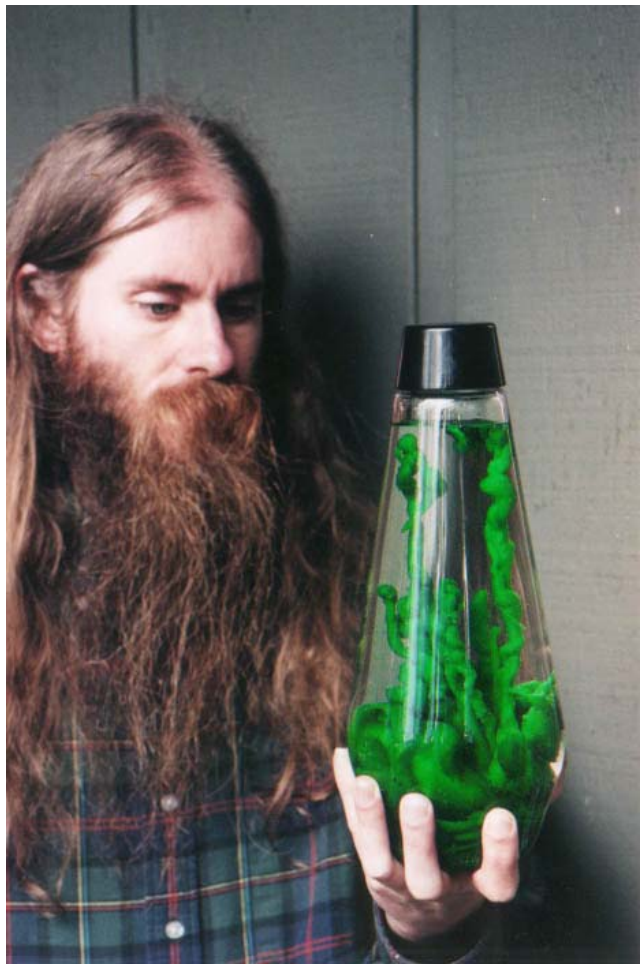
LAMP Tuning

Sean Walberg
Network guy, server ninja

LAMP

- Linux server
- Apache Web Server
- PHP/Perl application
- MySQL server



- Started off with 2 servers, 10 now
- 65m pageviews/month
- 20mbit/s of web traffic

Guiding Principles

- You can’t run a fast site on crappy hardware
- You can’t run a fast site on poorly tuned servers
- You can’t run a fast site using a bad application

- Application > tuning > hardware

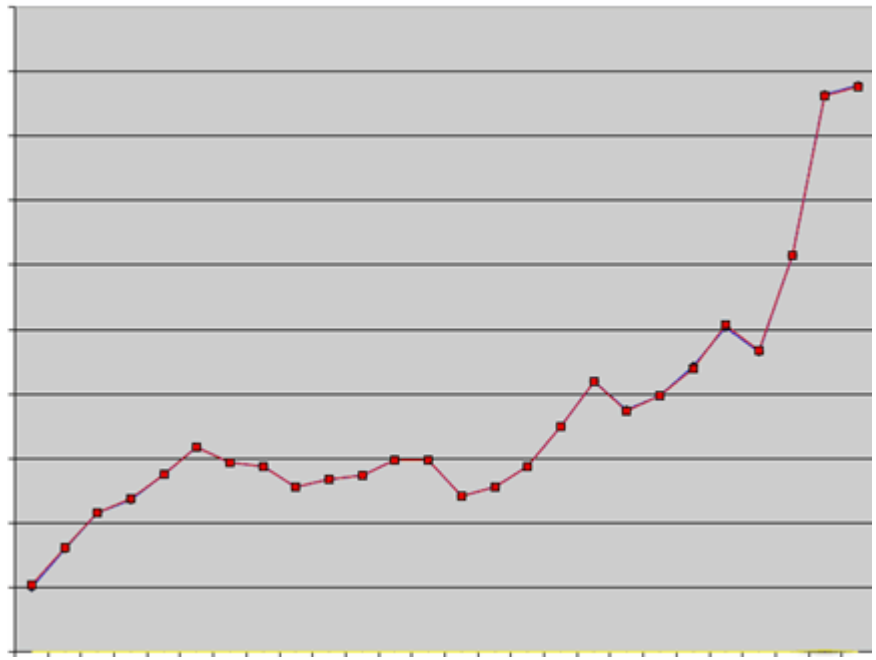
More on tuning

- If you can’t measure it, how do you know you’ve improved?
- Sometimes you just have to throw hardware at it, and/or fix your application.

