Équilibrage de Charge et Haute Disponibilité

pour applications Web Ruby On Rails

G.GASPARD, R.JACHNIEWICZ, J.LACAVA, V.MESLARD

Licence Professionnelle ASRALL, promotion 2009

29 avril 2009

Table des matières

Le projet

Équilibrage de Charge et Haute Disponibilité
pour applications Ruby On Rails

L'équilibrage de charge

- Pourquoi ? Répartir le travail
- Comment ? DNS ou reverse proxy
- Gains ? QoS, rapidité, flexibilité

La haute disponibilité

- Pourquoi ? Éviter les interruptions
- Comment ? Redondance
- Gains ? Productivité / Argent

Ruby On Rails

- Framework de développement Web
- Basé sur le langage Ruby
- Un bon compromis

Les solutions

Après avoir analysé plusieurs outils d'équilibrage et de haute disponibilité, voici ceux qui ont été étudié dans le cadre de ce projet.

Les solutions d'équilibrage

- LVS DR
- LVS Nat
- LdirectorD

LVS Dr

- Utilisation d'adresses IP publiques
- Pas d'isolation de serveurs
- Cluster de grandes tailles

LVS Nat

- Isolation du cluster
- Peu de configuration à mettre en place
- Serveurs multi plate-forme

LdirectorD

- Surveillance du pool de serveurs
- Requête sur une URL connue
- Réactivation automatique des serveurs up
- Interfaçage avec LVS

Les solutions de haute disponibilité

- Heartbeat
- DRBD
- MySql Replication

HeartBeat

- Partie du projet Linux HA
- Support de LdirectorD
- Prise en charge de défaillances réseaux

DRBD

- Mecanisme de réplication de données
- Réplication synchrone
- Configuration peut évidente

MySql Replication

- Intégré à MySQL
- Réplication asynchrone
- Un maître et un esclave

Structure mise en place

Étude de notre cas.

