

# Performance de QGIS Server

QGIS joue dans la cour des grands !

**camptocamp**<sup>▲</sup>

INNOVATIVE SOLUTIONS  
BY OPEN SOURCE EXPERTS

# Problématique

- Comment définir le concept de performances et le mesurer
- Historique
- Framework de tests de performances
- Quelques retours



# Performance : une définition

Indications chiffrées mesurant les possibilités maximales ou optimales d'un matériel, d'un logiciel, d'un système ou d'un procédé technique pour exécuter une tâche donnée.

Source : Wikipedia



# Performance : les mesures

- Selon le contexte, les performances incluent les mesures suivantes:
  - Un **faible temps de réponse** pour effectuer une tâche donnée
  - Un débit élevé (vitesse d'exécution d'une tâche)
  - L'efficacité : faible utilisation des ressources informatiques : processeur, mémoire, stockage, réseau, consommation électrique, etc.
  - Une haute disponibilité du système ou de l'application
  - Une bande passante élevée ou une faible latence
  - Le **passage à l'échelle** (scalabilité)
  - La capacité d'un canal de communication
  - La performance du stockage informatique



# Limite de tests de performance

- Pertinent seulement pour le cas étudié, attention à l'extrapolation (« *il est performant dans ce test, ce sera le cas avec mes données et mes cas d'utilisations* »)
- Bien faire attention à ce qui est testé
- Le rendu et les données doivent être identique
- L'optimisation doit être égale entre les différents éléments comparés



# Historique

- OSGeo a organisé un WMS Benchmark
  - Complexe à installer
  - Données à récupérer en différent endroit
- Volonté d'avoir quelque chose de facile à mettre en place, reproductible !



# Objectif de notre infrastructure

- Être reproductible : basé sur une infra docker
- Les mêmes données :
  - Stocké dans une base PostGIS
  - Sans optimisation avancée
  - Seulement le minimum de l'état de l'art : colonnes indexées
- Configuration des serveurs
  - mise en place des bonnes pratiques (pool de connexion si disponible, FastCGI)



# Limite actuelle de notre infra de test

- Optimisation à améliorer
  - Dangereux de comparer les services si une optimisation n'a pas été mise en place
- Tous les cas d'usage ne sont pas testés
- On ne compare pas les fonctionnalités, obligation de limiter par le bas



# Infrastructure de test

- Serveur :
  - CPU :
    - 8 cores
    - Intel Xeon E312xx
    - 2 GHz
  - RAM :
    - 16 Go.
  - Système :
    - Linux 3.16.0-4-amd64 x86\_64
- MapServer 7.0.\*
- GeoServer 2.9
- QGIS master (3.x)



