

VALTECH DAYS

17 MARS 2011 - PARIS

NoSQL : Le nuage souffle un nouvel « R » sur les SGBD

Claude Falguière

Senior Consultante
Valtech Paris

Herve Desauois

Directeur Opérationnel
Valtech Toulouse



Sommaire

Partie 1

Contexte et enjeux



Partie 2

Le tour du monde



Partie 3

Les 10 commandements



Sommaire

Partie 1 **Contexte et enjeux**



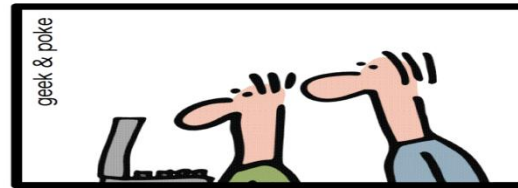
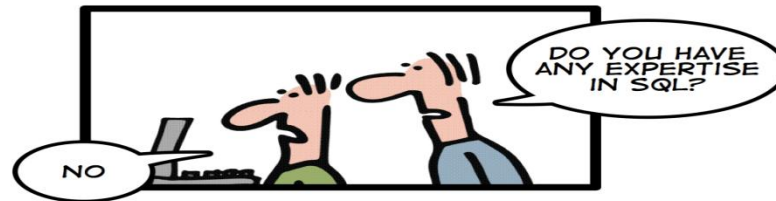
Partie 3 **Les 10 commandements**

Partie 2 **Le tour du monde**



NoSQL ?

HOW TO WRITE A CV



Leverage the NoSQL boom

NoSQL "Not only SQL"

- **Technologie relative aux bases de données**
Alternatives au SGBD relationnels pour gérer de gros volumes
- **Émergence printemps 2009 avec le Cloud Computing et le Web 2.0**
BigTable Google, Dynamo Amazon, Cassandra Facebook, Twitter
- **De nombreuses solutions Open Source**
Cassandra, CouchDB, MongoDB, Riak, Neo4J



