



Optimiser les performances d'un site web

Nicolas Chevallier
Camille Roux

Intellicore Tech Talks

- Des conférences pour partager son savoir
- Le mardi au CICA Sophia Antipolis
- <http://techtalks.intellicore.net>
- Aujourd'hui la 14ème conférence
- Vous voulez participer ?
 - techtalks@intellicore.net



Optimiser les performances d'un site web

Nicolas Chevallier
Camille Roux

Sommaire

- Optimisation de chaque page
- Optimisation de la navigation
- Optimisations complémentaires

Introduction > Pourquoi optimiser un site?

Bande passante

Charge serveur

L'expérience utilisateur

Très facile à mettre en place

Introduction > Sur quels paramètres agir?

Site	% Chargement du HTML
Skyrock	4,7
Orange	3,8
Free	6,2
Dailymotion	5,3

Optimisation de chaque
page

	CSS	Scripts	Images	Poids total	Requêtes
Skyrock	2	7	90	431,1ko	116
Orange	1	11	54	230,3ko	36
Free	2	6	54	226,9ko	64
Dailymotion	1	3	41	256,8ko	47

Diminuer le nombre de
requêtes

Diminuer le nb de requêtes > Regrouper scripts et CSS

Tout regrouper

"Compiler" les composants de chaque page

Regrouper intelligemment les scripts

Diminuer le nb de requêtes > Réduire le nb d'images

Images réactives (image map)

Images intégrées

Sprites CSS

Diminuer le nb de requêtes > Images réactives

- Cas d'utilisation : plusieurs liens sur une même image
- ```
<MAP NAME="DEMO">
 <AREA SHAPE="RECT" HREF="partie1.php" COORDS="54,81,83,107">
 <AREA SHAPE="CIRCLE" HREF="partie2.php" COORDS="70,126,15">
 <AREA SHAPE="DEFAULT" HREF="default.php">
</MAP>

```

Diminuer le nb de requêtes > Images intégrées

```

```

**A éviter (incompatible avec IE)**

Réduire la taille des  
composants



## Réduire la taille des composants > Introduction

- Supprimer les commentaires
- Réduire les URL
- Compresser les composants (gzip)

## Réduire la taille des composants > Compresser (1/2)

- A partir de HTTP/1.1
  - => Accept-Encoding : gzip, deflate
  - <= Content-Encoding : gzip

## Réduire la taille des composants > Compresser (2/2)

### Consomme du temps CPU

#### Compresser le texte

HTML / XML /JSON

CSS

Scripts

#### Ne pas recompresser

Images

PDF

Vidéos

Compresser si > 1 - 2ko

## Réduire la taille des composantes > Compactage

- Pour CSS et JS
- Suppression des caractères inutiles:
  - Espaces superflus
  - Commentaires

## Réduire la taille des composantes > Obfuscation

- Compactage + renommage des noms de variables et de fonctions
- Risques :
  - Bugs
  - Maintenance
  - Debug

## Réduire la taille des composants > Résumé

JS non compressé	JS compacté	JS obfusqué	JS gzip	JS compacté + gzip	JS obfusqué + gzip
20,2 ko	13,7 ko	13,1 ko	5,8 ko	3,8 ko	3,7 ko
<b>100%</b>	<b>67%</b>	<b>65%</b>	<b>29%</b>	<b>19%</b>	<b>18%</b>

Placer les feuilles de style  
en début de page

## Feuilles de style > Cas normal

- Affichage progressif de la page
- Rafraîchissement du contenu à chaque chargement de feuille de style
  - Phénomène de scintillement de la page



## Feuilles de style > Page Blanche

- CSS en bas de page
- IE
  - Nouvelle fenêtre
  - Actualisation
  - Page d'accueil

