Optimiser les performances Drupal par le cache

Meetup Drupal Lyon - 21 mars 2013

Acquia™

Qui suis-je?

Client Advisor, Tier 2

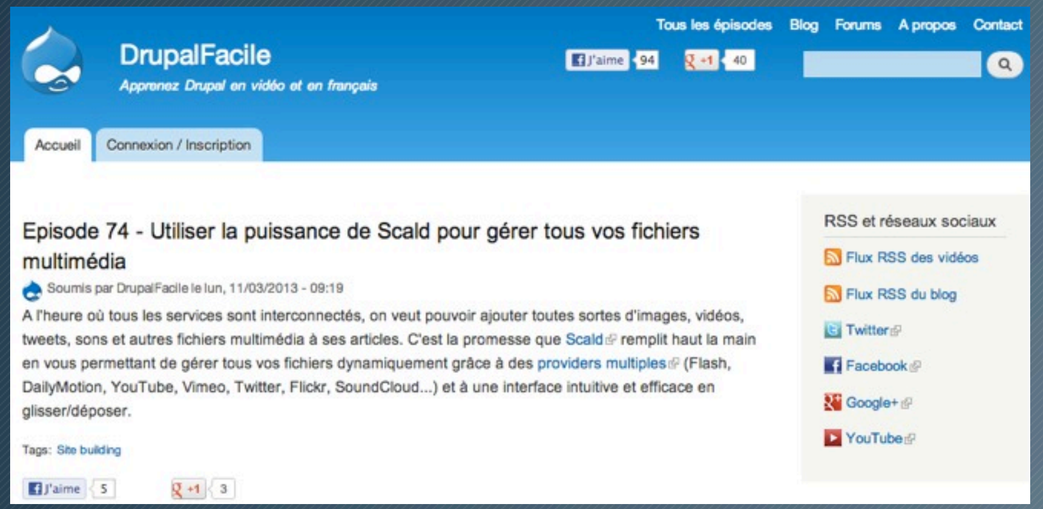
- **Produits** : Acquia Cloud (Enterprise), Drupal Gardens, Drupal Commons, Acquia Dev Desktop...
- **Offres** : audits, conseil, support et expertise Drupal
- **Nos clients** : Twitter, Intel, Ebay, Paypal, Al Jazeera, World Economic Forum, gouvernements, institutions, organisations, etc..

@AurelienNavarre

- 70+ tutoriels vidéos Drupal 7+ en français
- 600+ visites par jour / 1400+ abonnés
- 1400+ abonnés YouTube / 325k+ vues

www.drupalfacile.org

@DrupalFacile



Acquia™

Le cache en quelques mots

- Le cache correspond à du **stockage temporaire de données**. C'est le plus souvent le résultat d'une opération gourmande stockée en **mémoire** (ex : memcache, Redis, Varnish...) ou sur le **disque** (ex : boost...) que l'on souhaite renouveler le moins souvent possible.

- **Le cache améliore considérablement les performances d'un site web**. Gérer une requête de page Drupal complète (page, blocs, menus, thème...) est une opération gourmande. Plus on pourra recycler les opérations, plus le site sera optimisé.

- Une mise en cache efficace est la **seule façon de survivre avec succès à une forte pointe de traffic** ou à une attaque DOS ou DDoS.

- **Ne stockez jamais en cache des données pérennes**. La définition même du cache est de pouvoir détruire à tout moment les données.

Les différents composants du cache Drupal

Acquia™

Les tables de cache

```
mysql> SHOW TABLES LIKE 'cache_%';

+----------------------------------------------+
| Tables_in_D7 (cache_%)                       |
+----------------------------------------------+
| cache_apachesolr                             |
| cache_block                                  |
| cache_bootstrap                              |
| cache_field                                  |
| cache_filter                                 |
| cache_form                                   |
| cache_image                                  |
| cache_menu                                   |
| cache_metatag                                |
| cache_page                                   |
| cache_path                                   |
| cache_token                                  |
| cache_update                                 |
| cache_views                                  |
| cache_views_data                             |
+----------------------------------------------+

23 rows in set (0.00 sec)
```

- Le cache de Drupal se compose d'une table **cache** et d'une multitude de tables de la forme **cache_***
- **cache_form** est une table à part et ne correspond pas à du stockage temporaire comme les autres (formulaires du site en cours de remplissage)
- Utilisez la fonction **cache_set()** pour stocker les données en cache :

```php
<?php

function cache_set($cid, $data, $bin = 'cache', $expire =
CACHE_PERMANENT) {
  return _cache_get_object($bin)->set($cid, $data, $expire);
}
```

Acquia™

Le registre de thème

- C’est le meilleur ennemi du thémeur. Le registre de thème de Drupal garde en cache les données telles que le fichier **.info**, les **fichiers tpl.php** mais aussi les **hooks du thème** (ex : hook_preprocess()).

