

Performance Tuning and Optimization for high traffic Drupal sites

Khalid Baheyeldin
Drupal Camp, Toronto
May 11-12, 2007





Agenda



- Introduction
- The LAMP Stack
 - Linux, Apache, MySQL, PHP
- Drupal
 - Database queries
 - Modules
 - Caching
- Measurement and monitoring tools
- What can go wrong?
- Questions, discussion





About 2bits



- Based in Waterloo, Ontario
- Active member of the Drupal community since 2003
- Member of security and infrastructure teams
- 24+ modules on drupal.org
- Listed on Drupal.org's service providers section
- Maintain modules that run on drupal.org (donations, feature, lists)
- Google Summer of Code mentoring (2005, 2006, 2007)





2bits Services



- Clients mainly in USA and Canada
- Subcontracting development projects
- Customization of existing modules
- Development of new modules
- Installation, upgrades
- Automated backups
- Performance tuning and optimization





About Khalid



- Developing for computers for way too long (22 years), Drupal since 2003
- Core contributions
 - Site maintenance feature
 - Logging and alerts in Drupal 6
 - Several patches
- Member of
 - Drupal security team
 - webmasters team
 - infrastructure team
- Co-founder of 2bits
- Blog at <http://baheyeldin.com>
- Contributed modules
 - Adsense
 - Userpoints
 - Nodevote
 - Job search
 - Favorite nodes
 - Flag content
 - Stock API and module
 - Custom Error
 - Currency
 - Image watermark
 - Site menu
 - Email logging and alerts
 - Second Life
 - Technorati
 - Click thru
 - Referral





The Iron



- Physical server matters
 - Dedicated
 - VPS
- Not applicable to shared hosting
- Dual Opterons kick ass
- Lots of RAM (caching the file system and the database, as much as possible)
- Multiple disks if you can
- Always mirrored!





Multiple Servers



- One database server + multiple web servers
- Can use DNS round robin
- Or proper load balancers (commercial, free)
- Even a reverse proxy (squid)
- Do it only if you have the budget
 - Complexity is a running cost
 - Tuning a system can avoid (or delay) the split





The LAMP stack



- Most commonly used stack for hosting Drupal and similar applications
 - Linux
 - Apache
 - MySQL
 - PHP
- Most of this presentation applies to *BSD as well. Parts apply to Windows.





Linux



- Use a proven stable distro (Debian, Ubuntu)
- Use recent versions (no Fedora Core 4 please)
- Be a minimalist
- Install only what you need
 - (e.g. No X11, no desktop, No PostgreSQL if you are only using MySQL, ...etc.)
- Balance “compile your own” vs. upgrades





Apache

- Most popular, supported and feature rich
- Other web servers
 - lighttpd (lighty)
 - Popular with Ruby
 - 1MB per process
 - Recent memory leaks
 - nginx
 - More stable than lighty (no leaks)





Apache



- Cut the fat
 - Enable only `mod_php` and `mod_rewrite`
 - Disable everything else (java, python)
 - May need extended status for Munin
- Tune MaxClients
 - Too low: you can't serve a traffic spike (Digg, Slashdot)
 - Too high: your memory cannot keep up with the load, and you start swapping (server dies!)





Apache (cont'd)



- KeepAlive
 - 5 to 10 seconds OK
 - More than that, it ties up processes
- Allowoverrides
 - Set to None
 - Move Drupal's .htaccess contents to vhosts
- mod_gzip/mod_deflate
 - Compromise of CPU usage vs. Bandwidth usage





MySQL



- Most popular database for Drupal
- Not the best database from the technology point of view (ACID, transactions, concurrency), but still adequate for the job
- Various pluggable engines





MySQL Engines



- MyISAM
 - Faster for reads
 - Less overhead
 - Poor concurrency (table locking)
- InnoDB
 - Transactional
 - Slower in some cases
 - Better concurrency
 - Oracle owns the engine now ...





MySQL Engines



- Two new engines, owned by MySQL AB
 - Falcon. Not mature enough to match InnoDB, benchmarks show it is still slow
 - SolidDB.
- PBXT
 - PrimeBase XT





MySQL tuning



- Query cache
 - Probably the most important thing to tune
- Table cache
 - Also important
- Key buffer





PHP



- Use a recent version
- Install an Op-code cache / Accelerator
 - eAccelerator
 - APC
 - Xcache
 - Zend (commerical)
- APC vs. eAccelerator benchmark on 2bits.





