

# Scalable Web Programming

---

CS193S - Jan Jannink - 2/18/10

# 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

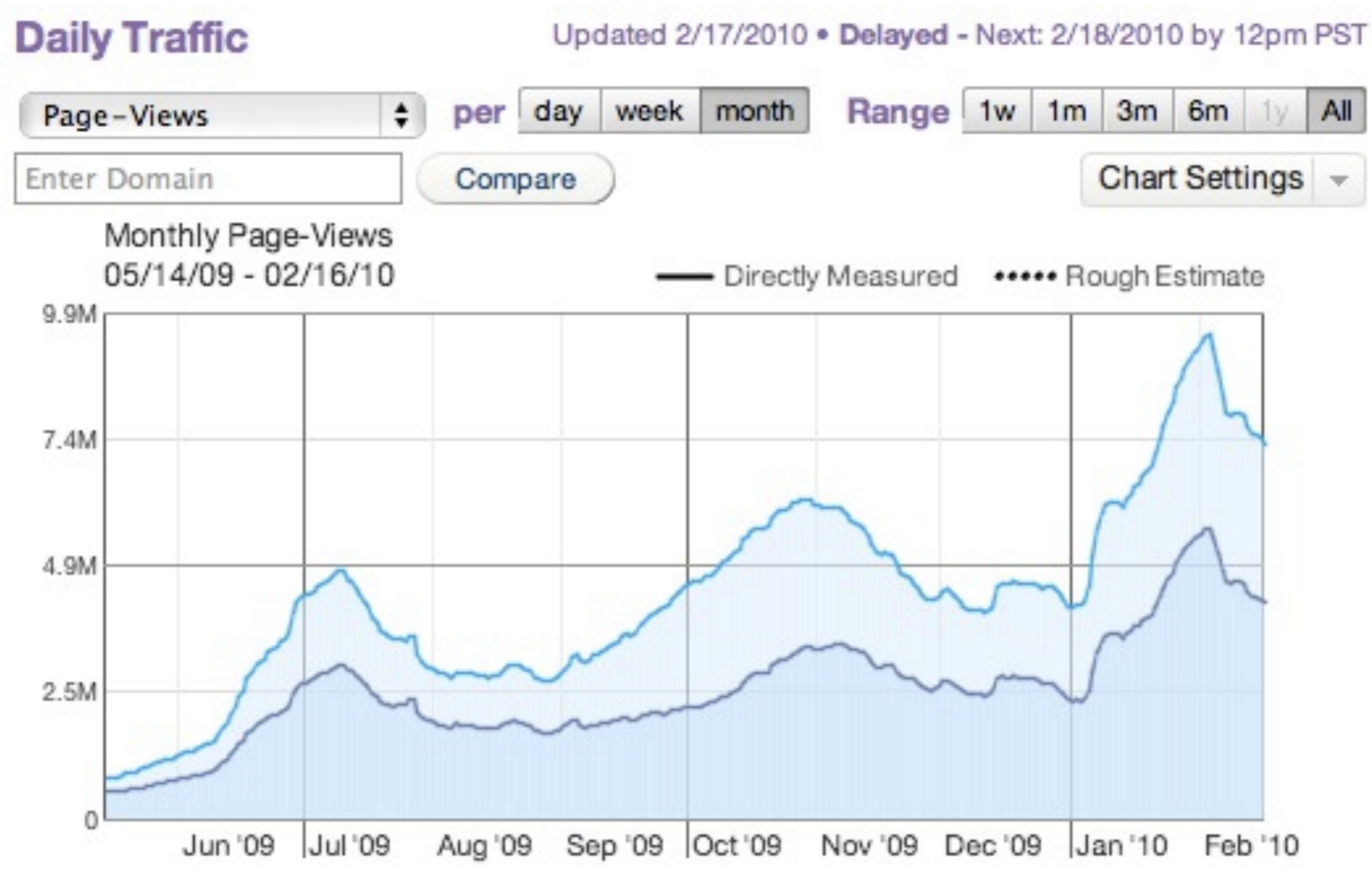
# Administrative Stuff

---

- 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 on 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

