

One Server, 3.4M page views a day

Khalid Baheyeldin

DrupalCamp Toronto

October 2010

<http://2bits.com>

2bits





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 contributions to Drupal core
 - Site maintenance, `hook_watchdog()`
- Over 37 modules contributed on drupal.org
 - Userpoints, Nagios, SecondLife, Favorite Nodes, Flag Content, Nodevote, ...





About Khalid (cont'd)



- Volunteer with various teams (webmasters, infrastructure, security, ...)
- Permanent 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 heavy ...?
 - Database intensive ...?





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 (past)



- 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 (present)



- July 2010
 - 3.42 page views / day (peak)
 - 832,744 visits / day (peak)
 - 92 million page views / month
 - 22.96 million visits / month
- Stats
 - 111,379 registered users
 - 15,906 nodes





Site Traffic



Thursday, April 8, 2010

Visits: 540,997

Pageviews: 2,799,536

Thursday, July 8, 2010

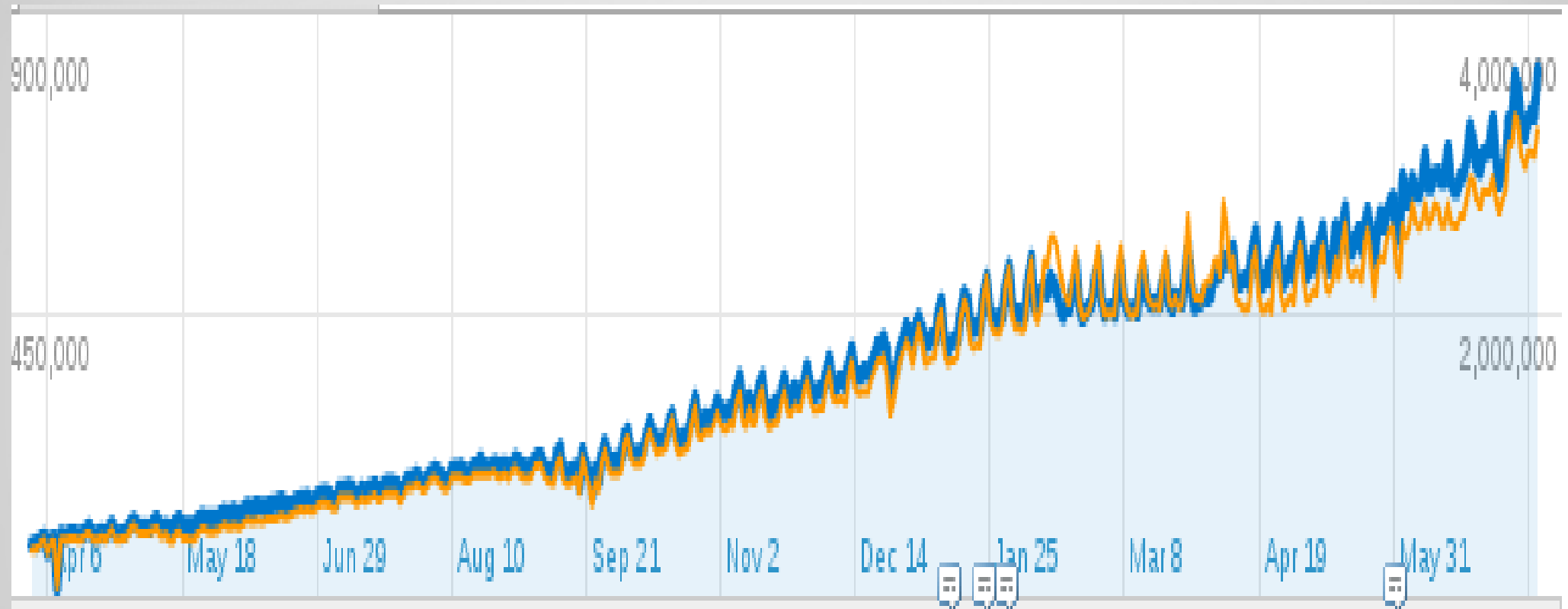
Visits: 832,744

Pageviews: 3,420,107





Site Traffic





Site Traffic (now)



- Back to normal levels (recent weeks)
 - 2.5M page views a day peak, 2.1M non-peak.
- September:
 - 66.3 M page views
 - 19M visits





How did we do it?



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





How did we do it?



Surprise!