- Les thèmes Drupal les plus populaires (Zen, Omega, Fusion...) proposent souvent une case à cocher pour automatiquement purger le registre de thème et faciliter la vie du thémeur. **Attention à ne jamais activer cette fonctionnalité en production !**

- Plusieurs modules permettent de vider le registre de thème (Admin menu, Devel...) mais aussi drush (*drush cc all* ou mieux car plus spécifique, *drush cc theme-registry*).

- Les fonctions drupal_theme_rebuild() sous Drupal 7 ou drupal_rebuild_theme_registry() sous Drupal 6 vous permettront d’implémenter les mêmes opérations.

Le registre de code

- Introduit avec Drupal 7, le registre de code est un **inventaire de toutes les classes et interfaces pour tous les modules activés et fichiers de Drupal core**.

- Le registre de code stocke simplement le/les chemin(s) défini(s) dans une classe ou une interface et charge le/les fichier(s) lorsque c'est nécessaire.

- Les modules peuvent désormais déplacer tout leur code dans un fichier séparé (*include*) pour les classes qui ne sont pas régulièrement utilisées. Drupal les chargera alors à la demande, ce qui optimisera les performances puisqu'aucun code superflu ne sera inutilement "parsé" par PHP.

- La fonction registry_rebuild() ou le "module" Registry Rebuild permettent de débugger un site qui renvoie un WSOD (White Screen Of Death) lorsque (par exemple) des modules ont été déplacés et que Drupal ne trouve plus les classes correspondantes.

Les modes de cache

- **Désactivé** : très bien pour le développement...mauvais pour tout le reste. Ne désactive que le cache de page, pas tout le cache.
- **Normal** : stocke toutes les versions des pages mises en cache dans la base de données et les sert aux utilisateurs anonymes.
- **Aggressive** : ce mode de cache fait que Drupal va éviter le chargement (*boot*) et dé-chargement (*exit*) des modules activés lorsqu'il sert une page en cache. Cela augmente les performances mais peut avoir des effets indésirables. Réglage masqué depuis Drupal 7 que l'on pourra ré-introduire avec un $conf dans settings.php
- **External** : mode recommendé pour Pressflow derrière un reverse proxy tel que Varnish. N'est plus nécessaire depuis Drupal 7 mais on peut le ré-introduire avec External cache.

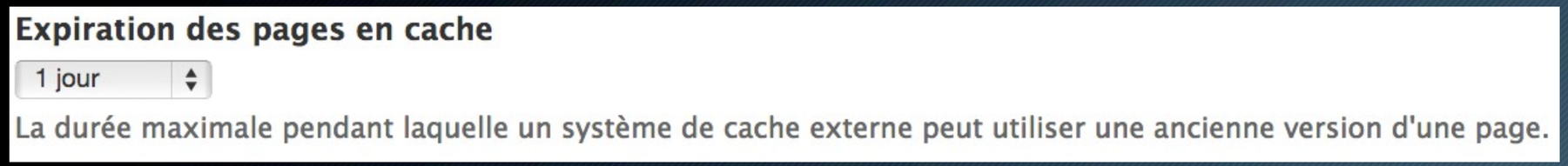
Durée de vie minimale du cache

- C’est le temps **pendant lequel une page sera mise en cache même si du nouveau contenu est ajouté**.

- **La durée de vie minimale du cache doit en principe être réglée à la plus haute valeur possible**. On peut avoir jusqu’à 1 jour par défaut dans Drupal core, ou jusqu’à 1 an grâce au module Cache Lifetime Options.

- Il y néanmoins certains cas (notamment avec memcache) où ce réglage sera contre-productif et on pourra recommander de le désactiver complètement. Notez que cache_lifetime sera complètement supprimé de Drupal 8.

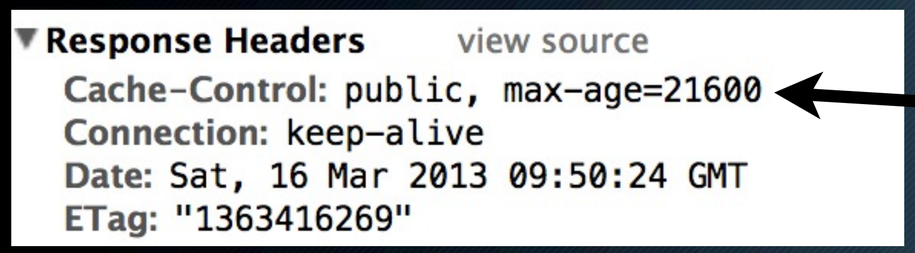
Expiration des pages en cache



- C’est le réglage qui permet de définir **pendant combien de temps un reverse proxy tel que Varnish mettra en cache les pages**.

- Vous retrouverez cette information via le header HTTP "*Cache-Control: max-age*".

- **L’expiration des pages en cache doit être réglée à la plus haute valeur possible**.