---

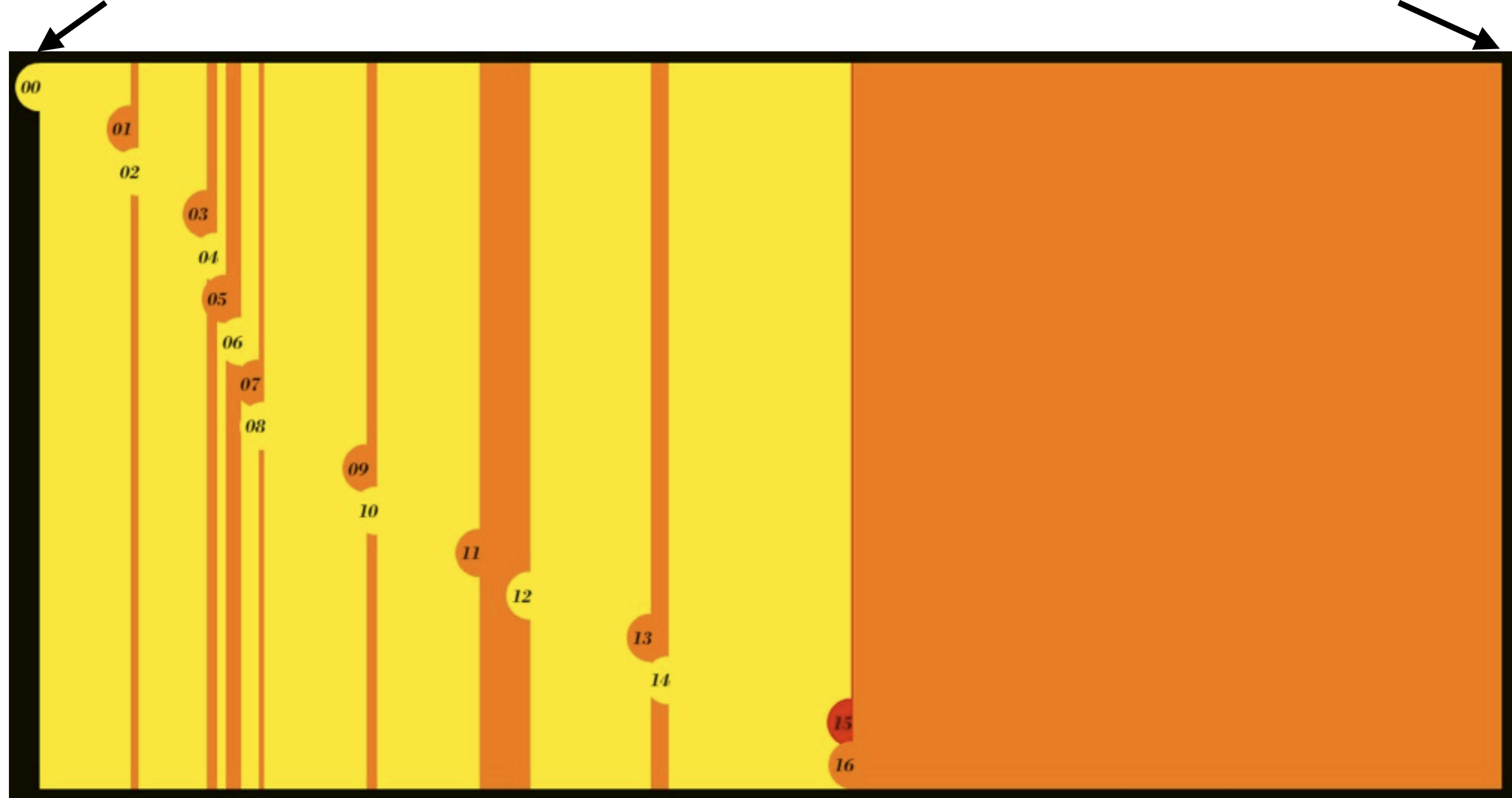
- Texting/Paging is cheap so set it up
- Ping/email once a day to verify that the monitor system is live
- Avoid cascading error notifications
  - when things go bad you don't want to sift through 100s of messages
  - good error prioritization helps failure diagnosis accuracy
- Status change notifications should always be actionable
  - gratuitous messaging causes active disregard of the system