Structure mise en place

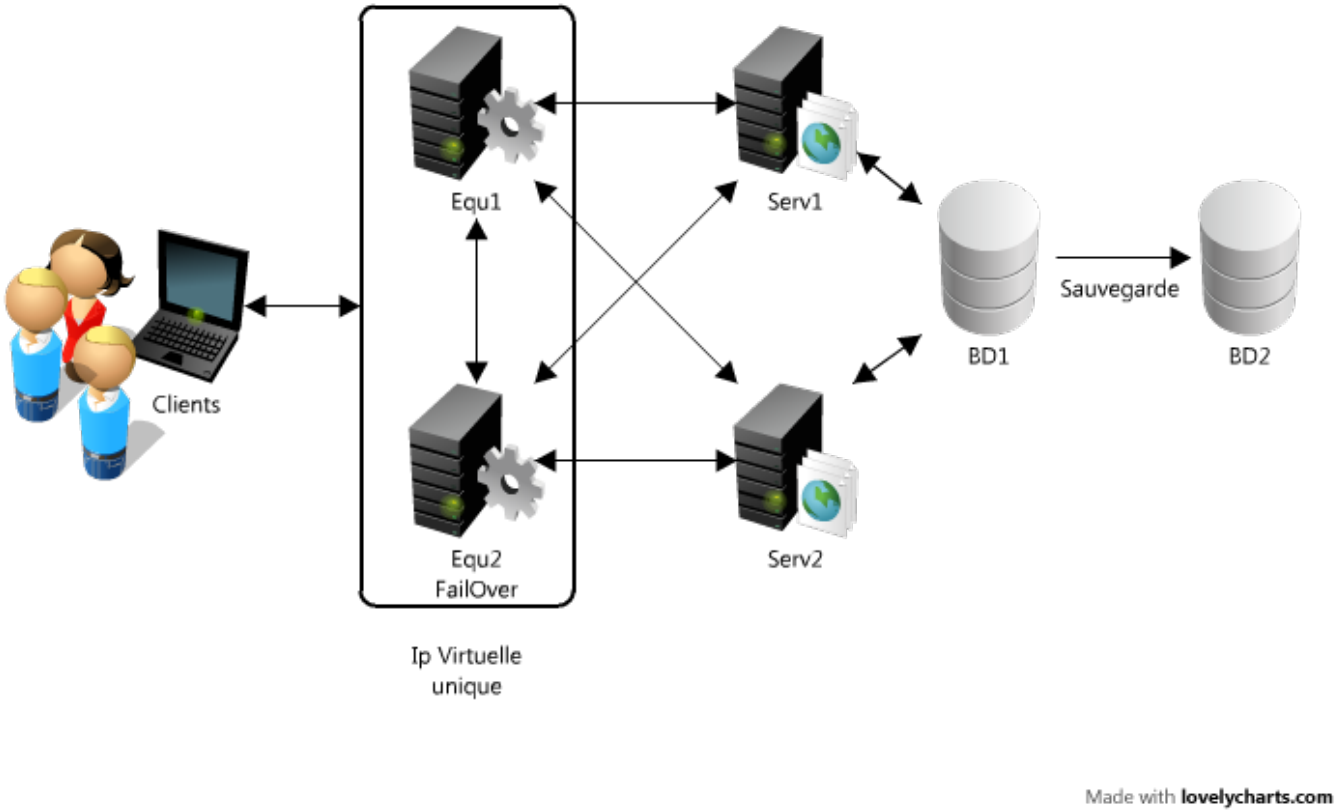

FIG.: Structure finale

Protocoles de tests

Les protocoles de tests permettent de mettre en évidence les réponses à divers types d'utilisation des serveurs Web. Il est ainsi possible de tester :

- Une montée en charge brutale.
- Une montée en charge réaliste.
- La commutation des équilibreurs de charge.
- La réplication de bases MySQL.

Point de vue client

Tout l'intérêt d'une solution d'équilibrage de charge hautement disponible pour le client, réside dans le fait d'obtenir une navigation plus fluide sans être conscient de la présence de ce cluster.
Intérêts :

- Gain de temps et de fluidité.

Point de vue administrateur

Pour un administrateur en revanche, l'interêt est bien plus concret puisqu'il
s'agit d'optimiser la disponibilité de l'application Web tout en réduisant les ressources utilisées sur les serveurs. Celà permet d'économiser le matériel tout en gagnant en performance.
Interêts :

- Économie de ressources.
- Augmentation de la durée de vie des serveurs.
- Optimisation de la disponibilité des serveurs.

Applications utilisées

Afin de réaliser des tests de montée en charge, nous avons déterminés une liste d'outils libres très pratiques :

- Apache Benchmark
- Siège
- Httperf
- Tsung

Application : Apache Benchmark

- Concurrence des connexions.
- Détermine le nombre de connexions / secondes.

Application : Httperf

- Concurrence des connexions.
- Gestion des sessions.
- Version simplifiée : HTTP_Load.

Application : Siège

- Concurrence des connexions.
- Gestion des sessions.
- Gestion de scénarios.

Application : Tsung

- Concurrence des connexions.
- Gestion des sessions.
- Utilisation en cluster.
- Basé sur le langage Erlang. (langage orienté concurrentiel)
- Support de nombreux protocoles (WebDav, SOAP, MySQL, Jabber, Html, . . .)
- Gestion de scénarios.
- Génération de graphiques et rapports.