## Feuilles de style > <link> vs @import

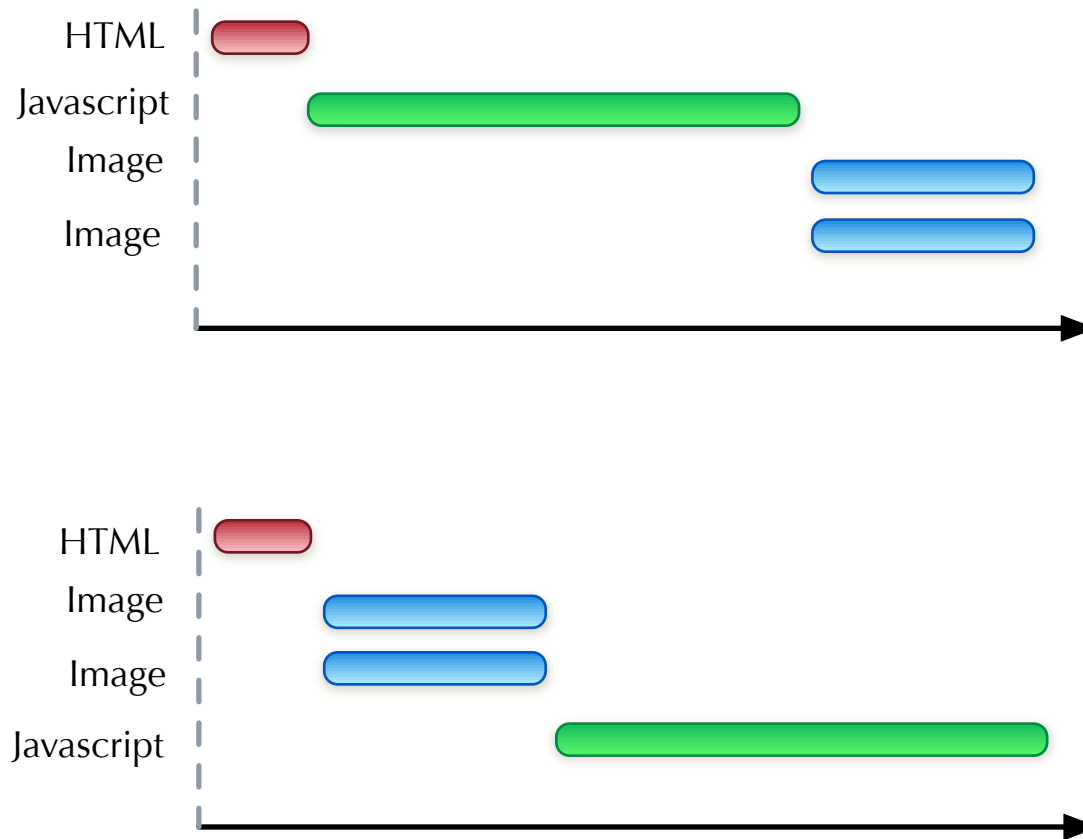
- <link rel="stylesheet" href="/styles/css.css" type="text/css" media="screen" />
- <style type="text/css">  
    @import url(/styles/css.css);  
</style>

## Feuilles de style > @import

- Ordre de chargement non contrôlé
  - Page blanche (même si en haut de la page)
  - Moins rapide que <link>
  - Pas compatible avec les vieux navigateurs
- ➔ Utiliser <link>

Déplacer les scripts en bas  
de page

## Déplacer les scripts > Comparaison



## Déplacer les scripts > Pourquoi ce phénomène?

- Pendant le chargement d'un script, le téléchargement en parallèle est désactivé
- Un script peut modifier la page (`document.write`)
- Garantie de l'exécution dans le bon ordre des scripts

# Rappel

- ✓ Diminuer le nombre de requêtes
- ✓ Compresser le contenu
- ✓ Placer les feuilles de style en haut de la page
- ✓ Placer les scripts en bas de la page

# Optimisation de la navigation



## Optimisations de la navigation > Pas de mise en cache

### Requêtes inutiles

### Renvoi systématique du composant

Empty Cache	Primed Cache
4.4K 1 HTML/Text    2.7K 1 XMLHttpRequest    33.8K 3 JavaScript Files    2.6K 1 Stylesheet File    0.3K 3 CSS Images    4.7K 3 Images	4.4K 1 HTML/Text    2.7K 1 XMLHttpRequest    33.8K 3 JavaScript Files    2.6K 1 Stylesheet File    0.3K 3 CSS Images    4.7K 3 Images
<b>48.6K Total size</b>	<b>48.6K Total size</b>
<b>12 HTTP requests</b>	<b>12 HTTP requests</b>

