



# HHVM / TYPO3

## Compatibilité & Performance

## ➤ Typo3, HHVM et Nxtweb en quelques mots

➤ **Typo3** est un CMS utilisé par environ 500 000 sites web à travers le monde. Les promoteurs et utilisateurs de **Typo3** mettent en avant sa flexibilité, son excellente internationalisation, sa maturité et ses performances. **Typo3** est particulièrement adapté à la réalisation de sites d'entreprise ou de grands sites institutionnels.

➤ **HHVM** est un interpréteur PHP développé par les équipes de Facebook. Sa raison d'être est la diminution du temps de calcul consacré à l'exécution du code PHP. C'est un projet open-source, disponible sur github depuis 2010.

Il a été conçu initialement pour exécuter le code du site Facebook et présentait à l'origine de nombreuses lacunes dans le traitement des CMS les plus courants.

Peu à peu, pour un certain nombre de plateformes, **HHVM** s'est rapproché d'une iso-fonctionnalité totale avec son prédécesseur, l'interpréteur **Zend**.

Côté performance, les résultats sont au rendez-vous. Il est aujourd'hui admis qu'**HHVM** est l'interpréteur PHP le plus efficace du marché.

➤ **Nxtweb** est une startup dont le cœur de métier est d'optimiser les performances des sites internet.

Les gains en efficacité que nous obtenons bénéficient à tous les acteurs de la toile : propriétaires, développeurs, hébergeurs et utilisateurs.

Par ailleurs, en diminuant drastiquement les besoins en ressources machine, notre action améliore nettement le bilan environnemental du web.

**Nxtweb** a fait d'**HHVM** son principal outil d'optimisation. Nous corrigeons ses bugs résiduels et développons des extensions accélératrices dédiées.

Autrement dit, nous sommes à même de mettre en place les sites Wordpress, Prestashop, Magento, etc... voire **Typo3** ?, les plus rapides au monde.

## ➤ HHVM peut-il améliorer les performances des sites Typo3 ?

Cette question n'est pas nouvelle. Dès avril 2014, au vu des résultats très positifs obtenus sur les plateformes concurrentes, les mainteneurs du CMS ont lancé plusieurs actions dans le but d'aboutir à la compatibilité **HHVM** de **Typo3**.

Les dernières informations disponibles sur le sujet semblent indiquer que ce projet n'a pas été mené à bien.

Pour nous forger notre propre opinion sur la question, nous avons décidé de reprendre le problème à zéro :

➤ en passant les tests de non-régression **PHPUnit**

➤ en mesurant la plus-value obtenue sur le frontend du site "**The Official Introduction Package**".

Nous nous servons ici de **Typo3** pour éclairer la nature de notre travail d'optimisation et de maintenance fonctionnelle.

Un travail sans doute un peu complexe à présenter autrement que par une étude de cas.

Nous intervenons en effet à la croisée de l'outil **HHVM**, des CMS PHP et de leur utilisation dans la construction de sites web particuliers.

## ➤ Evaluer la compatibilité : Les tests unitaires

Les promoteurs de **Typo3** ont une démarche qualité très appréciable.

Ils ont notamment créé une suite de tests de non-régression afin de valider le bon fonctionnement des évolutions successives.

En 2014, c'est très naturellement cette suite de tests qui a été utilisée pour établir un premier aperçu de la compatibilité **HHVM** de **Typo3**.

On constate alors que 95,5% des 6458 tests de non-régression se terminent correctement.

<http://blog.macopedia.co/post/running-typo3-cms-on-hhvm-part2/#.VemkBmztkmo>

La dernière log du testeur résume la situation d'alors :

```
FAILURES!  
Tests: 6458, Assertions: 10675, Failures: 71, Errors: 45, Incomplete: 17, Skipped: 153.
```

C'est à dire que 95.57% des tests se sont bien déroulés, 1.1% ont échoué, 0.7% ont provoqué une erreur, 0.26% ont été incomplets et 2.37% n'ont pas été exécutés car non pertinents sur l'environnement utilisé.

En août 2015, nous constatons que les dernières versions du testeur, de **Typo3** et de **HHVM** produisent cette sortie :

```
FAILURES!  
Tests: 7268, Assertions: 9379, Errors: 4, Failures: 6, Skipped: 144.
```

### › Les tests en erreurs

Ce sont des tests qui n'ont pu être menés à bien du fait d'une incompatibilité **HHVM** du code du testeur. Ces incompatibilités sont traitées au fil de l'eau par les développeurs **Typo3**.