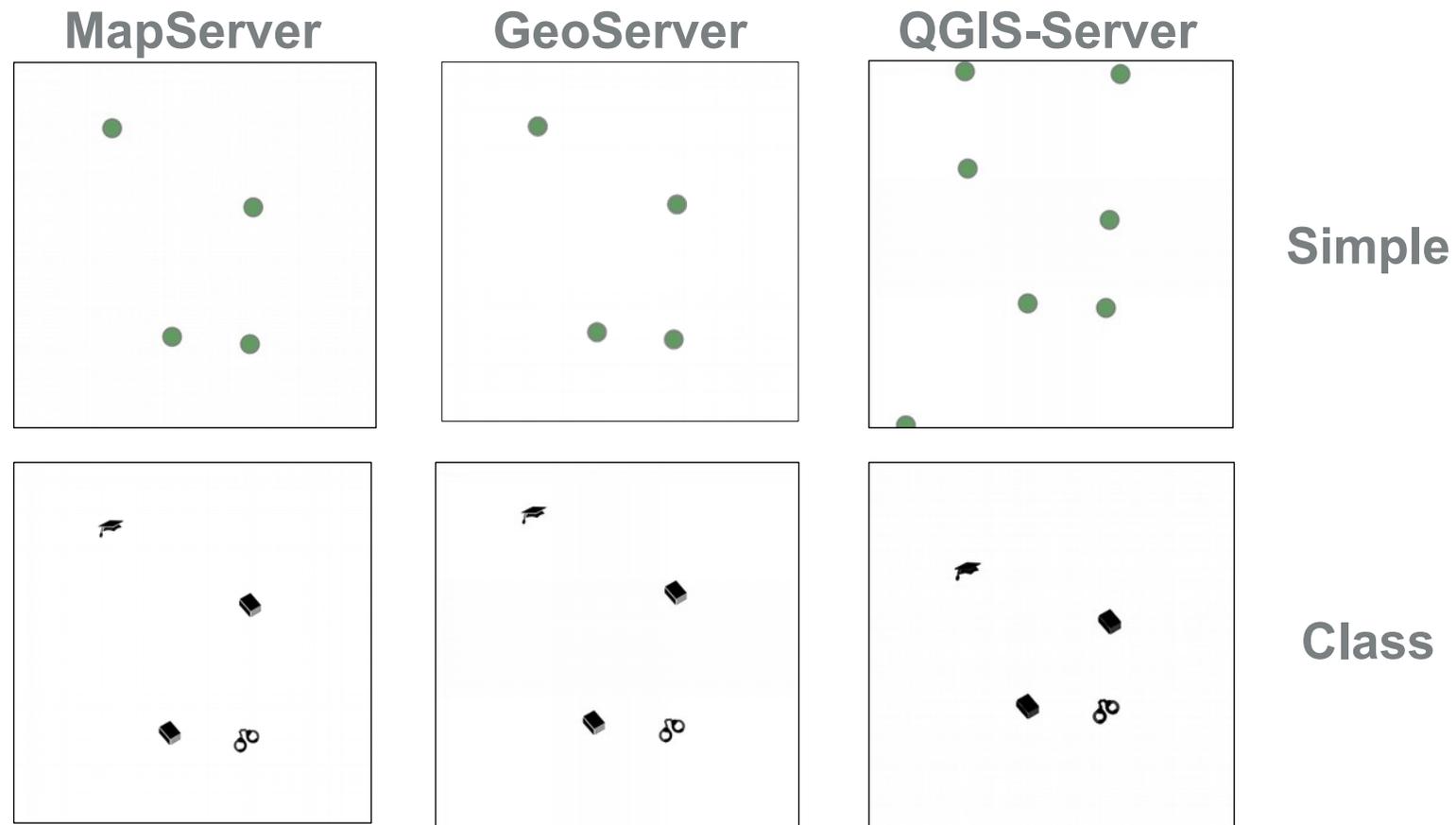
# Les couches

- Ponctuel : amenities
- Linéaire : route
- Surfacique : bâtiments



# Couches ponctuelles

## ■ Amenities



# Couches linéaires

- roads

MapServer



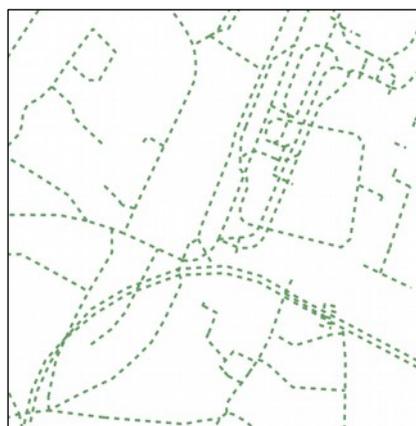
GeoServer



QGIS-Server



Simple



Hatched



# Couches surfaciques

## ■ Buildings

MapServer



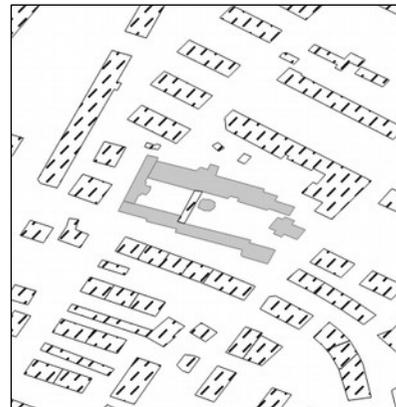
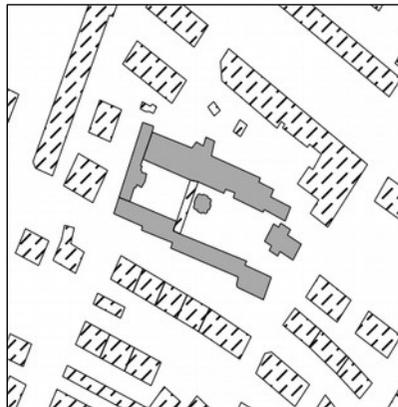
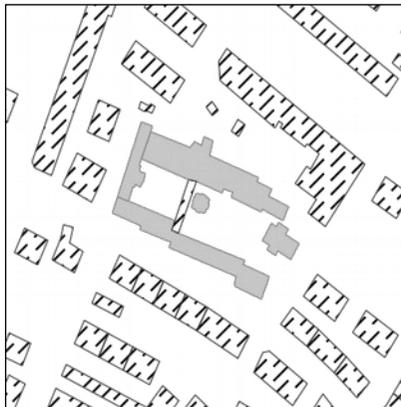
GeoServer



QGIS-Server



Simple



Hatched



# Différence visuelle

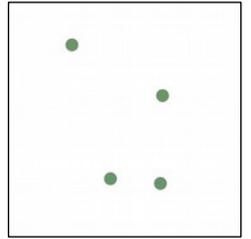
- MapServer et GeoServer très proche mais :
  - Bbox très légèrement différente
  - ~~Hachure difficile à reproduire~~
  - Teinte différente
- QGIS :
  - Problème de la projection CRS:84