Scale

What happens when your traffic doubles? Triples?

Digg.com?
Perez Hilton?
TV mentions?



The one server solution

httpd
mysqld
cpanel

Blog1
Blog2
…
BlogN

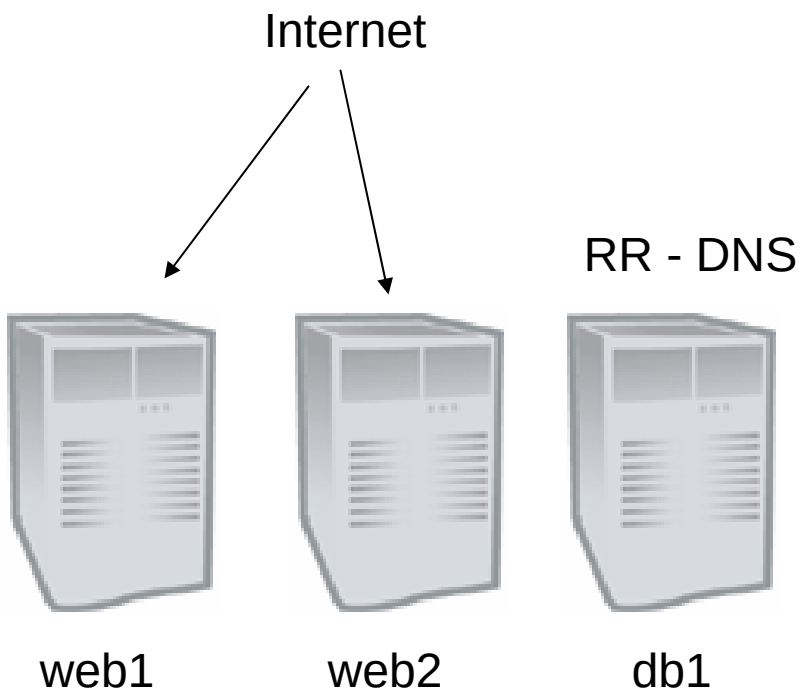
Time to grow!