Pour pouvoir réaliser tous les tests nous avons :

› appliqué le patch <https://github.com/sebastianbergmann/phpunit-mock-objects/commit/6e4f3076f790b391396778d11a3a5843313ac350>

› pour le test `scanAvailablePackagesKeepsExistingPackageConfiguration`, appliqué la modification suivante dans le fichier `./typo3/sysex/core/Tests/Unit/Package/PackageManagerTest.php`

```
//patch nxtweb $packageStates = require('vfs://Test/Configuration/PackageStates.php');  
    $packageStates =  
eval(substr(substr(file_get_contents('vfs://Test/Configuration/PackageStates.php'),5),0,  
-2).";");
```

### › Les tests en echec

Il s'agit de tests qui se sont déroulés entièrement mais dont le résultat ne correspond pas au résultat attendu.

```
checkDateWithInvalidDateValues with data set "Not Good" : correction par nxtweb du code  
hhvm  
checkDateWithInvalidDateValues with data set "HH:ii yyyy-mm-dd" : correction par nxtweb  
du code hhvm  
scanAvailablePackagesTraversesThePackagesDirectoryAndRegistersPackagesItFinds
```

Ces 3 cas d'erreurs correspondent à des incompatibilités réelles de **HHVM** avec le code **Typo3**.

Nous avons corrigé, en une quarantaine de lignes, les portions de code **HHVM** fautives.

## ➤ Les tests après corrections Nxtweb

Après correction de l'outil de test et de HHVM par Nxtweb, il ne reste plus que 3 tests en échec :

```
registeredExtractorClassWithSamePriorityAreReturnedInSameOrderAsTheyWereAdded
registeredFileRendererClassWithSamePriorityAreReturnedInSameOrderAsTheyWereAdded
executeValueModifierReturnsModifiedResult
```

Ces erreurs correspondent à des "faux positifs" (comportement attendu erroné) qui sont en cours de traitement par les développeurs **Typo3** (<https://forge.typo3.org/issues/65315> et <https://forge.typo3.org/issues/65317>)

## ➤ Conclusion

A ce stade de notre étude, nous disposons donc du premier binaire **HHVM 100% compatible avec les tests unitaires Typo3.**

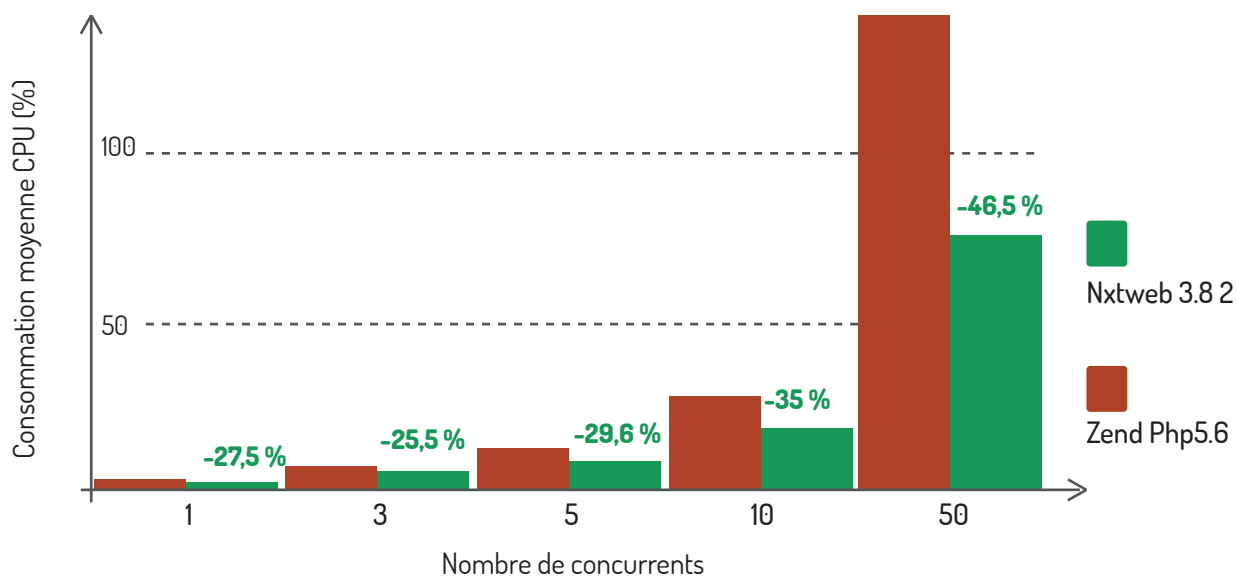
## ➤ Evaluer les performances

Notre protocole de test est le suivant :