Quelques résultats du point de vue client

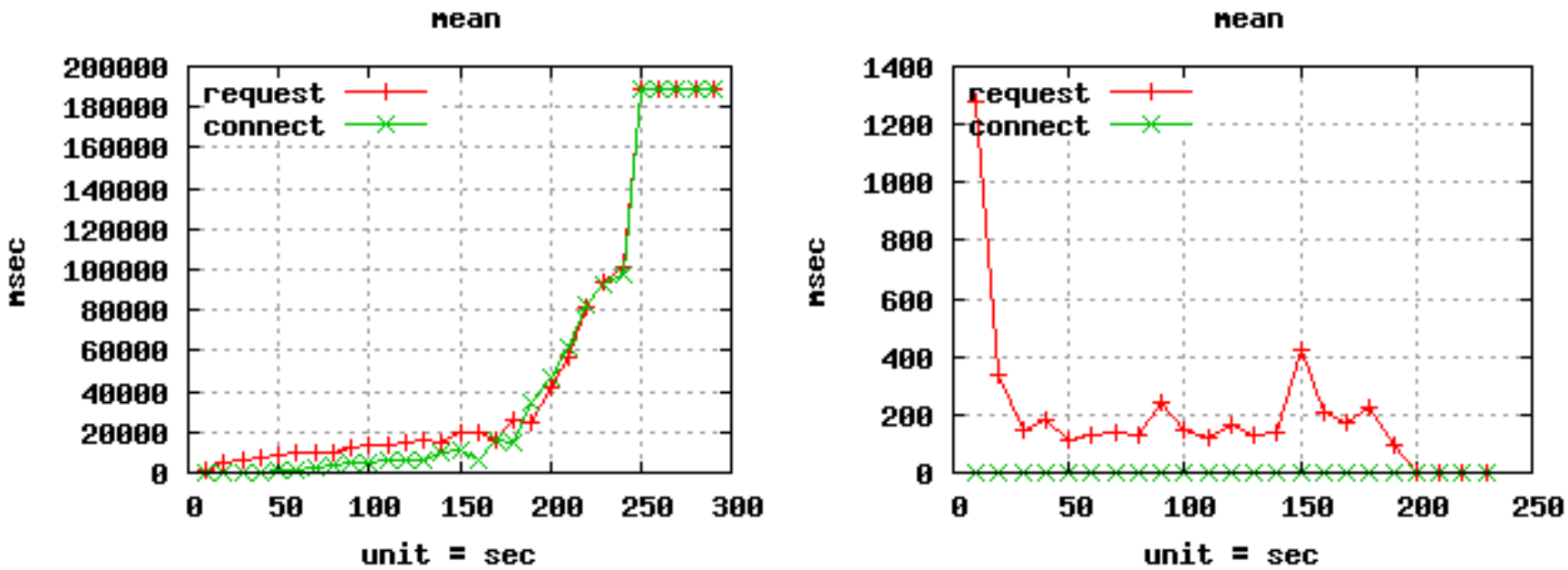

FIG.: Comparatif de performance PHP sans puis avec équilibrage

Quelques résultats du point de vue client

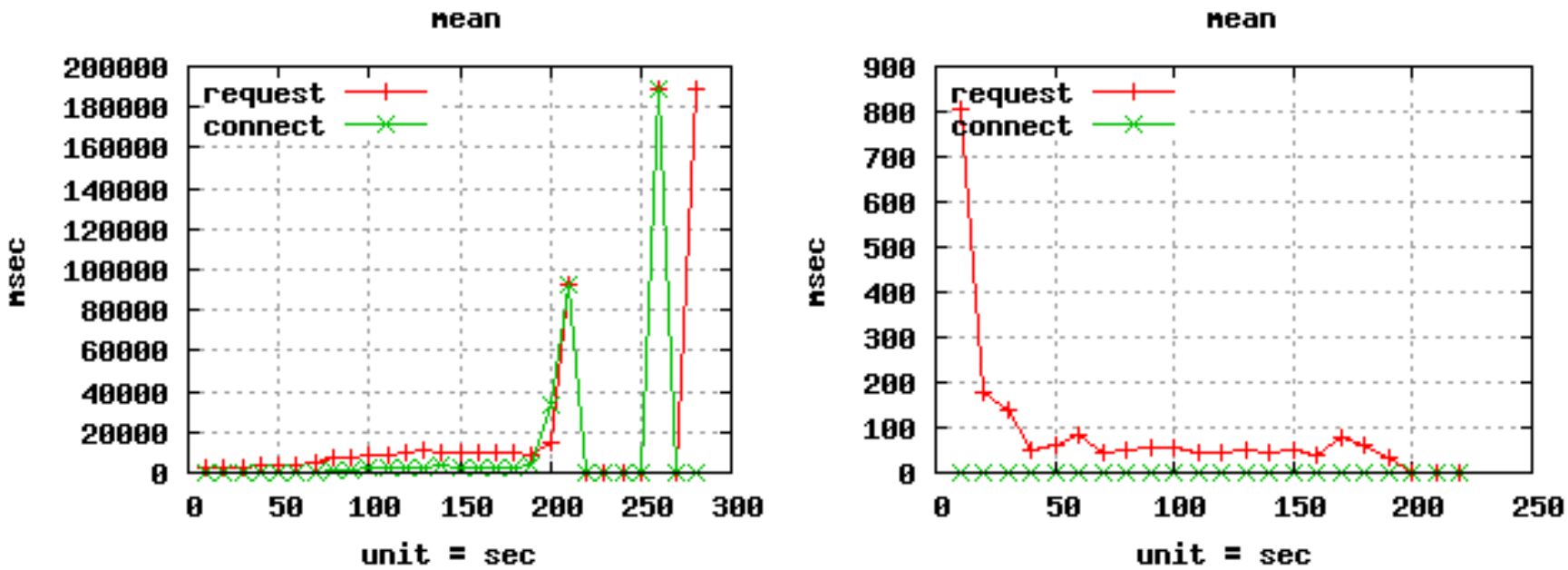

