

Stack Overflow behind the scenes. How it's made

Speaker : Oded Coster

Format : Conférence

Date : 21 avril 2016

Slides : <http://OdedCoster.com/Devoux.fr-2016>

Depuis 2 ans, Oded Coster travaille en tant que développeur sur Stack Overflow, une plateforme de questions / réponses bien connue des développeurs.

Au cours de cette conférence, Oded nous a présenté les dessous de Stack Overflow : architecture technique, outils et organisation.



Tous les chiffres présentés datent d'une seule journée (9 février 2016) :

- 209 millions de requêtes (2423 requêtes / secondes)
- 66 millions de page web
- 569 GB de data reçus
- 3.08 TB de data émises
- 504 millions de requêtes SQL
- 5 milliard de hits Redis
- 17 millions d'Elasticsearch request
- Temps de réponse moyen des pages de questions / réponses : 22.21 ms

Infrastructure :

- 4 Microsoft SQL Servers (dont 2 en lecture seule pour les réplicas) (384 Go, 12 cœurs x 2)
- 9 serveurs web IIS (64 Go, 12 cœurs x 2) : leur applicatif est bâti sur la plateforme .NET de Microsoft
- 2 serveurs Redis (256 Go, 10 cœurs x 2) : pas utilisé pour la persistance, seulement pour le cache. Oded précise que Stack Overflow peut tourner sans Redis mais sera très lentement.
- 3 Tag Engine Servers (solution maison qui gère les tags)
- 3 serveurs Elasticsearch (192 Go, 8 cœurs x 2)
- 4 load balancers HAProxy
- 2 réseaux

- 2 firewalls
- 4 routers
- 2 datacenters (bascule en cas de crash)

L'équipe Q&A

- Equipe composée de 10 personnes réparties dans 4 pays : Angleterre, Slovénie, Majorque et USA
- La société fonctionne sur le même mode : distribuée
- Tous les outils de communications sont en ligne :
 - Google Docs
 - Trello
 - Stack Chat : chat de StackOverflow (apparemment peu connu du public). Chat persistant. Au réveil, il regarde tout ce qui a été posté (fuseaux horaires différents)
 - Google Hangouts : complète le chat. Pratique pour de longues discussions, par exemple pour convaincre une personne.
- Des bots sur leur chat :
 - CI builds : avec le message de commit
 - Livraison en production, avec celui qui a cliqué sur le bouton
 - Volume d'exceptions inhabituel
 - Pour le fun : the wheel of blame
- Outils utilisés par la Team :
 - Visual Studio
 - Git
 - Gitlab : hébergé par leurs soins car si GitHub crash, l'activité des devs s'arrête
 - TeamCity pour la CI
 - SQL Server Management Studio
- Core platform sur la stack Microsoft :
 - C#
 - ASP.NET / MVC
 - IIS
 - SQL Server
- Autres outils hors stack Microsoft (se définissent comme agnostiques) :
 - HAProxy (CentOS)
 - Redis (CentOS)
 - Elasticsearch (CentOS)
 - Tag Engine (Windows)
- Monitoring and Alerting
 - MiniProfiler - in the browser
 - Opserver – dashboard
 - Grafana – dashboard
 - Bosun – alerting system
- Build process dans Teamcity :
 - Déclenchement d'un build lors d'un commit
 - Meta build : StackOverflow et StackExchange
 - Prod build : très réactif en cas de problème. Utilise les retours de la communauté, notamment via Twitter.

- Build process – Some details
 - Localisation (JS, C#, Razor views) => 5 langues
 - Compilation LESS + minification
 - Packaging JS + minification
 - Rolling build => assure le déploiement sur l'ensemble des serveurs avec 0 indisponibilité
- Depuis l'IHM de Stackoverflow, ils sont accés au détail du temps de réponse des requêtes, jusqu'à la requête Redis
- Comment c'est aussi rapide ?
 - SQL optimisé : forte expertise et investissement conséquent
 - Fast librairies : Dapper, Jil (sérialisation)
 - Caching : en font beaucoup. 2 niveaux de cache : 1^{Er} sur la mémoire du serveur et le 2nd dans Redis
 - Focus sur la performance
 - Equipe spécialisée comprenant le bas niveau
 - Ils mesurent tout

Leur point de vue sur le Cloud :

- Pas utilisé pour leur prod, même si beaucoup de personnes ont déjà travaillés sur le Cloud dans leur précédent job dans le Cloud.
- Trop cher
- Ne satisfait pas le haut niveau de performance auquel ils aspirent
- Latence non fiable
- Nécessitait un énorme investissement, notamment au niveau de la base de données. Cet cela, même si Azure dispose de SQL Server : la base de Stackoverflow dépasse la capacité de données possibles sur Azure.