



Drupal 7.

The highest performance for anonymous users

Who

Am I

- Evgeniy Maslovskiy
- Team lead at Bright Solutions
- Maintainer and co-maintainer of lots Drupal contribs
- @Spleshka



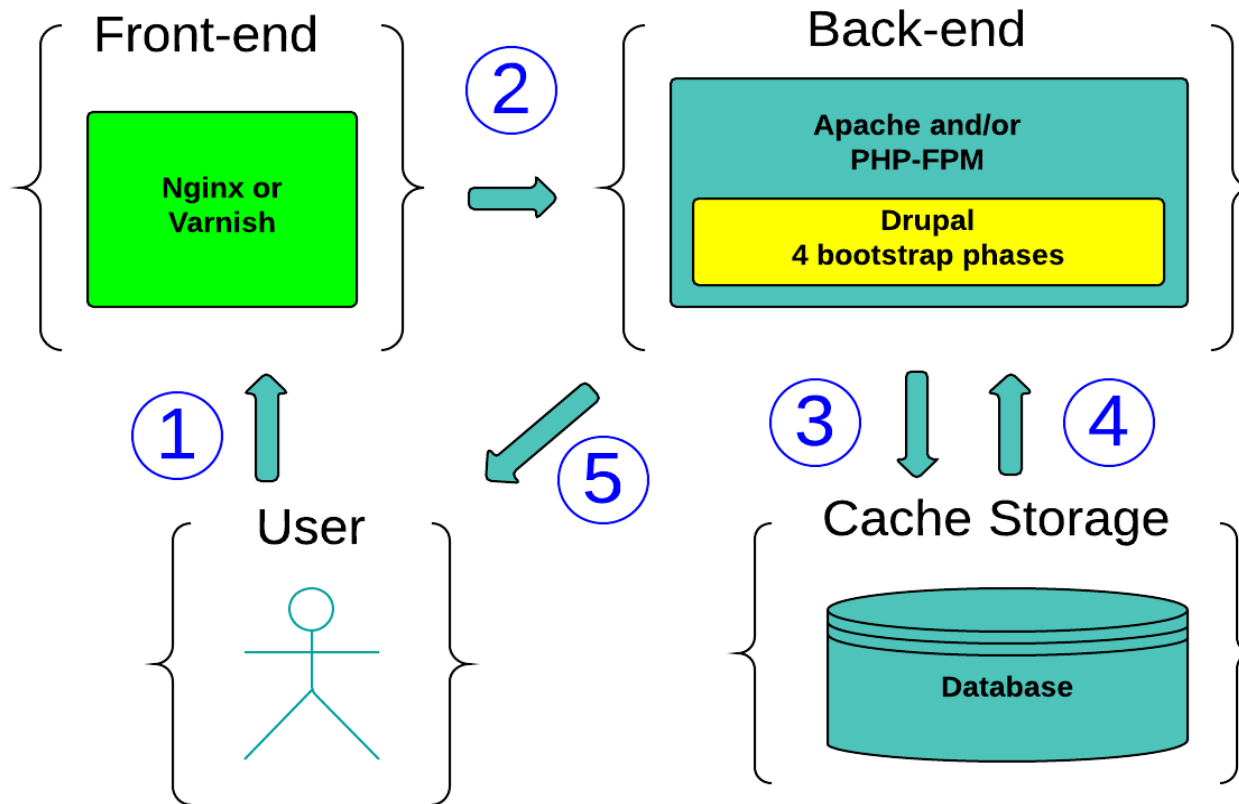
How

Websites get a lot of anonymous

- Search engines
- Spam systems
- DDoS attacks
- Unexpected traffic explosion
- Just a lot of anonymous users

How

Drupal handles anonymous by default



What

Simple ways to improve performance

- Memcache or Memcache Storage
- APC
- Boost
- File cache
- Redis

Drupal sample

Implementation of such method

```
$conf['cache_backends'][] = 'sites/all/modules/memcache_storage/memcache_storage.inc';
```

```
$conf['cache_default_class'] = 'DrupalDatabaseCache';
```

