

# One Server, 2.8 M page views a day

Khalid Baheyeldin

DrupalCon San Francisco 2010

<http://2bits.com>

The logo for '2bits' features the number '2' in a bright green color and the word 'bits' in a dark grey, stylized font. The '2' is positioned to the left of 'bits'. The entire logo is set against a light grey background and has a subtle reflection effect below it.



# Agenda



- Introduction
- Site overview
- Challenges
- Hardware
- Software
- Drupal
- Some graphs
- Lessons learned





# About Khalid



- 25 years in software development and consulting
  - Mainframe, UNIX, Linux, LAMP
- Drupal since 2003, develop, contribute, consult
- Several core contributions to the Drupal project
  - Site maintenance, `hook_watchdog()`
- Over 37 modules contributed on drupal.org
  - Userpoints, Favorite Nodes, Flag Content, Nodevote





# About Khalid (cont'd)



- Member of the General Assembly of the Drupal Association
- Co-Founder of the Waterloo Region Drupal Users Group (Southern Ontario, Canada)





# About 2bits.com



- Founded in 1999, Drupal since 2003
- Services
  - Drupal / LAMP performance optimization and tuning
  - Server provisioning for performance and uptime
  - Manage large sites for clients
  - Drupal custom module development
- Clients from USA, Greece, Argentina, China, Germany, ...
- In depth articles, and testimonials at <http://2bits.com>





# Is Drupal ...



- How many of us have heard:
  - “Drupal is slow ...”
  - “Drupal is resource intensive ...”
    - Memory hog ...?
    - CPU hog ...?





# Site overview



- Entertainment site
- Traffic heaviest in evening and weekends
- Revenue is advertising driven
- Alexa traffic rank (around 3,400)
- Netcraft site rank (top 250 to 450)
  - Higher than bit.ly, guardian.co.uk, bankofamerica.com, netflix.com





# Site Traffic 1 year ago



- March/April 2009
  - 404,000 page views / day peak
  - 96,000 visits / day peak
  - 10.5 million page views / month
  - 2.58 million visits / month
- Stats
  - 5,125 registered users
  - 7,682 nodes







# Site Traffic now



- March/April 2010
  - 2.799 page views / day (peak)
  - 581,000 visits / day (peak)
  - 70.1 million page views / month
  - 15.4 million visits / month
- Stats
  - 65,180 registered users
  - 14,652 nodes





