

# Benoît Jacquemont, SMILE : « PHP est devenu un vrai langage pour l'entreprise »

**Silicon.fr – Pensez-vous que le Big Data va concerner toutes les entreprises ou simplement les grands groupes, comme les géants du Web par exemple ?**

Benoît Jacquemont – Le Big Data concerne les entreprises qui traitent de très grosses quantités de données. C'est une problématique indépendante de la taille de la structure. Par exemple, chez Smile, nous sommes plus de 500 salariés. Mais le Big Data n'a pas tellement d'intérêt pour le volume de données que l'on génère. En revanche, des petites start-ups qui analysent l'ADN sont en plein dedans. Si je regarde nos clients, nous en avons qui utilisent MongoDB pour leurs sites de e-commerce avec plusieurs millions de produits. Mais cela n'aurait pas de sens s'il n'en avait que quelques dizaines de milliers. Donc non, cela ne va pas toucher tout le monde.

D'autant plus que les solutions traditionnelles permettent de traiter beaucoup de volumes. On peut même considérer que quelques millions de références, surtout pour des produits simples, peuvent être traitées avec ces solutions. Dans certains cas, on gère très bien une base de données de plusieurs millions – voire plusieurs milliards – de lignes avec du MySQL. Je n'ai pas le droit de vous parler du client en question, mais nous l'avons déjà fait. Ce n'était certes pas du temps réel, mais les réponses devaient être tout de même très rapides en partitionnant les données dans des tables plus petites pour avoir des indexes plus faciles à gérer.

En revanche, ce qui est vrai, c'est qu'il y a de fortes chances que les PME soient de plus en plus consommatrices de services associés au Big Data. Les PME seront à mon avis de plus en plus comme les particuliers qui font des recherches sur Google. Elles bénéficieront d'un service Big Data, mais ne feront pas nécessairement de Big Data.

**Silicon.fr – On parle actuellement beaucoup de MongoDB pour remplacer MySQL dans la pile LAMP. En termes de performances, en quoi cette solution dépasse-t-elle les acteurs traditionnels des bases de données relationnelles ?**

Benoît Jacquemont – Le problème des bases de données relationnelles traditionnelles, c'est la difficulté à faire de la « scalabilité horizontale ». Le deuxième problème, c'est qu'elles font assez peu de traitements côté base. On fait certes de l'agrégation, mais beaucoup de traitements statistiques se font côté applicatif. Des astuces existent mais il faut alors stocker les données sur un nœud principal pour pouvoir les traiter. L'avantage des solutions Big Data, c'est que les traitements et les données – les deux – sont distribués.

Imaginons que vous soyez sur cinq serveurs. Lorsque vous demandez un traitement sur les données en question, chaque serveur va faire le traitement sur ses données à lui. Et il y aura ensuite une agrégation. 99 % du traitement va donc être distribué en même temps que les données. C'est le principe du MapReduce (DEF).

Quant à MongoDB, c'est un des exemples de solutions Big Data, mais ce n'est pas nécessairement le plus adapté. Le projet libre le plus « full Big Data » – et qui d'ailleurs n'est pas intéressant sur une machine seule – c'est Hadoop.

MongoDB fait plus Big Data par « effet secondaire ». C'est une base de données qui a implémenté du sharding, mais à l'origine ce qui l'intéressait c'était le côté NoSQL (qui ne signifie pas « pas de SQL » mais « pas seulement (Not only) du SQL »). C'est-à-dire la souplesse du schéma et l'accès simplifié aux données.

**Silicon.fr – Voyez-vous arriver une solution complète open source pour le Big Data sur le modèle de ce qu'a été LAMP pour le Web dans la décennie précédente, c'est-à-dire simple, standardisée et efficace ?**

Benoît Jacquemont – Non, pas vraiment. Pour une raison principale. Le Big Data n'est pas un sujet simple. La pile LAMP est devenue populaire parce qu'elle a permis à plein de gens de faire des petits sites, très rapidement, sans se poser trop de questions.

A l'opposé, dans le Big Data, vous avez des problématiques qui sont intrinsèquement complexes. Néanmoins, on peut effectivement imaginer avoir une partie PHP en frontal – qui servirait à interroger une base MongoDB ou Hadoop. On n'aura pas de problème à communiquer avec ces bases-là. La communauté de LAMP est très active et on trouve de très nombreux connecteurs (notamment pour MongoDB ou CouchDb). Mais, ce ne sera que pour la présentation des données. Le traitement, lui, se fera sur des serveurs vraiment taillés pour le vrai Big Data.

**Silicon.fr – Ce que vous me disiez sur un modèle LAMP « boosté » avec MongoDB (ou autre), c'est que la stack pourrait certes servir à interroger les données, mais de toute façon il y aura derrière un ou plusieurs serveurs pour traiter les données Big Data elles-mêmes. C'est bien cela ?**

Benoît Jacquemont – Oui, exactement. Le traitement PHP ne serait de toute façon pas une bonne idée. Même les développeurs PHP considèrent que la durée de vie d'une mémoire PHP est de l'ordre de la seconde. Il n'y a pas eu beaucoup de travail qui a été fait de ce côté-là... même s'il y a des améliorations dans la 5.3. Donc l'idée c'est que les données soient traitées et manipulées côté serveur Big Data (avec du MongoDB, du CouchDB ou du Hadoop). Et qu'après, le résultat de la requête soit récupéré pour l'afficher au client. Dans ce cadre-là, le PHP est tout à fait intéressant.

**Silicon.fr – Et donc côté serveur qui traite les données, la stack open-source la plus simple pour le Big Data est du type Linux/MongoDB ou Linux/Hadoop...**

Benoît Jacquemont – C'est ça. Ces éléments-là se suffisent à eux-mêmes. A partir de là, vous communiquez depuis votre application avec ces données comme si c'était des bases de données classiques. Sauf que c'est du Big Data... Avec tout de même tout ce que cela implique comme gestion des traitements à envoyer vers le cluster.

## **Silicon.fr – Indépendamment du Big Data, quel avenir voyez-vous pour le modèle LAMP ?**

Benoît Jacquemont – Je pense que LAMP évolue de plus en plus vers le monde de l'entreprise. Cela se voit au niveau du PHP qui était perçu comme un langage de bidouilleurs. Il n'était pas très bien vu dans le monde professionnel. Il y est entré par la petite porte, parce que plein de produits étaient faits en PHP, comme les CMS. Mais aujourd'hui, grâce à des Frameworks comme Symfony 2 – qui est à l'état de l'art des Framework Web – PHP est devenu un vrai langage pour l'entreprise. Ce qui pourrait faire le plus concurrence à LAMP aujourd'hui dans son usage Web, ce sont les Framework JavaScript qui répondent aussi bien aux besoins côté Back-end avec Node.js que côté Front avec des Frameworks MVC. Mais le PHP continuera à mon avis à prédominer côté serveur. Et avec lui LAMP à séduire le milieu professionnel.

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