# Benchmark

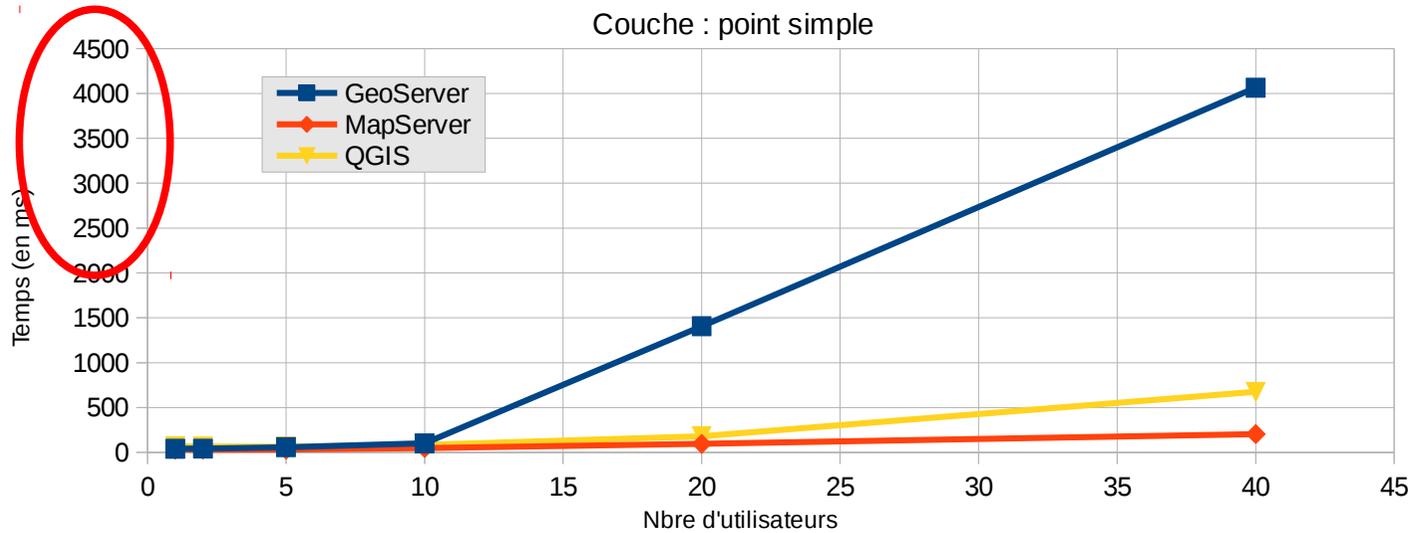


# Attention à l'optimisation !



Temps de génération d'une image

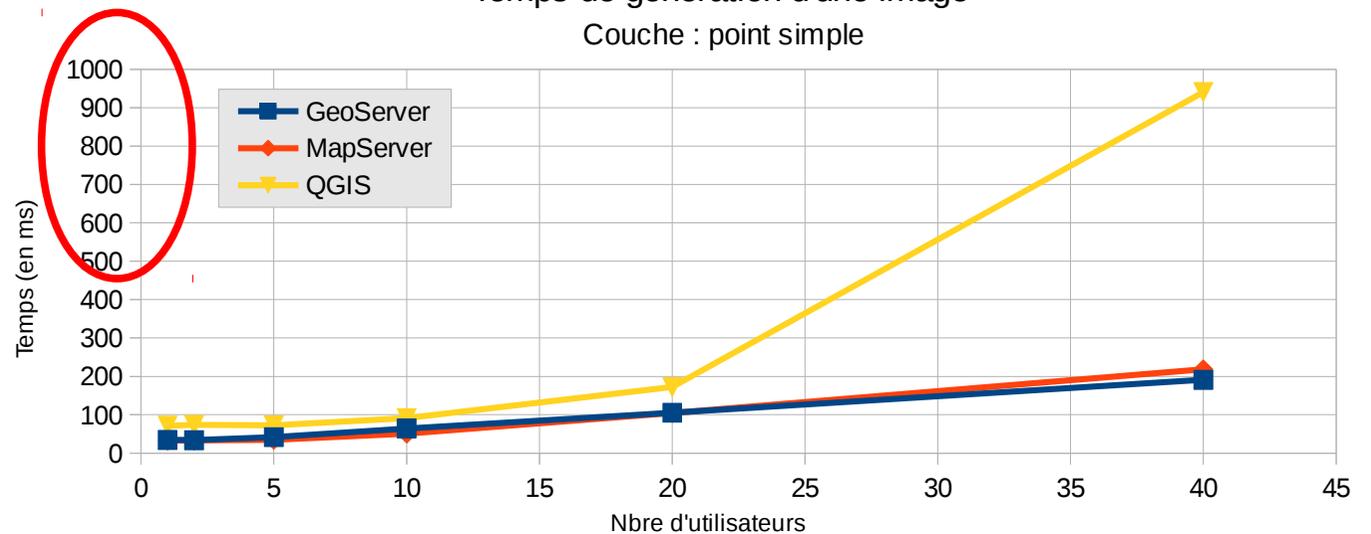
Couche : point simple



GeoServer out-of-box

Temps de génération d'une image

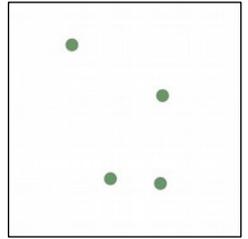
Couche : point simple



GeoServer optimisé

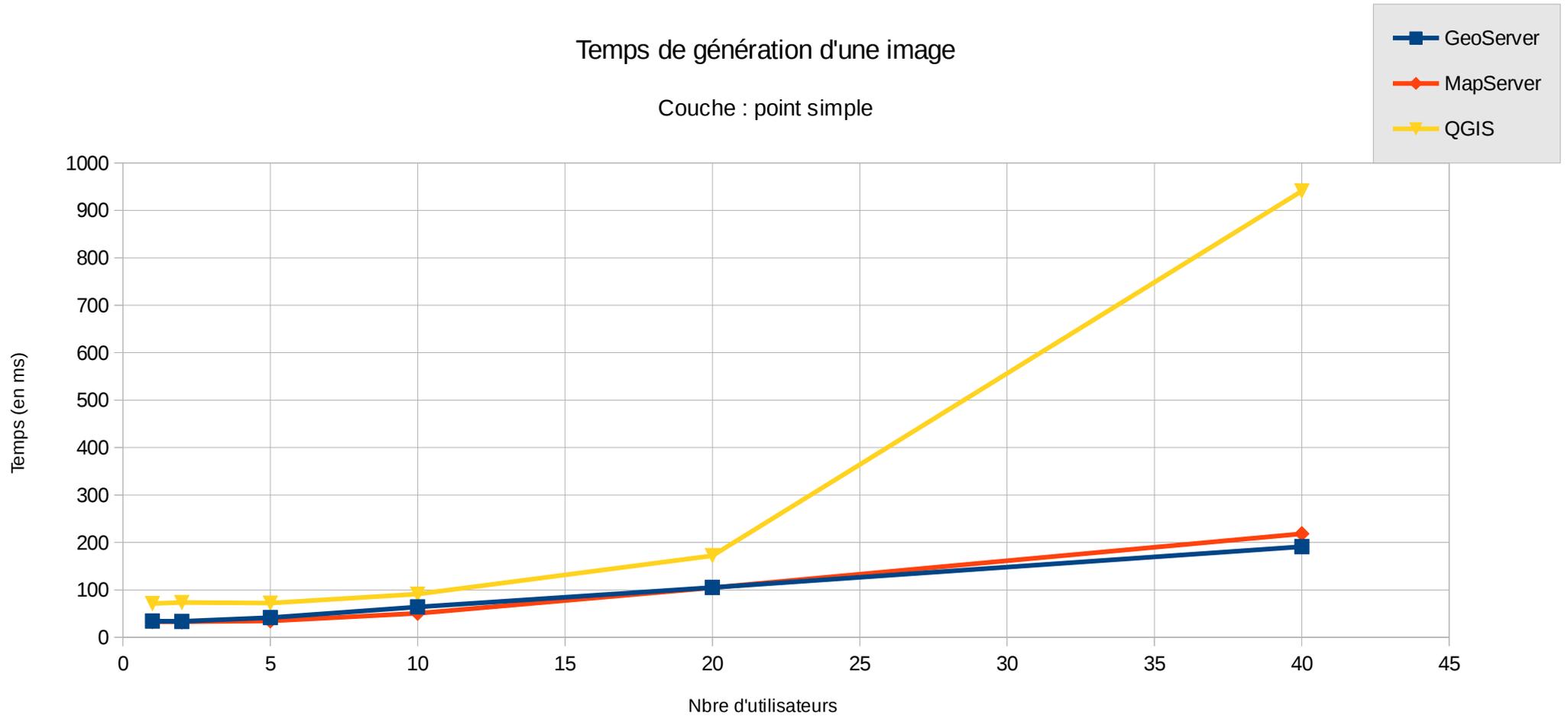


# Évolution du nombre d'utilisateurs

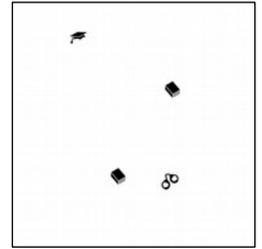