# Site Traffic



Sunday, April 18, 2010

**Visits: 581,828**

**Pageviews: 2,465,284**

Thursday, April 8, 2010

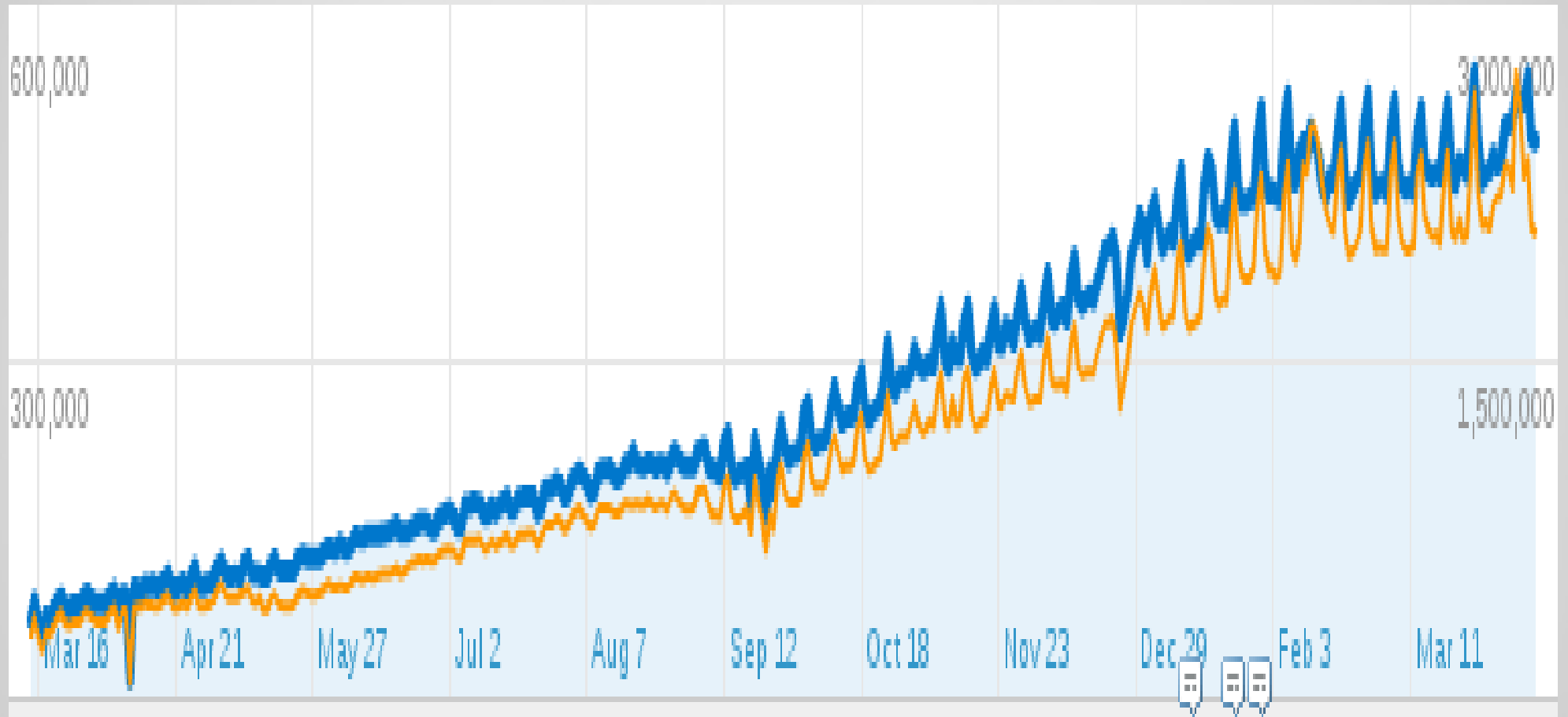
**Visits: 540,997**

**Pageviews: 2,799,536**





# Site Traffic





# How did we do it?



- Boost
- AdvCache
- AuthCache
- Pressflow
- Mercury
- Nginx
- PHP-fpm
- CDN
- Squid
- Varnish
- MongoDB
- Cassandra
- Amazon Cloud
- Server cluster/farm
- Solid State Disks





# How did we do it?



Surprise!

None of the above ....





# Challenges



- Site “locks up” every few days.
  - Excessive swapping under load
- “Regular” off the rack server:
  - Many services running (DirectAdmin, Courier, FTP)
  - Untuned: No PHP accelerator, no memcache
  - Lots of useless stuff that adds bloat (Apache modules)
  - Could not get stuff installed properly (Munin, current version of htop, ...etc.)





# Challenges



- Using core caching
  - Cache table had 50,000+ rows, and used 574MB, mainly cache\_menu entries
  - Cache page table has 34,000+ rows, and used 194MB (27,000+ were 404s!)
- One view on votingapi caused a slow query (2+ seconds, file sort and temporary table)
- Locking on the table level (users/sessions were MyISAM)





# Challenges



- Concern about fast rate of growth
- Need assistance scaling the site







# Performance Assessment **abits**

- Conducted a performance assessment engagement
  - Observation/Monitoring
  - Analysis
  - Recommendations
- Installed a new server from scratch
  - Only the stuff that is needed (no Java)
  - Custom tuned for Drupal





# Refactoring



- Conducted moderate refactoring of the site
- Removed all unnecessary modules
  - Less code to execute
  - Less memory to consume
  - Less database queries
- Removed on problematic view (long running query)
- Replace CCK with a custom module





# Hardware



- Initially on a 4-core server
- Now on
  - Dell PowerEdge 2950 Rack Mount Chassis
  - 2 X Quad Intel Xeon @ 2.5GHz, 1333MHz FSB
  - 8GB FB 800MHz Memory
  - PERC Integrated Raid Controller
  - 4 X 300GB, SAS, 15K RPM Hard Drive
  - RAID 1 Configuration
- \$400 per month





# Hardware



- Use current CPUs
- 64 bits
- More cores if all-in-one servers
  - Opterons
  - Xeons
- Not a Mac G5!





# Disks



- Fast disks (15,000 RPM)
- Separate spindles for
  - MySQL
  - Everything else
- Ideally ...
  - Root for operating system
  - Logs
  - Web Root
  - MySQL





# Software



- Use a server distro, install only what is needed
  - Ubuntu Server Edition 8.04 LTS 64bit
- Apache
  - MPM Worker (threaded, less memory for static)
  - Apache fcgid (less memory, less MySQL connections)
- PHP
  - FastCGI (stable with fcgid, 11-14 ms page time)
  - APC 3.0.19 (installed via pecl)





# Software



- Munin for monitoring, and historical resource usage
- Awstats for statistics (and comparison with Google Analytics)





# Drupal



- No commenting on the site
- Voting on nodes is enabled
- Book module for content hierarchy
- Only 44 modules enabled total
- 2 themes (garland for administration)







# Simplicity vs. Complexity

- Regular Drupal sites have 120 or more modules. Had nightmares on sites with 200 modules, then had a client with 231 modules!
- Simplicity is good for scalability, but also good for security updates, maintenance, version upgrades, and refactoring
- E. Dijkstra: Simplicity is a prerequisite for reliability
- Complexity is a disease! Caused by ...





# Chronic Featuritis



- Defined as:
  - “The pathological condition of the site's owner(s) having the burning desire to cram every feature available for download as a module on drupal.org, and asking for more features and new modules, in the hope that the site will be the next Facebook or Twitter ...





# Simplicity vs. Complexity **abits**

- Client with 300,000 page views per day, running on 4 web servers + 4 database servers
  - 6 seconds page load times!
  - “Only” 144 modules
  - Have their parallel application inside the theme (path routing, queries and all)
  - Calls to slow modules from all over the theme (relatedcontent)
  - Calls to `curl_exec()` and `shell_exec()` to load pages!