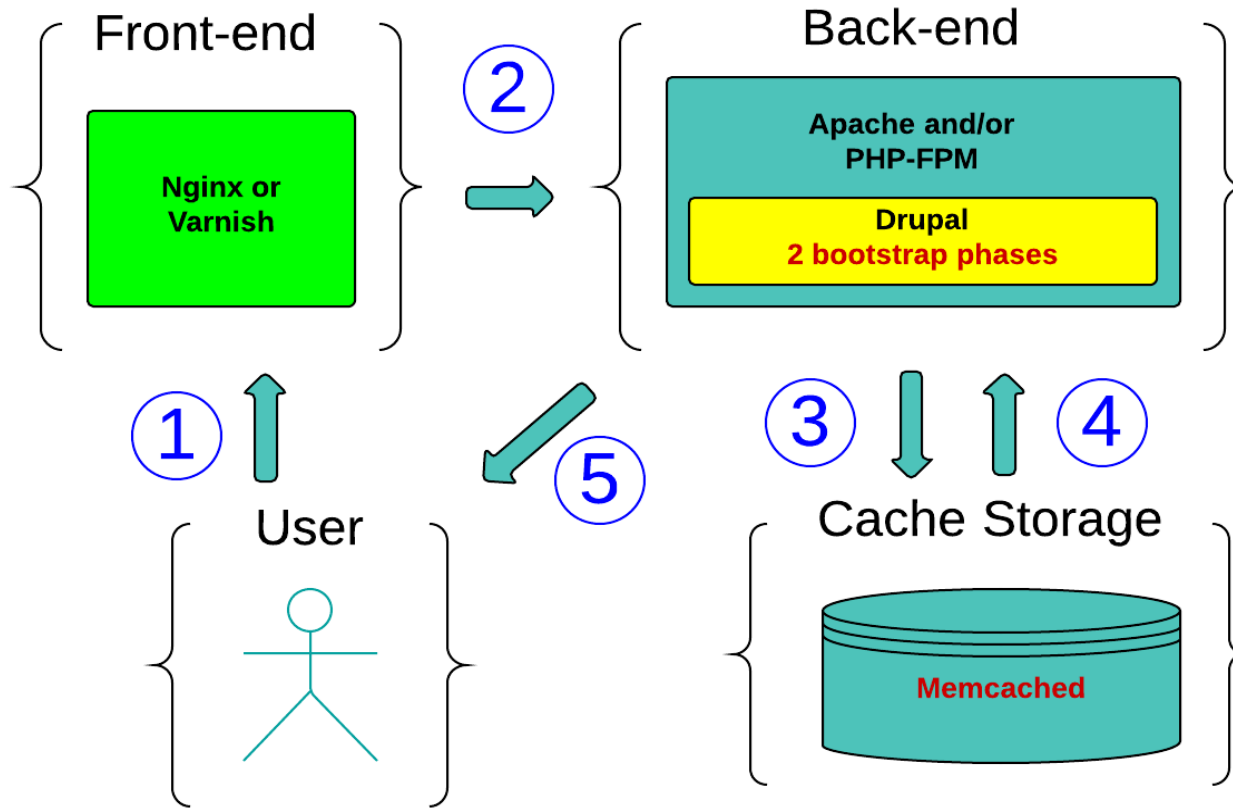
```
$conf['cache_class_cache_page'] = 'MemcacheStorage';
```

```
$conf['page_cache_without_database'] = TRUE;
```

```
$conf['page_cache_invoke_hooks'] = FALSE;
```

