SQL vs NoSQL : Synthèse

**2013**

Valentin

EISTI

02/10/2013



Sommaire

[INTRODUCTION : 3](#_Toc368521602)

[Partie 1 : Requêtes 4](#_Toc368521603)

[Partie 2 : Transactions 5](#_Toc368521604)

[Partie 3 : Cohérence 6](#_Toc368521605)

[Partie 4 : Evolutivité 7](#_Toc368521606)

[Partie 5 : Management 8](#_Toc368521607)

[Partie 6 : Schémas 9](#_Toc368521608)

[CONCLUSION : 10](#_Toc368521609)

# INTRODUCTION :

 Le but de cette synthèse est de résumer les points et concepts importants de la conférence de Google, présenté par Ken Ashcraft et Alfred Fuller, qui a pour titre : « SQL vs NoSQL : Battle of the Backends » disponible sur Youtube en cliquant [içi](https://www.youtube.com/watch?v=rRoy6I4gKWU). Cette conférence traite de l’utilisation de deux types de stockage de données qu’offre Google App Engine, à savoir le SQL et le NoSQL.

 Google App Engine (connu aussi sous l’abréviation GAE) est un outil permettant de construire des applications sur l’infrastructure de Google. GAE est une « Plate-forme en tant que service » qui est un des types de Cloud Computing où Google va maintenir la plateforme d’exécution des applications et l’entreprise cliente maintient les applications proprement dites.

 Ainsi, sur Google App Engine plusieurs types de stockage sont disponibles. Tout d’abord, « Google Cloud SQL » qui est un service Web fournissant une solution entièrement géré, hébergé et hautement disponible de stockage pour les applications. Et « Datastore » qui repose sur la même technologie qu’utilise Google pour ses applications et qui est une solution NoSQL permettant ainsi de supporter une grande quantité de donnée.

 Lors de cette conférence, les deux présentateurs comparent ces 2 solutions sur 6 divers caractéristiques :

* Les requêtes
* Les transactions
* La cohérence
* L’évolutivité
* La gestion/le management
* Les schémas

 

Logo Google App Engine

# Partie 1 : Requêtes

 La partie requête est importante car si l’utilisateur ne dispose pas d’un puissant langage de requêtes alors on accède plus difficilement aux données que l’on désire et on y accède moins rapidement.

 Ainsi, le Cloud SQL supporte les langages de manipulations SQL standards assurant ainsi une recherche, une modification, des ajouts, etc. de manière aisé sur la base de donnée. Le NoSQL ne supporte pas entièrement toutes les spécificités du langage SQL. En effet, certaines fonctions sont supportés mais la majeur partie des techniques avancé de ce que permet le langage SQL ne fonctionnera pas de la même manière via le NoSQL de Google. Néanmoins, les requêtes dans le NoSQL de Google pâtiront moins de l’augmentation du volume de données qu’une base de donnée uniquement SQL.

 Par exemple, le cloud SQL de Google permet d’utiliser des agrégations (group by) ce que ne permet pas directement le NoSQL du Datastore de GAE. Ainsi pour gérer ces aggréagation le NoSQL passe par des algorithmes de « MapReduce ». Le MapReduce est un algorithme pour les applications utilisant d’énormes quantités de données qui permet une répartition efficace du travail sur différents nœuds de calcul. En effet, la gestion des données sous une forme d’arbre et de couples « Clés-Valeurs » permet d’utiliser tout le potentiel de cette algorithme.



Schéma du MapReduce

 Malgré tout, le requêtage des données et beaucoup plus intuitif et facile d’accès sur MySQL de GAE que sur DataStore.

# Partie 2 : Transactions

 Les transactions sont un autre élément important du stockage des données. En effet, il faut par exemple s’assurer que les changements sont bien réalisés de manière atomique dans la base de données. La propriété d'atomicité assure qu'une transaction se fait au complet ou pas du tout : si une partie d'une transaction ne peut être faite, il faut effacer toute trace de la transaction et remettre les données dans l'état où elles étaient avant la transaction. L'atomicité doit être respectée dans toutes situations, comme une panne d'électricité ou une défaillance de l'ordinateur.

 La solution utilisant MySQL supporte bien évidemment les transactions et leur atomicité. C’est plus compliqué pour les solutions NoSQL qui ne les supportent pas toutes. Ainsi, la solution Datastore de Google App Engine utilise les « Entity Group » qui, comme son nom l’indique, groupe plusieurs entités sous un unique log de transaction. Néanmoins, ces « Entity Group » sont plus complexes à mettre en place qu’une simple Transaction en SQL qui réalise exactement le même travail. De plus, en cas d’échange entre Entity Group la limitation est à 5 Entity Group différentes en même temps pour continuer à assurer l’atomicité des transactions. Or, avec la solution MySQL, il n’y a aucune limitation.