FIG.: Comparatif de performance RoR sans puis avec équilibrage

Quelques résultats du point de vue administrateur

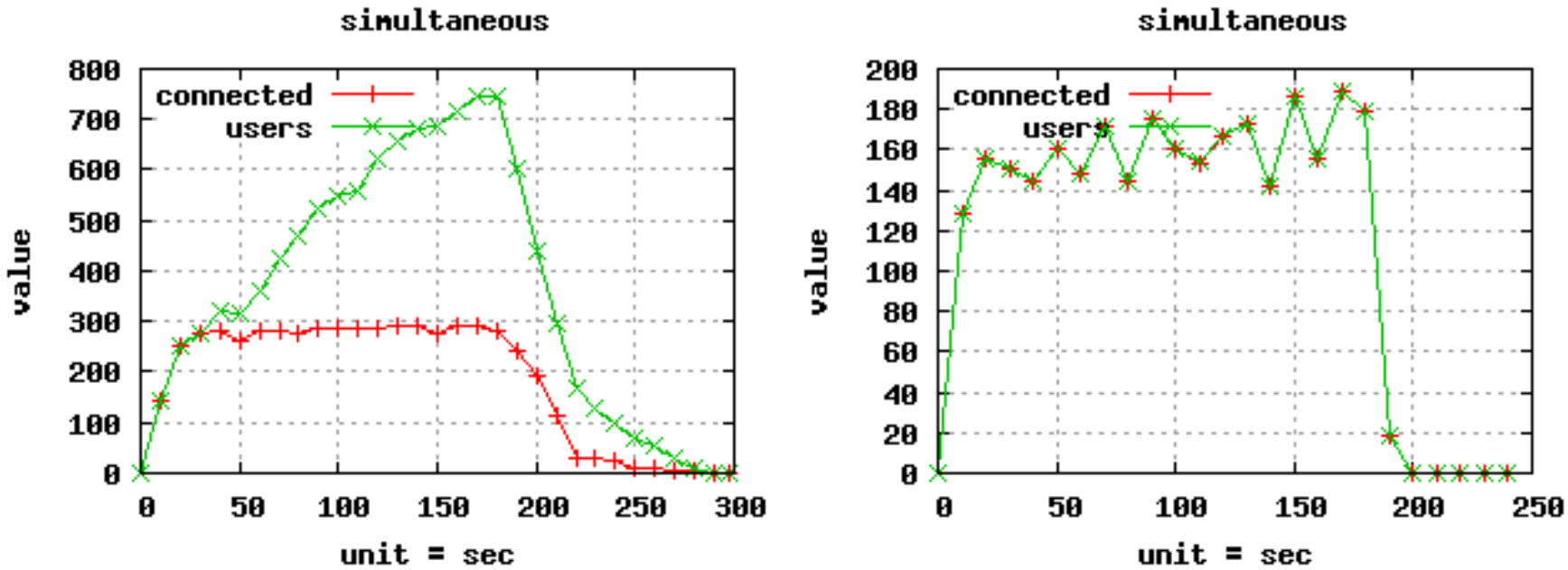

FIG.: Comparatif d'utilisateurs PHP sans puis avec équilibrage