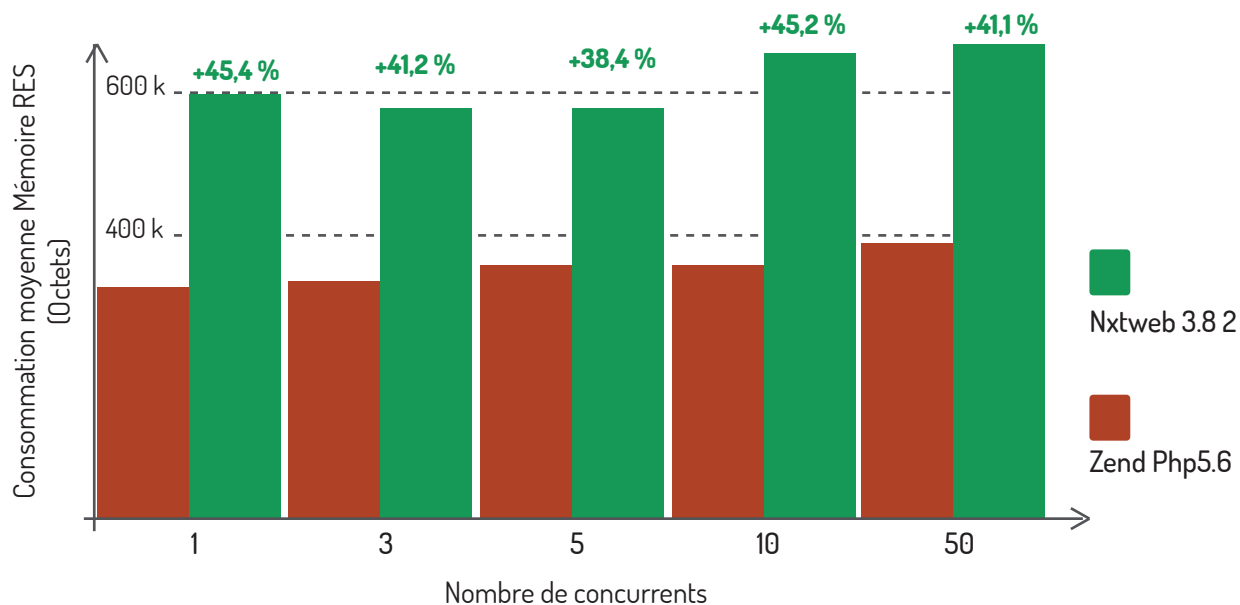
Nous interrogeons toutes les urls du front office du site "**Introduction package**" avec 1, 3, 5, 10 et 50 concurrents et nous mesurons les paramètres suivants : cpu, mémoire, temps de réponse obtenus avec php 5.6 puis avec **HHVM**.

Voici nos premiers résultats :