## Optimisations de la navigation > GET conditionnel

**Requêtes HTTP 304 si fichier non modifié**

**Requêtes inutiles**

Empty Cache	Primed Cache
4.4K 1 HTML/Text    2.7K 1 XMLHttpRequest    33.8K 3 JavaScript Files    2.6K 1 Stylesheet File    0.3K 3 CSS Images    4.7K 3 Images	4.4K 1 HTML/Text    2.7K 1 XMLHttpRequest    0.0K 2 JavaScript Files    0.0K 1 Stylesheet File    0.0K 3 CSS Images    0.0K 3 Images
<b>48.6K Total size</b>	<b>7.1K Total size</b>
<b>12 HTTP requests</b>	<b>11 HTTP requests</b>

## Optimisations de la navigation > Entêtes d'expiration

**Plus de requêtes**

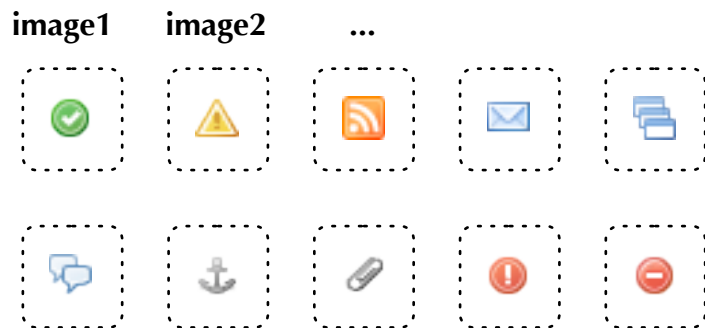
**Obliger de renommer le composant à chaque modif**

`script_20080605.js` plutôt que `script.js?20080605`

Empty Cache	Primed Cache
4.4K 1 HTML/Text    2.7K 1 XMLHttpRequest    33.8K 3 JavaScript Files    2.6K 1 Stylesheet File    0.3K 3 CSS Images    4.7K 3 Images         <b>48.6K Total size</b> <b>12 HTTP requests</b>	4.4K 1 HTML/Text    2.7K 1 XMLHttpRequest    0.0K 1 JavaScript File         <b>7.1K Total size</b> <b>3 HTTP requests</b>

# Optimisations complémentaires

## Optimisations complémentaires > Sprite CSS



**10 images 6,1ko**

image



CSS

```
.sprite-anchor { background-position: 0 -5px; }
.sprite-accept { background-position: 0 -26px; }
.sprite-attach { background-position: 0 -47px; }
.sprite-error { background-position: 0 -68px; }
.sprite-exclamation { background-position: 0 -89px; }
.sprite-feed { background-position: 0 -110px; }
.sprite-email { background-position: 0 -131px; }
.sprite-comments { background-position: 0 -152px; }
.sprite-application_cascade { background-position: 0 -173px; }
.sprite-delete { background-position: 0 -194px; }
#container li { background: url(csg-4856bd30bc355.png) no-repeat top left; }
```