# Red Orange Yellow Blue and Green .com

---

3-12-02

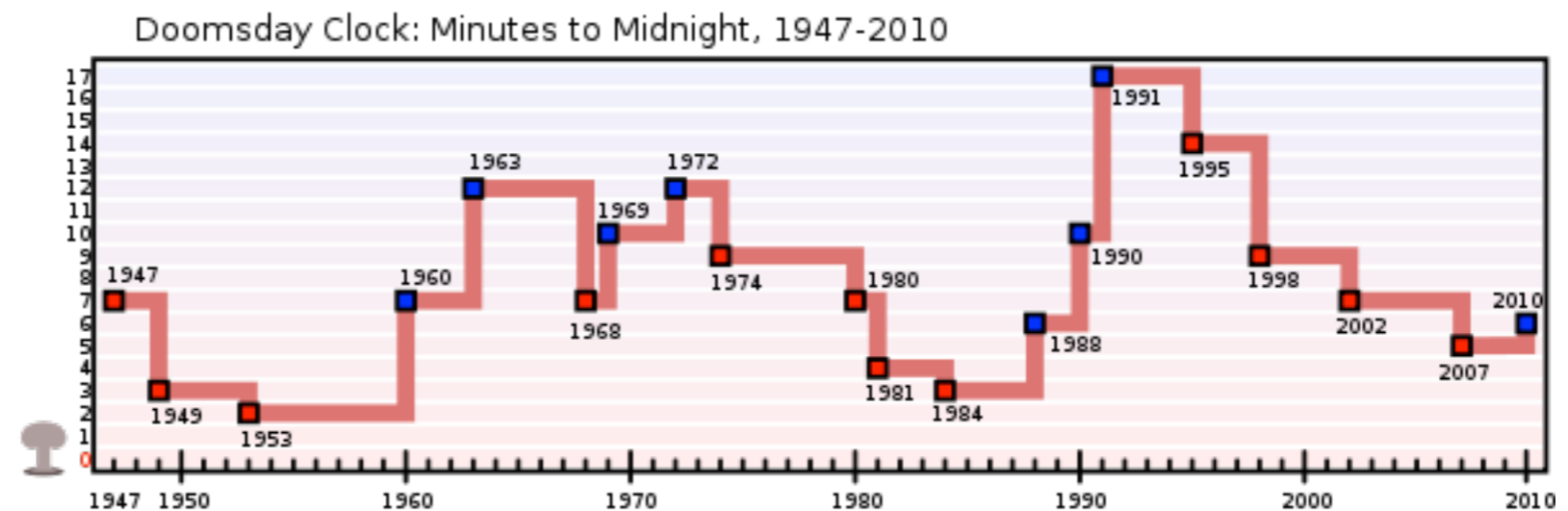
2-18-10



# 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

**Current Network Status**

**Current Network Status**  
 Last Updated: Fri Jan 11 11:48:27 CST 2008  
 Updated every 90 seconds  
 Nagios® 3.0rc1 - [www.nagios.org](http://www.nagios.org)  
 Logged in as nagiosadmin

[View History For all hosts](#)  
[View Notifications For All Hosts](#)  
[View Host Status Detail For All Hosts](#)

