

Alvin Richards, 10gen : « Aujourd'hui le M de L.A.M.P., c'est MongoDB »

Silicon.fr – MongoDB s'inscrit dans le courant NoSQL. Dans quelle mesure le modèle SQL vous semble-t-il dépassé ?

Alvin Richards – Au cours des 30 dernières années, un système de gestion de bases de données relationnelles (SGBDR) était la seule option possible pour stocker des données persistantes. Mais les changements dans la charge de travail et les nouvelles exigences dans les rythmes de diffusion des produits ont créé un besoin pour un autre type de bases de données.

Ces nouvelles contraintes exigent des bases qui soient optimisées pour pouvoir traiter de grandes quantités de données, avec des taux d'exploitation élevés. Des bases qui s'adaptent à un cadre de développement agile et qui soient compatibles avec des environnements Cloud. C'est de là que viennent les bases non relationnelles.

Elles sont différentes dans le sens où elles n'ont pas besoin de schémas de table fixes. Elles ne nécessitent pas non plus d'opérations de jointure puisqu'elles stockent des données non structurées. Et puis elles sont conçues pour être scalables horizontalement. La plupart d'entre elles peuvent être qualifiées de « base de données orientées documents ».

Silicon.fr -En terme de performances, en quoi votre solution dépasse-t-elle les acteurs traditionnels des bases de données relationnelles ?

Alvin Richards -La manière dont les logiciels sont faits a radicalement changé depuis que les SGBDR ont été inventés. Aujourd'hui, on utilise des méthodes de développement itératives, dont le but est d'avoir des déploiements continus et des cycles de réalisations très courts.

Pour y arriver, le stockage des données des applications doit lui aussi être flexible. Or la structure rigide des bases de données traditionnelles ne correspond pas vraiment à l'approche agile. A chaque fois que vous sortez une fonctionnalité, le schéma des données a lui-aussi très souvent besoin d'être modifié. Plus la base est grande, plus cela devient complexe. Et plus cela ralentit le processus. Si les livraisons de code sont très fréquentes, la planification des migrations de schéma et la maintenance d'une telle base deviennent tout simplement impossibles.

MongoDB a été conçu dès le départ pour répondre à ces nouveaux modes de travail et à ces nouveaux environnements. Il a complètement changé la manière de modéliser, de stocker et d'accéder aux données.

MongoDB représente les données comme dans des fichiers JSON. Ce qui permet une scalabilité horizontale sur plusieurs serveurs sans avoir à faire des jointures ou à passer par un protocole de validation à 2 phases (2-phase commit ou 2PC), raison opérationnelle pour laquelle les SGBDR ne sont pas scalables de cette manière.

Silicon.fr -Et par rapport aux autres acteurs NoSQL comme CouchDB ?

Alvin Richards -Tous les projets NoSQL essayent de répondre à leur manière à cette problématique de scalabilité. La particularité de MongoDB est qu'il est simple à comprendre et simple à utiliser pour les développeurs. JSON est simple à utiliser avec tous les langages modernes populaires comme Java, Python, C#, Node.js (NDT : JavaScript), C++, [PHP](#), etc.

Pour faciliter la transition depuis un SGBDR, MongoDB assure également une cohérence immédiate des données. De telle sorte que si une donnée est écrite puis lue, vous avez la certitude de lire ce que vous venez d'écrire. Tous ces objectifs ont fait de MongoDB une base facile à adopter et simple à l'usage. C'est fait son succès.

Silicon.fr -Plusieurs personnes appellent à remplacer la brique MySQL du modèle LAMP par MongoDB ; celà vous parait-il pertinent, notamment dans une démarche Big Data ?

Alvin Richards -Les développeurs veulent du changement. Ils ont besoin d'outils plus simples et plus faciles pour développer leurs systèmes. Quant aux responsables financiers, ils veulent que leur entreprise soit compétitives, mais avec des budgets raisonnables. L'open-source répond bien à tout cela. Mais vous avez aussi besoin d'une base avec une bonne persistance des données, une base qui permette de réaliser des représentations flexibles, qui offre une scalabilité simplifiée ; le tout pour que l'organisation puisse sortir des projets plus rapidement et pour moins cher. Tout ce que permet MongoDB. Et que [MySQL](#) ne peut pas faire avec autant de flexibilité.

Imaginons par exemple que vous ayez un catalogue produits. Vous aurez certainement besoin de stocker des données qui varient en fonction des pays où vous vendez ce produit. Avec MongoDB vous n'avez qu'à ajouter ces attributs dans l'Objet "produit" et c'est tout. Avec un SGBDR, vous pourrez avoir à rentrer ces informations dans plusieurs tables et plusieurs relations, ce qui ajoute une complexité considérable. Tout cela pour résoudre un problème métier très simple. Nous avons plus de 1.5 millions de téléchargements de MongoDB par an. La majorité est déployée sur Linux, pour faire des applications Web. En fait, aujourd'hui, le « M » de LAMP : c'est MongoDB.

A lire : [Dossier : Le Big Data va t'il forcer le modèle L.A.M.P. à muter ?](#)