**css + 1 image 4,5 ko**

## Optimisations complémentaires > Sprite CSS



# Optimisations complémentaires > Pré chargement



Accès à la page 1

**Chargement**

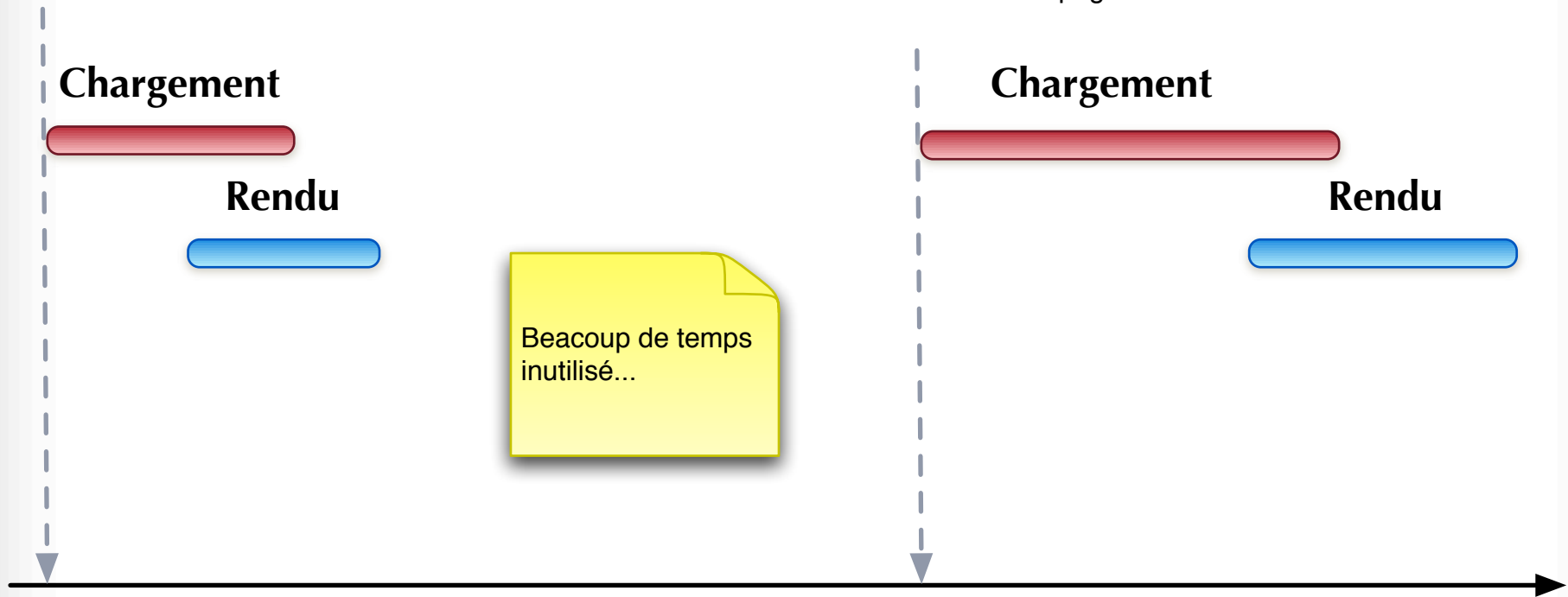
**Rendu**

Beaucoup de temps  
inutilisé...

Accès à la page 2

**Chargement**

**Rendu**



# Optimisations complémentaires > Pré chargement

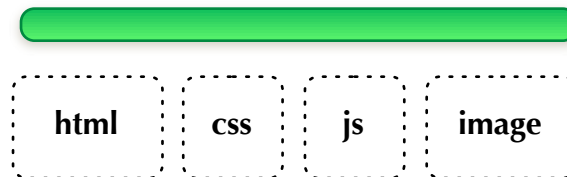


Accès à la page 1

**Chargement**

**Rendu**

**Pré chargement**



Accès à la page 2

**Chargement**

**Rendu**

Préchargement de 3 images avec MooTools :

```
new Asset.images(['image1','image2','image3'], {});
```



## Optimisations complémentaires > AJAX

### **XML vs JSON**

JSON plus léger

JSON plus rapide à parser/évaluer en javascript

### **GET vs POST**

GET plus rapide et logique dans la plupart des cas

### **Mettre en cache**

5 minutes? 10 minutes? 1 heure?