Ici le cache Varnish est de 6h

Acquia™

Optimiser les réglages du max-age

- Dans certains cas bien précis, on peut définir le **max-age** de façon granulaire dans **settings.php**, basé sur le chemin de la requête entrante :

```php
<?php

// Règle le max-age à 5mn pour la page des news
if ($_SERVER['SCRIPT_URL'] == '/news') {
   $conf['page_cache_maximum_age'] = 300;
}

// Règle le max-age à 1h pour le blog
if (strpos('/blog', $_SERVER['REQUEST_URI']) === 0 {
   $conf['page_cache_maximum_age'] = 3600;
}
```

Cache des blocs

- Le cache des blocs est la **façon la plus simple de mettre en cache des éléments pour les utilisateurs authentifiés**.

- Vous ne pourrez pas mettre en place le cache de blocs si votre site utilise un module qui tire parti de hook_node_grants() - C'est le cas pour Content Access, Domain Access, Forum Access, Organic groups (si *og_access* est activé)...et bien d'autres.

Comment déterminer si un module utilise hook_node_grants() ?

```
$ drush fn-hook node_grants (si Devel est activé)
$ drush pml | egrep '(content_access|forum_access|og_access|domain)'
```

Acquia™

Aggrégation et compression CSS et JS

- Sans ce réglage activé, Internet Explorer 9.0 et versions précédentes ne pourront le plus souvent pas charger les feuilles de style de Drupal.
- Cela provient du fait que **IE ne peut charger que 31 feuilles de style à la fois**.
- IE10 peut aller jusqu'à 4095 feuilles de style et règle donc ce problème.

- Tout comme l'aggrégation et compression CSS, c'est un réglage de cache qui augmente les performances.
- Attention cependant car du mauvais code JS dans un seul fichier peut "casser" le JS sur toutes vos pages.
- Utilisez le module Speedy pour optimiser le chargement des fichiers JS du core.

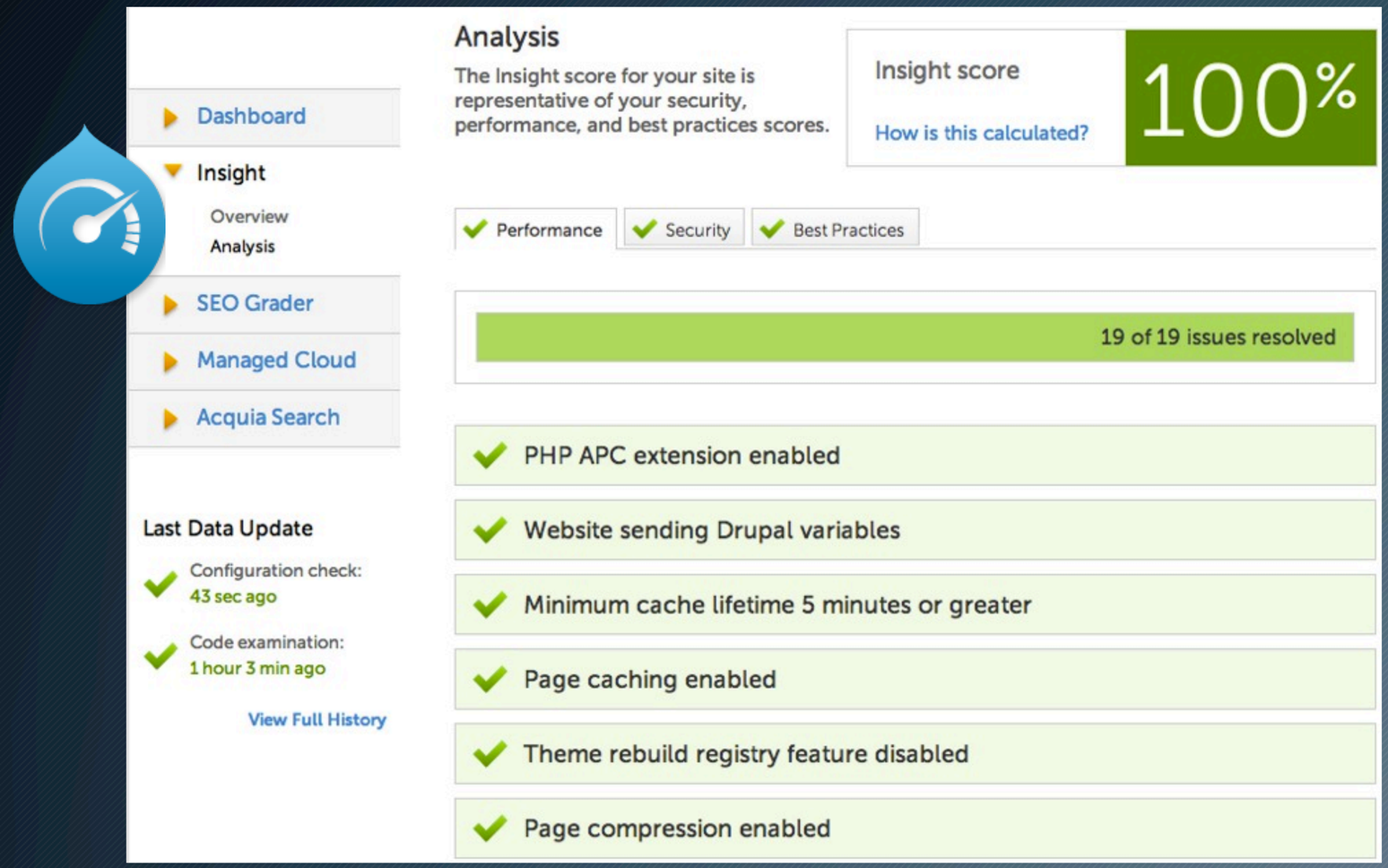
Acquia Insight - Quel score auriez-vous ?

