

Veille Technologique



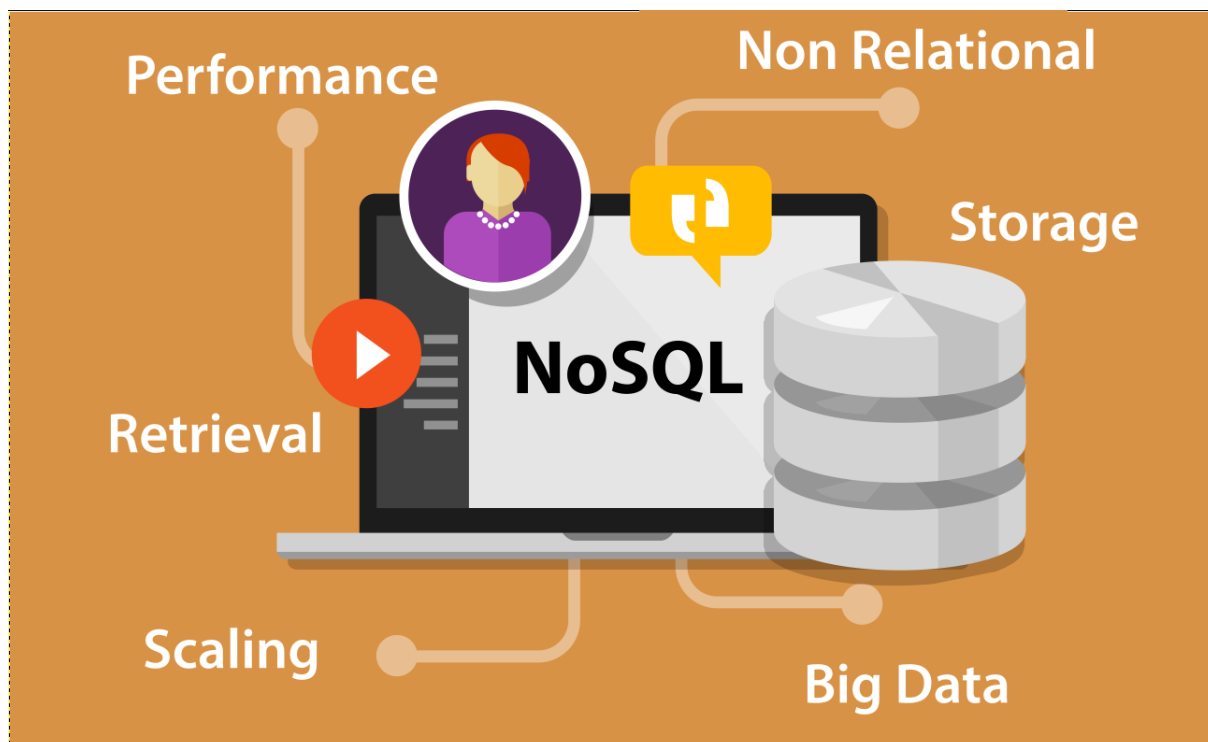
La veille technologique est une activité qui met en œuvre des techniques d'acquisition, de stockage et d'analyse d'informations, concernant un produit ou un procédé, sur l'état de l'art et l'évolution de son environnement scientifique, technique, industriel ou commercial, afin de collecter, organiser, puis analyser et diffuser les informations pertinentes qui vont permettre d'anticiper les évolutions, et qui vont faciliter l'innovation.

Plus simplement, la veille technologique consiste à se tenir au courant de l'actualité afin d'être à même de faire de meilleurs choix, d'innover, de créer...

Pour ma veille technologique j'ai décidé de m'informer principalement sur le NoSql ainsi que d'autres sujets comme les montres connectées et le langage RPG mais ceux-ci sont présents sur mon compte pearltrees.

J'ai utilisé google alert afin de recevoir des alertes/actualités par mail quotidiennement à propos des sujets que j'ai choisis:

A screenshot of the Google Alerts web interface. At the top left is the Google logo. To the right are icons for a grid, a notification bell, and a profile letter 'N'. Below the logo is the word 'Alertes' in a large font, followed by the subtitle 'Recevez des alertes lorsque du contenu susceptible de vous intéresser est publié sur le Web'. A search bar contains the text 'Créer une alerte à propos de...'. Below the search bar is a list titled 'Mes alertes (3)' with a gear icon for settings. The list contains three entries: 'montres connectées', 'rpg ibmi', and 'no sql'. Each entry has a pencil icon for editing and a trash can icon for deleting.



La définition du NoSQL

En informatique et en bases de données, NoSQL désigne une famille de systèmes de gestion de base de données (SGBD) qui s'écarte du paradigme classique des bases relationnelles. L'explicitation du terme la plus populaire de l'acronyme est Not only SQL (« pas seulement SQL » en anglais) même si cette interprétation peut être discutée¹.

