Class announcements

- FINAL EXAM
- Tuesday 3/15/2011
- Time: 8:00-11:00 a.m.

Click to add text
Student Presentations
What does a network do?

- Transport data from one host to another
  - Host allocation
  - Routing

- Millions of users/applications/hosts share the same network
  - Resource sharing
  - Congestion control
Network congestion

Traffic can overload links
- Failure of statistical multiplexing

Congestion must be limited in some fashion
Carried traffic

Congestion instability

Network “capacity”

Increasing portion of network traffic is resent packets

Offered traffic

Social optimum
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?
Akamai Case
Web Caching

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

http://www.ucsc.edu

Local Cache

Web server
Web Caching can also happen at Proxy Server at ISP.
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 coarse grained than telephone billing.
Internet Bottlenecks

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

- **Backbone** - Plentiful, but some shortage within metropolitan areas. NAP (network access points)

- **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...**
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
Akamai EdgeSuite

Local Office or ISP

Large Company
Web Server

INTERNET
NSP 2
NSP 1

Movies in Santa Cruz after 8pm?

Web Page
Movie A
Movie C

Construct Page
Web Page
Movie A
Movie B
Movie C

Movie A
Movie B
Movie C

Web Page
Movie C
Movie C
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
Technology

- 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>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>
We count in base 10 by powers of 10:

\[ 10^1 = 10, \quad 10^2 = 10 \times 10 = 100, \quad 10^3 = 10 \times 10 \times 10 = 1000, \]
\[ 10^6 = 1000000 \]

Computers count by base 2:

\[ 2^1 = 2, \quad 2^2 = 2 \times 2 = 4, \quad 2^3 = 2 \times 2 \times 2 = 8, \quad 2^{10} = 1024, \quad 2^{20} = 1048576 \]

So in computers, the following units are used:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 kilobyte (KB)</td>
<td>1,024 bytes</td>
</tr>
<tr>
<td>1 megabyte (MB)</td>
<td>1,048,576 bytes (1,024*1,024)</td>
</tr>
<tr>
<td>1 gigabyte (GB)</td>
<td>1,073,741,824 bytes</td>
</tr>
<tr>
<td>1 terabyte (TB)</td>
<td>1,099,511,627,776 bytes</td>
</tr>
<tr>
<td>1 petabyte (PB)</td>
<td>1,125,899,906,842,624 bytes</td>
</tr>
</tbody>
</table>
Applications and the Organization

Click to add text
Build vs. Buy?

Purchase off the Shelf
+ less time and cost
+ benefits of using a “standard” solution
+ support available
- must mold org to app
- no potential for competitive advantage

Outsource
+ developers not as familiar with org as you
+ more opportunity for customizing than off the shelf
- contractor may share knowledge with competitors
- contractor may have too much bargaining power

Make
+ most customizable of 3
+ easier iteration between conceptualization and development needed
- most risky
- org may lack competency to do it
Application Lifecycle

It is important to think beyond acquiring an application

- How do we come with the idea?
- How do we architect it?
- How do we implement?
- How do we extend and maintain it?

For this reason, the software engineering community came up with:

- Application Lifecycle Model
Application Lifecycle

Stages:
1. Conceptualization
2. Analysis
3. Architecture Design
4. Development Evolution
5. Testing and Evaluation
6. Deployment
7. Operations, Maintenance, and Upgrade
1) Conceptualization

What is the vision?

- What are the objectives?
- What is the business case?

- EXAMPLE: HHC to inform flight attendants which passengers are low and high value.
  - Present diagram to FA’s
  - HHC customer info updated wirelessly at gate
  - Also has reporting function for misbehaving passengers.

- Business Case:
  - Increase repeat business from high value customers.
1) Conceptualization -- Example:
2) Analysis

- Describe what the application will do.

- Enough info to allow “stakeholders” to review idea
  - Management, end-users, operators and administrators, maintenance organization, suppliers and customers

- Don’t make highly detailed specifications

- Describe scenarios in which it is used
  - “Use Cases”
2) Analysis -- Example

- **Example: Scenarios:**
  - **NORMAL FUNCTION**
    - When at gate, passenger data of next flight are sent to HHC
    - HHC displays info on color coded seat map
    - If FA clicks on seat she/he gets more info about passenger
  - **REPORTING FUNCTION**
    - FA wants to report that passenger in 13F is bad.
    - FA clicks “report pass.” button followed by 13f
    - HHC finds from its data that John Doe is in 13f
    - When HHC is in radio range of WiFi AP, HHC tells server that John Doe is bad.
3) Architecture Design

- Decompose the application into subsystems
  - Hardware, software
  - Try use commercial off the shelf subsystems
  - Try to use standard infrastructure layers
    - Operating system, network, middleware, etc.
3) Architecture Continued

- Define the functionality, interaction and interfaces of subsystems
- While doing this, consider
  - **Scalability**
    - How easily can we increase the number of users and maintain performance? (increasing level of activity and capability)
  - **Extensibility**
    - How easily can we add new features in the future?
  - **Administration**
    - How much work will it take by humans to keep this running properly?
    - (Remember Sun thin vs fat client discussion)
4) Development Evolution

- **Develop the details**
  - Develop/program custom subsystems
  - Have contractor build outsourced pieces
  - Put together with off-the-shelf components (integration)

- **Incremental**
  - Start with simplest implementation and get it working
  - Later add more features.
5) Testing

- A must!

- If architected well, we can test subsystems independently.

- Alpha test - offline test of prototype

- Beta test - test in intended environment with cooperative users
  - Example - give HHC to initial group of FA’s
6) Deployment

- **Convert from previous processes if necessary**
  - Example: CISCO ERP (all at once)
  - Or, you could do incrementally

- **Train users**
  - Example: Frito-Lay HHC

- **Data importation**
  - (if necessary)
7) Operations, Maintenance, Upgrade

- Continuous Process
- Maintain Application
- Repair Problems
- Correct performance shortcomings *(Cisco ERP)*
- Add features
Application Lifecycle Model concluding remarks

- ALM rarely followed precisely
- Many times projects loop between stages
- ALM followed more closely in larger companies

Alternative:

- Rapid Iterative Prototyping
  - (Cisco did some of this in the ERP case.)