# Drupal modules



- Removed unnecessary modules (CCK)
- Custom modules (only 2)
  - One is form alters and customizations
  - The other is a CCK replacement for a specific content type





# Drupal modules



- captcha, image\_captcha, comment\_subject, avatar\_selection, votingapi, fivestar, fivestar\_comment, taxonomy\_browser
- nodewords, nodewords\_basic
- page\_title, profile\_csv, session\_expire
- token, token\_actions
- views, views\_ui, views\_export
- Devel





# Drupal modules



- customerror, admin\_menu, memcache\_admin, blockcache\_alter, googleanalytics, fckeditor, local\_menu, pathauto





# Drupal



- Block cache enabled
  - With block\_cache\_alter module
- CSS aggregation enabled
  - Reduced the number of hits considerably





# Taxonomy Browser



- Can be detrimental to the site if misconfigured
- Form alters to limit the number of terms to search on
- Otherwise the SQL query from hell with 30 joins!







# memcache



- Essential for the site
- Page cache with cache lifetime 15 minutes
- Extended it by having our own customcache.inc
  - Replacements of certain strings in the cached page
  - Add timestamps and generation times





# memcache bins



- cache 8MB
- cache\_block 64MB
- cache\_filter 64MB
- cache\_form 128MB
- cache\_page 768MB
- cache\_menu 128MB
- cache\_update 8MB
- cache\_views 4MB
- cache\_views\_data 2MB
- cache\_content 2MB
- cache\_nodewords 2MB





# memcache



- Watch for the input format in nodes and blocks.
  - PHP format is not cached
  - Some use it to embed javascript for ads and such
- Created a new Raw format (no line break)
- Now nodes and blocks are cached ...





# Fast Path Cache



- Drupal has a feature called fast path cache
- Your .inc implements a function called `page_cache_fastpath()`
- Experimented with it, but certain things were broken (mainly forms due to tokens, search, login)
- Reverted back to regular page cache





# syslog



- Available since Drupal 6.x
- Uses the pluggable `hook_watchdog()`
- Reduces the load on the database
- Writes to the file system
- Drawback: web user can't see the log entries (has to be checked from the file system)
- Applications available to push the log somewhere else





# Patches



- Patching is bad, unless ....
  - Kept to a minimum
  - Properly managed (VCS)
- #106559: URL alias auto white list (reduce the number of queries for aliases)
- #246653: Duplicate menu router problem. Used LOCK TABLES patch. Solved since Drupal 6.16 (locking framework implemented)





# Fast 404s



- Regular 404 handling in Drupal causes scalability issues
  - Full bootstrap of Drupal (CPU and database)
  - Logs to the watchdog (database load)
- Check in settings.php if it is a static file, and exit right there, with a 404. Much less resources.
- Issue #76824 specifically for that. (wink wink Dries!)





# Crawlers



- Found some stupid crawlers that can hit the site hard
  - Microsoft URL Control
  - LucidMedia ClickSense
- Modified settings.php to send them HTTP code 418 (I'm a teapot) and exit early
- Monitor logs and block IP addresses that are resource hogs (search on 418)







# Apache's Access Log



- Lately, Apache's access log has been growing too big due to increased traffic
  - Uses too much disk space
  - Hits the disk often
- Added rules to Apache's configuration to exclude logging for jpg, png, gif, js, css, ico





# MySQL



- # Separate disk spindle
- `datadir = /data/mysql`
- `innodb-file-per-table`
- `innodb_buffer_pool_size = 256M`
- `long_query_time = 2`





# MySQL



- `key_buffer` = 48M
- `key_buffer_size` = 128M
- `sort_buffer_size` = 2M
- `read_buffer_size` = 2M
- `read_rnd_buffer_size` = 8M
- `join_buffer_size` = 4M
- `table_cache` = 750





# MySQL



- `thread_stack` = 128K
- `thread_cache_size` = 64
- `max_allowed_packet` = 16M
- `max_connections` = 100
- `max_heap_table_size` = 256M
- `tmp_table_size` = 256M
- `query_cache_size` = 128M
- `query_cache_limit` = 4M





# InnoDB



- Locking is often a problem with heavily updated tables
- The following tables were converted to InnoDB
  - Users
  - Sessions
  - votingapi\_vote
- We use innodb-file-per-table
- Other sites need additional tables to be InnoDB





# Slow Queries



- Due to heavy tuning of the entire stack, we don't have many of them.
- We check them once a week (maat-kit script)
- Nothing alarming anymore





# Slow Queries Example

- 84% of slow queries were:
  - `SELECT COUNT(*) FROM sessions WHERE uid = 0 AND unix_timestamp() - timestamp < (60*15)`
  - 3.6 seconds!
- Changed to
  - `SELECT COUNT(*) FROM sessions WHERE uid = 0 AND timestamp > (unix_timestamp() - (60*15))`
  - 30 milliseconds!





# Monitoring



- Munin
  - CPU, disk, memory, load, I/O, Apache, MySQL, memcache, ...etc.
  - We added scripts for number of logged in users, and anonymous users







# Miscellany



- Elisiya Cron
  - Allows different hook\_cron to be run at different frequencies and different times of day)
  - e.g. Search (1hour) vs. Job queue (1 minute)
- Queue mail module
  - Emails sent from cron, don't hold up pages or timeout if you have lots of users.