# En savoir plus

## Outils

★ **YSlow** <http://open.yahoo.com/yslow/>

**AOL PageTest** <http://pagetest.wiki.sourceforge.net/>

**Générateur de Sprite CSS Online** <http://spritegen.website-performance.org/>

**Générateur de Sprite CSS Java et PHP** <http://smartsprites.osinski.name/>

## Sites web

**Yahoo! Exceptionnal Performance** <http://developer.yahoo.com/performance/>

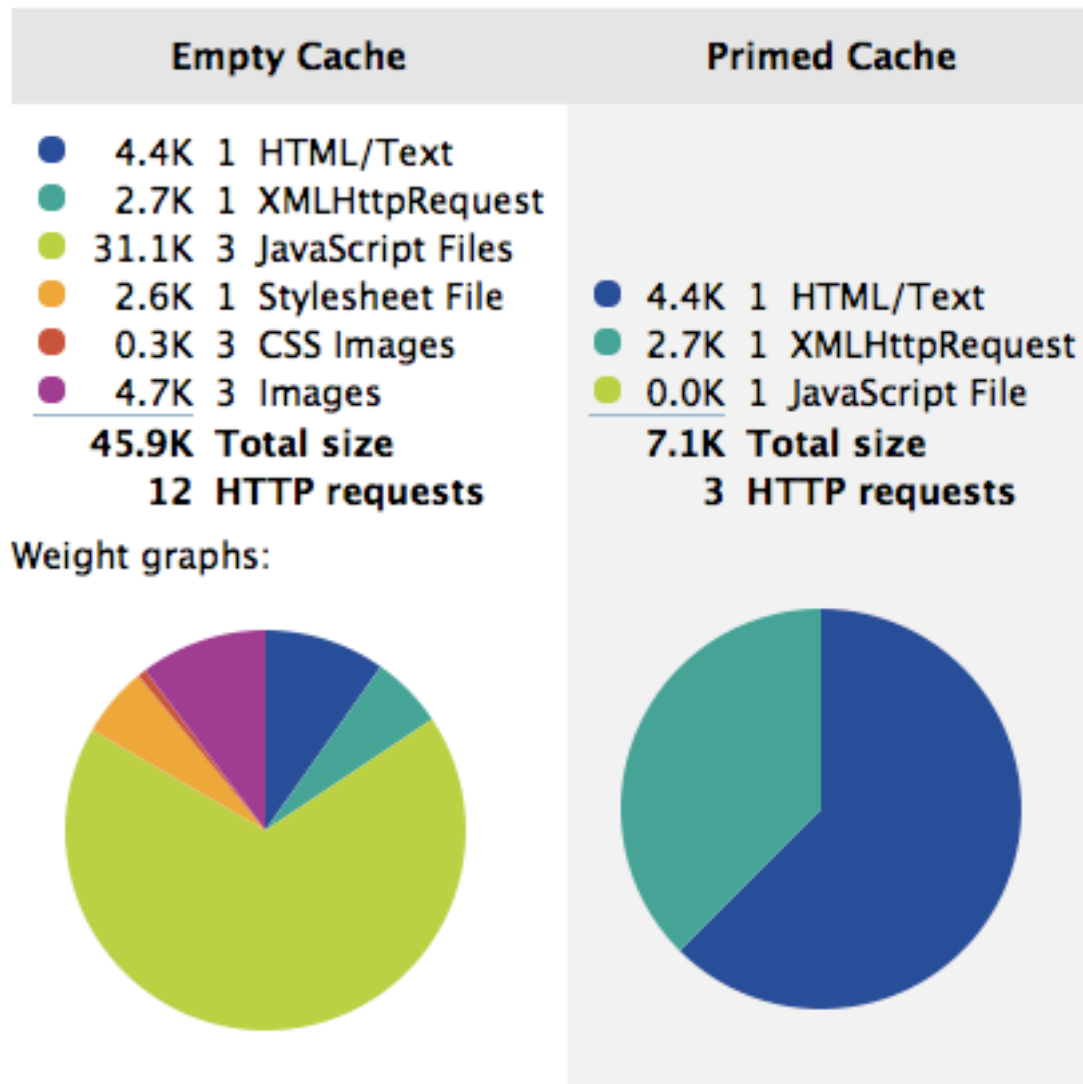
★ **Performance Web** <http://performance.survol.fr/>