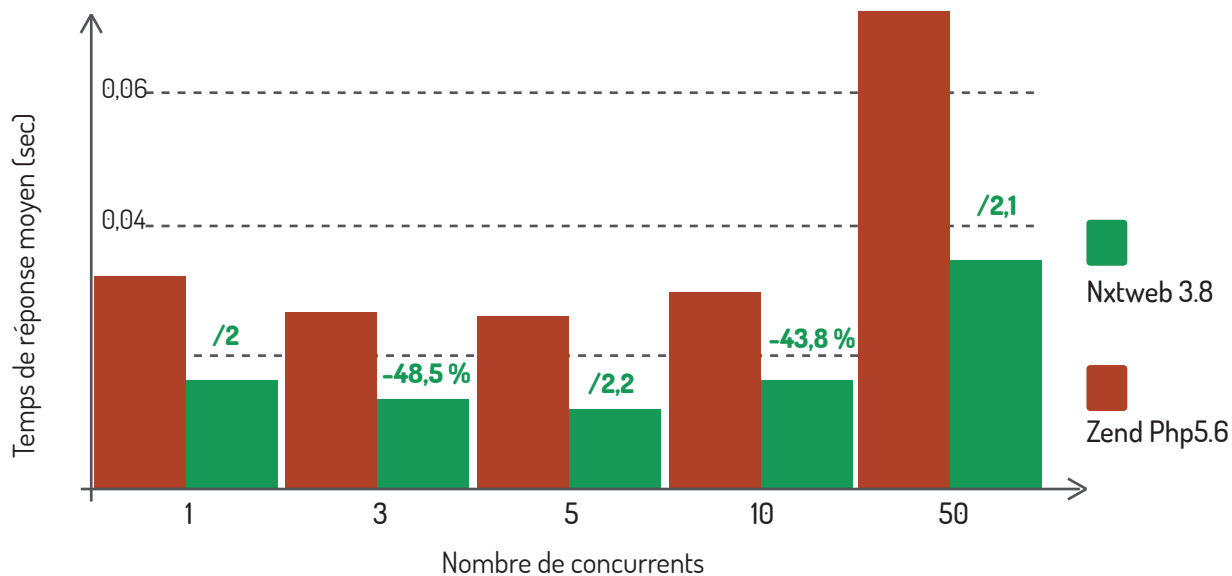
### Comparaison des consommations moyennes CPU (%)