None of the above





Challenges (before)



- 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 bloat (e.g. Apache modules)
 - Could not get stuff installed properly (Munin, old version of htop, ...etc.)





Challenges (before)



- Was 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 using votingapi caused a slow query (2+ seconds: file sort + temporary table)
- Locking on the table level (users/sessions were MyISAM)





Challenges (before)



- Concern about fast rate of growth (was at 4K page views a day then!)
- 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's Drupal codebase
- Removed all unnecessary modules
 - Less code to load/execute
 - Less memory to consume
 - Less database queries
- Removed problematic view (long running query)
- Replaced CCK with a custom module (one content type with a few fields)





Hardware



- Initially on a 4-core server for about 6 months
- Then on
 - Dell PowerEdge 2950 Rack Mount
 - 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 (2 + 2)
- \$400 per month





Hardware



- Use current CPUs
- 64 bit CPU with 64 bit distro
- More cores if all-in-one servers
 - Opterons
 - Xeons
- Not a Mac G5!





Disks



- Fast disks (15,000 RPM)
 - Better than run of the mill 7,200 RPM
- Separate spindles for MySQL and 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
- Recently moved to 10.04 LTS 64bit with PHP 5.3.
- Could be done because limited number of modules (44)





Software



- Apache
 - MPM Worker (threaded, far less memory for static content)
 - Apache fcgid (less memory, less MySQL connections)
- PHP
 - FastCGI (stable with fcgid, 11-15 ms page execution time)
 - APC 3.0.19 (installed via pecl)





Software



- Munin for monitoring, and historical resource usage
 - Why did the graph change?
 - What does that mean?
- Awstats for statistics
 - And comparison with Google Analytics





Drupal



- Book module for content hierarchy
- Only 44 modules enabled total
- Comment module is not enabled
- Voting on nodes is enabled
 - VotingAPI + fivestar
- 2 themes
 - Garland for administration





Simplicity vs. Complexity **abits**

"Simplicity is a prerequisite for reliability"
-- Edsger Dijkstra 1975

"Complexity is a disease"
-- Dries Buytaert, May 2006





Simplicity vs. Complexity **abits**

- Regular Drupal sites have 90 to 120 modules (or more).
- Saw 180 modules, and then had nightmares on sites with 200 modules, then had a client with 231 modules, then another with 233!
- Complexity is a indeed 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 ...





Complexity (another client) **abits**

- Client with 300K page views per day, 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!





Simple vs. Complex



- **Scalability/Performance:** More components = slower + more resource usage.
- **Maintainability:** Less moving parts to maintain, e.g. Security updates.
- **Testability:** Easier to test less parts.
- **Upgradability:** You will have less headaches when you need to upgrade to a newer version.
- **Refactoring:** Easier to refactor a site that has fewer components





KISS

abits

Keep It Simple Stupid ...

It pays off in the long term ...





Drupal modules



- Removed unnecessary modules (including CCK)
- Custom modules (only 2)
 - One is your regular form alters and customizations
 - The other is the 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 (marked difference in the Apache “requests per seconds” graph)





Taxonomy Browser



- Can be detrimental to a site if misconfigured
 - Checkboxes and lots of tags
 - Users can cause the SQL query from hell with 32 tables joins ... server melts ...
- Solution:
 - Use form alters to limit the number of terms to search on





memcache



- The most essential component 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



- Watch for the input format in nodes and blocks.
 - PHP format is not cached
 - Get rid of PHP nodes and blocks regardless!
 - Some site admins use so they can embed javascript for ads and such
- Created a new Raw format (no line break)
- Now nodes and blocks are cached ...





memcache bins (past)



- 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 bins (present)

- Memcache 6.x-1.5
 - Simplifies configuration tremendously
 - Single bin on port 11211
- Do **not** use memcache 6.x-1.6, even though it is a security release!
 - Just don't enable the memcache_admin module included with it.





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, such as login, and search)
- 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 (or another server)
- Drawback: web user can't see the log entries (has to be checked via command line or some other tool)
- Applications available to push the log somewhere else





Patches



- Patching is bad, unless
 - Kept to a minimum
 - Properly managed (Version Control System)
- #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, nudge nudge, 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



- Recently, Apache's access log has been growing too big due to increased traffic
 - One week, 133.9M hits, 40GB uncompressed, 2.8GB compressed
 - Quickly running out of disk space
 - Too much disk writes
- Added rules to Apache's configuration to exclude logging for jpg, png, gif, js, css, ico
 - Now only 500 to 900MB per week, compressed





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
- Only 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.