# Partie 3 : Cohérence

 La cohérence est un élément important. En effet, la propriété de cohérence assure que chaque transaction amènera le système d'un état valide à un autre état valide. Tout changement à la base de données doit être valide selon toutes les règles définies, incluant mais non limitées aux contraintes d’intégrités aux rollbacks en cascade, aux déclencheurs de base de données, et à toutes combinaisons d'évènements.

 Ainsi, dans le cas de la solution utilisant MySQL les données sont répliquées dans deux autres Datacenter avant que la demande de l’utilisateur soit traitée. Si le Datacenter A devient inutilisable, les deux autres Datacenter peuvent assurer le traitement. Cela permet d’assurer la non-perte des données en toutes circonstances.



Schéma de la réplication des datacenter dans GAE utilisant MySQL

 Dans la solution NoSQL, une réplication transactions et donc des « Entity Groups » est assurée. Néanmoins, il n’est pas assuré (pour que 100milli secondes au maximum) que l’utilisateur est accès aux données les plus récentes à tout moment.

# Partie 4 : Evolutivité

 Lors de cette conférence, l’évolutivité de ces solutions en fonctions de l’augmentation de la quantité de données, de la volumétrie, est abordé. Ainsi, si le cloud SQL de Google fonctionne bien pour de grandes applications il devient limité face au potentiel du NoSQL pour des applications a portée mondial. Ainsi, dans cette conférence, le cas de l’application « Angry BIrds » est pris en exemple (sous le nom détourné de « Disgruntled Pigeons) pour montrer la puissance de l’outil NoSQL dans ces cas là.

 En effet, le NoSQL utilisé via GAE s’appuie sur l’infrastructure de Google, à savoir « Google File System » qui permet à l’application de pouvoir être aussi gourmande qu’elle le désire sans limite de volumétrie de données. Mais aussi, « BigTable » qui est une base de données orientée colonne, haute performance.

# Partie 5 : Management

 Dans les deux cas de figures, les outils présentés à cette conférence bénéficie de l’infrastructure et de la puissance de Google. En effet, Google gère les patchs, les installations, les serveurs. Il est donc facile de gérer, de manager sa solution.

 Néanmoins, la solution NoSQL de Google à besoin de moins de configuration et d’étapes de créations que la solution SQL. En effet, le Datastore est prêt a accueillir les données de l’applications dès la créations de l’application.

# Partie 6 : Schémas

 Les Schémas permettent de définir quels sont les types de nos données, comment sont liées les données entre elles, et donc la structure des données.

 Via le SQL il est très facile de changer la structure des données, en utilisant la commande « ALTER » par exemple. Néanmoins, les données deviennent indisponibles le temps de la modification de la structure. En NoSQL, le problème ne se pose pas puisque le changement se produit dans l’application et que l’impact est immédiat au niveau du traitement des données.

Un exemple est pointé avec la fausse application de partage de document « DropRectangle » (qui est un clin d’œil à « DropBox ») les modifications de structures de la base de données sont plus faciles à mettre en places via la solution NoSQL puisqu’il suffit d’utiliser les « Entity Groups » pour résoudre certains des problèmes de changement de structures que l’on peut rencontrer.

# CONCLUSION :

 Le tableau suivant résume, selon les deux conférenciers, les forces de chaque solution par rapport à l’autre.



Tableau de comparaison Datastore Cloud SQL

 La conclusion de cette conférence est que chaque solution présente ses forces et faiblesses selon l’utilisation que l’on désire en faire. Ainsi, à mon sens, les solutions SQL sont très efficaces et plus efficaces que les solutions NoSQL dans la majeur partie des cas grâce à leur facilité pour accéder aux données via des requêtes, à leur gestion des transactions et à la cohérence des données. Néanmoins l’excellente évolutivité qui assure que le système reste efficace malgré l’augmentation de la volumétrie ou des demandes d’accès aux données, les facilités de management et de gestions des schémas via cette relations clés-valeurs et ce stockage orienté colonne, permet au NoSQL d’être efficace sur les plus grandes volumétries et sur des applications très utilisés mondialement.

 J’estime donc, comme ce que présente les deux conférenciers, que la meilleur solution peu provenir d’une utilisation conjointe de ces deux solutions en utilisant les points forts de l’un et de l’autre comme le montre l’exemple « Greg’s List » (qui est une comparaison effectué avec le site de petites annonces « Craiglist »).

 D’une manière plus général et en réfléchissant en dehors de Google App Engine, Je pense donc que les solutions hybrides utilisant les forces du SQL et du NoSQL combinés vont se multiplier avec le développement des projets NoSQL (qu’ils soient privés ou open-source).