Temps de génération d'une image

Couche : point simple

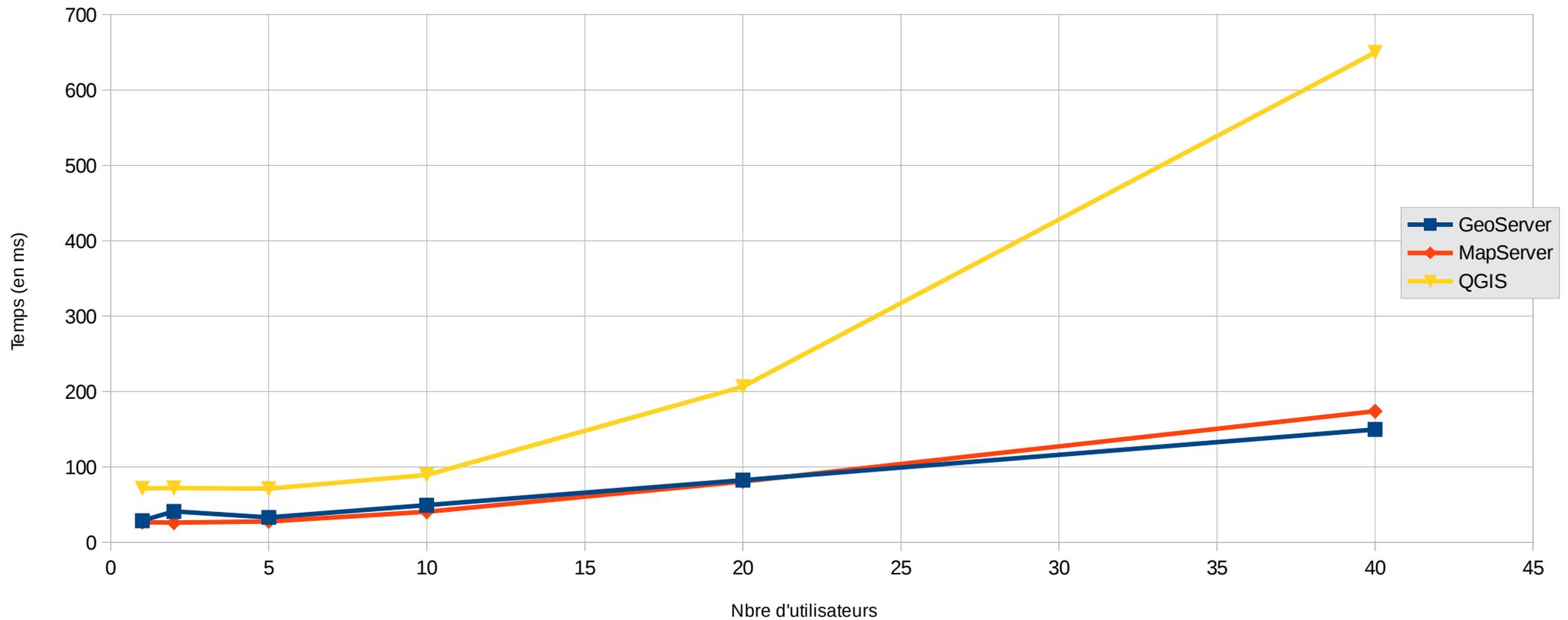


# Évolution du nombre d'utilisateurs



Tems de génération de l'image

Couche : point avec plusieurs classes

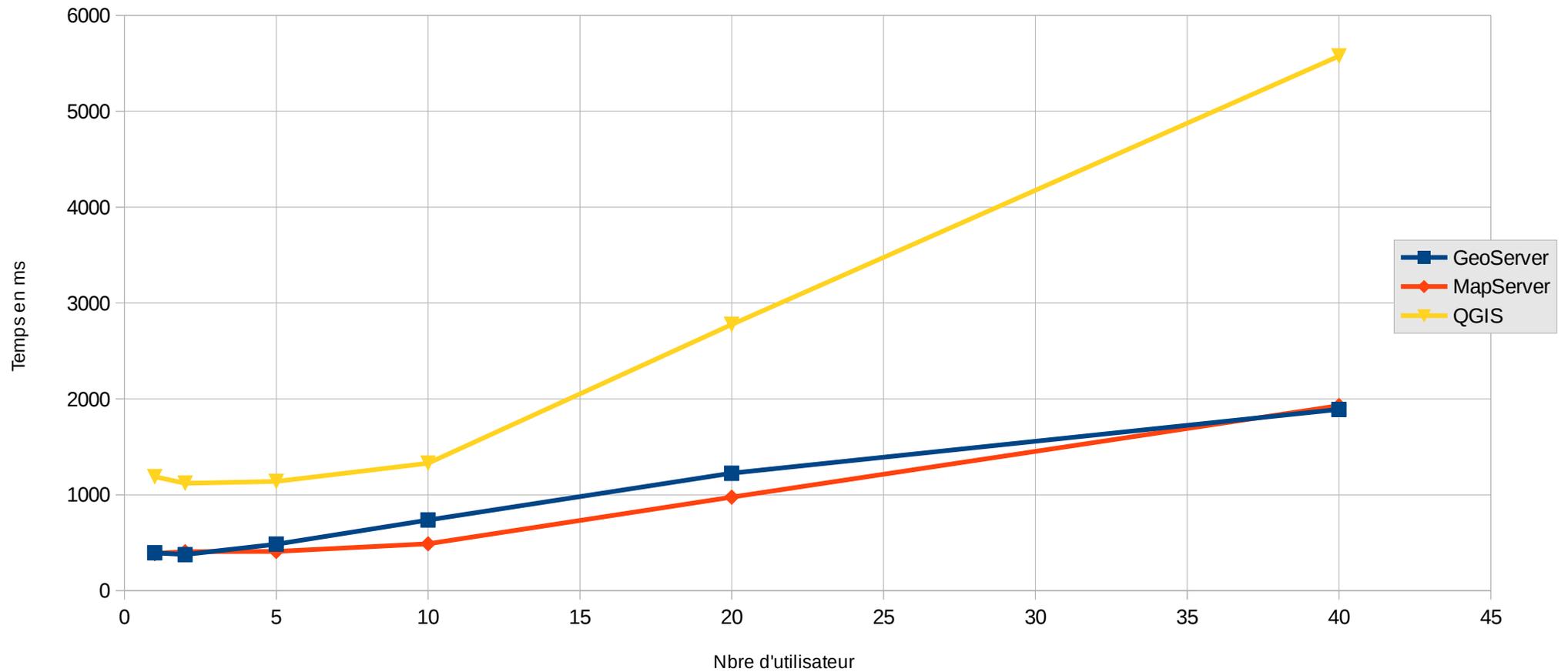


# Évolution du nombre d'utilisateurs

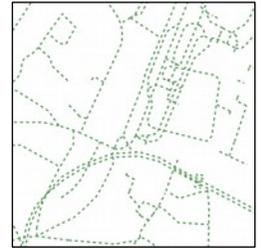


Temps de génération d'une image

Couche : linéaire simple

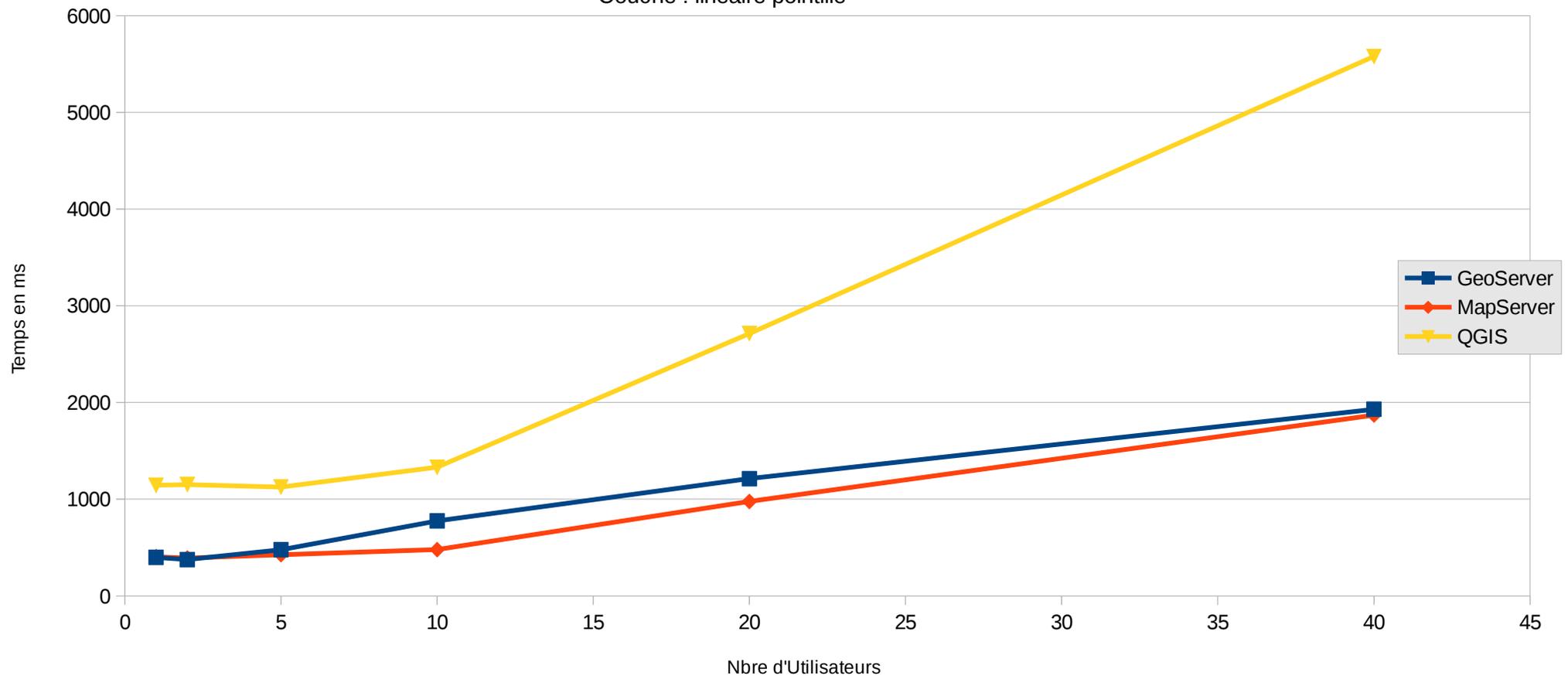