L'écosystème NoSQL

Document

Colonne

Clé Valeur

Graphe



Return On Invest



Coût faible



Gros Volume

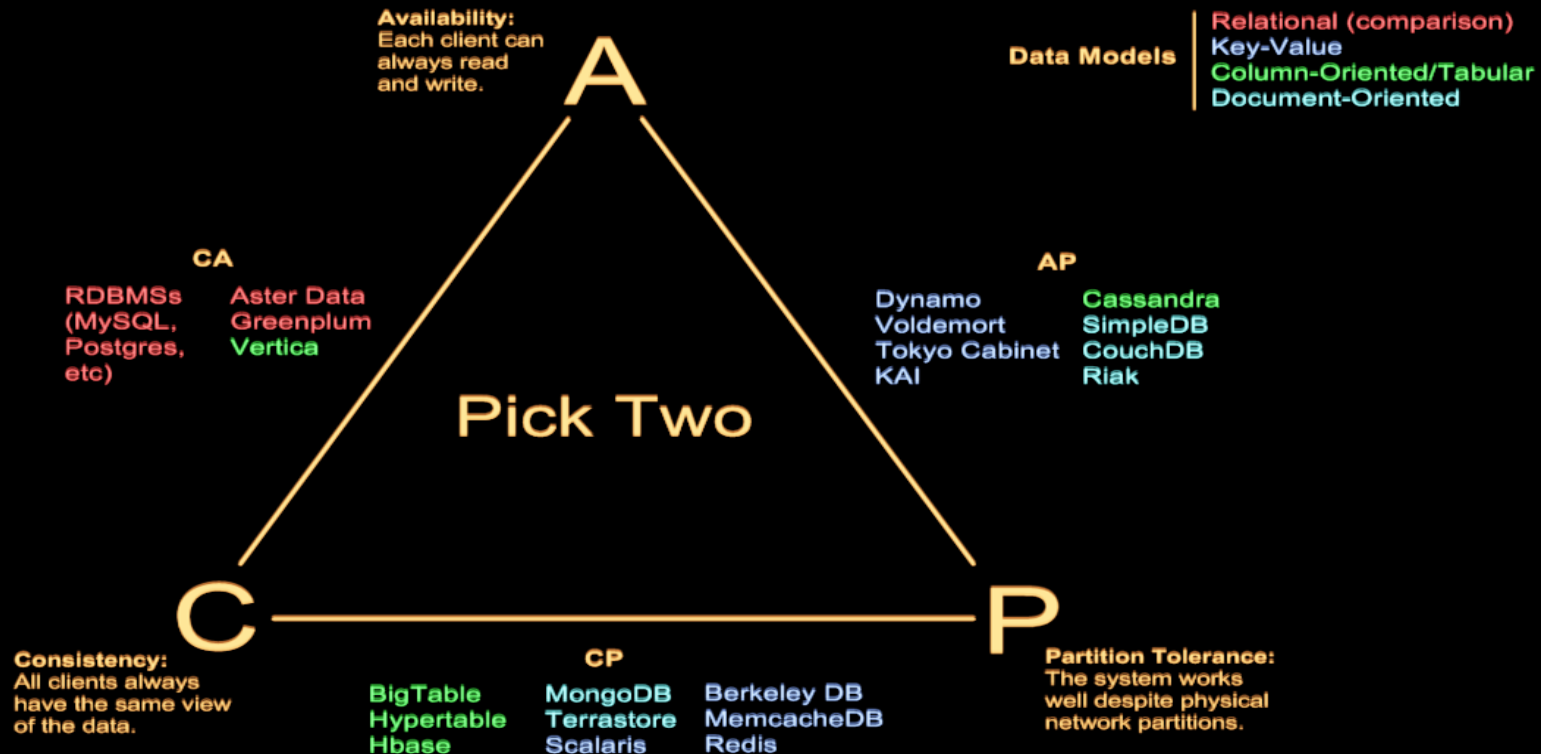


Haute Disponibilité

Alternatives intéressante au solution de type RAC
(Real Application Cluster)

Le CAP Theorem (Eric Brewer)

Visual Guide to NoSQL Systems

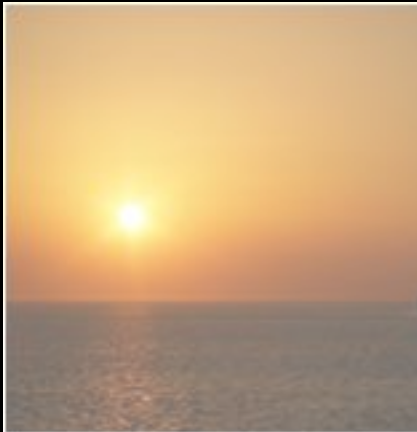


Sommaire

Partie 2 Le tour du monde

Partie 1 Contexte et en

Partie 3 commandements



Les bases Clé Valeur



Implémentations très nombreuses

Structure de données très simple

- Map

Clé	Valeur
Clé	Valeur
Clé	Valeur

Base simple à créer

Redis, BigTable, Voldemort, MemcacheDB ...

Les bases Colonnes



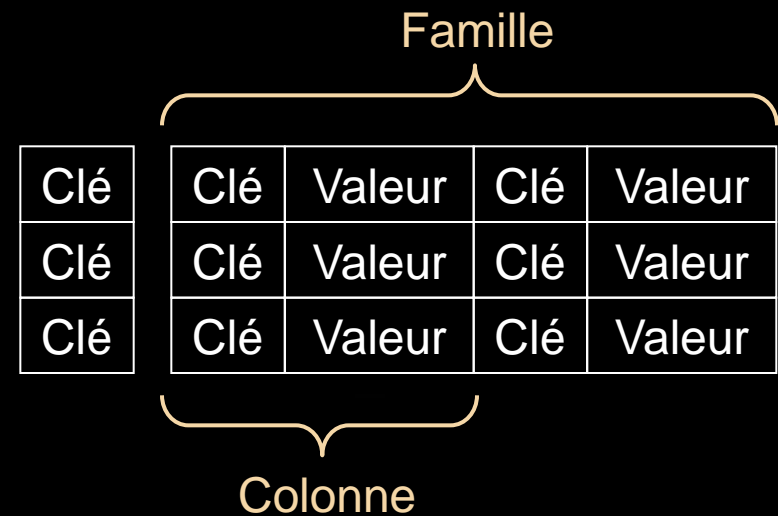
Une table est définie par des **familles** de colonnes

- Chaque famille peut avoir un nombre quelconque de colonnes
- Les colonnes sont représentées par des couples clés-valeur