Analyse des données

- Examination de la configuration
- Analyse du code (hacks, updates...)

Recommandations diverses

- Performances
- Sécurité
- Bonnes pratiques Drupal
- SEO Grader (partenariat Volacci)



Acquia™

Le cache, côté système

Acquia™

Varnish, pour les utilisateurs anonymes

- Varnish, c'est tout simplement du cache HTTP stocké en mémoire, autrement connu sous le terme "*accélérateur HTTP*".
- Il décuple le nombre de visiteurs anonymes concurrents qui peuvent accéder à un site web (ce qu'on appelle "*scalability*").
- Non seulement les requêtes des visiteurs anonymes sont plus rapides, mais elles évitent en plus complètement aux serveurs webs d'avoir à les exécuter, ce qui leur donne plus de ressources pour gérer les requêtes des utilisateurs authentifiés qui, par définition, ne peuvent utiliser Varnish.

Une mine d'info : les headers HTTP

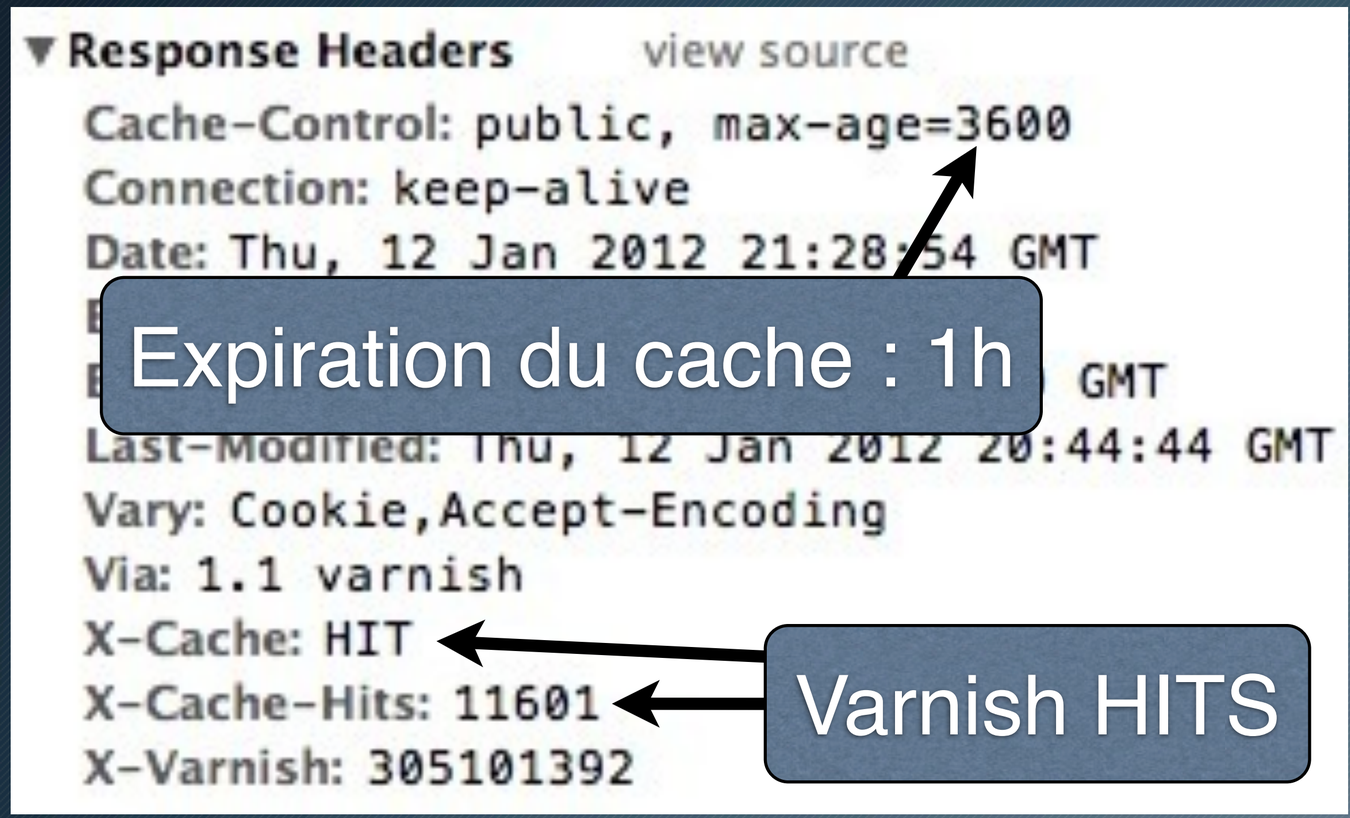
Quels outils utiliser ?