# Évolution du nombre d'utilisateurs



Temps de génération d'une image

Couche : linéaire pointillé

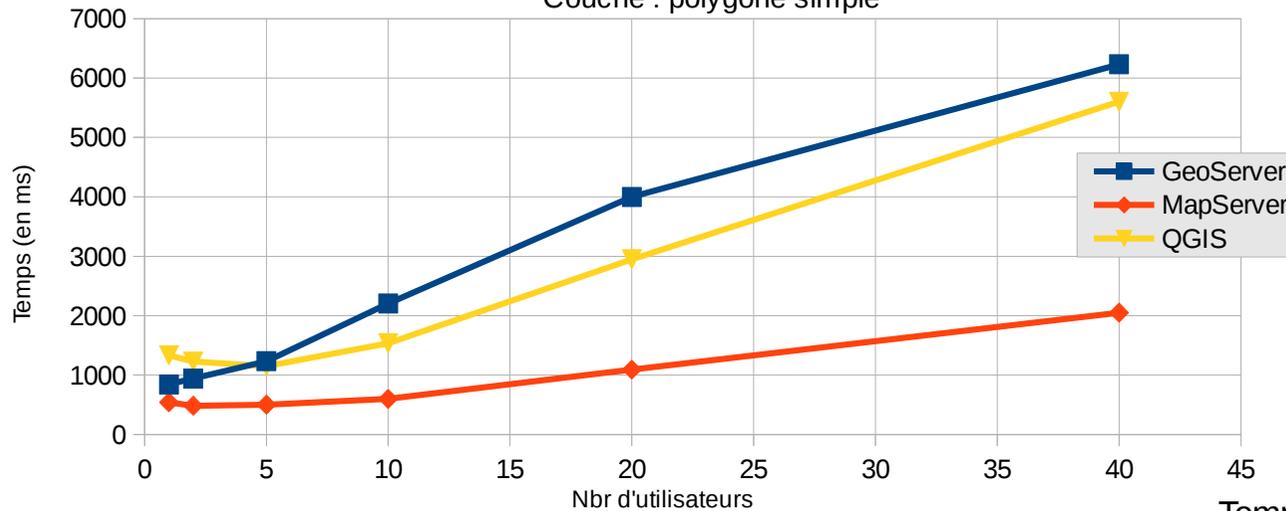


# Attention à l'optimisation !



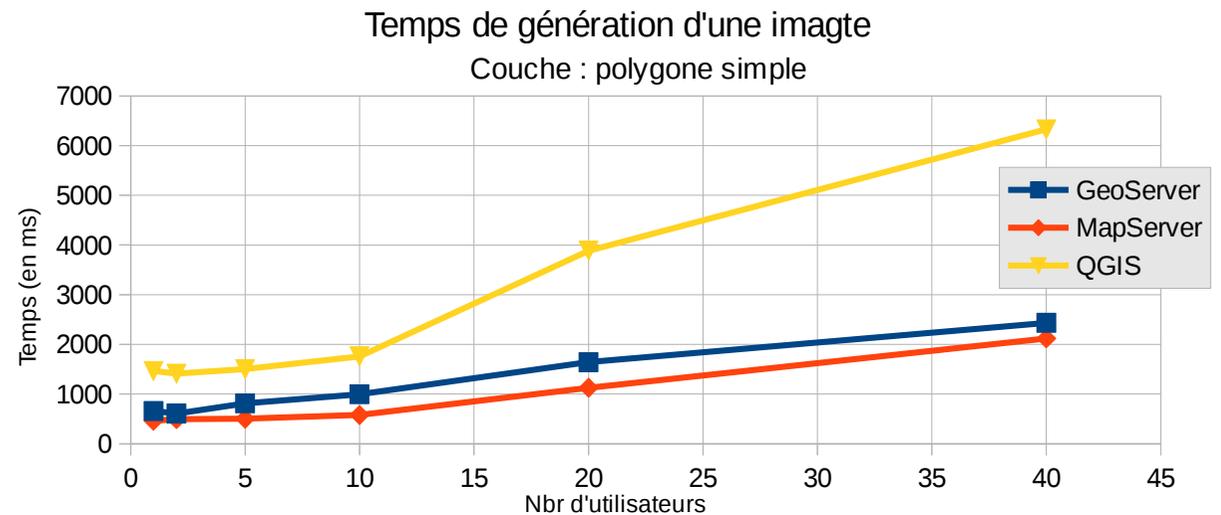
Temps de génération d'une image

Couche : polygone simple



GeoServer out-of-box

GeoServer optimisé

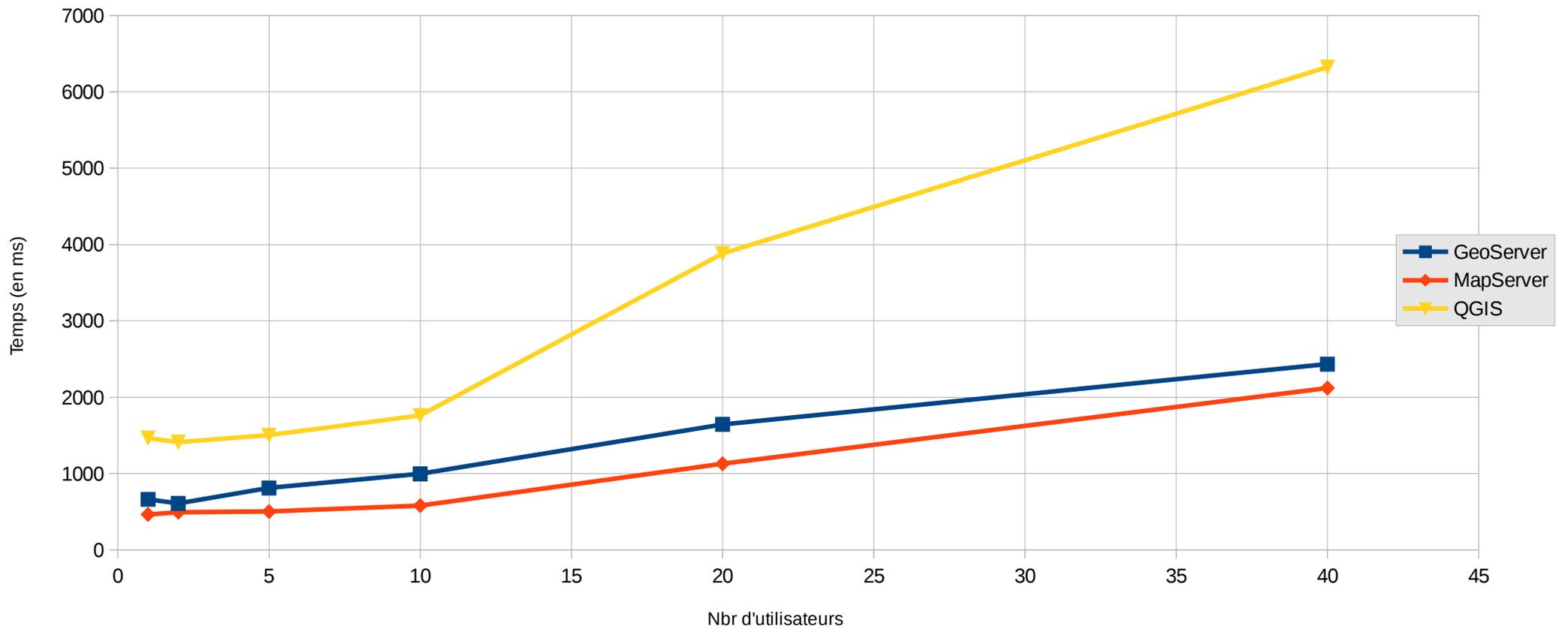


# Évolution du nombre d'utilisateurs

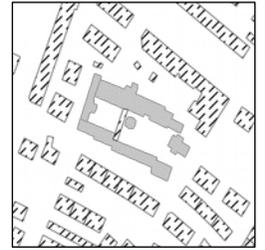