Host Status Totals			
Up	Down	Unreachable	Pending
17	0	0	0

All Problems		All Types	
0	17		

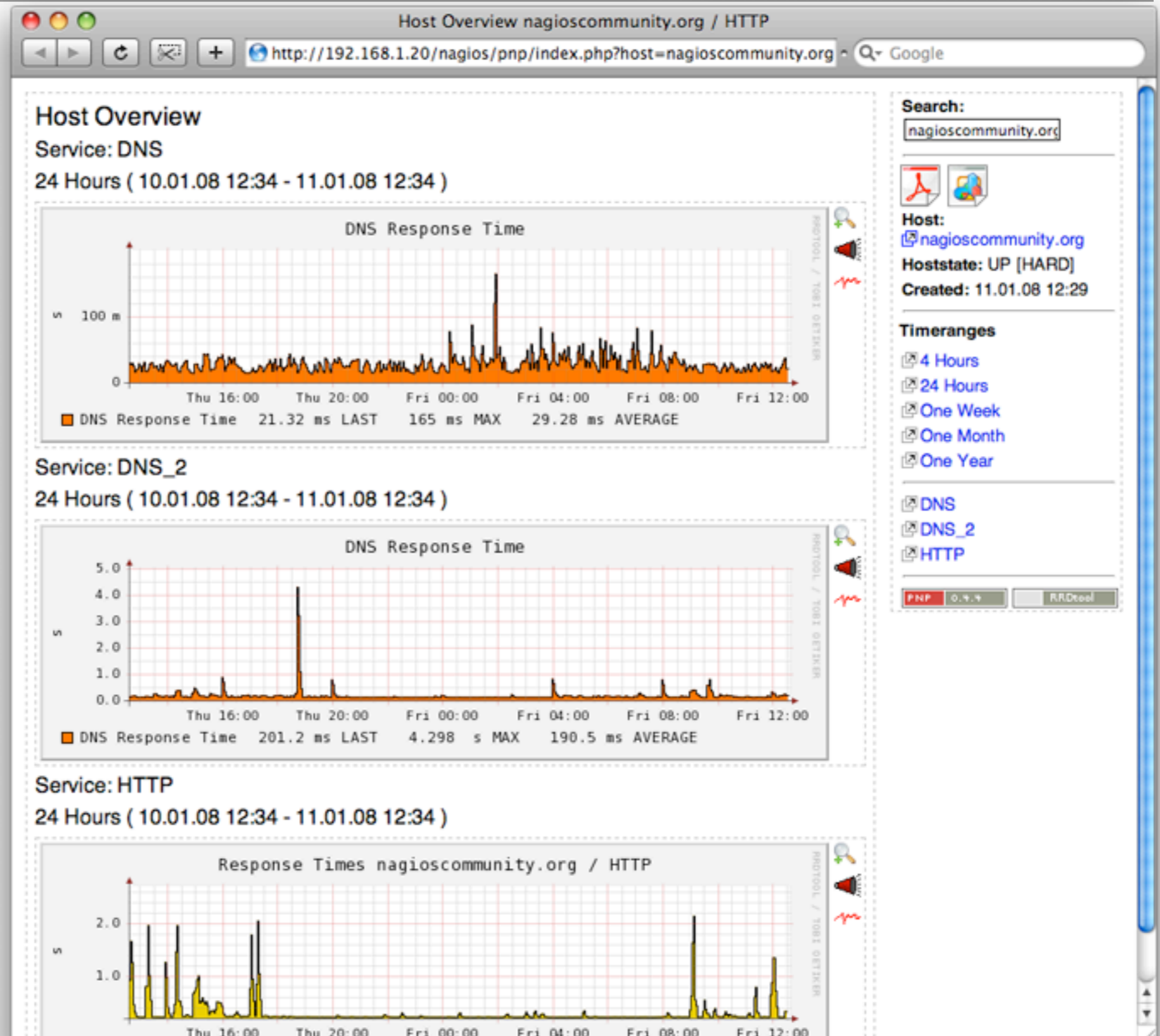
Service Status Totals				
Ok	Warning	Unknown	Critical	Pending
169	4	0	2	0