Op-code caches



- Benefits
 - Dramatic speed up of applications, specially complex ones like Drupal
 - Significant decrease in CPU utilization
 - Considerable decrease in memory utilization
 - The biggest impact on a busy site
- Drawbacks
 - May crash often
 - Use logwatcher to auto restart Apache





mod_php



- Normally, Apache mod_php is the most commonly used configuration
- Shared nothing
 - No state retained between requests
 - Less issues
- Stay with mod_php if you can.
- Can be as low as 10-12MB per process
- Saw it as high as mid 20s+





PHP as CGI



- CGI is the oldest method from the early 90s.
- Forks a process for each request, and hence very inefficient.
- Some hosts offer it by default (security) or as an option (e.g. running a specific PHP version).
- Don't use it!





Fast CGI



- FCGI is faster than CGI (uses a socket to the PHP process, not forking)
- Mostly with Lighttpd and nginx, since it is the only way to run PHP for those servers, but also with Apache
- There are some cases (e.g. drupal.org itself)
- Better separation of permissions (e.g. Shared hosting)
- If you have one server and one Linux user, permissions may not be an issue.





Drupal



- Mainly database bottlenecks
- Bottlenecks are worked on as they are found by the community
- Some modules known to be slow
- Not all sites affected by all bottlenecks





Watchdog



- Avoid errors (404s on graphics, favicon)

```
TIME STATE      INFO
24  updating  DELETE FROM watchdog WHERE timestamp < 1176392718
24  Locked    INSERT INTO watchdog (uid, type, message, severit
19  Locked    INSERT INTO watchdog (uid, type, message, severit
14  Locked    INSERT INTO watchdog (uid, type, message, severit
11  Locked    INSERT INTO watchdog (uid, type, message, severit
6   Locked    INSERT INTO watchdog (uid, type, message, severit
```

- Optional in Drupal 6 (syslog as an option)





Sessions



- Heavily used in high traffic sites

TIME	STATE	INFO
28	Locked	UPDATE sessions SET uid = 0, hostname = '212.154.
28	Copying to t	SELECT ... FROM sessions WHERE timestamp >= 11776
28	Locked	SELECT ... FROM users u INNER JOIN sessions s ON
27	Locked	UPDATE sessions SET uid = 0, hostname = '222.124.
27	Locked	UPDATE sessions SET uid = 0, hostname = '201.230.
27	Locked	SELECT ... FROM users u INNER JOIN sessions s ON
27	Locked	SELECT ... FROM users u INNER JOIN sessions s ON
27	Locked	SELECT ... FROM users u INNER JOIN sessions s ON
27	Locked	SELECT ... FROM users u INNER JOIN sessions s ON
27	Locked	SELECT ... FROM users u INNER JOIN sessions s ON





Drupal (cont'd)



- Disable modules that you do not need.
- Enable page caching
 - May expire too often on a busy site, causing slow downs!
- Consider caching modules
 - FS Fastpath
 - boost
- Make sure cron runs regularly
- Enable throttle
 - Be wary about throttle and cache





Puggable caching



- Using \$conf variable in settings.php
 - 'cache_include' => './includes/yourcache.inc'
- Allows you to have a custom caching module
- Caching using memcached is being worked on
- Tip: can be used to disable cache for development (stub functions)





Slow modules



- Statistics module
 - Adds extra queries
 - Even slower on InnoDB (COUNT(*) slow)
 - Disable Popular Content block
- gsitemap (XML sitemap)
 - Had an extra join, patch accepted
- Aggregator2
 - Abandoned!
- Many more ...





Measure and Monitor



- How do you know you have a problem?
 - Users complain (site is sluggish, timeouts)?
 - Losing your audience? Loss of interest from visitors?
- Tools for various tasks





Top



- Classic UNIX/Linux program
- Real time monitoring (i.e. What the system is doing NOW)
- Load average
- CPU utilization (user, system, nice, idle, wait I/O)
- Memory utilization
- List of processes, sorted, with CPU and memory
- Can change order of sorting, as well as time interval, and many other things





vmstat



- From BSD/Linux
- Shows aggregate for the system (no individual processes)
- Shows snapshot or incremental
- Processes in the run queue and blocked
- Swapping
- CPU user, system, idle and io wait





netstat



- Shows active network connections (all and ESTABLISHED)
- `netstat -anp`
- `netstat -anp | grep EST`





mtop, mytop



- mtop

- Like top, but for MySQL
- Real time monitoring (no history)
- Shows slow queries

- mytop

- Similar to mtop

- **SHOW FULL PROCESS LIST**





mysqlreport / db tuning



- Mysqlreport
 - Perl shell script
 - Displays statistics
 - No recommendations
- Db tuning
 - A shell script that reads variables from MySQL
 - Annoying use of colors
 - Useful recommendations





Graph monitoring



- Munin

- Nice easy to understand graphs.
- History over a day, week, month and year
- CPU, memory, network, Apache, MySQL, and much more
- Can add your own monitoring scripts

- Cacti

- Similar features





Drupal tools



- Devel module
 - Total page execution
 - Query execution time
 - Query log
 - Memory utilization
- Trace module
 - More for debugging, but also useful in knowing what goes on under the hood





What can go wrong?



