Weekly Syllabus

1. Scalability: *(Jan.)*

2. Agile Practices

3. Ecology/Mashups

4. Browser/Client

5. Data/Server: *(Feb.)*

6. Security/Privacy

7. Analytics*

8. Cloud/Map-Reduce

9. Publish APIs: *(Mar.)*

10. Future

* assignment due
Administrative Stuff

• First team submissions due now

• About a dozen RSVPs for the March 10 Demo Lunch
  • Still a little more outreach to do

• On a separate note
  • class action suit filed against Google for Buzz
  • one click Buzz removal option
Tuesday Lecture Review

• Study your users, build features to meet their wants

• Find a positive feedback growth cycle, then iterate
When Analysis Becomes a Business Model

- Typos cause lost revenue on eBay, lost Google referrals
  - classic arbitrage business becomes possible
- Typosquatters get revenue from ads clicked by people seeking another site
- Resell cheaply bought items on eBay due to misspelled marketing copy
- Repackage and optimize content for better search engine ranking
- Optimize keyword choices for online advertising
- Autocompletion in search engines will alter this phenomenon
Internal Analytics

- Failures, Errors and Bugs oh my!
  - development progress tools
  - site health

- Why is my site slow?
  - server edition
  - browser edition
Make Internally Generated Data Communicate

• Measure all data generated in development process
  • prioritize bug fixes to be a part of ongoing development
  • identify trouble spots in feature set
  • make data accessible to entire team

• Record release process activity
  • automate as much as possible
  • deploy test environments using same process
Site Health Analytics

• Log all site activity

  • Separate logging DB / table

    • log DB has a write many, read few usage pattern

• Consolidate server process logs

  • DB, web server, cache, OS/process load

  • Add 64 bit random ‘salt’ to key to uniquify timestamps

• Eventually plan for multiple server logs
Make Server Data Communicate

• Use Nagios + Lucene or Splunk to improve access to server events

• Make site failures quickly diagnosable, fixable
  • Send alerts for unusual failures

• Use log files as a measurement of server activity
  • Plan server scaling based on load
  • Iteratively optimize worst performing operations

• Can support 1,000,000 monthly unique users with 2 servers
Dos and Don’ts of Site Monitoring

• Texting/Paging is cheap so set it up

• Ping/email once a day to verify that the monitor system is live

• Avoid cascading error notifications
  • when things go bad you don’t want to sift through 100s of messages
  • good error prioritization helps failure diagnosis accuracy

• Status change notifications should always be actionable
  • gratuitous messaging causes active disregard of the system
Homeland Security Advisory System Example

• Eight years of unchanging information gets ignored by people
  • No one really feels the need to adjust their behavior any more
  • Has the system actually affected security in any way?

• Similar in lack of information value to the doomsday clock
Important IT Analytics Basics