Quelques résultats du point de vue administrateur

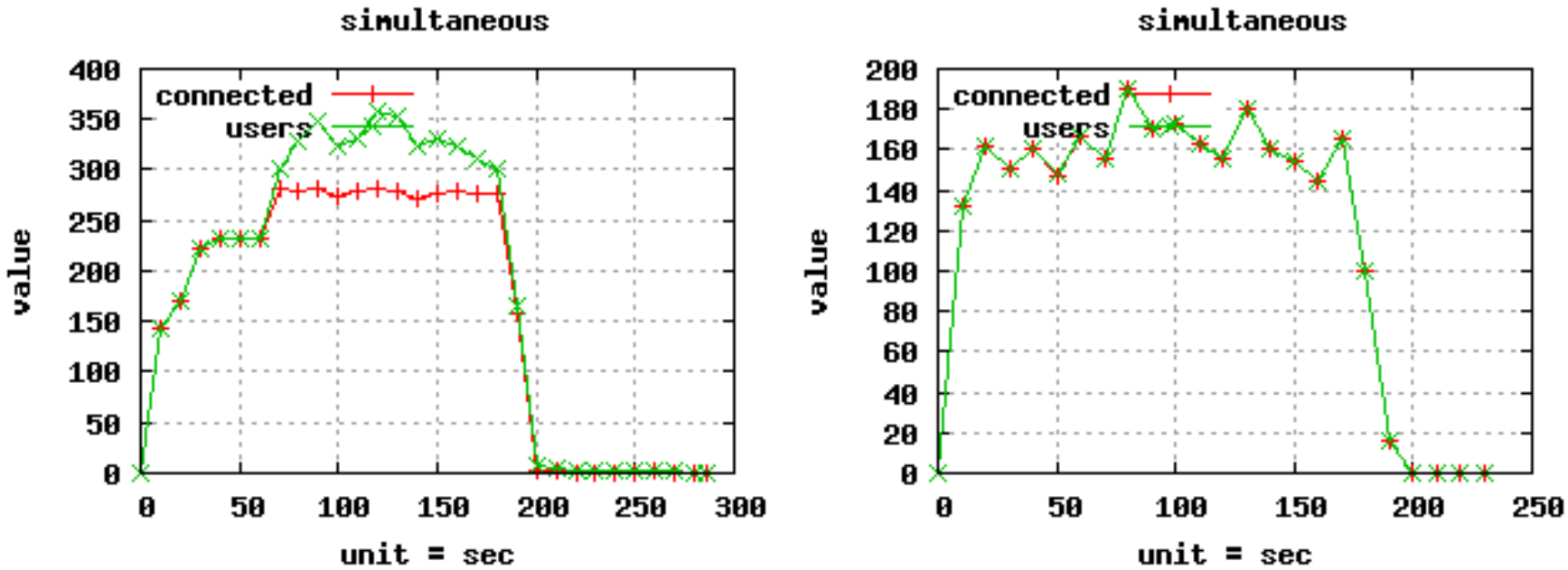

FIG.: Comparatif d'utilisateurs RoR sans puis avec équilibrage

Quelques résultats du point de vue administrateur