- CPU usage is too high
- Memory over utilization
- Too much disk I/O
- Too much network traffic





CPU



- Find out who is using the CPU?
- Find out which type (user, system, wait I/O)





CPU



- If it is an Apache process, the op-code cache will help, unless you have a bug.
- If it is MySQL, then some of that is normal (intensive queries), otherwise
 - tune the indexes
 - split the server to two boxes.
 - Tune the query cache
- If it is something else, and consistent, then consider removing it.





CPU 100%



- Output from Top

top - 10:16:58 up 75 days, 59 min, 3 users, **load average: 152.70, 87.20, 46.98**

Tasks: **239 total, 157 running**, 81 sleeping, 0 stopped, 1 zombie

Cpu(s):100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 2075932k total, 1558016k used, 517916k free, 13212k buffers

Swap: 1574360k total, 49672k used, 1524688k free, 442868k cached

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
659	www-data	21	0	61948	14m	4060	R	3	0.7	0:14.35	apache2
960	www-data	20	0	62084	14m	4076	R	3	0.7	0:10.51	apache2
989	www-data	20	0	62036	14m	4052	R	3	0.7	0:09.95	apache2

.... hundreds of them





CPU 100%



- Vmstat output

```
# vmstat 15
```

```
procs -----memory-----      ----cpu----
 r  b   swpd   free   buff  cache   us   sy  id  wa
152  0   40868 1190640 13740 465004  22   6  71  2
153  0   40868 1190268 13748 464996 100  0  0  0
155  0   40868 1189740 13756 464988 100  0  0  0
154  0   40868 1189540 13768 465044 100  0  0  0
```

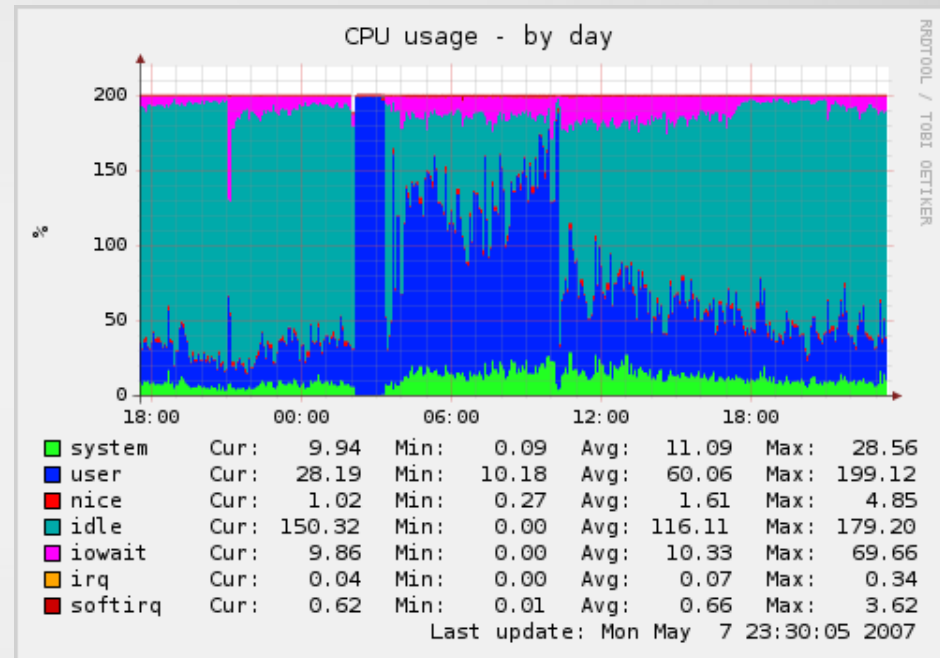




CPU 100%



- What was it?
- eAccelerator (svn303 + PHP 5)
- Attempt to get over PHP crashes
- Note CPU utilization (100%, then high, then dropped low when good version used)





Memory



- Swapping means you don't have enough RAM
- Excessive swapping (thrashing) is server hell!
- Reduce the size of Apache processes
- Reduce the number of Apache processes (MaxClients)
- Turn off processes that are not used (e.g. Java, extra copies of email servers, other databases)
- Buy more memory! Cost effective and worth it.