Optimisé pour l'accès par colonne

Représentation plus **flexible**

- One to many
- Grand nombre de colonnes
- Sparse data



HBase, Cassandra

Les bases Document



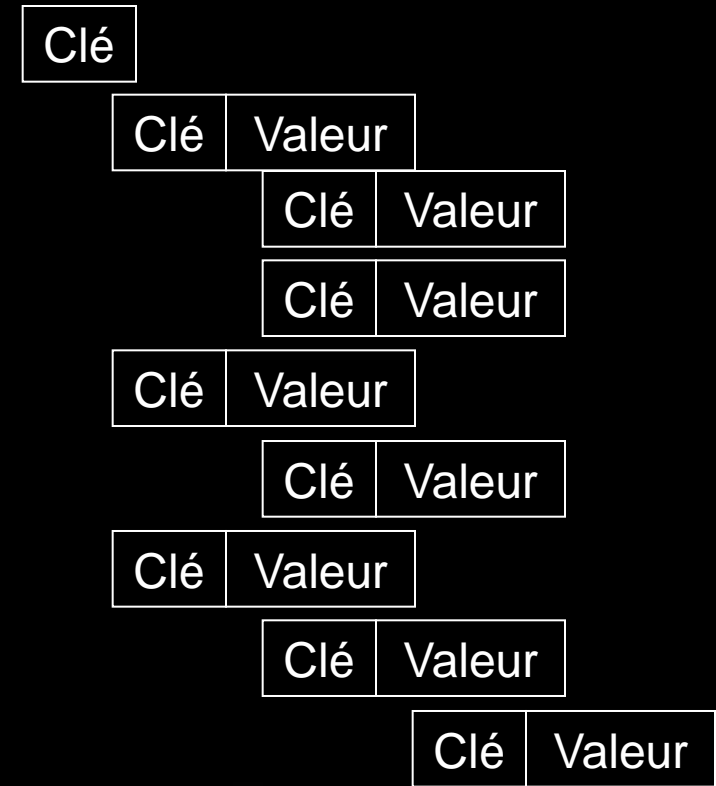
La clé correspond à un document soit XML soit JSON

Retrouver avec une seule clé un ensemble d'informations structurées de manière hiérarchique

- L'utilisateur, ses statuts, ses amis

L'équivalent en relationnel impliquerait beaucoup de jointures

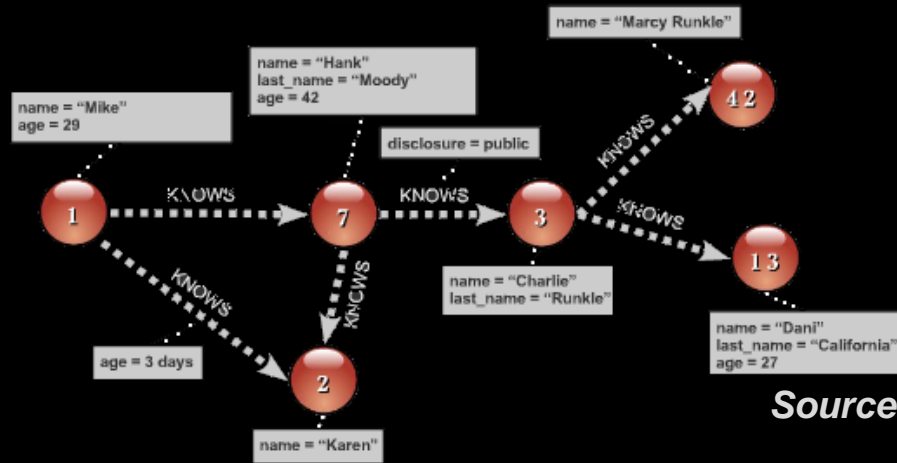
MongoDB, CouchDB



Les bases Graphe



Reposent sur la notion
de nœuds
et de relations
et de propriétés



Source: neo4J

Traitement des données de réseaux sociaux

- l'utilisateur, ses amis, les amis de ses amis

En phase avec les outils du web sémantique (RDF, SparQL)

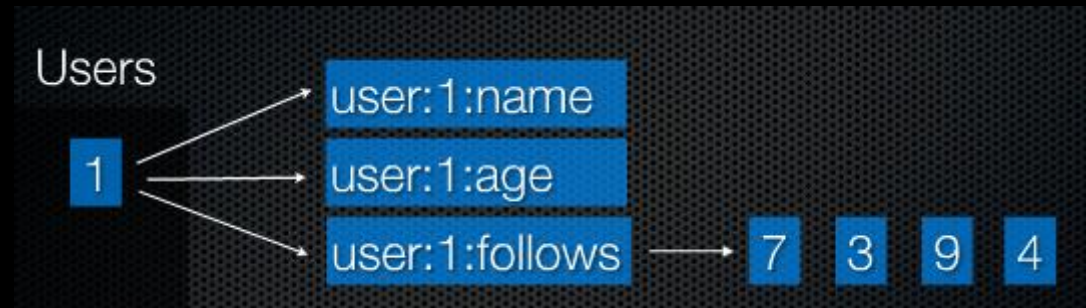
Neo4J

Use case : accès rapide



Redis à Skyrock

Calculer en temps réel les amis connectés



20h30 : [1,4,6,9,10,15,17,25]

20h29 : [1,6,9,10,15,17,25]

20h28 : [4,6,9,10,13,15,25]

20h27 : [1,4,6,9,10,13,18,25]

20h26 : [1,4,6,9,10,18,20]

UNION : [10,1,20,4,13,15,6,18,9,17,25]

follows : [7, 3, 9, 4]

INTER : [9, 4]

