**

**Web Caching**

- Speed up web page loading by storing previously seen components locally
- **http://www.ucsc.edu**
- **Cache on Hard Drive**

**Quality of service (QoS)**

- **Packet latency**
  - Time until packet delivered at destination
  - Transmission time, propagation time, queuing delay, processing time
- **Packet loss**
- **Packet corruption**
  - Payload only
  - Normally network will not deliver corrupt packet
Transport services

“Raw” packet service is not what is needed by most applications
Transport services “condition” packet service by adding layers
- Reliable delivery
- Message service
- Session
- Time stamps
- etc

Packet latency affects transport service QoS

Stream of packets

Audio coder

Packet latency

Audio decoder

End-to-end delay

Reliable delivery

Add acknowledgement for each packet
Lost packet can be detected by missing ACK
Lost packet can be retransmitted

Tradeoff:
- Reliable delivery for greater latency
- Latency-sensitive applications must abandon reliable delivery (e.g. remote conferencing)

QoS Guarantees

Source and network enter “session contract”:
- Source promises not to exceed specified traffic parameters for that session
  - Rate and burstiness
- Network promises to limit impairments such as latency, loss, and corruption

Achieving QoS

TCP/IP offers only best-effort
- Every connection gets “best-effort” service

Achieving maximum latency guarantees
- Reserve resources
- Or attach priorities to packets
  - Contract may allow network to delay or discard low-priority packets when necessary
- Application may guarantee traffic “shape”
  - e.g., steady flow rather than bursts

Achieving QoS

- Increase the capacity of the network a lot
- TDMA instead of statistical multiplexing
  - That way traffic from one connection does not effect the quality of another
  - But, we lose the benefits of statistical multiplexing.
- Priority Scheduling?
  - Analogy: first class check-in vs. coach check-in
Quality of Service (QoS) - metrics

- Latency – the time it takes a packet to travel from a sender to receiver.
- Throughput – the rate of the connection in bps.
- Loss – the fraction of packets that get lost.
- Jitter – How much the latency varies over time.

Big advantage of pricing

Congestion pricing uses incentives rather than forced control or policies to affect user/application behavior
- Market mechanism like other goods and services
User/application can determine freely and independently whether use of network during periods of congestion is warranted
Major objection is cost of monitoring and billing

Pricing Today

End User
- Flat Rate most common
  - pays an ISP a flat rate per month
  - does not depend on use

Pricing Alternatives

- Usage based pricing
  - Charge some amount per megabit sent.
  - Those that use more thus, pay more.
  - This is done today for the phone network, as well as data connections over cell phones.
  - Advantages?

Pricing Alternatives

- QoS based pricing
  - Pay a high price for guaranteed QoS
    - Guaranteed throughput, low loss and latency.
  - Pay a lesser price for not-guaranteed (best effort) service.
  - Advantages:
    - QoS costs provider more, so user should pay more.
    - Might improve provider revenue.
  - Disadvantages:
    - Complexity.

Pricing Alternatives

Congestion Pricing
- Idea studied a lot in research community,
  - Pay more when links are congested.
  - This gives an incentive to reduce usage.
  - If fined grained enough, congestion prices could be an alternative to TCP congestion control.
  - Some proposed schemes allow users with greater needs to outbid those with less need.
**Congestion Pricing**

**Advantages?**
- More revenue for provider
- Allows users with sensitive applications to pay more to get the service they require.

**Disadvantages?**
- Complexity
  - This is why it has not caught on.

---

**Cost based Pricing?**

**What are the difficulties with this?**

- Large Fixed Costs.
- Small marginal costs.

---

**Pricing and Accounting**

**What's the incentive for using low quality service?**
- Why should user accept greater latency if less is an option?
- Why should application try to minimize bandwidth, or shape its traffic?
- Answer is good citizenship, or pricing incentives

---

**Pricing within the Internet**

- Customer pays an ISP
- 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.