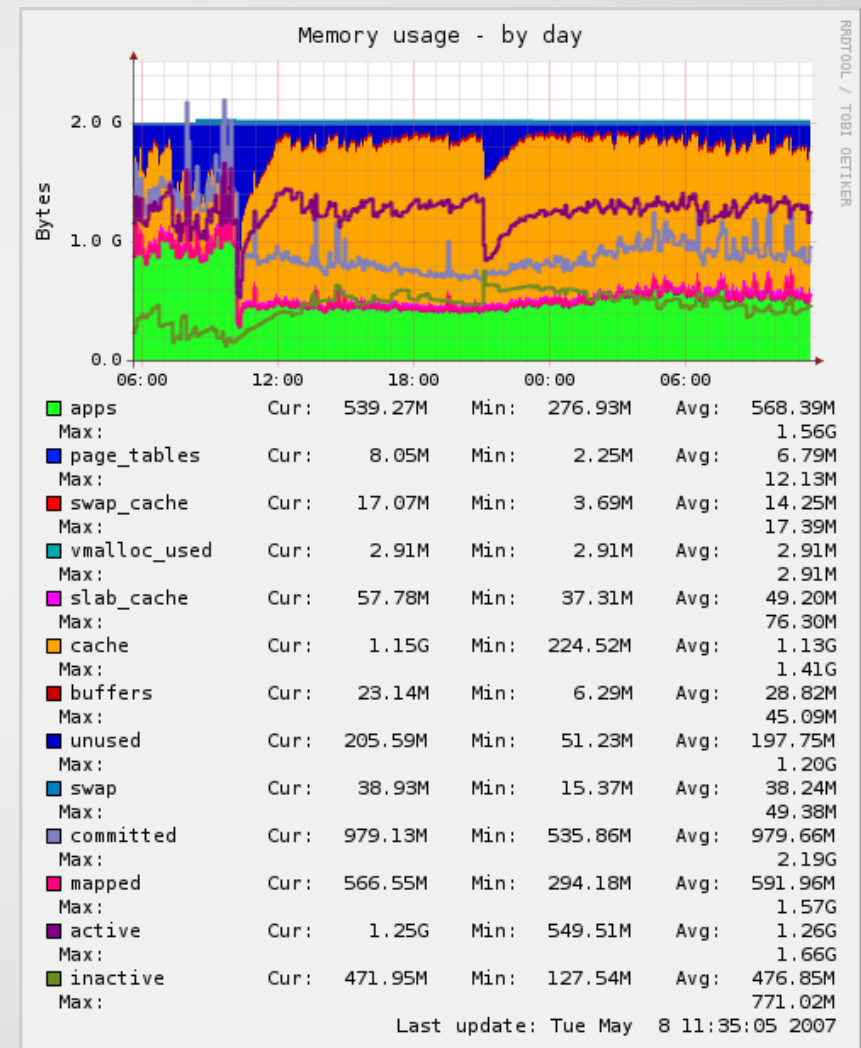




Memory



- Impact on memory usage when there is no op-code cache vs. with an op-code cache





Disk I/O



- First eliminate swapping if get hit by it.
- Get the fastest disks you can. 7200 RPM at a minimum.
- Turn off PHP error logging to `/var/log/*/error.log`
- Consider disabling watchdog module in favor of syslog (Drupal 6 will have that option), or hack the code
- Optimize MySQL once a week, or once a day





Network



- Normally not an issue
- Occasionally you will have a stubborn crawler though
- Or even a DdoS
- Or worse, extortion
- Can eat up resources, including network





Digg front page?



- On Good Friday, adsoftheworld.com was on Digg's front page.
- The founder wrote about it
http://creativebits.org/webdev/surviving_the_digg_effect
- Survived the digg well.
- Another server (untuned) got digged twice and died





Resources and Links



- General

- <http://2bits.com/articles/drupal-performance-tuning-and-optimization-for-large-web-sites.html>
- http://www.lullabot.com/articles/performance_and_scalability_seminar_slides

- Apache

- <http://httpd.apache.org/docs/2.0/misc/perf-tuning.html>

- MySQL

- <http://www.mysqlperformanceblog.com/>
- <http://dev.civicaactions.net/moin/CodeSprint/SanFransiscoMarch2007/PerformanceAndScalabilitySeminar>





Conclusion



- Questions?
- Comments?
- Discussions?