httpd	httpd mysqld cpanel
BlogN+1 BlogN+2 …	Blog1 Blog2 … BlogN

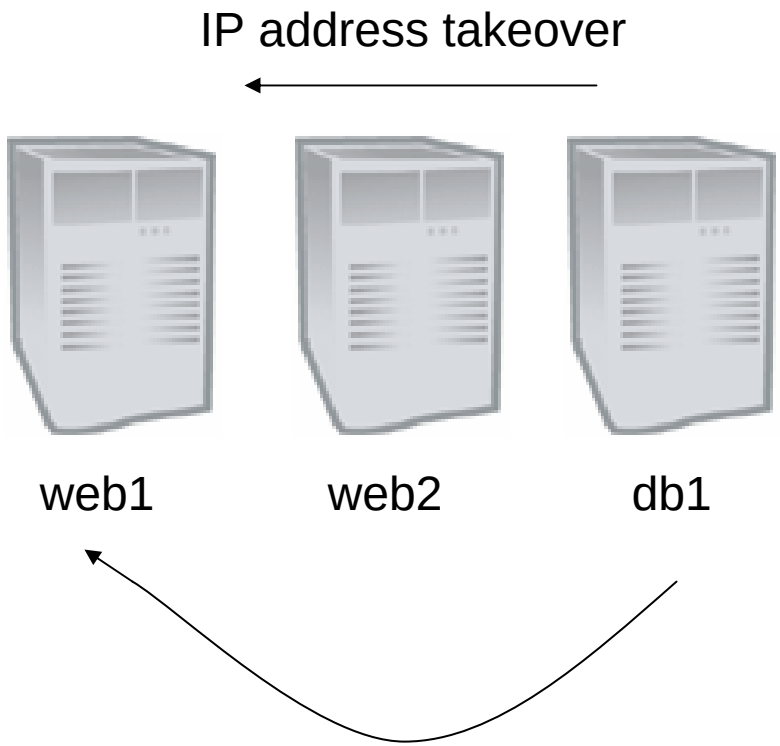
That wasn’t working

- Frequent database crashes
- No failover
- No load sharing
- One cpanel admin, one manual admin

Let’s get another server!



Believe it or not, that wasn't too bad.



What about MySQL?

I forgot to mention I fixed that before the third server came in.

It just makes for a better story to get into that now.

And for that matter, what about PHP and Apache?

MySQL tuning in 4 slides

Fix your fscking queries!

```
[mysqld]
; enable the slow query log, default 10 seconds
log-slow-queries
; log queries taking longer than 5 seconds
long_query_time = 5
; log queries that don't use indexes
; MySQL 4.1 and newer only
log-queries-not-using-indexes
```

Look at server-slow.log, run mysqldumpslow, learn to use EXPLAIN, and fix what needs fixing.

All the hardware and tuning in the world won’t fix bad queries

Query Caching

Store the results of queries in RAM (disabled by default)

query_cache_size = 64M ; or whatever

```
mysql> SHOW STATUS LIKE 'qcache%';
+-------------------------+------------+
| Variable_name           | Value      |
+-------------------------+------------+
| Qcache_free_blocks      | 5216       |
| Qcache_free_memory      | 14640664   |
| Qcache_hits             | 2581646882 |
| Qcache_inserts          | 360210964  |
| Qcache_lowmem_prunes    | 281680433  |
| Qcache_not_cached       | 79740667   |
| Qcache_queries_in_cache | 16927      |
| Qcache_total_blocks     | 47042      |
+-------------------------+------------+
8 rows in set (0.00 sec)
```

Cache table file descriptors

```
mysql> SHOW STATUS LIKE 'open%tables';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| Open_tables   | 5000  |
| Opened_tables | 195   |
+---------------+-------+
```

MySQL has 5000 tables open, ready for use.

195 tables had to be opened because a cached one was not available.

Is that good? Depends on how long the server has been up!

`table_cache = 5000`

Cache MyISAM keys

```
mysql> show status like '%key_read%';
+-------------------+-----------+
| Variable_name     | Value     |
+-------------------+-----------+
| Key_read_requests | 163554268 |
| Key_reads         | 98247     |
+-------------------+-----------+
```

```
Reads / read requests = miss rate (miss == disk access)
```

Key efficiency should be >= 99.9%. Increase buffer size with key_buffer

How big are your .myi files? That’s a good starting point for key_buffer.

I said 4 slides!