### Comparaison des consommations moyennes Mémoire RES



### Comparaison des moyennes des temps de réponse

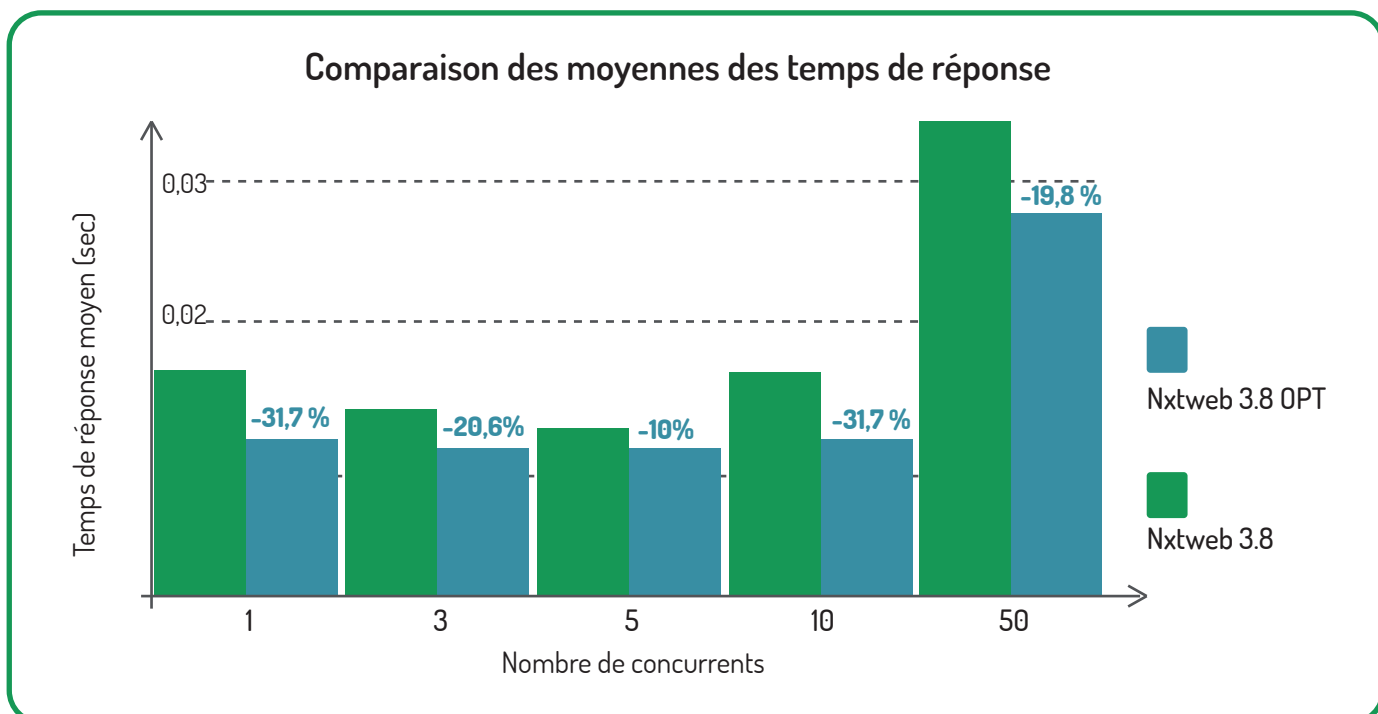
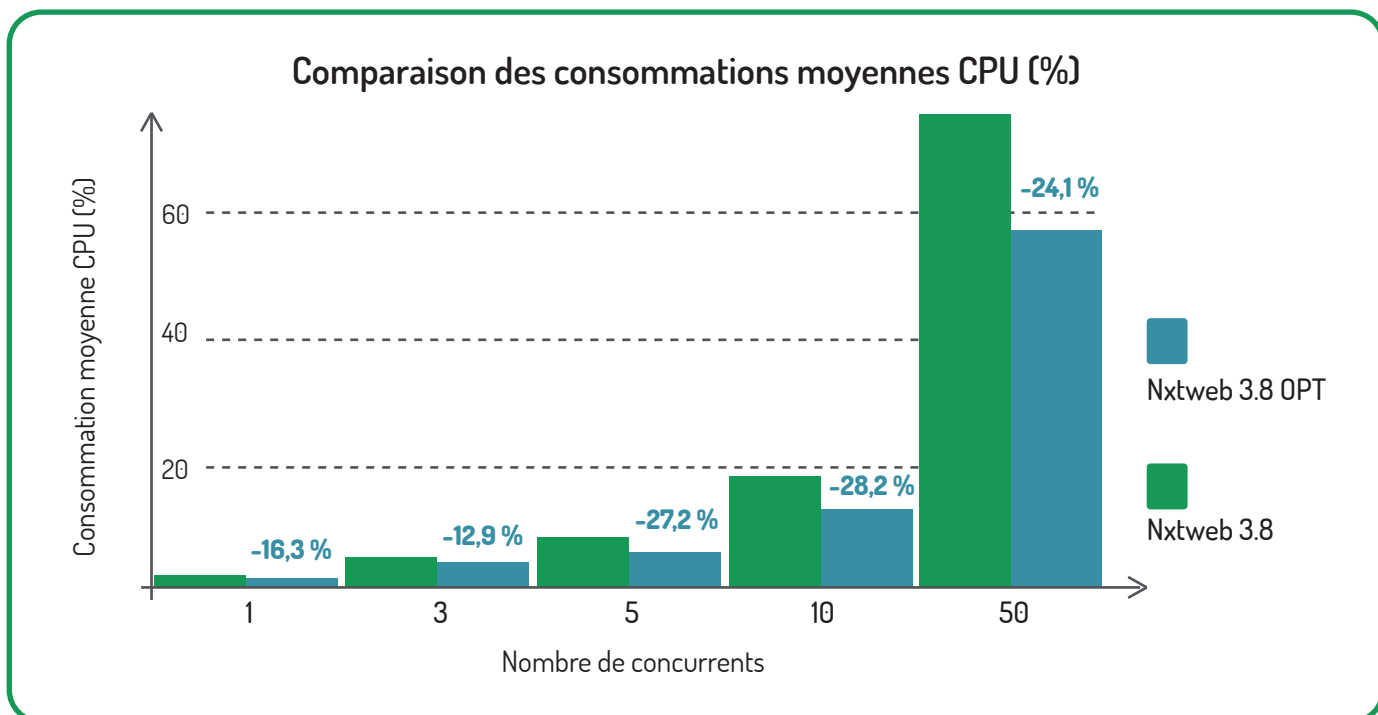