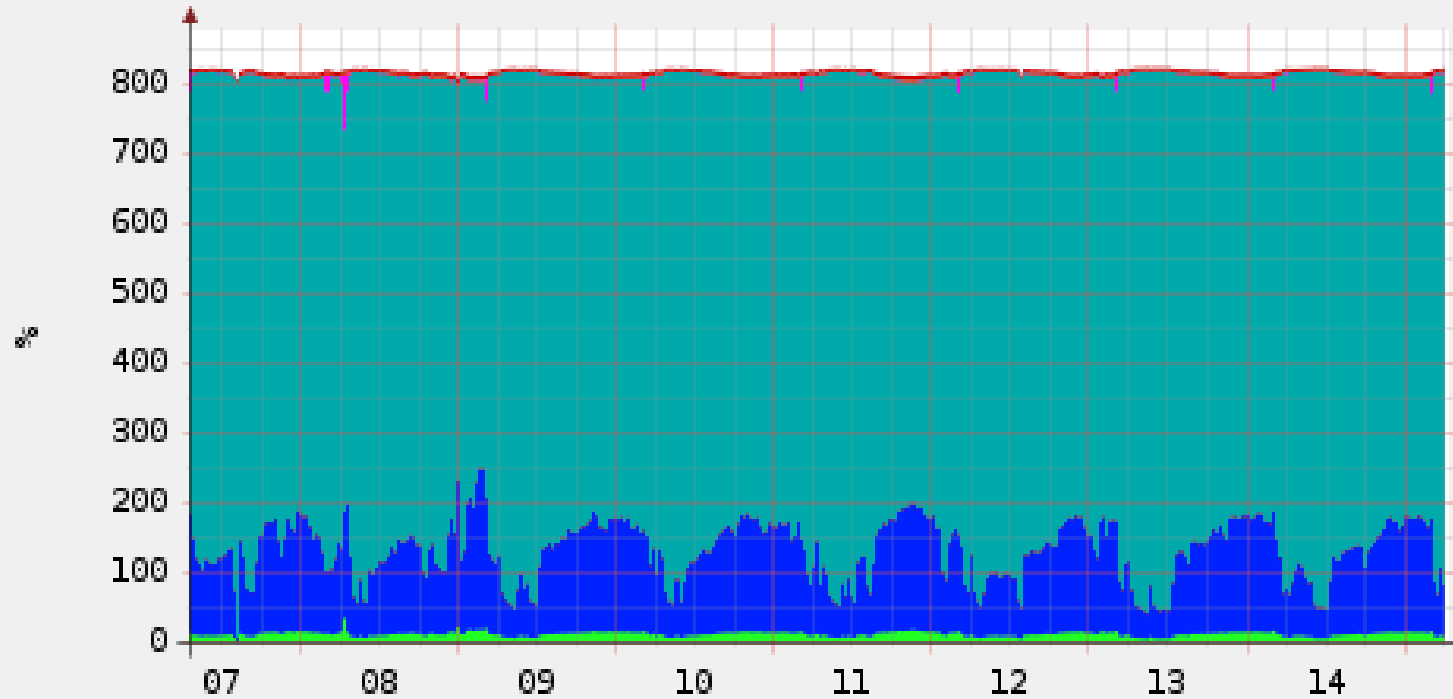




# CPU



CPU usage - by week



RRDTOOL / TOBI OETIKER

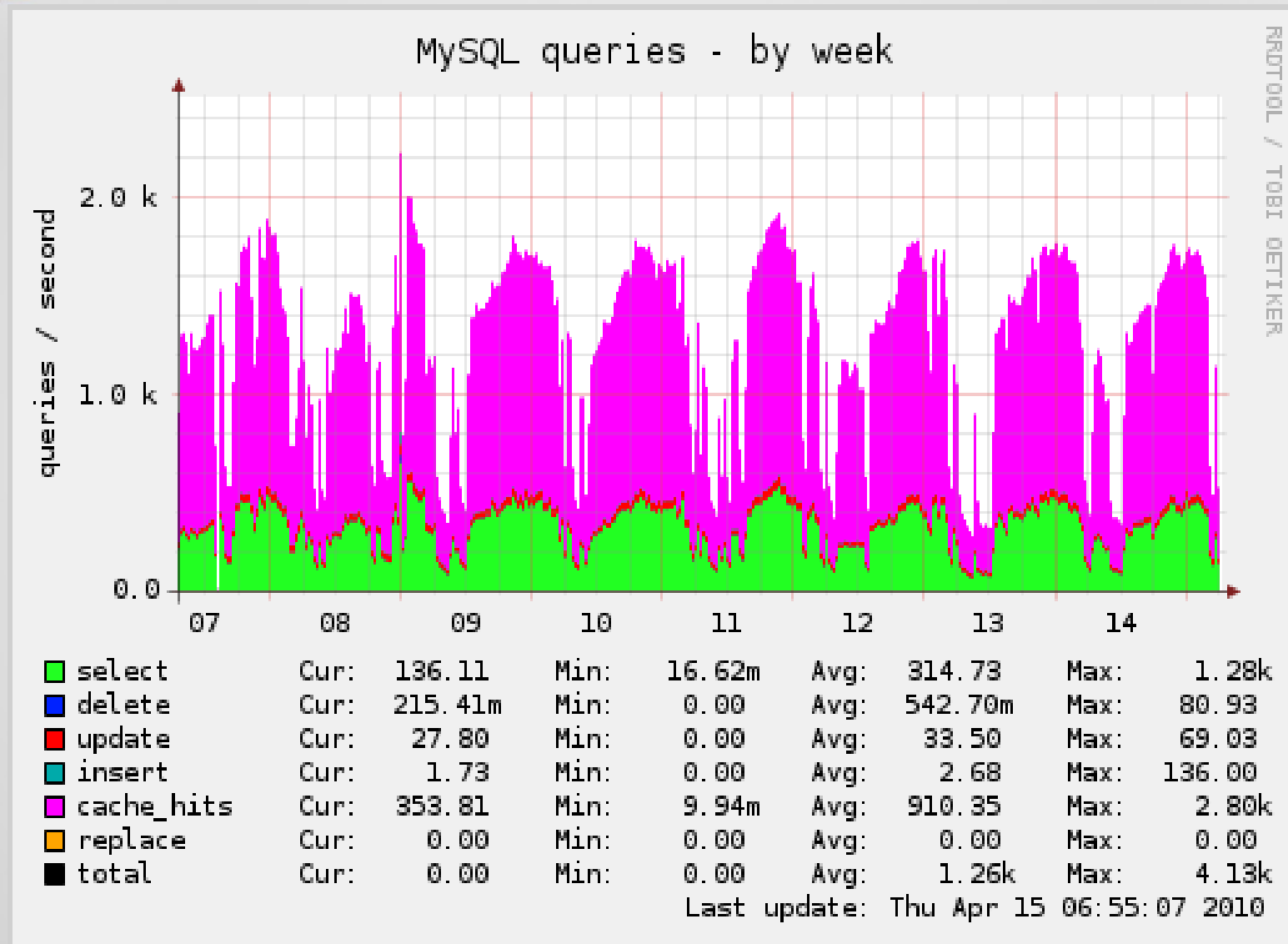
■ system	Cur:	7.69	Min:	0.22	Avg:	11.24	Max:	37.50
■ user	Cur:	73.11	Min:	2.72	Avg:	119.45	Max:	366.18
■ nice	Cur:	0.71	Min:	0.33	Avg:	0.74	Max:	1.24
■ idle	Cur:	731.91	Min:	400.42	Avg:	678.43	Max:	800.91
■ iowait	Cur:	3.23	Min:	0.00	Avg:	1.69	Max:	124.62
■ irq	Cur:	0.46	Min:	0.00	Avg:	0.55	Max:	1.02
■ softirq	Cur:	2.61	Min:	0.00	Avg:	3.55	Max:	7.30

