

Build a High-Performance Server in “No Time At All”™

Chris Skene

The Problem

- Drupal is slow
- MySQL is slow
- Apache is slow
- mod_php is slow

The Drupal 6 problem

- Drupal is slow
- MySQL is slow
- Apache is slow
- mod_php is slow
- PHP 5.2 is slow

The solution

- Make Drupal faster
- Use MongoDB + MySQL
- Use Nginx instead of Apache
- Use PHP FastCGI Process Manager
- Use PHP 5.3
- (Use Drupal 7)

When to use this?

- All the time

Especially...

- Many concurrent logged-users
- Anywhere Varnish is not available/practical

But not...

- For Drupal 6 (unless you are a masochist)

GETTING SET UP

The Server

- Linode 512 MB slice
- 4 cores
- 10112 MB storage
- 256 MB swap
- Fremont data centre

The configuration

- Ubuntu 12.04 LTS 64 Bit
- PHP 5.3.10
- MySQL 5.5
- MongoDB
- Nginx 1.1.19
- PHP-FPM

The solution

- Make Drupal faster
- Use MongoDB (+ MySQL)
- Use Nginx instead of Apache
- Use PHP FastCGI Process Manager
- ~~Use PHP 5.3~~
- (Use Drupal 7)

INSTALLATION WALK- THROUGH

Getting set up

- Install Ubuntu 12.04 (or upgrade 10/11)
- Good install guides in the Linode Library
 - Initial setup
 - Security
- Install MySQL

```
apt-get install mysql-server mysql-client
```
- 16 minutes

Nginx

- Asynchronous web server
 - Can serve many clients simultaneously (000's)
 - Minimal memory footprint
 - Potentially much faster than Apache
 - **AND DOESN'T SLOW DOWN**
-
- No .htaccess (use site .conf file instead)
 - copy config from <http://wiki.nginx.org/Drupal>

Installing Nginx

```
apt-get install nginx  
service nginx start
```

Nginx options

In nginx.conf:

- set worker processes to 2 x server cores

Advanced Nginx/Drupal config available at:

<https://github.com/perusio/drupal-with-nginx>

Complex, lots of options, requires some background knowledge. Remember to use `nginx -t` to debug.

The solution

- Make Drupal faster
- Use MongoDB (+ MySQL)
- ~~Use Nginx instead of Apache~~
- Use PHP FastCGI Process Manager
- ~~Use PHP 5.3~~
- (Use Drupal 7)

PHP5-FPM

- Apache loads all of PHP on EVERY REQUEST
- PHP-FastCGI + FPM loads PHP once for many requests, pooling connections
- Dynamic PHP process spawning
- Gracefully start and stop PHP processes
- An alternative to spawn-fcgi

Install PHP5-FPM

```
apt-get install php5-fpm  
service php5-fpm start
```

Configure PHP5-FPM sockets (optional)

- Open FPM configuration

```
nano /etc/php5/fpm/pool.d/www.conf
```

- Change...

```
listen=127.0.0.1:9000 to...
```

```
listen=/var/run/php5-fpm.sock
```

```
/etc/init.d/php5-fpm restart
```

The solution

- Make Drupal faster
- Use MongoDB (+ MySQL)
- ~~Use Nginx instead of Apache~~
- ~~Use PHP FastCGI Process Manager~~
- ~~Use PHP 5.3~~
- (Use Drupal 7)

Get extra PHP things

```
apt-get install php5-mysql php5-curl php5-gd  
php5-idn php-pear php5-imagick php5-imap  
php5-mcrypt php5-memcache php5-ming php5-ps  
php5-pspell php5-recode php5-snmp php5-  
sqlite php5-tidy php5-xmlrpc php5-xsl
```

```
/etc/init.d/php5-fpm restart
```

In php.ini:

```
cgi.fix_pathinfo = 0;
```

MongoDB

- Document store (NoSQL), not Relational store (MySQL)
- Like MySQL, supports replication,
- Unlike MySQL, supports sharding, atomic in-place updates
- Suits “high-volume problems” and anything “document like” (e.g. nodes)
- Fast!

Installing MongoDB

- Import the 10gen key:

```
apt-key adv --keyserver keyserver.ubuntu.com  
--recv 7F0CEB10
```

- Add 10gen to `/etc/sources.list`:

```
deb http://downloads-distro.mongodb.org/  
repo/ubuntu-upstart dist 10gen
```

Install MongoDB (cont...)

```
apt-get update  
apt-get install mongodb-10gen
```

```
service mongodb start
```

The solution

- Make Drupal faster
- ~~Use MongoDB (+ MySQL)~~
- ~~Use Nginx instead of Apache~~
- ~~Use PHP FastCGI Process Manager~~
- ~~Use PHP 5.3~~
- (Use Drupal 7)

Configure Nginx

```
touch /etc/nginx/sites-available/drupal.conf
```

copy config into drupal.conf from:

<http://wiki.nginx.org/Drupal>

set fastcgi_pass value to socket from FPM:

`/var/run/php5-fpm.sock`

or IP:Port combination:

`127.0.0.1:9000`

Configure Nginx (cont...)

Change the root directory

```
ln -s /etc/nginx/sites-available/  
drupal.conf /etc/nginx/sites-enabled
```

```
nginx -t
```

```
service nginx restart
```

Get Drupal

- Download Drupal into your server directory
- Change ownership to Nginx user (usually www-data)
- Install!

Get Drupal

```
cd /var
mkdir www
cd www
wget http://ftp.drupal.org/files/projects/
      drupal-7.12.tar.gz
tar -xzf drupal-7.12.tar.gz
mv drupal-7.12 drupal
chown -R www-data:www-data www
```

The solution

- Make Drupal faster
- ~~Use MongoDB (+ MySQL)~~
- ~~Use Nginx instead of Apache~~
- ~~Use PHP FastCGI Process Manager~~
- ~~Use PHP 5.3~~
- ~~(Use Drupal 7)~~

MongoDB for Drupal

<http://drupal.org/project/mongodb>

- Cache
- Field storage
- Session
- Lock
- Watchdog
- Queue
- Blocks
- API

Usage

- Cache, session, lock, queue and watchdog can be turned on straight away
- Field storage migrates existing content when saved (test)
- Block modules requires some care (test)
- Follow readme.txt instructions (test)
- Bleeding edge (test, test, test)

Field Storage Caveats

- Can't be disabled
- No way to migrate content back to MySQL
- No Views integration AT ALL

but

- API is EntityFieldQuery aware...

The solution

- ~~Make Drupal faster~~
- ~~Use MongoDB (+ MySQL)~~
- ~~Use Nginx instead of Apache~~
- ~~Use PHP FastCGI Process Manager~~
- ~~Use PHP 5.3~~
- ~~(Use Drupal 7)~~

How long did it take?

- Ate dinner
- Wrote a presentation
- Made a cup of tea
- Checked on my son
- Watched “Shakespeare Wallah” on DVD

How long did it take?

1 hour 15 minutes

Other things you can do

- Use MySQL replacements
 - Maria DB
 - Percona Server
- APC
- Memcache
- Varnish

@xtfer

Christopher Skene

chris@xtfer.com