Monitoring d’application web avec Librato et statsd

# Introduction

Librato est un service de monitoring de performances.

A la différence de services hébergés comme cacti, munin, graphite, il affiche tout type de mesure que vous lui envoyez. Donc pas seulement des mesures hardwares, mais aussi fonctionnelles.  
Les mesures sont de type : counter et gauge.

Son principal concurrent est New Relic.

On peut vouloir surveiller plusieurs indicateurs, afin de détecter les tendances en temps réel.  
De la santé de plusieurs serveurs (cpu, ram), aux métriques fonctionnelles.  
Cela peut être le cas du nombres de parties jouées d’un jeu en lignes, ou, dans notre cas, le suivi du nombre de commandes passées par rapport au nombre de visiteur.

# Librato

Metric : un nom, une valeur : une sonde  
Instrument : un graphique représentant une ou plusieurs métriques à mettre en corrélation.  
Dashboard : Regroupement d’un ou plusieurs instruments afin de pouvoir etre visualisés en un coup d’œil.

L’exemple va porter sur la création d’une mesure portant sur cpu\_temp. C’est une gauge puisqu’elle indiquera la valeur à un moment donnée.

http://labs.local/librato/site/post/cpu\_temp/65.4  
Retrouver la metric  
Créer un instrument  
Ajouter au dashboard

Voir le dashboard  
Explication sur le fonctionnement (curl)  
Appel des fonctions d’envoi (plusieurs fois)

Retour sur la vue du dashboard

AB et vue du dashboard en même temps

$ ab –n 10000 <http://labs.local/librato/site/home>

Problématique : Si on fait une requête en curl a chaque visite, le site ne sera pas scalable.  
Pour pallier à cela, on utilise un collecteur de statistiques.

A Mediastay, sur une application nodejs, on utilisait statsd.  
Le principe : stats est un démon en nodejs qui ecoute un port UDP.

On crée un socket udp en javascript, ou en php, et on envoit les informations à statds.  
Il les collecte, et les envoie periodiquement à librato.  
Avec UDP, on accepte de perdre des paquets, mais en contrepartie et on peut traiter des milliers d’appels concurrents sans problème de charge.

# Conclusion

On peut répondre a des problématiques de monitoring en temps réel grâce à ces deux outils, sans perte de perfomances liées à la montée en charge.

Pour en savoir plus :

<http://www.getapp.com/metrics-application>

<https://github.com/etsy/statsd/>

<https://github.com/librato/statsd-librato-backend>