Firebug pour Firefox

Webkit Inspector

Ou via cURL...

```
$ curl -s -D /dev/stderr http://site.com
```



Acquia™

Memcache, pour les utilisateurs authentifiés

- Définition de http://memcached.org/ - "*Memcache est du stockage mémoire sous forme clé-valeur pour de petits morceaux de données arbitraires (chaînes, objets) qui résultent de requêtes de base de données, appels d'API, ou rendus de pages*."

- Le module Drupal Memcache API and Integration stocke les tables de base de données qui commencent par "*cache*" en mémoire. Il peut optionnellement stocker les sessions (pas encore fonctionnel sous Drupal 7).

- Puisque qu'il stocke les données en mémoire, Memcache est beaucoup plus rapide que MySQL qui écrit sur le disque et possède des mécanismes de cache moins puissants.

Memcache, en action

- Intégration typique de memcache dans **settings.php**. Notez que **cache_form** est envoyé vers la base de données et non memcache !

```
$conf['cache_backends'][] = './sites/all/modules/contrib/memcache/
memcache.inc';
$conf['cache_default_class'] = 'MemCacheDrupal';
$conf['cache_class_cache_form'] = 'DrupalDatabaseCache';
```

Est-ce que memcache fonctionne correctement ?

```
$ watch -td '(echo stats ; echo quit) | nc `hostname -s` 11211 |
grep get_hits '

STAT get_hits 17413371
...
STAT get_hits 17414399
```

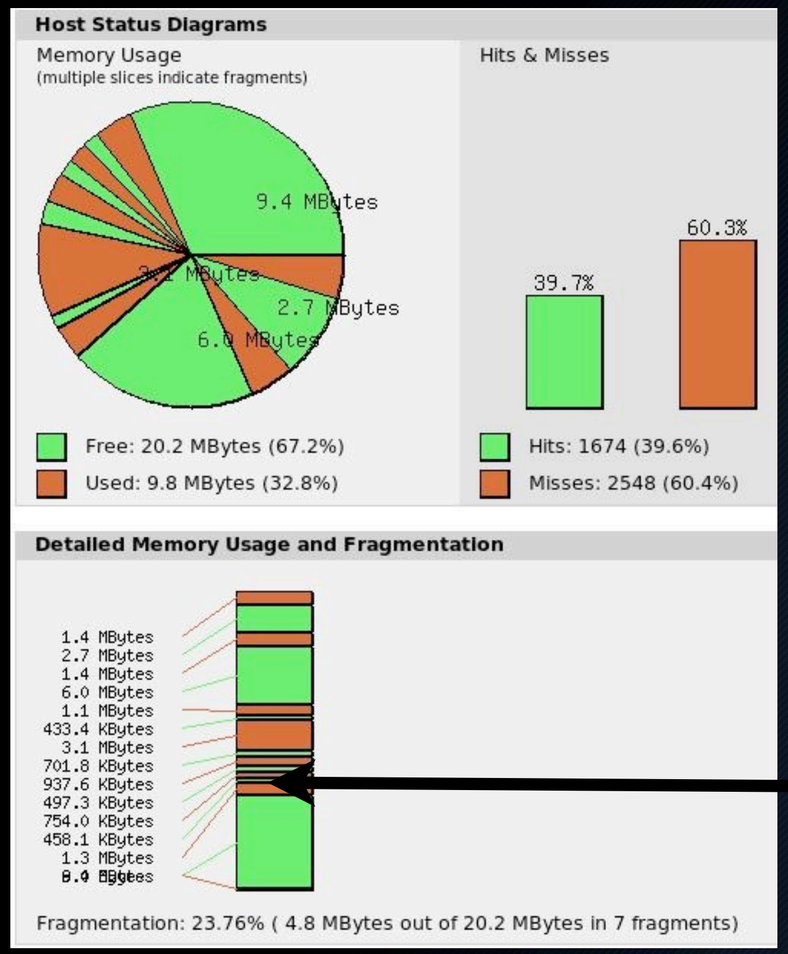
Acquia™

APC, le cache “intermédiaire” de PHP

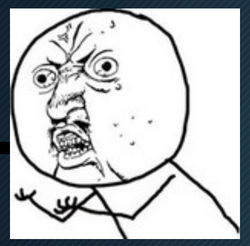
- PHP étant est un langage interprété, chaque accès à une page produit 4 opérations : chargement, parsing, compilation, puis enfin, exécution.
- Le cache “op-code” en élimine 3 en gardant uniquement la version compilée du script en mémoire pour la ré-utiliser la prochaine fois qu’il sera demandé.
- Le gain de performances CPU/RAM est significatif et immédiat - quasiment sans configuration - ce qui fait d’APC un incontournable de l’optimisation de performances.