La définition exacte de la famille des SGBD NoSQL reste sujette à débat. Le terme se rattache autant à des caractéristiques techniques qu'à une génération historique de SGBD qui a émergé à la fin des années 2000/début des années 2010². D'après Pramod J. Sadalage et Martin Fowler, la raison principale de l'émergence et de l'adoption des SGBD NoSQL serait le développement des centres de données et la nécessité de posséder un paradigme de bases de données adapté à ce modèle d'infrastructure matérielle³.

L'architecture machine en clusters induit une structure logicielle distribuée fonctionnant avec des agrégats répartis sur différents serveurs permettant des accès et modifications concurrentes mais imposant également de remettre en cause de nombreux fondements de l'architecture SGBD relationnelle traditionnelle, notamment les propriétés ACID.

Le No-SQL est une solution de stockage de données indexées qui contrairement à ses prédécesseurs relationnels de type Oracle, Sybase, MySQL etc., ne répond pas à une définition précise. Il fait clairement table rase des anciens standards définis par les bases de données relationnelles.

De fait, le développement parallèle du No-SQL montre bien qu'il s'agit d'une technologie alternative et non pas concurrente. Pour simplifier, No-SQL est un dérivé des moteurs de recherche.

En effet, Google a été confronté très tôt à des problèmes de volumétrie et de performance. Les technologies traditionnelles ne pouvant pas adresser cette problématique, des solutions ad hoc pour le stockage indexé de données sur des clusters répartis ont été développées. Par

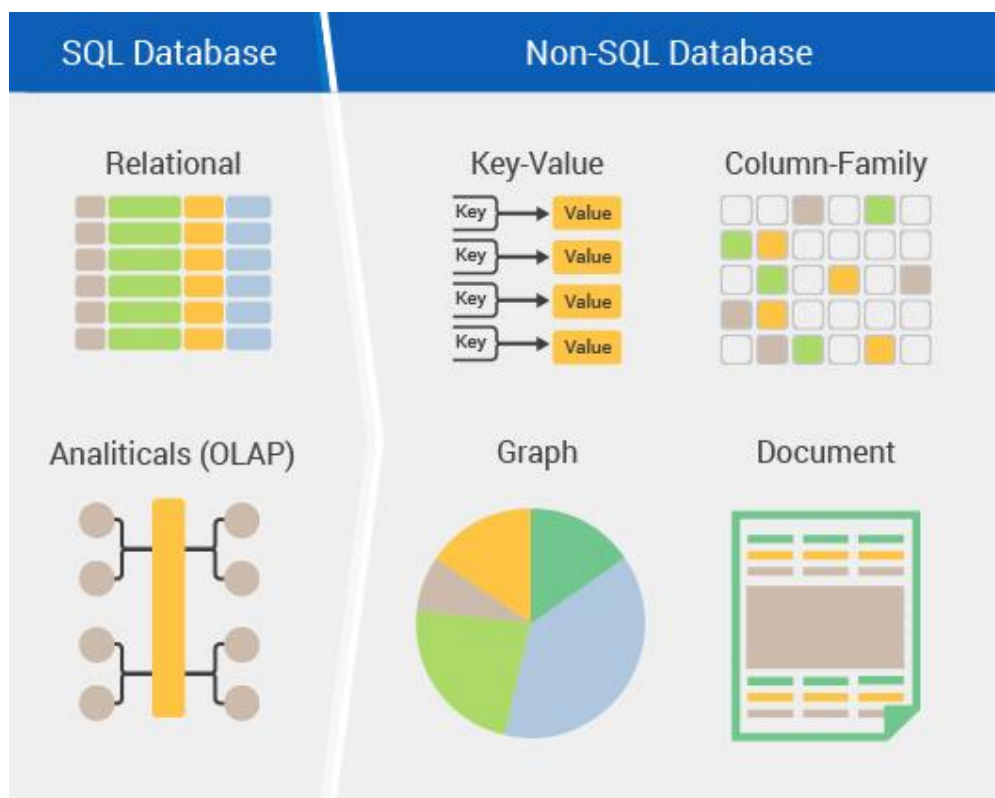
la suite, des frameworks furent créés pour étendirent ce type de stockage à des applications sortant du cadre de la recherche par mot clé.

No-SQL n'a donc aucune parenté avec le SQL (d'où son nom) et ce n'est que récemment que les développements tous azimuts ont proposé une définition moins polémique (Not Only SQL).

Pourquoi le ?



Le NoSQL est apparu afin de contrer la dominance des bases de données relationnelles dans le domaine de l'internet. En effet, un des problème récurrent des bases de données relationnelles est la perte de performance lorsque l'on doit traiter un très gros volume de données. De plus, la multiplication des architectures distribués a apporté le besoin de disposer de solution s'adaptant nativement aux mécanismes de réplication des données et de gestion de la charge.



Les familles de NoSQL

Une remarque qui est revenue régulièrement lors des sessions de cette conférence est que le NoSQL est très souvent réduit au concept de simple association clé-valeur et d'accès put/get. En réalité, chaque famille apporte une forme de représentation des données différente, chacune ayant ses spécificités et simplifiant la manipulation d'un certain type de données.