Last update: Thu Apr 15 06:55:07 2010





# MySQL queries





# Reg. Users (15 min)



Users - Logged-in on the site - by week



■ Users = Logged-in on the site

Cur:	Min:	Avg:	Max:
120.73	0.00	145.07	240.41

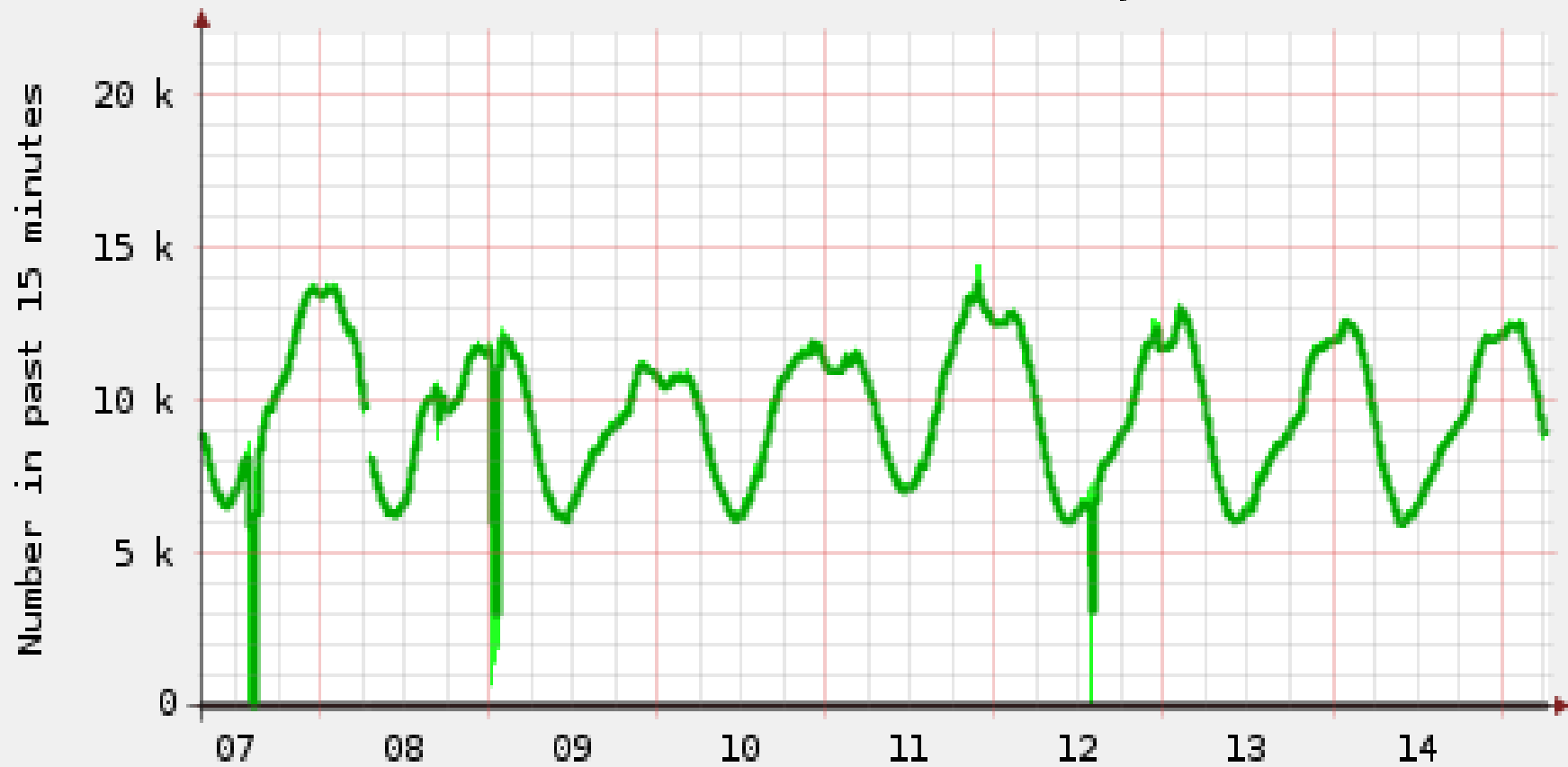
Last update: Thu Apr 15 06:40:10 2010



# Anon. Users (15 mins)



Users - Anon on the site - by week