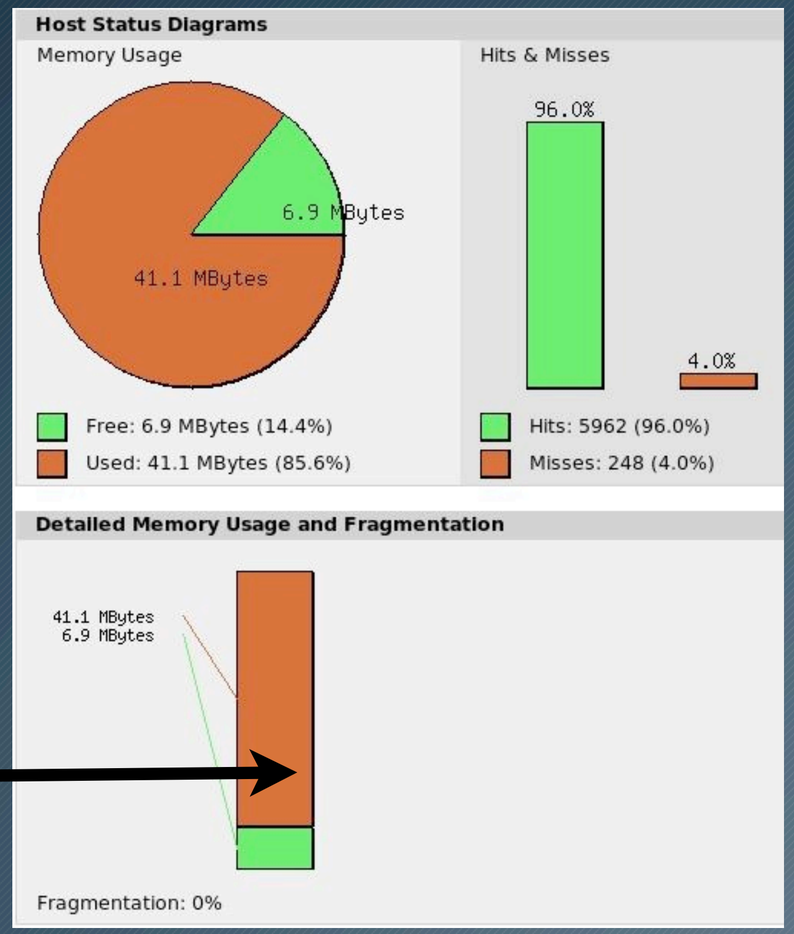
Acquia™

Surveiller la fragmentation APC



Plus la fragmentation APC sera basse, plus vous tirerez efficacement parti du cache...





Acquia™

Qu’est-ce qui invalide les caches ?

Acquia™

Drupal 6...

Solution :

Pressflow™

- Supprime le cookie de session de D6
- Gère le cache externe (Varnish)
- Mise en cache des alias d'URL
- Cookie cache bypass (forms, CAPTCHAs)
- ...

...ou Drupal 7 !

Acquia™

Trouver les cookies de session

Certains modules sont connus pour créer des cookies de session (SESS)

- Mobile Tools
- Mollom/CAPTCHA (quand un formulaire a été soumis avec succès)
- Views (avec filtre exposé et case à cocher “*Se rappeler du choix*”)
- Flags (pour les “flags” anonymes)
- Ubercart

- Liste disponible à https://pressflow.atlassian.net/wiki/pages/viewpage.action?pageId=589829
- Quelques solutions à l’adresse https://pressflow.atlassian.net/wiki/display/PF/Modules+that+break+caching,+and+how+to+fix+them

Rechercher du code qui établit un cookie de session

```
$ grep -inr --color=auto "_SESSION\['.*'\] = " * --exclude=\*.
{svn,po,html,xml,csv,xls,pdf,install,patch,css,js,txt}
```

Acquia™

Tirer parti l'API de Drupal 7

On peut utiliser la fonction drupal_add_http_header() pour manipuler les headers HTTP :

```
drupal_add_http_header('Cache-Control', 'public, max-age=0');
```

Une alternative est de définir un cookie header **NO_CACHE** à 0 :