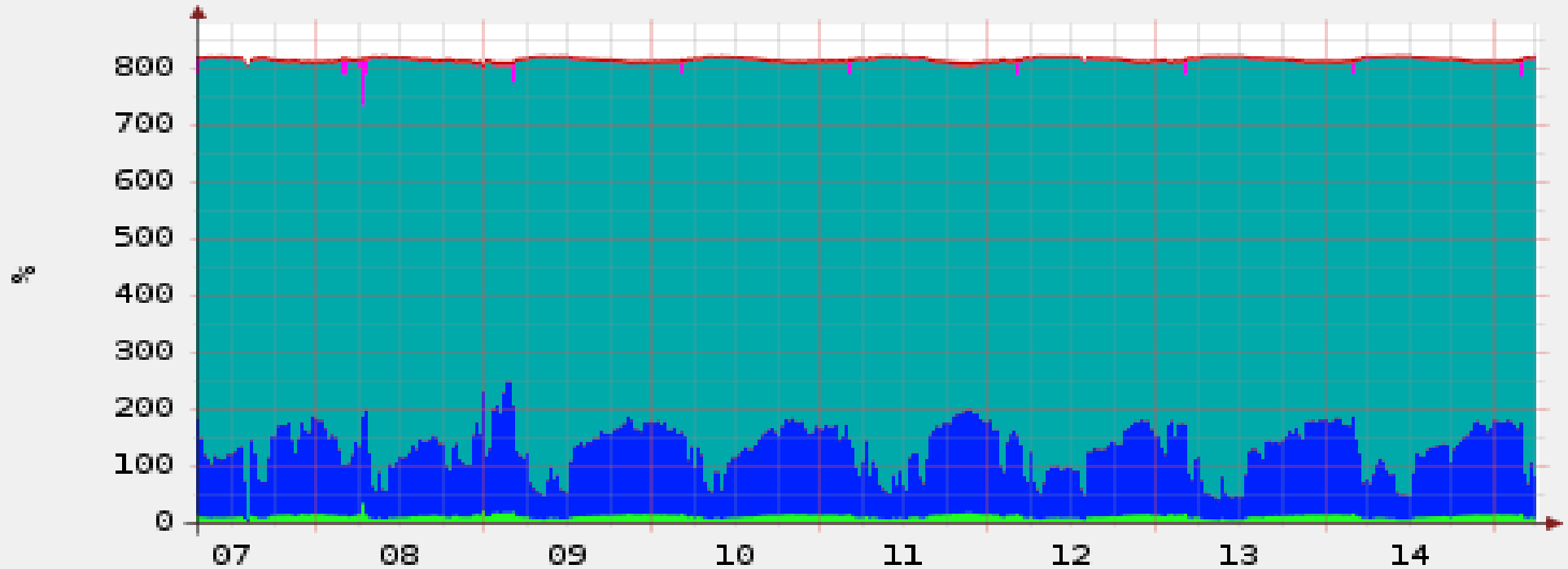




CPU



CPU usage - by week



■ 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

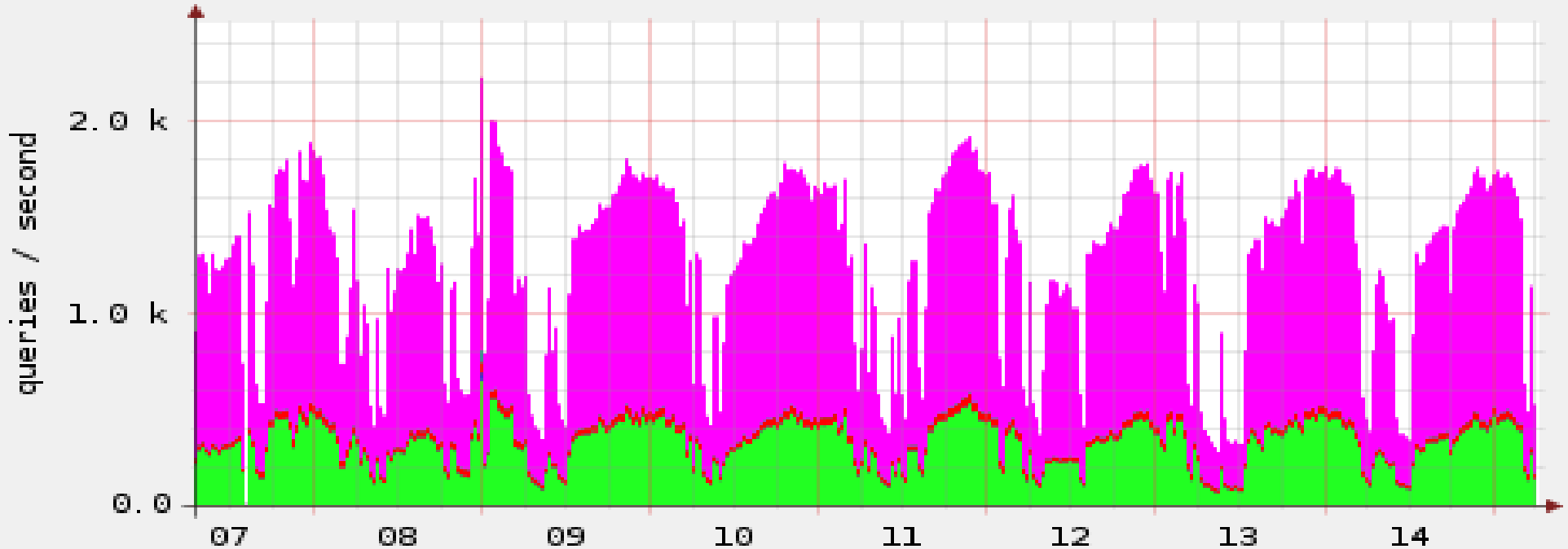











MySQL queries



MySQL queries - by week



RRDTOOL / TOBI OETIKER

 select	Cur:	136.11	Min:	16.62m	Avg:	314.73	Max:	1.28k
 delete	Cur:	215.41m	Min:	0.00	Avg:	542.70m	Max:	80.93
 update	Cur:	27.80	Min:	0.00	Avg:	33.50	Max:	69.03
 insert	Cur:	1.73	Min:	0.00	Avg:	2.68	Max:	136.00
 cache_hits	Cur:	353.81	Min:	9.94m	Avg:	910.35	Max:	2.80k
 replace	Cur:	0.00	Min:	0.00	Avg:	0.00	Max:	0.00
 total	Cur:	0.00	Min:	0.00	Avg:	1.26k	Max:	4.13k