# YSlow > Audit des performances

## Performance Grade: B (87)

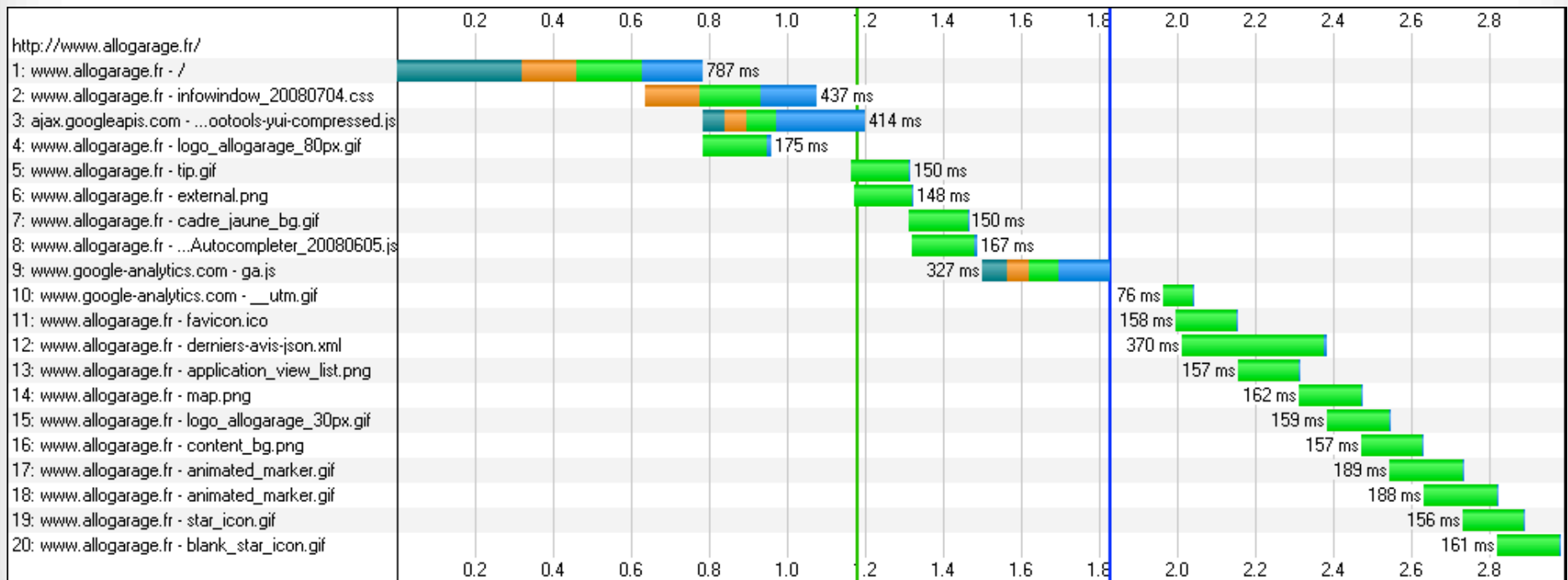
A	1. Make fewer HTTP requests
F	2. Use a CDN ▸
B	3. Add an Expires header ▸
A	4. Gzip components
A	5. Put CSS at the top
A	6. Put JS at the bottom
A	7. Avoid CSS expressions
n/a	8. Make JS and CSS external ▸
A	9. Reduce DNS lookups
A	10. Minify JS
A	11. Avoid redirects
A	12. Remove duplicate scripts
A	13. Configure ETags

## YSlow > Mise en cache



# YSlow & AOL PageTest > Chargement d'une page

► <b>www.allogarage.fr</b>	allogarage.fr	5 KB	45ms
► <b>infowindow_200</b>	allogarage.fr	3 KB	175ms
► <b>mootools-yui-cc</b>	ajax.googleapis.com	20 KB	483ms
► <b>CityAutocomplet</b>	allogarage.fr	3 KB	138ms
► <b>ga.js</b>	google-analytics.com	10 KB	334ms
► <b>derniers-avis-js</b>	allogarage.fr	3 KB	314ms
<b>6 requests</b>		<b>41 KB</b>	<b>1.63s</b>



# Conclusion

## **Optimiser!**

**1 ou 2 jours nécessaires**

**Très simple à mettre en place**

**Expérience utilisateur améliorée**

# Questions?

## Blog

**Intellicore Tech Talks** <http://techtalks.intellicore.net/>

**Blog de Camille Roux** <http://www.camilleroux.com/>

## Email

**Camille Roux** [contact@camilleroux.com](mailto:contact@camilleroux.com)

**Nicolas Chevallier** [nicolas.chevallier@b2b-net.com](mailto:nicolas.chevallier@b2b-net.com)

WWW.INTELLICORE.NET

Tous droits réservés Intellicore Services © 2008