■ Users - Anon on the site

Cur:	Min:	Avg:	Max:
8.95k	0.00	9.55k	14.36k

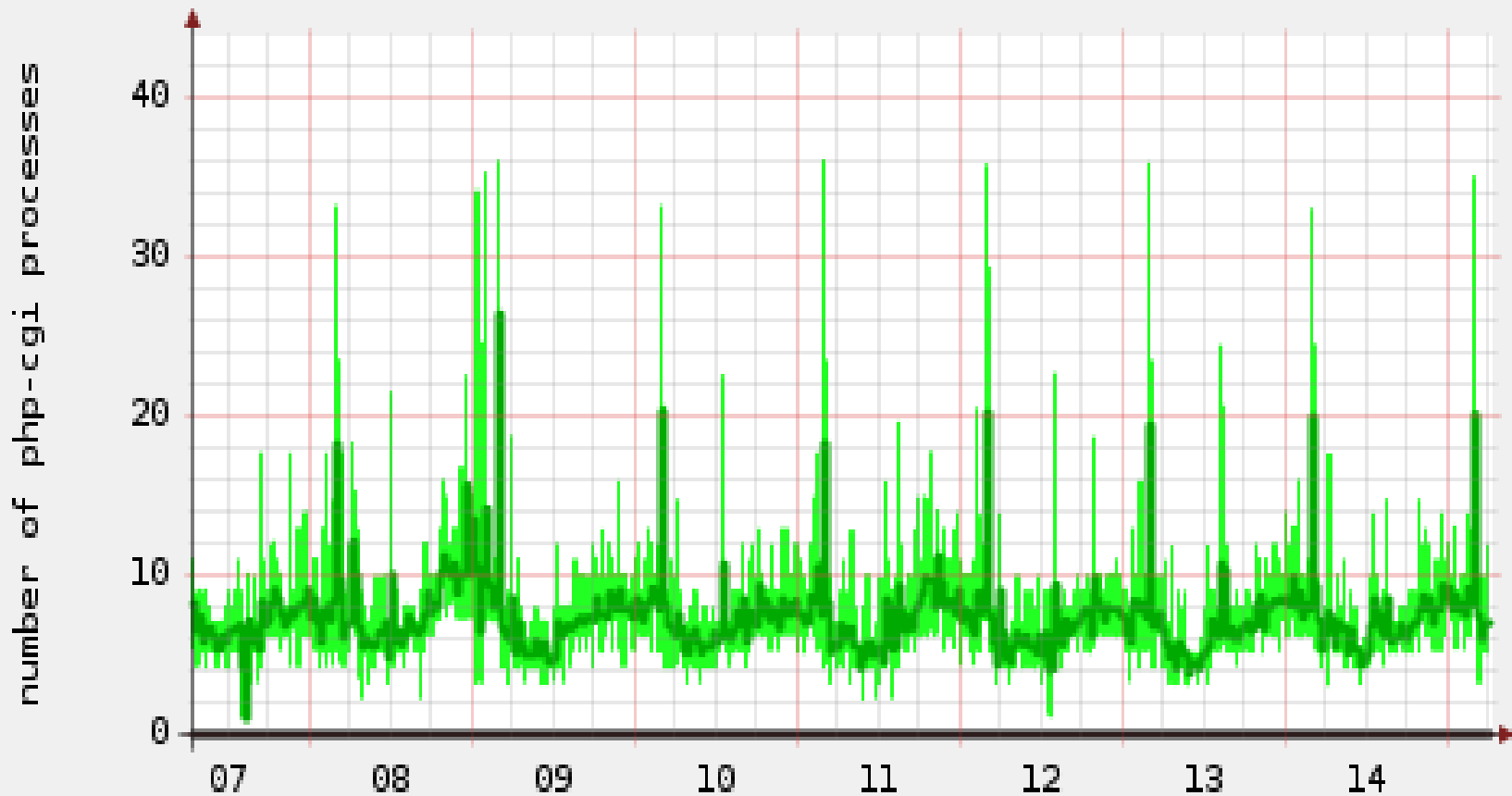
Last update: Thu Apr 15 06:55:07 2010



# # of PHP procs.



Number of php-cgi processes - by week



■ php-cgi Cur: 6.99 Min: 1.00 Avg: 7.43 Max: 35.95  
Last update: Thu Apr 15 06:40:10 2010