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

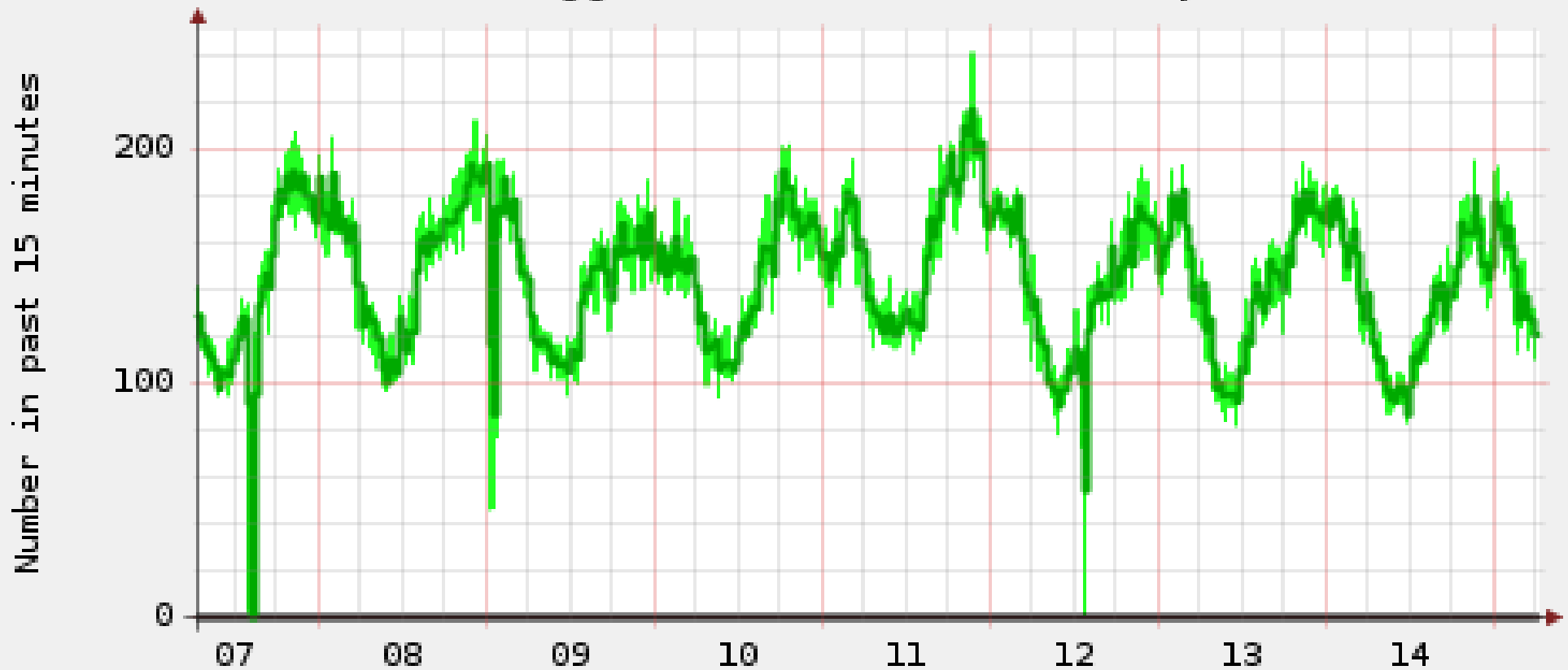




Logged users (15 min)