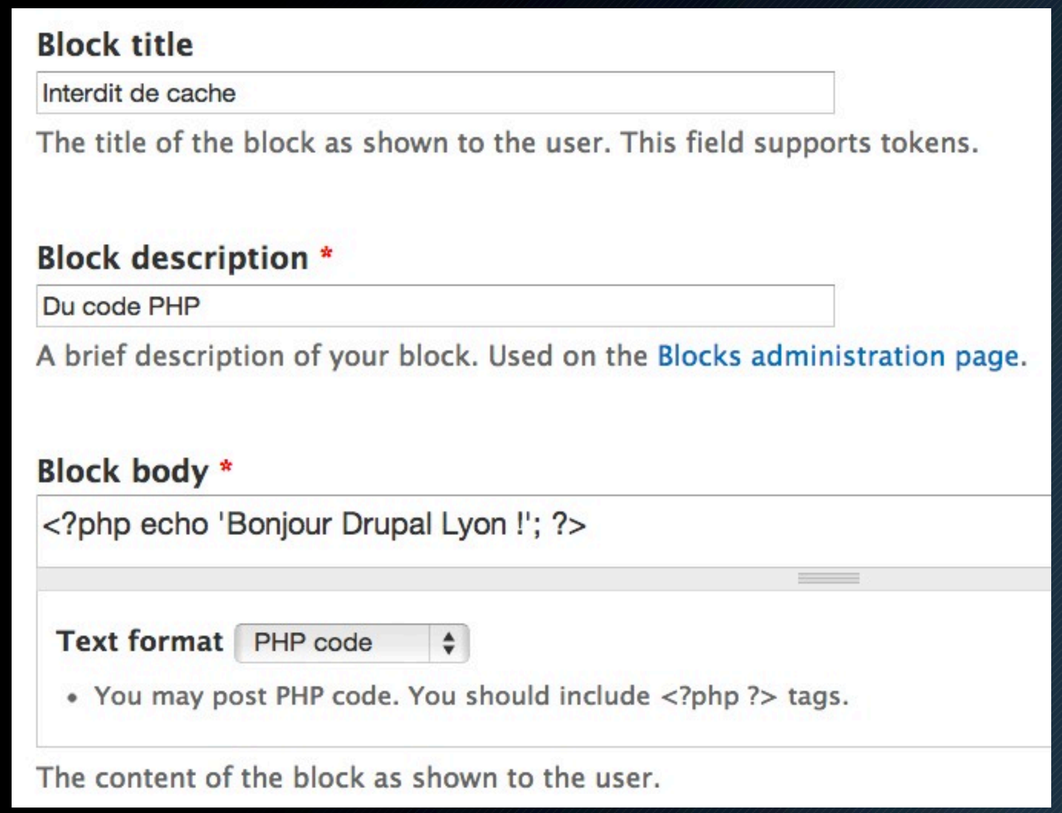
```
drupal_add_http_header('NO_CACHE=0');
```

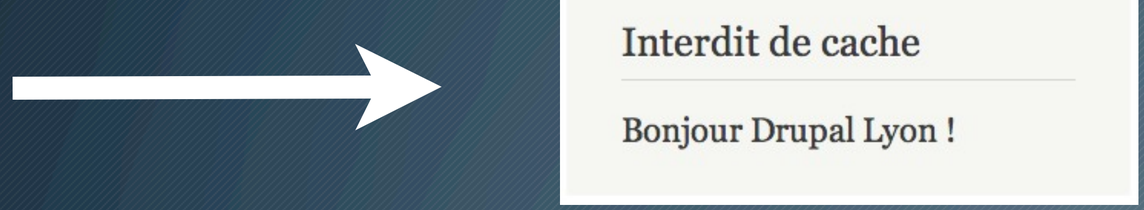
Autrement on peut également utiliser une variable **$GLOBALS** :

```
$GLOBALS['conf']['cache'] = CACHE_DISABLE;
```

Attention au format d’entrée PHP

- PHP étant dynamique, Drupal ne mettra jamais en cache les éléments qui en contiennent
- Attention donc au code PHP que vous pouvez ajouter aux nodes, blocs, views, rules...





Acquia™

Attention à variable_set()

- Vu que variable_set() est très pratique pour stocker des paramètres de Drupal, on a tendance à en abuser dans les modules...et parfois dans template.php !
- N'oubliez pas que cela a pour effet de vider le cache des variables dans **cache_bootstrap** et peut causer de réels problèmes de performance.

```php
<?php
function variable_set($name, $value) {
  global $conf;
  db_merge('variable')->key(array('name' => $name))->fields(array('value'
=> serialize($value)))->execute();
  cache_clear_all('variables', 'cache_bootstrap');
  $conf[$name] = $value;
}
```

Acquia™

Modules et techniques pour aventuriers

Acquia™

Entity Cache

- Permet de transférer les entités du core vers l'API de cache de Drupal.
- Nécessite du code pour supporter les entités custom/contrib

```
mysql> SHOW TABLES LIKE 'cache_entity%';

+----------------------------------+
| Tables_in_d7 (cache_entity%)     |
+----------------------------------+
| cache_entity_comment             |
| cache_entity_file                |
| cache_entity_node                |
| cache_entity_taxonomy_term       |
| cache_entity_taxonomy_vocabulary |
| cache_entity_user                |
+----------------------------------+

6 rows in set (0.00 sec)
```

- http://drupal.org/project/entitycache
- http://drupal.org/project/entitycache_flush

Acquia™

Cache de Views

- Views content cache : implémente un plugin de cache intelligent pour Views qui permet de mettre un affichage de vue en cache jusqu'à ce que le contenu change.