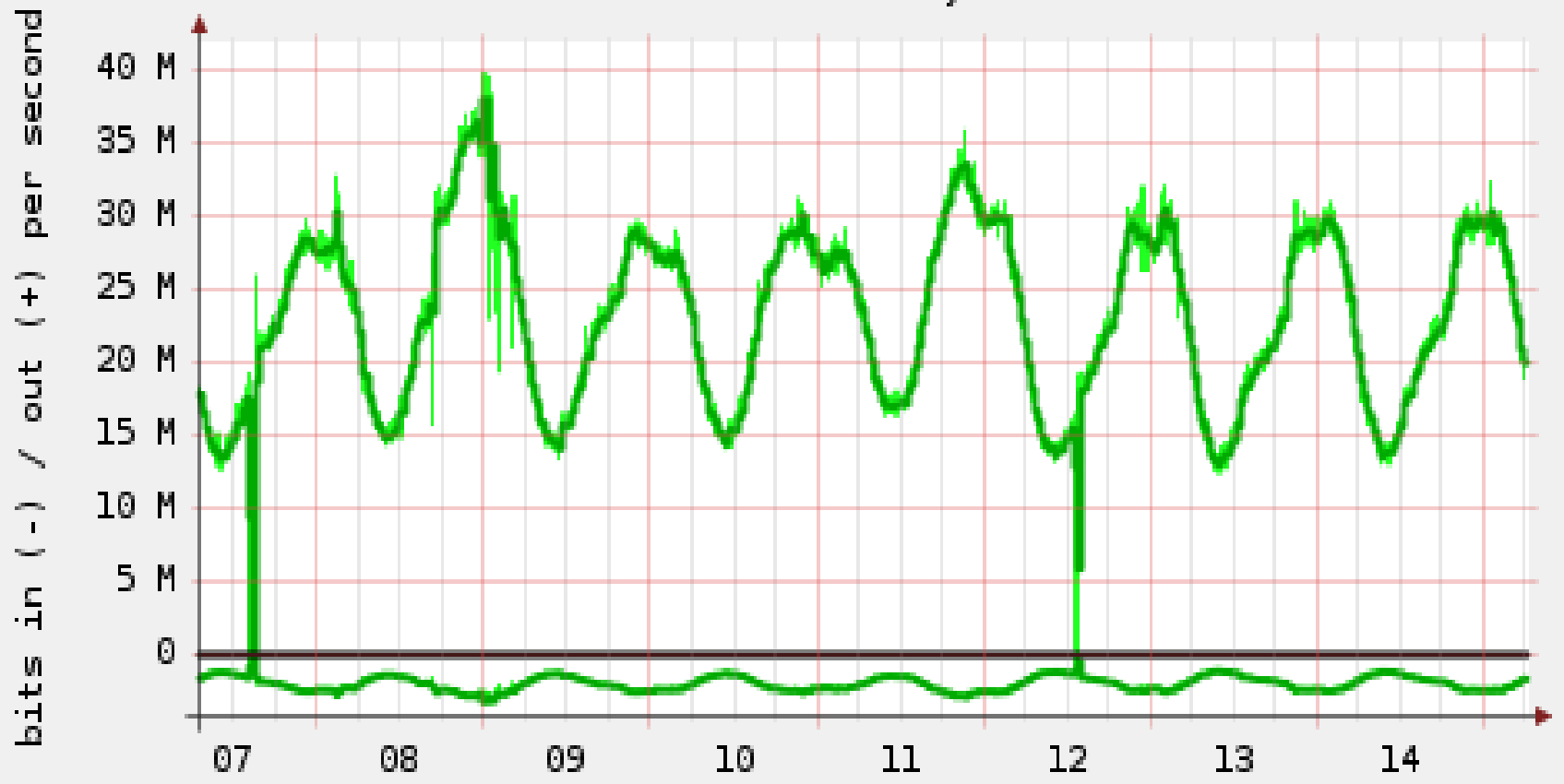


# Ethernet port



eth0 traffic - by week

RRDTOOL / TOBI OETIKER



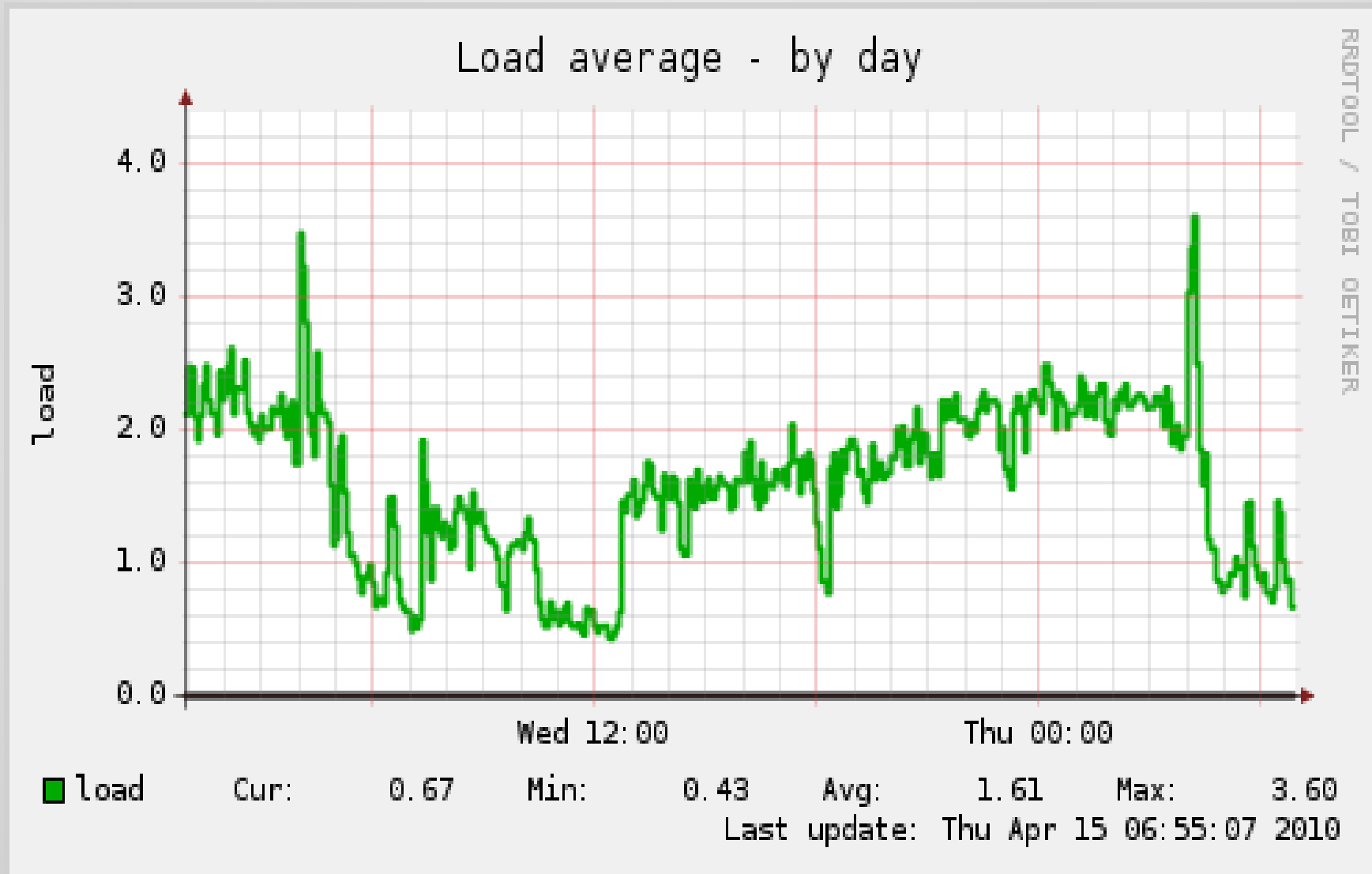
	Cur (-/+)	Min (-/+)	Avg (-/+)	Max (-/+)
■ bps	1.67M/ 19.98M	0.00 / 0.00	2.00M/ 23.03M	3.50M/ 39.65M

Last update: Thu Apr 15 06:45:03 2010





# Load average







# The Future?



- No idea if the site will continue to grow, or has reached a plateau
- Varnish with HTTP headers for cache changes?
- Two servers, one for the database and the other for Apache/PHP/Drupal?





# The Future?



- Drupal 7:
  - Fields in core (more overhead)
  - Lots of optimizations too (thanks to “catch”  
Nathaniel Catchpole, #1 contributor for 7!)





# Lessons Learned



- Drupal can scale well
- Each site is unique
- Complexity is an abomination ... resist the urge!
- Simplicity is a virtue ... practice and preach it!
- Measure and monitor, don't fly blind
- Diagnosis before treatment





# Survey



## What did you think?

### Step 1)

Locate this session on the DCSF site

<http://sf2010.drupal.org/conference/schedule>

### Step 2)

Click the “Take Survey” link