Temps de génération d'une image

Couche : polygone simple

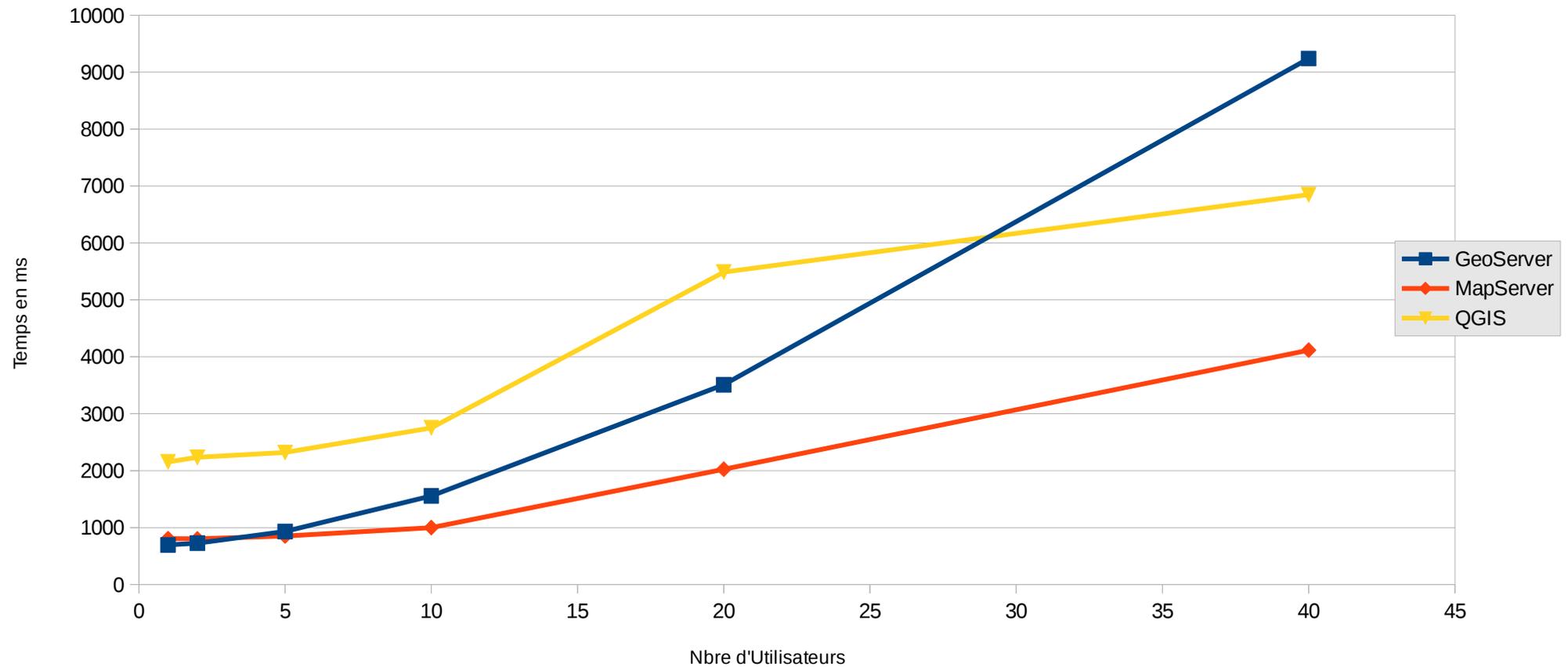


# Évolution du nombre d'utilisateurs



Temps de génération d'une image

Couche : polygone hachuré



# 1ere conclusion

- Amélioration des performances avec GeoServer
  - Pas out of the box
  - Plugins Control flow module manquant ?
  - Désactivé certaines valeurs par défaut
- QGIS plutôt performant mais moins que les autres
- MapServer plutôt performant, out of the box



# Autres considérations

- QGIS server :
  - Erreurs lors des tests :
    - ~200 au total
    - notamment sur les polygones – ~135 erreurs
  - L'écart type du temps de réponse augmente avec le nombre d'utilisateurs
    - Double de ceux de MapServer et GeoServer



# Conclusion / todo

- Il manque un test QGIS 2.18 vs 3.x
  - Des évolutions ont été apportées cette année par Camptocamp sur les performances
- Ce test permettra au prochain code sprint de s'assurer qu'il n'y a aucune régression (voire une amélioration)
- Automatiser la génération des graphiques



# Information projet

Fork me on GitHub

- Github : [http://github.com/camptocamp/ms\\_perfs](http://github.com/camptocamp/ms_perfs)
- Contactez moi : [yves.jacolin@camptocamp.com](mailto:yves.jacolin@camptocamp.com)
- PR bienvenues



to camp 

camp **to** camp

INNOVATIVE SOLUTIONS  
BY OPEN SOURCE EXPERTS