Users - Logged-in on the site - by week



RRDTOOL / TOBI OETIKER

■ Users = Logged-in on the site

Cur:
120.73

Min:
0.00

Avg:
145.07

Max:
240.41

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

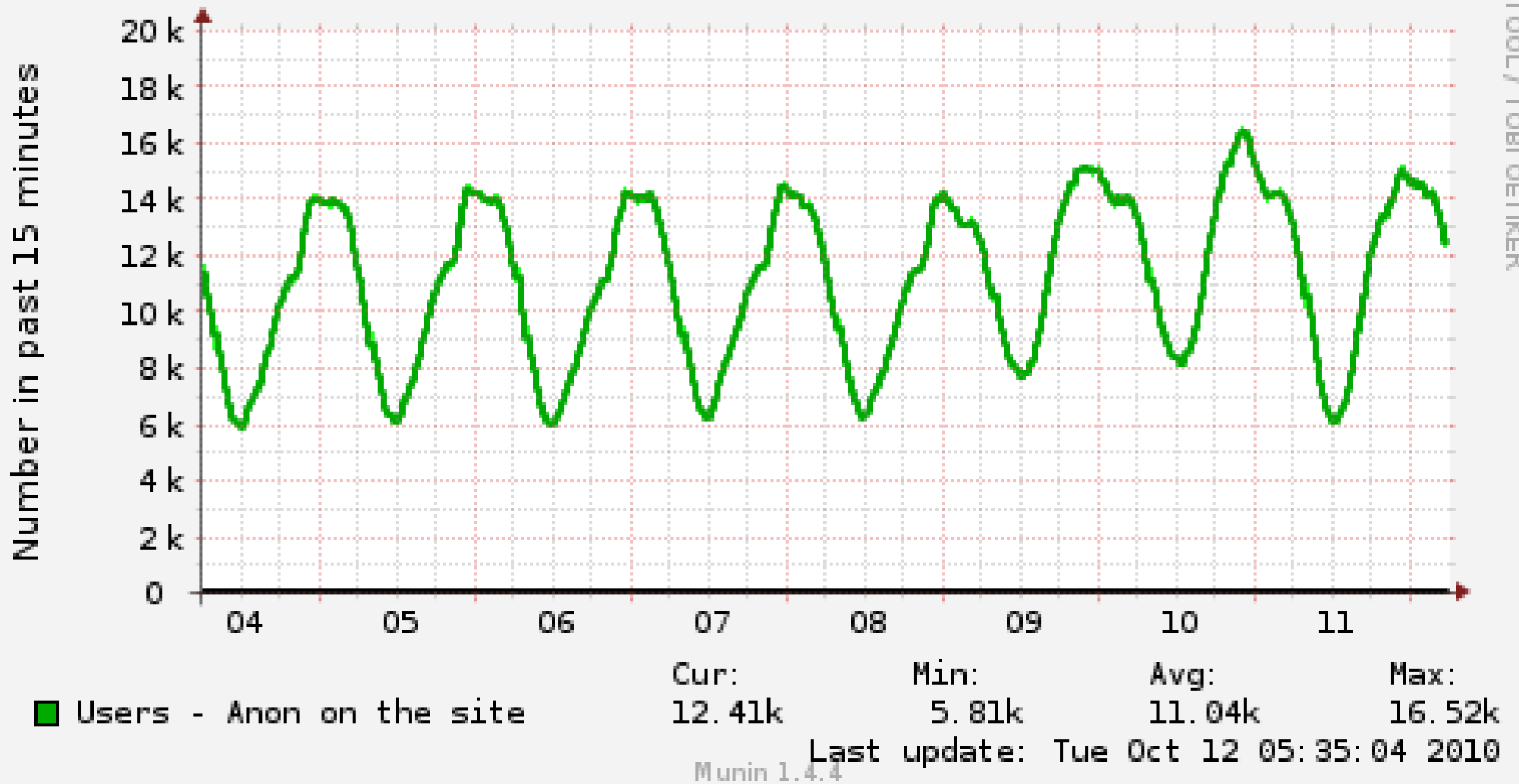




Anon. Users (15 mins)