All Problems		All Types	
6	175		

**Service Status Details For All Hosts**

Host ↑↓	Service ↑↓	Status ↑↓	Last Check ↑↓	Duration ↑	Attempt ↑↓	Status Information
<a href="#">ayamon.com</a>	<a href="#">DNS</a>	OK	01-11-2008 11:45:08	2d 1h 48m 21s	1/3	DNS OK: 0.017 seconds response time. ayamon.com returns 208.64.136.202
	<a href="#">FTP</a>	OK	01-11-2008 11:44:11	0d 0h 14m 16s	1/3	FTP OK - 10.261 second response time on port 21 [220 ProFTPD 1.3.0 Server (4Admin(tm) FTP Server) [208.64.136.202]]
	<a href="#">HTTP</a>	OK	01-11-2008 11:48:06	0d 23h 0m 21s	1/3	HTTP OK HTTP/1.1 200 OK - 10363 bytes in 0.433 seconds
	<a href="#">IMAP</a>	OK	01-11-2008 11:46:36	2d 1h 46m 51s	1/3	IMAP OK - 0.202 second response time on port 143 [* OK [CAPABILITY IMAP4rev1 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.]
	<a href="#">PING</a>	OK	01-11-2008 11:46:34	0d 1h 42m 21s	1/3	OK - 208.64.136.202: rta 97.770ms, lost 0%
	<a href="#">SMTP</a>	OK	01-11-2008 11:44:37	1d 18h 58m 51s	1/3	SMTP OK - 0.401 sec. response time
<a href="#">dev1</a>	<a href="#">/ Disk Usage</a>	OK	01-11-2008 11:47:35	1d 23h 42m 21s	1/3	DISK OK - free space: / 6497 MB (60% inode=88%):
	<a href="#">//dev1/html</a>	OK	01-11-2008 11:48:08	1d 23h 40m 46s	1/3	Disk ok - 6.34G (57%) free on \\DEV1\HTML
	<a href="#">/boot Disk Usage</a>	OK	01-11-2008 11:48:02	1d 23h 41m 21s	1/3	DISK OK - free space: /boot 223 MB (91% inode=99%):
	<a href="#">/dev/sda S.M.A.R.T.</a>	OK	01-11-2008 11:47:36	1d 23h 40m 51s	1/3	Id= 1, Status=11 (PreFailure , OnLine ), Value=200, Threshold= 51, Passed
	<a href="#">/home Disk Usage</a>	OK	01-11-2008 11:48:09	1d 23h 40m 19s	1/3	DISK OK - free space: /home 2437 MB (84% inode=93%):
	<a href="#">/store Disk Usage</a>	OK	01-11-2008 11:45:23	1d 23h 44m 19s	1/3	DISK OK - free space: /store 683 MB (28% inode=99%):
	<a href="#">/tmp Disk Usage</a>	OK	01-11-2008 11:45:23	1d 23h 44m 19s	1/3	DISK OK - free space: /tmp 1109 MB (97% inode=99%):
	<a href="#">Backups: Home Dirs</a>	OK	01-11-2008 11:44:40	1d 23h 43m 49s	1/3	/store/backups/homedirs/root.tar.gz is OK (0d 5h 41m 40s old, 184094422 bytes)
	<a href="#">Backups: Mondo Rescue</a>	OK	01-11-2008 11:45:08	1d 23h 43m 19s	1/3	/store/backups/mondo/mondorescue-1.iso is OK (4d 8h 22m 2s old, 730595328 bytes)
	<a href="#">Backups: MySQL</a>	CRITICAL	01-11-2008 11:47:18	2d 1h 45m 50s	3/3	CRITICAL: mysql_2008-01-02_07h00m.Wednesday.sql.gz is too old (9d 4h 47m 16s old)
	<a href="#">Backups:</a>	OK	01-11-2008 11:46:08	1d 23h 42m 20s	1/3	/store/backups/system/etc.tar.gz is OK (0d 6h 45m 52s

# 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

# High Performance Websites - Steve Souders

---

- 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