- Views argument cache : plugin de cache de vues spécifiquement conçu pour les affichages qui utilisent des arguments. Correspond donc à un cas d'usage très particulier, plutôt que de purger le cache pour toute une vue avec plusieurs affichages.

Par défaut....

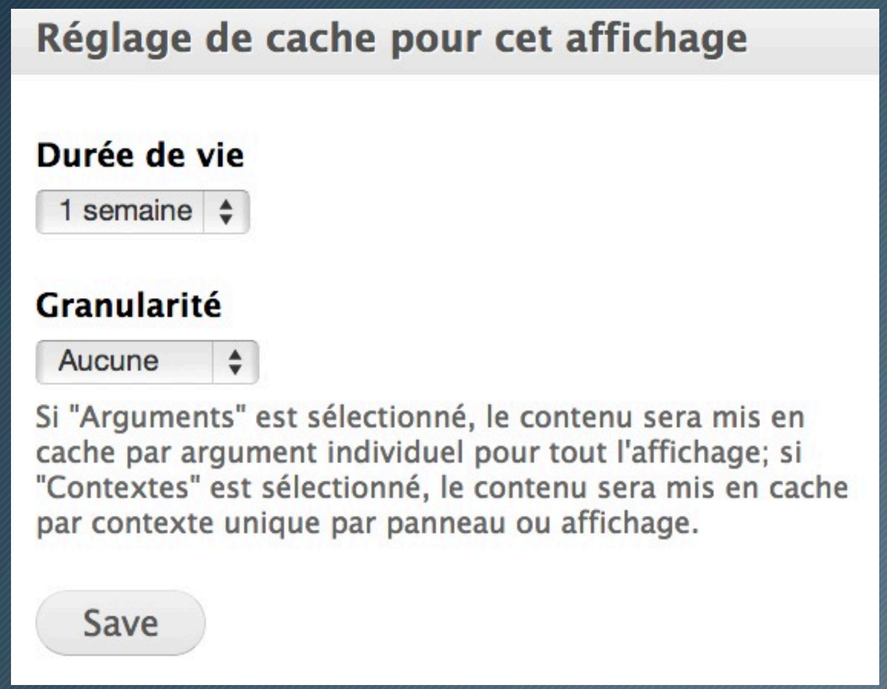
Acquia™

Cache de Panels

- Panels Content Cache : permet de mettre en cache des Panels et panneaux d'affichage Ctools et de les mettre à jour automatiquement lorsque le contenu des Panels change.

- Panels Hash Cache : met en cache les Panels et affichages Ctools à partir d'un hash, ce qui fait que le cache expire automatiquement quand un élément change (node , user, terme de taxonomie...)

Par défaut....



Acquia™

D’autres modules intéressants

- Cache Actions : permet de vider le cache de Drupal, CSS/JS, Views et Panels spécifiques via des Rules.
- Cache Audit : fournit une interface de commande drush pour rapidement passer en revue les réglages de cache d’un site pour Drupal core, Views et Panels.
- Cache Warmer : fournit une interface de commande drush qui visite une liste données d’URIs d’un site Drupal basé sur la fraîcheur du contenu. Utilise la technique dite du “microcaching”.
- Boost : choix idéal pour un hébergement mutualisé. Stocke des versions HTML statiques des pages sur disque. Attention néanmoins aux disques “cloud” (GFS, EBS...) où les performances d’écriture disque baissent rapidement.

Checklist des caches

- Cache de page, cache de blocs, aggrégation CSS et JavaScript, cache des modules (Views, Panels, Date...)
- Pour que Pressflow fonctionne (bien) avec un reverse proxy cache préférez le **cache externe**
- Jusqu'à Drupal 7.4, pour faire fonctionner Varnish vous devez ajouter la ligne suivante dans **settings.php** : `$conf['page_cache_invoke_hooks'] = FALSE;`
- Monitorez vos HITS Varnish avec **Firebug**, **Webkit Inspector** ou **cURL**
- Monitorez les **get_hits** et **get_misses** memcache avec la commande :

```
$ watch "(echo stats ; echo quit ) | nc SERVER_ID 11211"
```

NE PURGEZ PAS LES CACHES (Drupal, Varnish) AUX HEURES DE POINTE !

Acquia™

Merci. Questions ?

Acquia™