Users - Anon on the site - by week

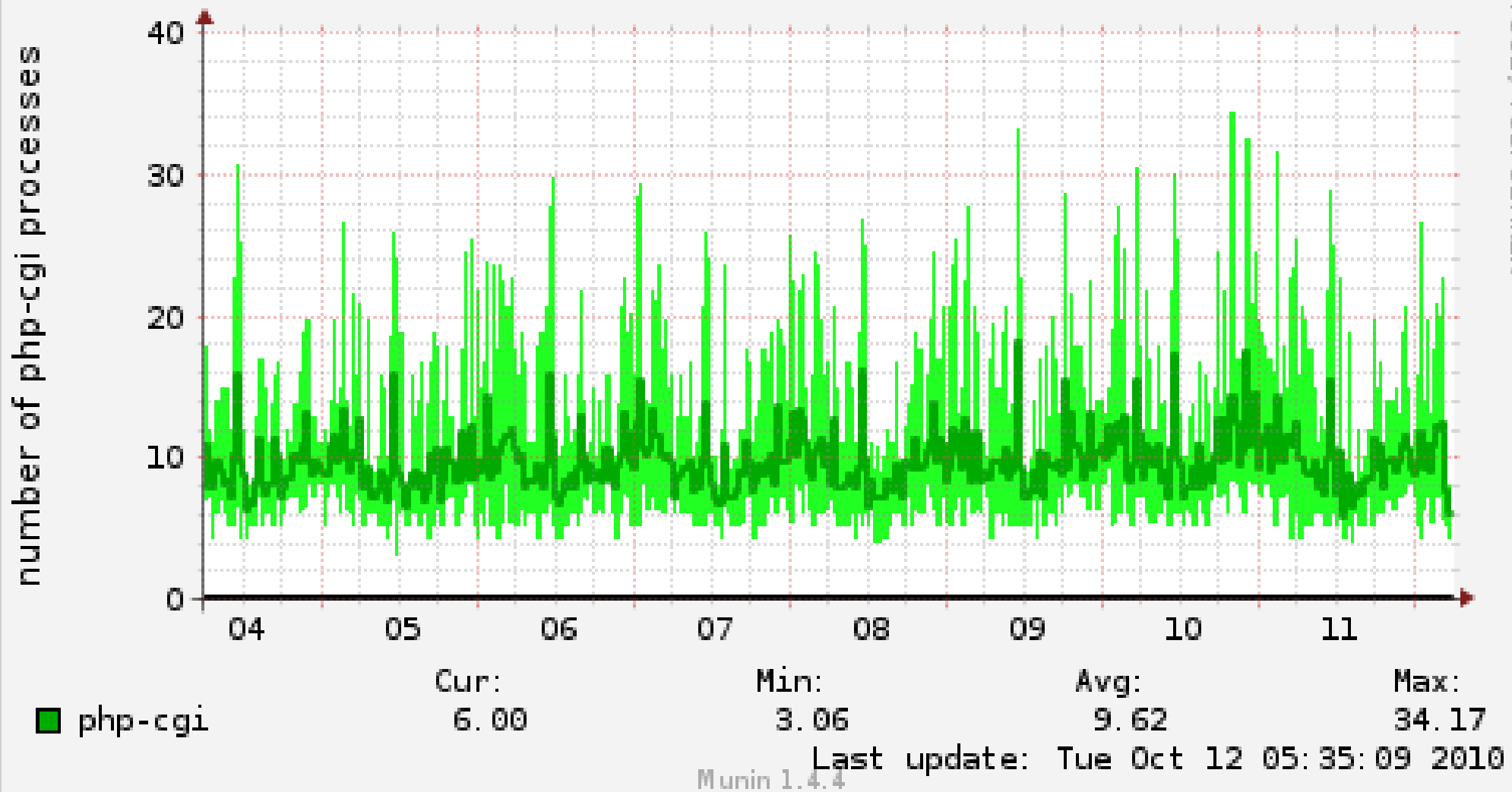




of PHP procs.