There’s a lot more to it. ☹

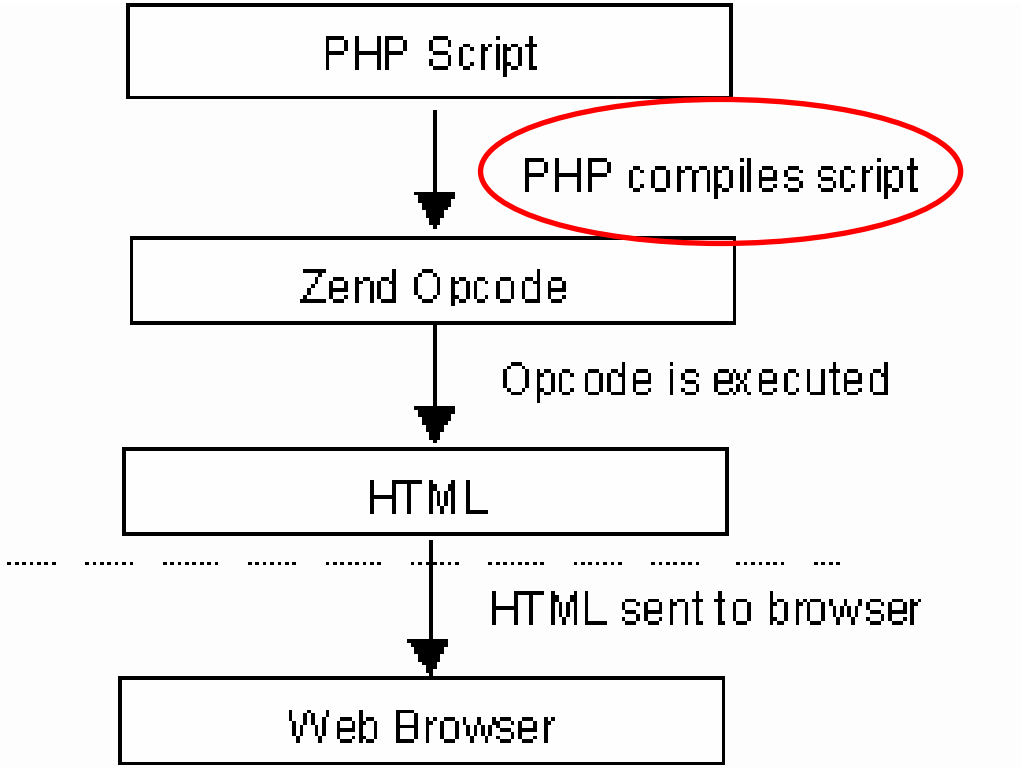
Fix those 4 things is a good start.

Read my article on dW for other tuning parameters.

PHP

There are some knobs to tweak

But if you aren't running an opcode cache, you're wasting your (CPU) time.



Original source unknown

The opcode cache also saves disk access

Compiled scripts stored in memory

(yea, you’d better watch your usage too)

Apache 

Don’t allow too many connections

Avoid .htaccess if you can, scope queries if you must

Read my article on dW for all the gory details

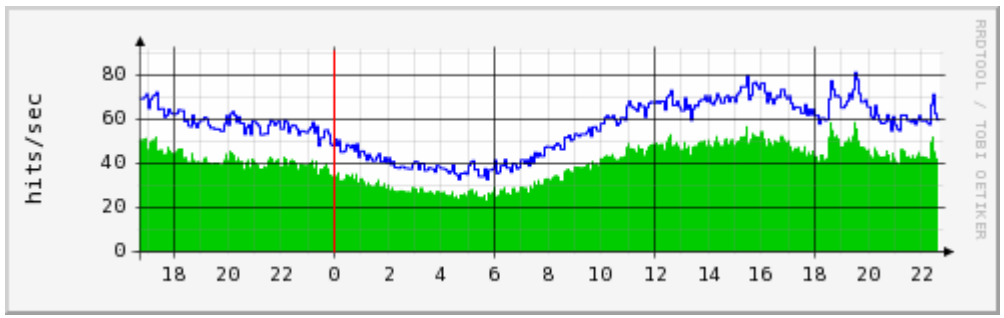
LAMP**S**

Inbound requests

Misses

Squid caches output
of web servers

Web Slaves



X 2

Wow, that DB is busy!

Internal API was causing 5 million trips to the web servers and database per day.

Tweak the application to use memcached

Memcached uses idle memory from servers to build a distributed hash table

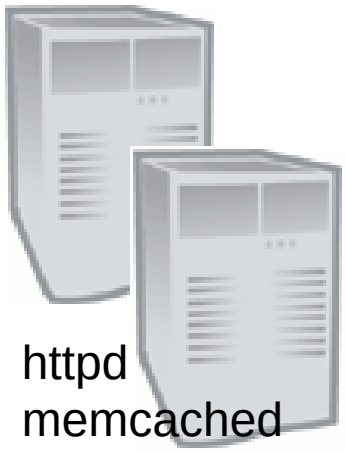
mysqld

httpd
memcached

function getblogroll($channelid)

1. Is the “blogroll:$channelid” in memcached?
2. Yes? Great!
3. No? Query DB/REST/whatever
4. Store results in memcached as “blogroll:$channelid” with an expiry

But it’s even better than that!