• .log files - /var/log & slow query log

• cron
  • schedule log cleanup

• syslog
  • centralize log data management

• nagios
  • liveness checks, inventory, alerts, etc.
Nagios Screenshots

<table>
<thead>
<tr>
<th>Host</th>
<th>Service</th>
<th>Status</th>
<th>Last Check</th>
<th>Duration</th>
<th>Attempt</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>ayamon.com</td>
<td>DNS</td>
<td>OK</td>
<td>01-11-2008 11:45:08</td>
<td>2d 1h 48m 21s</td>
<td>1/3</td>
<td>DNS OK - 0.017 seconds response time. ayamon.com returns 208.64.136.202</td>
</tr>
<tr>
<td></td>
<td>FTP</td>
<td>OK</td>
<td>01-11-2008 11:44:11</td>
<td>0d 0h 14m 16s</td>
<td>1/3</td>
<td>FTP OK - 10.261 second response time on port 21 [220 ProFTPD 1.3.0 Server (4Admin(tm) FTP Server) [208.64.136.202]]</td>
</tr>
<tr>
<td></td>
<td>HTTP</td>
<td>OK</td>
<td>01-11-2008 11:48:06</td>
<td>0d 2h 0m 21s</td>
<td>1/3</td>
<td>HTTP OK HTTP/1.1 200 OK - 10363 bytes in 0.433 seconds</td>
</tr>
<tr>
<td></td>
<td>IMAP</td>
<td>OK</td>
<td>01-11-2008 11:46:36</td>
<td>2d 1h 46m 51s</td>
<td>1/3</td>
<td>IMAP OK - 0.202 second response time on port 143 [* OK [CAPABILITY IMAPrev1 UIDPLUS CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT QUOTA IDLE ACL ACL2=UNION STARTTLS] Courier.IMAP ready. Copyright 1998-2004 Double Precision, Inc. See COPYING for distribution information.]</td>
</tr>
<tr>
<td></td>
<td>PING</td>
<td>OK</td>
<td>01-11-2008 11:46:34</td>
<td>0d 1h 42m 21s</td>
<td>1/3</td>
<td>OK - 208.64.136.202: rta 97.770ms, lost 0%</td>
</tr>
<tr>
<td></td>
<td>SMTP</td>
<td>OK</td>
<td>01-11-2008 11:44:37</td>
<td>1d 1h 58m 51s</td>
<td>1/3</td>
<td>SMTP OK - 0.401 sec. response time</td>
</tr>
<tr>
<td>dev1</td>
<td>/ Disk Usage</td>
<td>OK</td>
<td>01-11-2008 11:47:35</td>
<td>1d 2h 3m 21s</td>
<td>1/3</td>
<td>DISK OK - free space: / 6497 MB (60% inode=88%)</td>
</tr>
<tr>
<td></td>
<td>/dev1.html</td>
<td>OK</td>
<td>01-11-2008 11:48:08</td>
<td>0d 2h 44m 46s</td>
<td>1/3</td>
<td>Disk ok - 6.34G (57%) free on /DEV1/HTML</td>
</tr>
<tr>
<td></td>
<td>/boot Disk Usage</td>
<td>OK</td>
<td>01-11-2008 11:48:02</td>
<td>0d 2h 41m 21s</td>
<td>1/3</td>
<td>DISK OK - free space: /boot 223 MB (91% inode=99%):</td>
</tr>
<tr>
<td></td>
<td>/dev/vda</td>
<td>OK</td>
<td>01-11-2008 11:47:36</td>
<td>1d 1h 45m 31s</td>
<td>1/3</td>
<td>Id: 1, Status=11 (PreFailure , Online), Value=200, Threshold=51, Passed</td>
</tr>
<tr>
<td></td>
<td>/home Disk Usage</td>
<td>OK</td>
<td>01-11-2008 11:48:09</td>
<td>0d 2h 40m 19s</td>
<td>1/3</td>
<td>DISK OK - free space: /home 2437 MB (84% inode=93%):</td>
</tr>
<tr>
<td></td>
<td>/store Disk Usage</td>
<td>OK</td>
<td>01-11-2008 11:45:23</td>
<td>1d 2h 3m 49s</td>
<td>1/3</td>
<td>DISK OK - free space: /store 683 MB (28% inode=99%):</td>
</tr>
<tr>
<td></td>
<td>/tmp Disk Usage</td>
<td>OK</td>
<td>01-11-2008 11:45:23</td>
<td>1d 2h 3m 49s</td>
<td>1/3</td>
<td>DISK OK - free space: /tmp 1109 MB (97% inode=99%):</td>
</tr>
<tr>
<td></td>
<td>Backups: Home Dirs</td>
<td>OK</td>
<td>01-11-2008 11:44:40</td>
<td>0d 2h 43m 49s</td>
<td>1/3</td>
<td>/store/backups/homedirs/root.tar.gz is OK (0d 5h 41m 40s old, 184094422 bytes)</td>
</tr>
<tr>
<td></td>
<td>Backups: Mondeo Rescue</td>
<td>OK</td>
<td>01-11-2008 11:45:08</td>
<td>0d 2h 2m 30s</td>
<td>1/3</td>
<td>/store/backups/mondorescue-1.so is OK (4d 8h 22m 2s old, 730595328 bytes)</td>
</tr>
<tr>
<td></td>
<td>Backups: MySQL</td>
<td>CRITICAL</td>
<td>01-11-2008 11:47:16</td>
<td>2d 1h 45m 50s</td>
<td>3/3</td>
<td>CRITICAL: mysql_2008-01-02_07h00m.Wednesday.sql.gz is too old (9d 4h 47m 16s old)</td>
</tr>
<tr>
<td></td>
<td>Backups:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Nagios Screenshots
Server Performance

- Automate and time server reboot procedure
  - it is becoming possible to completely boot Linux in 5-10 seconds
  - bootchart visualizes the boot process
- Virtual memory page swapping can kill a busy server very quickly
  - stay below 90% usage for best performance
- Don’t let average server load rise above 50% of capacity
  - allow 2-1 margin for peak usage
Database Performance

- Analyze query logs
  - find ratio of read vs. write DB access

- Slow query log
  - even an infrequent slow query can kill performance of fast queries

- An index addition/change can improve performance by orders of magnitude

- Good rule of thumb
  - Force all queries to return no more than some fixed k items
Load Testing & Stress Monitoring

- Difficult to realistically simulate high load situations

- Develop & refine a model of system bottlenecks
  
  - Network - max out the connection with large data requests
  
  - can affect bandwidth costs

- Web Server - max out connections

- Cache memory - max out server memory

- Database - max out connections & query load
• Reduction
  • fewer requests, DNS lookups
  • minify javascript
  • gzip components
  • eliminate duplication

• Organization
  • CSS placement, use & expiration, script placement, use & expiration
Worth Checking Out

• Logging howto
  • http://www.campin.net/newlogcheck.html

• Nagios
  • http://www.nagios.org/

• Firebug, Yslow
  • http://www.getfirebug.com/
  • http://developer.yahoo.com/yslow/
Q & A Topics

- Project next steps

- Scalable engineering through bootstrapping
  - start small
  - build tight feedback process
    - test
    - communicate
  - get over mistakes quickly, expand successes
Scalable Web Programming

CS193S - Jan Jannink - 2/18/10