## > Aller au-delà

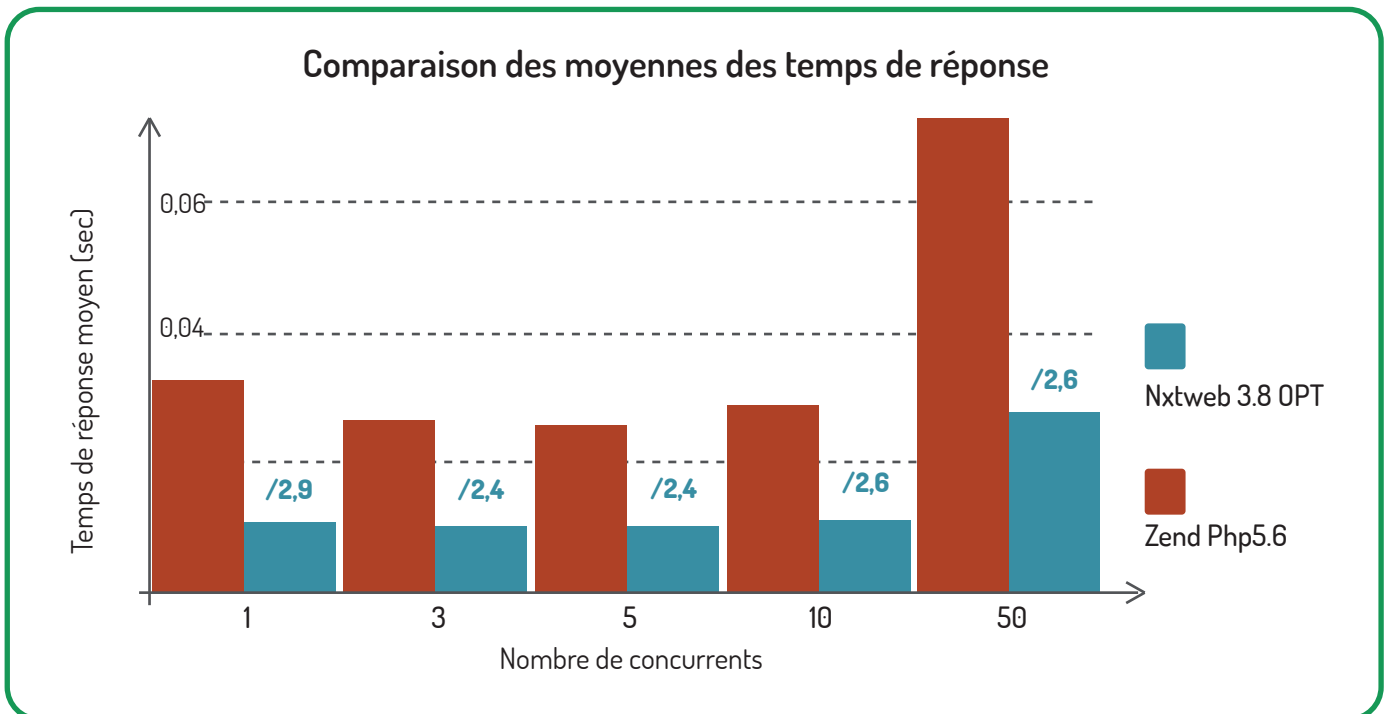
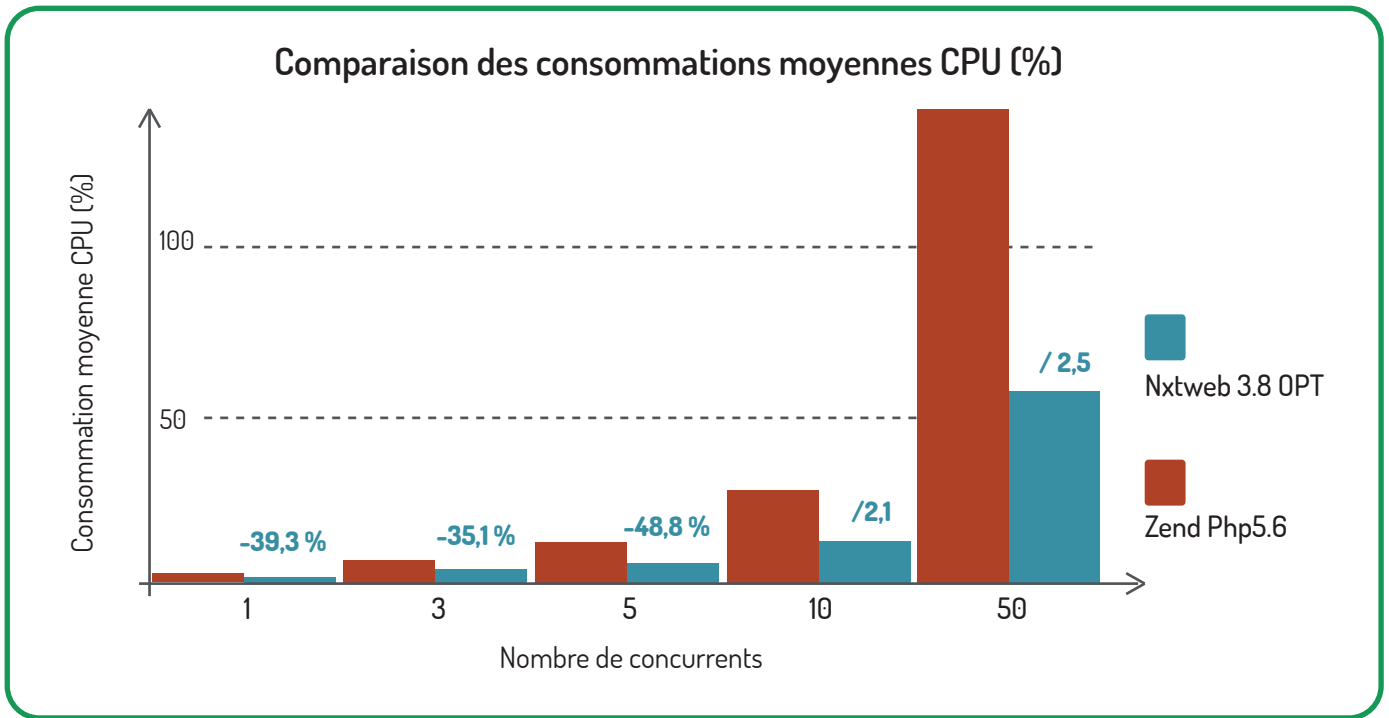
Pour **Nxtweb**, les gains obtenus grâce à **HHVM** ne sont qu'un point de départ.