→ Les bases de données clé-valeur

La représentation en clé-valeur est la plus simple et est très adaptée aux caches ou aux accès rapides aux informations. Elle considère la valeur stockée comme un bloc de données opaque, la base de données étant agnostique de son contenu. Ce postulat permet en général d'atteindre des performances bien supérieures dans la mesure où les lectures et écritures sont réduites à un accès disque simple.

Les implémentations les plus courantes sont [Riak](#), [Redis](#), [Voldemort](#).

L'article [Bases de données clé-valeur et Riak](#) permet d'aller plus loin sur ce sujet.

→ Les bases de données orientées document

La représentation en document est particulièrement adaptée au monde du Web. Il s'agit d'une extension du concept de clé-valeur qui représente la valeur sous la forme d'un document. Un document contient des données organisées de manière hiérarchique à l'image de ce que permettent XML ou JSON. Étant consciente du contenu qu'elle stocke, la base de données peut alors effectuer des indexations de différents champs et offrir des requêtes plus élaborées.

Les implémentations les plus courantes sont [CouchDB](#) et [MongoDB](#).

L'article [Bases de données orientées documents et MongoDB](#) permet d'aller plus loin sur ce sujet.

→ Les bases de données orientées colonnes

La représentation orientée colonnes s'oppose à la représentation des tables dans les bases de données relationnelles. En effet, les SGBDR manipulent les colonnes d'une ligne d'une manière statique. Les bases de données orientées colonnes ont une vision bien plus flexible permettant d'avoir des colonnes différentes pour chaque ligne et de multiplier de manière conséquente le nombre de colonnes par ligne. Il en résulte une capacité à stocker des listes d'informations pour chaque clé, et à accéder à des intervalles de colonnes.

Les implémentations les plus courantes sont [HBase](#) et [Cassandra](#).

L'article [Bases de données orientées colonnes et Cassandra](#) permet d'aller plus loin sur ce sujet.

→ Les bases de données orientées graphe

La représentation en graphe permet la modélisation, le stockage et la manipulation de données complexes liées par des relations non-triviales ou variables. Si le cas d'utilisation classique de ce type base de données est le stockage des informations des

réseaux sociaux, elle est à même de modéliser de nombreuses situations du monde réel qui ne pourraient être représentées que de manière réductrice dans une base de données relationnelle.

Les implémentations les plus courantes sont [Neo4j](#), [HypergraphDB](#) et [FlockDB](#). L'article [Bases de données graphe et Neo4j](#) permet d'aller plus loin sur ce sujet.

→ Les autres bases de données

D'autres types de bases de données existent tels que les bases de données objet, les bases de données hiérarchiques ou encore les *datagrids* que l'on différencie des systèmes de persistance clé-valeur précédemment évoqués. Ils ne sont en général pas évoqués lorsque l'on parle de NoSQL aujourd'hui.

L'adoption du NoSQL

Les projets NoSQL sont très actifs en ce moment. Leur communauté s'élargit et leur nombre de commetteurs augmente. Parallèlement les événements traitant de ce sujet se multiplient, l'intérêt pour ce type de stockage croît peu à peu dans les entreprises et les annonces d'adoption de ces technologies par quelques 'grands du Web' se sont multipliées. Ceci donne aux NoSQL un positionnement peu commun pour des technologies émergentes puisque leur bon fonctionnement dans des conditions extrêmes leur confère une fiabilité déjà reconnue.

Conclusion

Ceci conclut cette synthèse de l'état actuel du monde NoSQL.

Les bases de données NoSQL ne sont donc pas la solution miracle pour répondre à toutes les problématiques de stockage sur le web ou ailleurs. Il est surtout très important de bien comprendre ce que le choix d'une base de données de ce type va avoir comme conséquences en termes d'architecture logicielle et complexité de développement.

Quel avenir pour le NoSQL ?

A première vue, le principal désavantage du NoSQL est sa jeunesse. Comparé aux systèmes matures et connus que sont les bases relationnelles, il ne fait pas le poids. Les outils de supervision ne sont pas très développés pour le moment et cela peut constituer un frein à une mise en production sur des applications critiques. Le NoSQL a aussi une image de système à déployer uniquement sur des applications nécessitant d'être très performante et travaillant sur de gros volumes. Sorte de Rolls-Royce des bases de données que seules de grosses compagnies peuvent s'offrir. Malgré cela, le NoSQL reste une technologie attrayante. D'une part elle apporte de l'agilité au niveau du développement et de la maintenance. D'autre part, elle remet au goût du jour des solutions qui avaient été éclipsées par l'avènement des bases relationnelles.

Il ne s'agit pas non plus de tout basculer vers le NoSQL mais plutôt de s'orienter vers des solutions hybrides en couplant une base relationnelle avec une base NoSQL.

Nourhene Chettaoui.