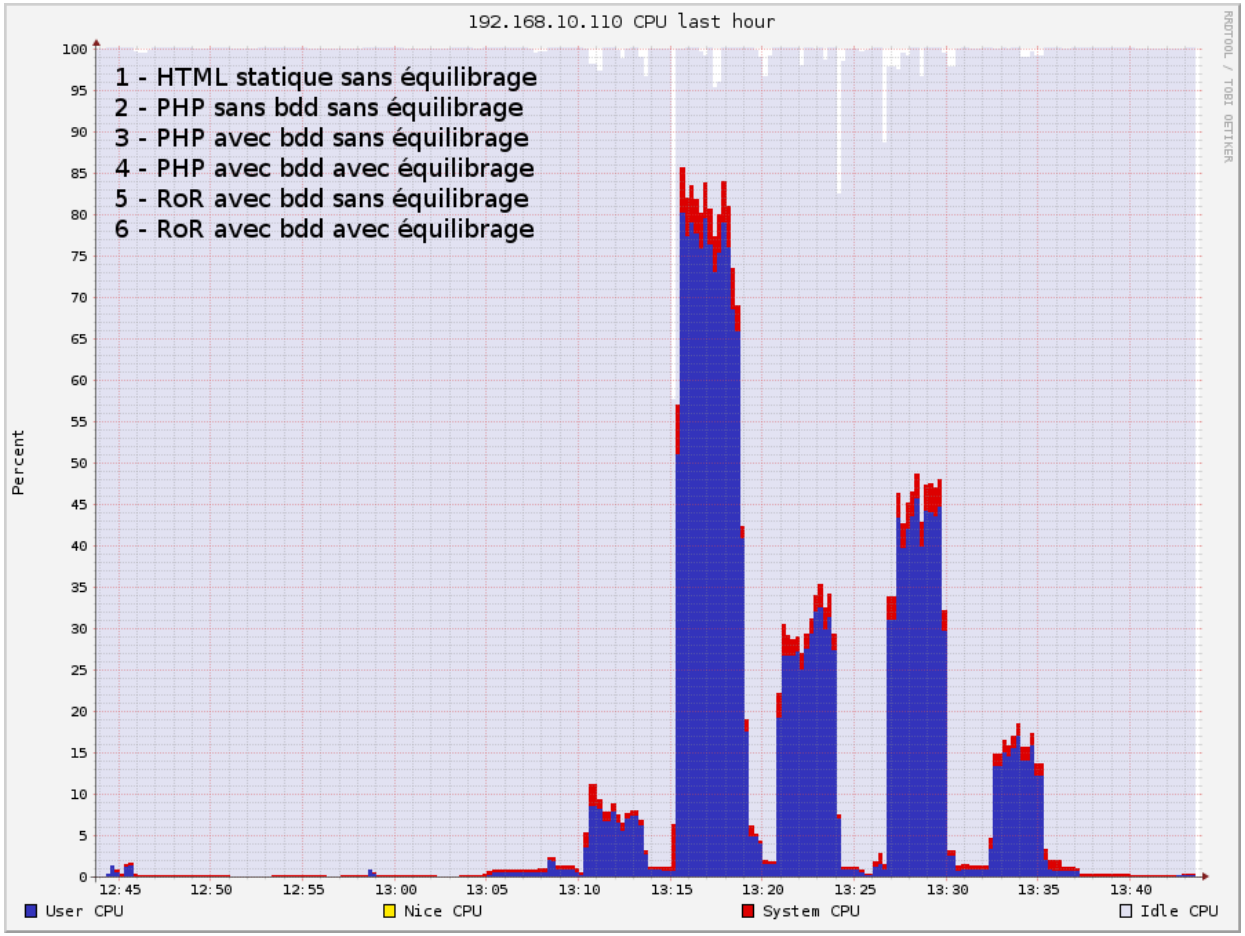

FIG.: Ressources CPU du serveur surveillé

Quelques résultats du point de vue administrateur

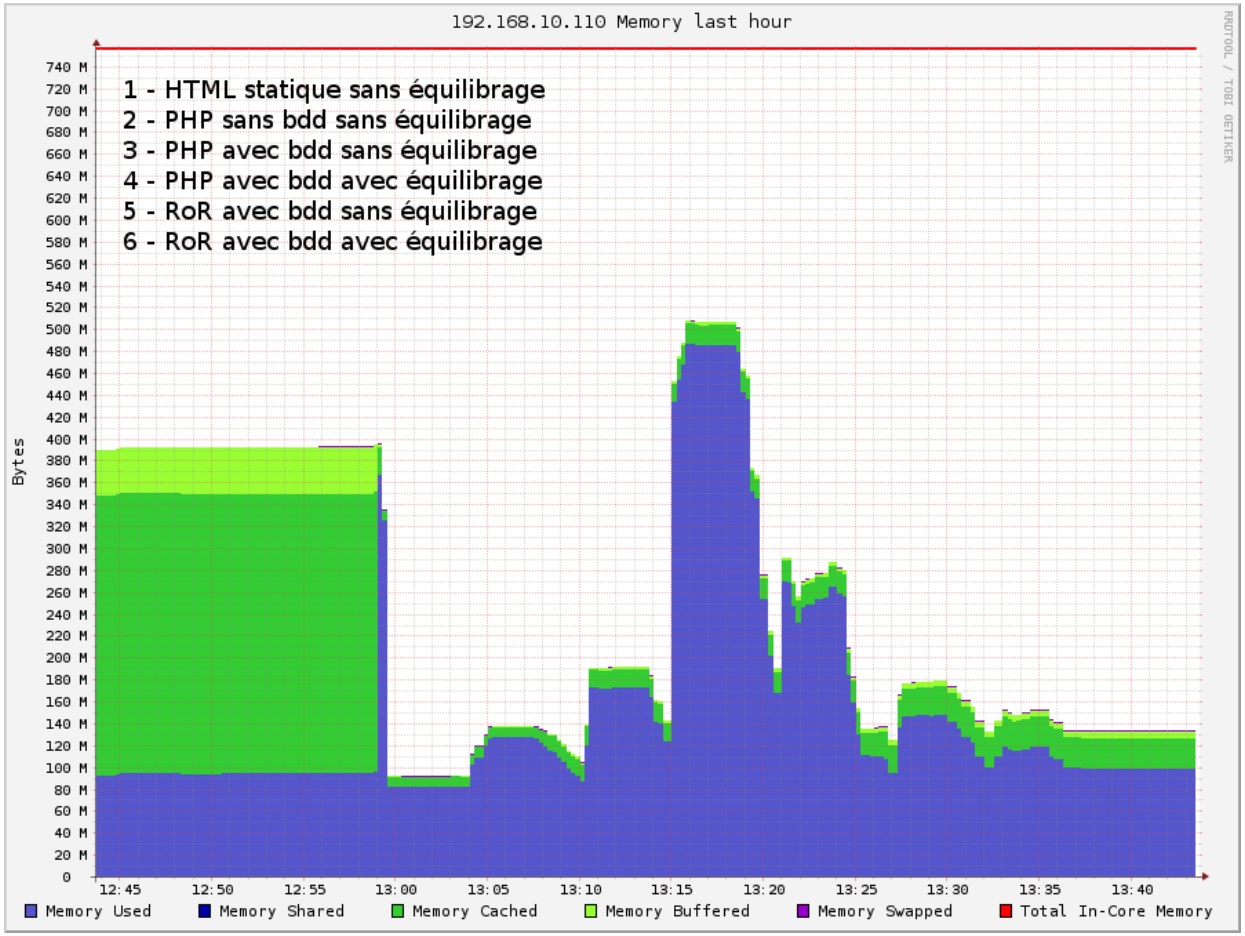

FIG.: Ressources mémoire du serveur surveillé

Conclusion

- Gains importants aussi bien pour clients que l'administrateur.
- Peu vite devenir coûteux (redondance matérielle).
- Mise en place presque transparente.
- Indispensable dans toute grande infrastructure.