What

That simple methods improves



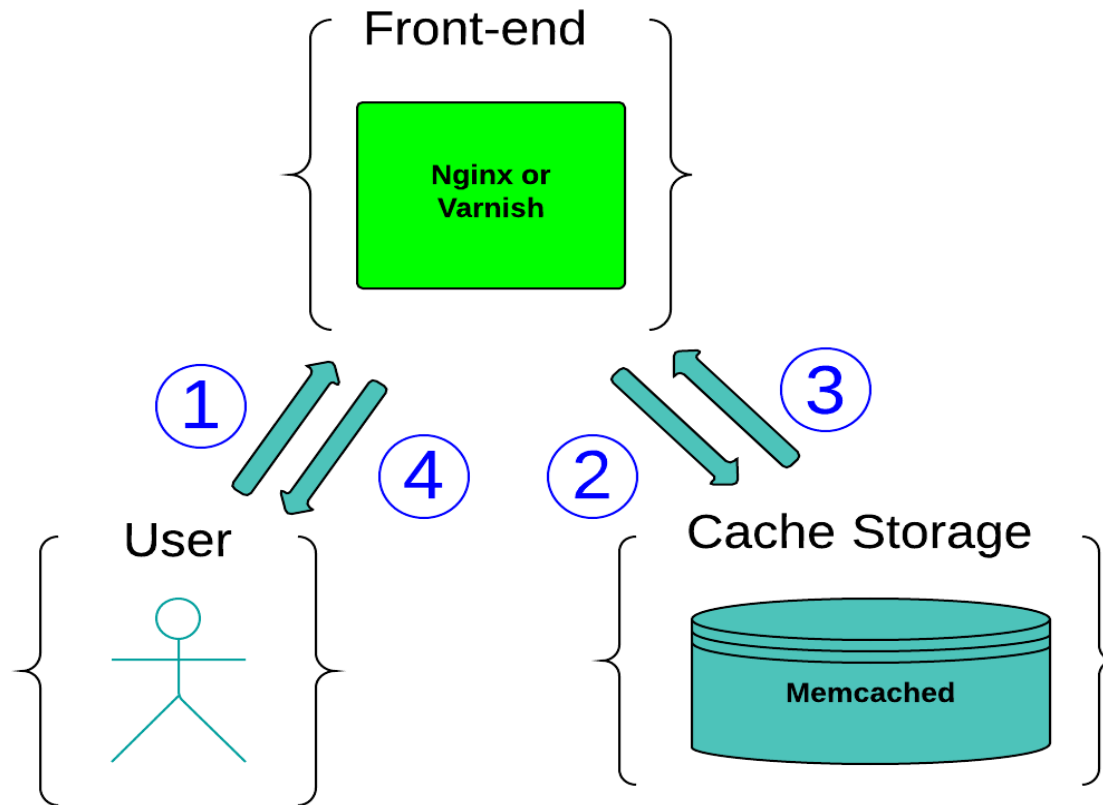
How

Can we avoid request to a backend

Configure proxy
server to work with
caching backend

How

Perfect workflow looks like



Drupal sample

Implementation

Move all cached data (except form cache) to memcache storage.

```
$conf['cache_backends'][] = 'sites/all/modules/memcache_storage/memcache_storage.inc';  
$conf['cache_default_class'] = 'MemcacheStorage';  
$conf['cache_class_cache_form'] = 'DrupalDatabaseCache';
```

Advanced usage of Drupal page cache.

```
$conf['cache_backends'][] = 'sites/all/modules/memcache_storage/memcache_storage.page_cache.inc';  
$conf['cache_class_cache_page'] = 'MemcacheStoragePageCache';
```

Enable storing of plain HTML text instead of Drupal usual cache object.

```
$conf['memcache_storage_external_page_cache'] = TRUE;
```

Never expire page cache.

```
$conf['memcache_storage_page_cache_custom_expiration'] = TRUE;  
$conf['memcache_storage_page_cache_expire'] = 0;
```

Nginx sample Implementation

Set output format.

```
default_type text/html;
```

Set a proper key for search in memcached pool.

```
set $memcached_key "cache_page-$scheme://$server_name$uri$is_args$args"; # without prefix
```

```
# set $memcached_key "PREFIX-cache_page-$scheme://$server_name$uri$is_args$args"; # with prefix
```

Set gzip flag to 1 if page compression in Drupal is enabled.

```
memcached_gzip_flag 1;
```

Route nginx to memcached socket (or ip).

```
memcached_pass unix://var/run/memcached/memcached.socket;
```

If cache is not found - pass request to the backend.

```
error_page 404 502 = @backend;
```

What

Issues you may face

- Keep pages up to date

Solve it with Cache Expiration module

- Need for up-to-date block on all page

Solve it with SSI/ESI

What

Requirements has this approach

- You should have an access to server configuration
- You should be (or you should pay) a skilled Drupal architect

Which

Benefits does this approach have

- Incredibly fast response
- Significantly decrease server load
- Save a lot of RAM
- Much less database connections
- Could be scaled
- Advanced configurations

Why

Not native proxy cache

- Storage
- Cache management
- Scalability
- Drupal API



brightsolutions

we make innovation in reality. in web and mobile.