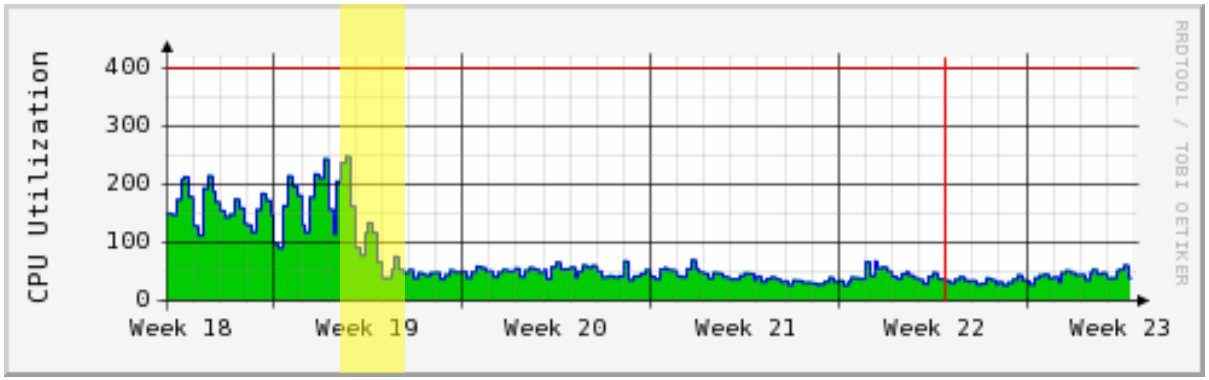
httpd
memcached

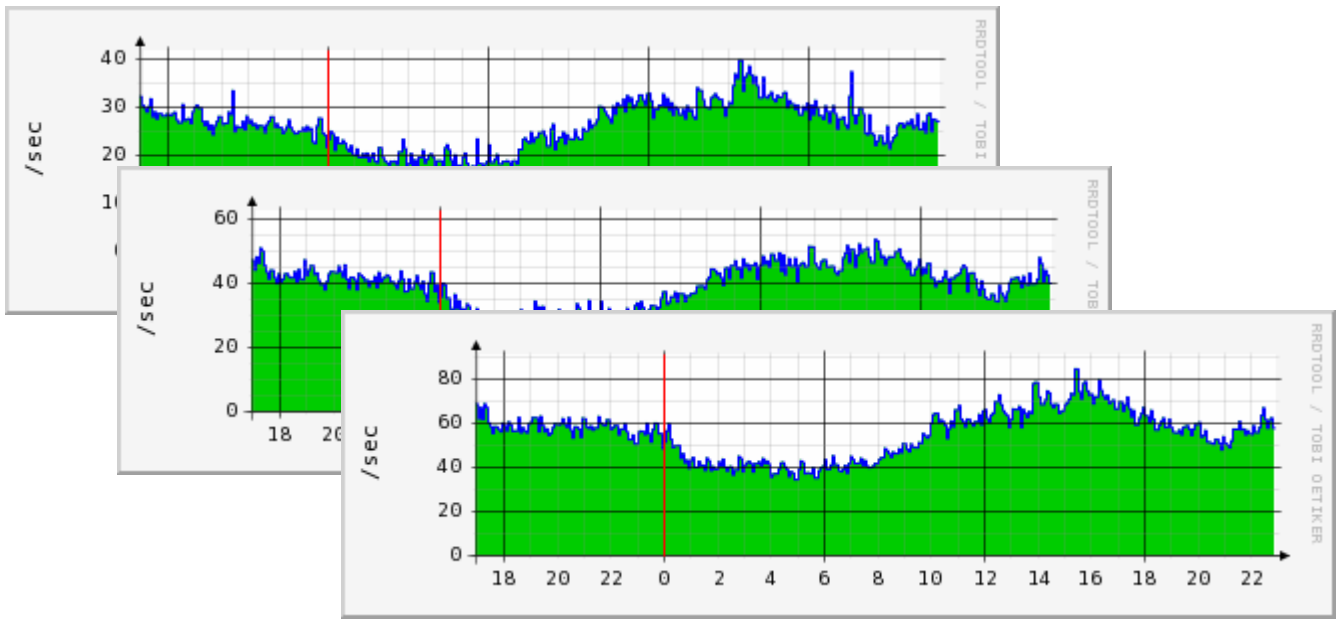
httpd
memcached

```
function getfrommemcached($key) {

   $servertocheck = hash($key) % $numservers
   ...

}
```