Nous avons profilé les différents traitements du front **Typo3** et avons créé un accélérateur **HHVM** dédié à ce CMS.

Les gains obtenus par rapport à la version **HHVM** de base sont les suivants :



Comparé à php 5.6, nous obtenons :



#### ➤ Conclusion

A ce stade de nos tests, nous disposons, à configuration et paramétrage égaux, du site **Typo3 "Introduction package"** le plus rapide au monde ! ;-).

## ➤ Bilan

Nous avons montré

- que l'isofonctionnalité **HHVM/Zend** pour **Typo3** est accessible dès aujourd'hui.
- qu'**HHVM** divise plus ou moins par deux les consommations CPU et les temps de réponses du front **Typo3** "Introduction Package".

En ce qui concerne les performances et la consommation énergétique, faute de temps/budget, cette étude est très incomplète.

- Nous n'avons pas eu le loisir de bencher le backoffice. Il faut pourtant s'attendre sur cette partie de l'application à des résultats beaucoup plus positifs encore. En effet, plus les traitements php sont importants et plus l'écart se creuse entre **HHVM** et **Zend**. Or, les requêtes backoffice, qui ne peuvent bénéficier des mécanismes de cache qui réduisent au strict minimum les traitements front-office, sont très gourmandes en CPU.

Pour certains clients de **Nxtweb**, tous CMS confondus, la plus value d'**HHVM** réside d'ailleurs dans l'accélération du backoffice : leurs gains de productivité pendant la phase d'intégration sont impressionnants.

- L'optimisation que nous avons réalisée est très sommaire. Avec un peu de temps et l'aide d'une expertise **Typo3**, il est certainement possible d'aller beaucoup plus loin.

Toujours faute de temps, nous n'avons pas pu mesurer l'impact d'**HHVM** sur la consommation électrique du site **Typo3**. Nos précédents tests nous ont montré que la diminution de la consommation CPU moyenne donnait un ordre de grandeur de la diminution de consommation énergétique induite. On peut donc raisonnablement tabler sur une division par 2 de cette consommation.

En résumé, nos tests ont porté sur une partie seulement d'un site **Typo3** de démonstration.

Il serait imprudent d'affirmer que nos conclusions s'appliquent à tous les front/back de tous les sites **Typo3** aujourd'hui en production.

Néanmoins, nous pensons avoir établi que la migration d'un site opérationnel a de grandes chances de s'avérer payante.

Il est aujourd'hui admis que l'efficacité est un enjeu crucial pour les sites marchands. Il faut rappeler que les sites institutionnels ont également tout à gagner à améliorer leurs performances.

Optimiser la réactivité d'un site institutionnel, c'est lui donner plus de visibilité, augmenter son trafic et le nombre de pages vues par visite. C'est également, en fluidifiant son utilisation, lui offrir une identité web dynamique et positive. Enfin, diminuer son impact environnemental, c'est participer à la révolution greenIT et pouvoir revendiquer une démarche éco-responsable.