Number of php-cgi processes - by week

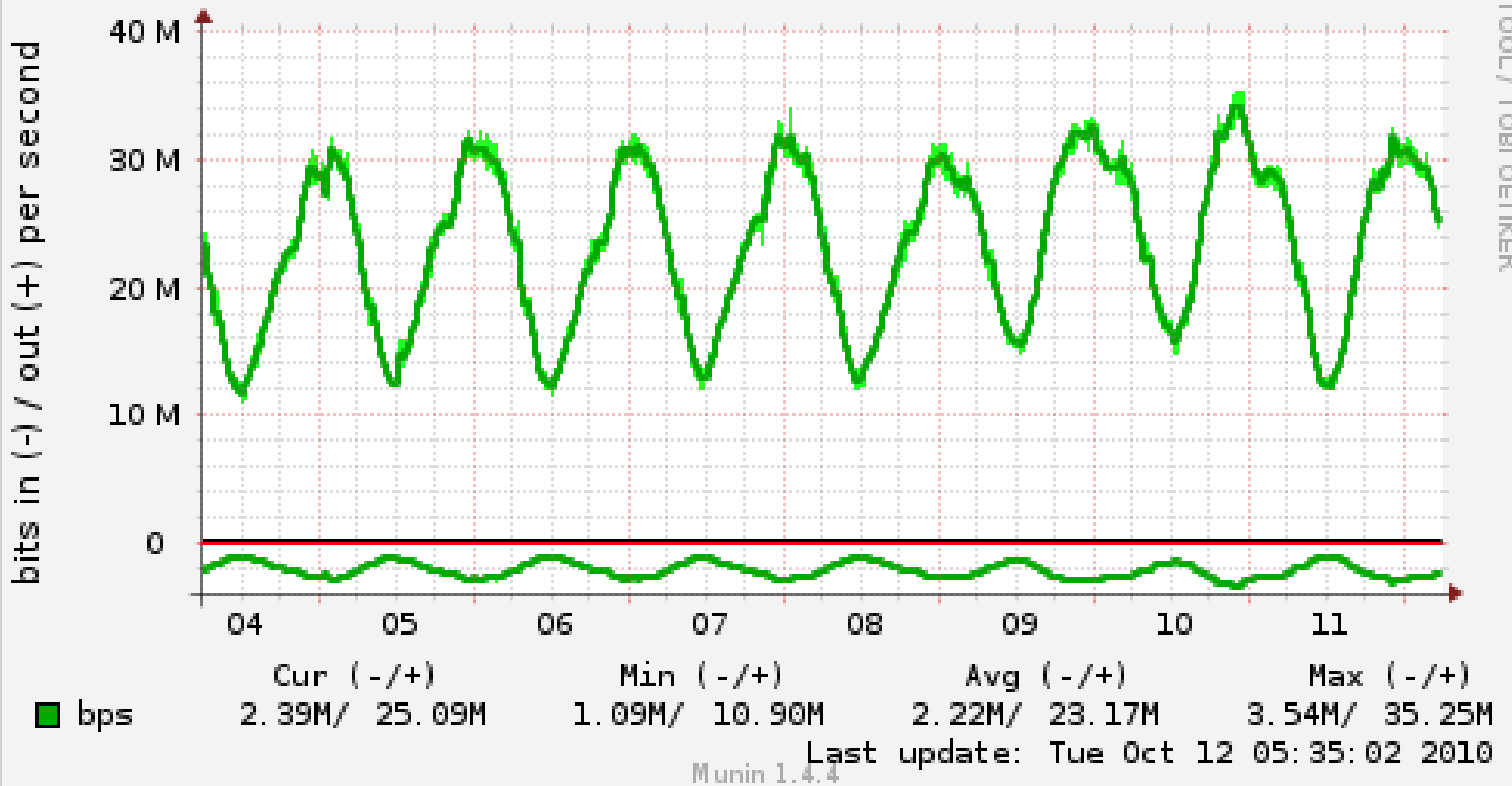




Ethernet port



eth0 traffic - by week

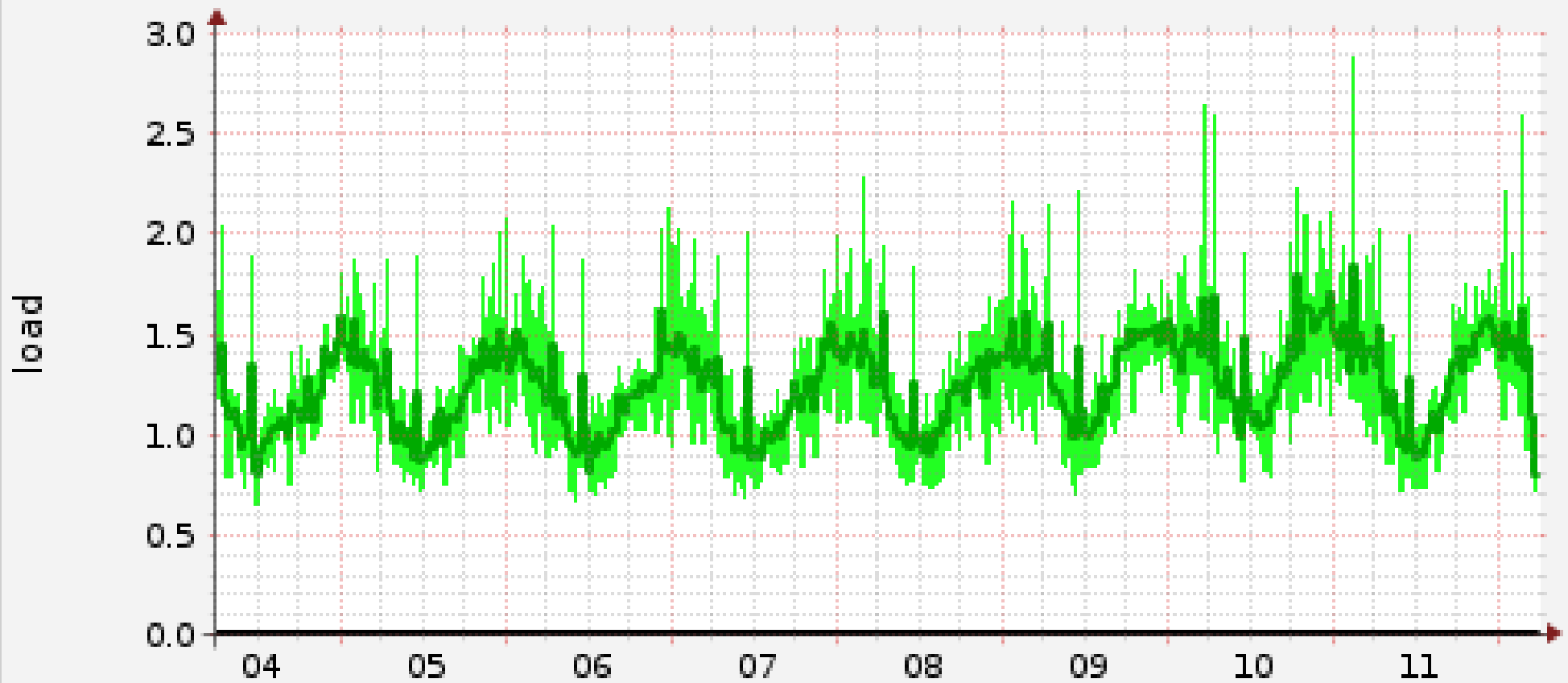




Load average



Load average - by week



RRDTOOL / TOBI OETIKER

■ load

Cur:	Min:	Avg:	Max:
0.78	0.64	1.24	2.87

Last update: Tue Oct 12 05:35:04 2010
Munin 1.4.4





The Future?



- Traffic subsided a bit in the last 2 months after several weeks of peaking.
- We don't know for sure if it has reached a plateau or not.
- When we need it, we will go for 2 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, and many others.)





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





Resources



- Many articles on performance, tuning and optimization <http://2bits.com>





Questions?



Questions, comments, ...?