Client libraries take care of hashing and figuring out if servers are alive
PECL::memcached

The results


CPU Utilization
400
300
200
100
0
Week 18
Week 19
Week 20
Week 21
Week 22
Week 23
RRDTOOL / TOBI OETIKER


/sec
40
30
20
/sec
60
40
20
0
18
/sec
80
60
40
20
0
18
20
22
0
2
4
6
8
10
12
14
16
18
20
22
RRDTOOL / TOBI OETIKER

This has nothing to do with tuning

But you want to be spending your time growing your business.

Not tuning

Not managing systems

So automate everything (blog creation, server configuration, etc)

while (in_business) $servers++

They’re like rabbits, you know.

My checklist to add a new server was a page long.

And every time I wanted to update a configuration file…

1. Define policies
2. Configure servers to fetch policies
3. Let ‘er go

Cool things about Cfengine

```
copy:

   webslaves::
       $(masterfiles)/usr/local/b5media
              dest=/usr/local/b5media
              owner=root group=0 mode=0644
              backup=true
              recurse=inf
              type=checksum
              inform=true syslog=true
              server=$(policyhost)
```

Keeps a local directory synchronized from the master

What’s a web slave?

```
classes:
 # based on server name
 nonwebslaves = ( b5media_db1 b5media_db2 )
 webslaves = ( any -nonwebslaves )
 # based on functions
 nisservers = ( FileExists(/var/yp/b5media) )
 nisclients = ( any -nisservers )
```

Editing files

```
nisclients::
        { /etc/yp.conf
        AppendIfNoSuchLine "ypserver lwdb2"
        AppendIfNoSuchLine "ypserver lwdb1"
        }
```

Sequences

```
shellcommands:
 restartmemcached::
                "/sbin/service memcached restart"
editfiles:
 memcached::
        { /etc/sysconfig/memcached
   AutoCreate
   BeginGroupIfNoLineContaining  "OPTIONS"
     EmptyEntireFilePlease
     Append "OPTIONS=\"-d -l ${global.ipv4[eth1]} -u nobody -t 1 -m 2048\""
     DefineInGroup "restartmemcached"
   EndGroup
   LocateLineMatching "^OPTIONS=.*"
   # These lines had better be identical!!! probably move to a var
   ReplaceLineWith "OPTIONS=\"-d -l ${global.ipv4[eth1]} -u nobody -t 1 -m
    2048\""
   DefineClasses "restartmemcached"
}
```

Package Management

```
packages:
  webslaves::
     php version=5.1.6-3 cmp=eq
          elsedefine=needsphpupgrade

shellcommands:
  needsphpupgrade::
     "/bin/echo I need a php upgrade"
```

Building a new host

1. Copy update.conf from another host to /var/cfagent/inputs
2. Run cfagent
3. (optional) add host name to cfrun.hosts on master

Getting started

1. Read the quick start and get one host updating itself
2. Think of a single task, Google it, figure it out
3. Find the reference manual for all the keywords (broken up by action), and get going
4. Configure classes, deploy to other servers

Links

- http://seanwalberg.com
- http://del.icio.us/SeanW/
  - Cfengine, tuning, performance tags
- http://danga.com/words/

Did you find that exciting?

b5media is looking for a full time Server Ninja.

- Work from home (or Toronto)
- Lots of interesting things to do
- Small team, freedom to do what you want