Share Nothing, pas de transactionnel

Distribution

Partionnement / Sharding

Replication

Versionning

Pas de langage de requête, pas de jointures

Interface de type get/put

- `put user:1:name claud`
- `get user:1:name`
- `list user*:name`

Les valeurs sont des tableaux d'octets ou des types prédéfinis

API propriétaires

Bindings dans différents langages

Apache Thrift, Avro, parfois JPA ou JDO

Des compléments SQL Like : Pig, Hive

Use case : Données massives, parallélisation



Gestion des logs chez Twitter

12 millions de tweets / jour

7 To de logs par jour

Double plusieurs fois par an

Saturation des systèmes de gestion de logs traditionnels

- Pertes de logs
- Perte d'information



7 disques standards

14 disques de laptop

**24,3h d'écriture sur
un disque à 80Mo/s**

Analytics des tweets chez Twitter

Quoi ?

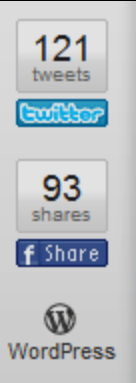
- Compte les actions sur les URL (clic, favoris, retweet)
- Comptabilise par annonceur/campagne/tweet_id

A quoi ça sert ?

- Cette url a été retweeté 121 fois (en temps réel)
- Les analytics des promoted tweets
- Le monitoring interne de cette activité

Exigences

- 100 000 Write/s
- 10 000 Read/s
- >100 To
- < 100ms



121 tweets

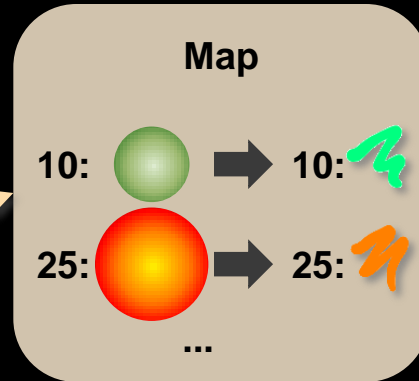
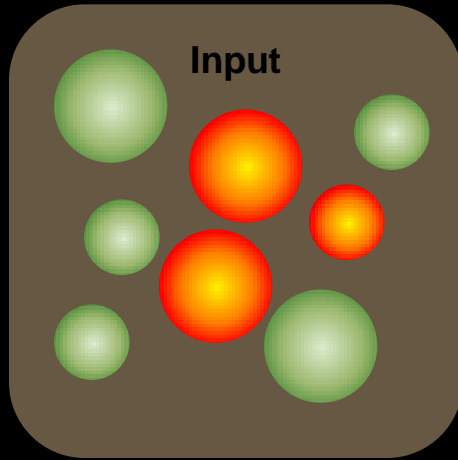
93 shares

NoSQL

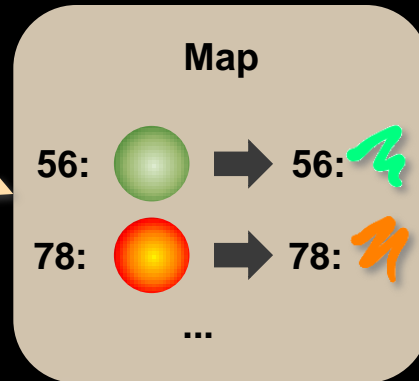
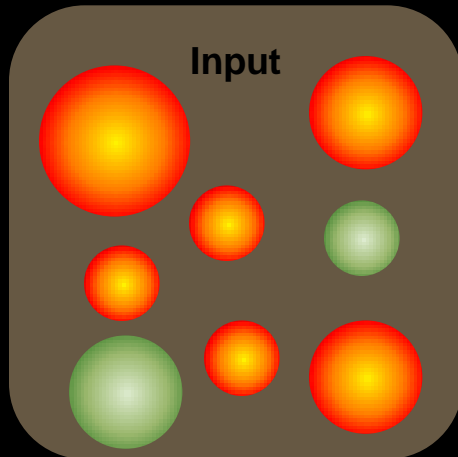
Kevin Weil -- @kevinweil
Analytics Lead, Twitter

WordPress

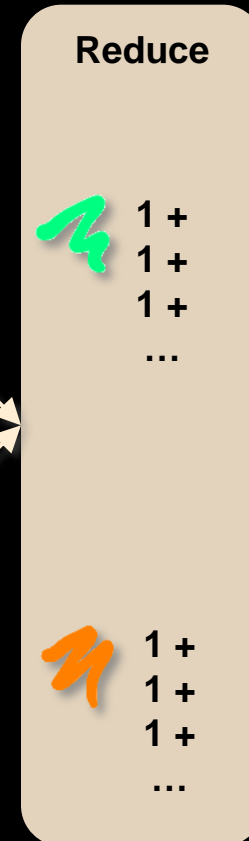
Map Reduce



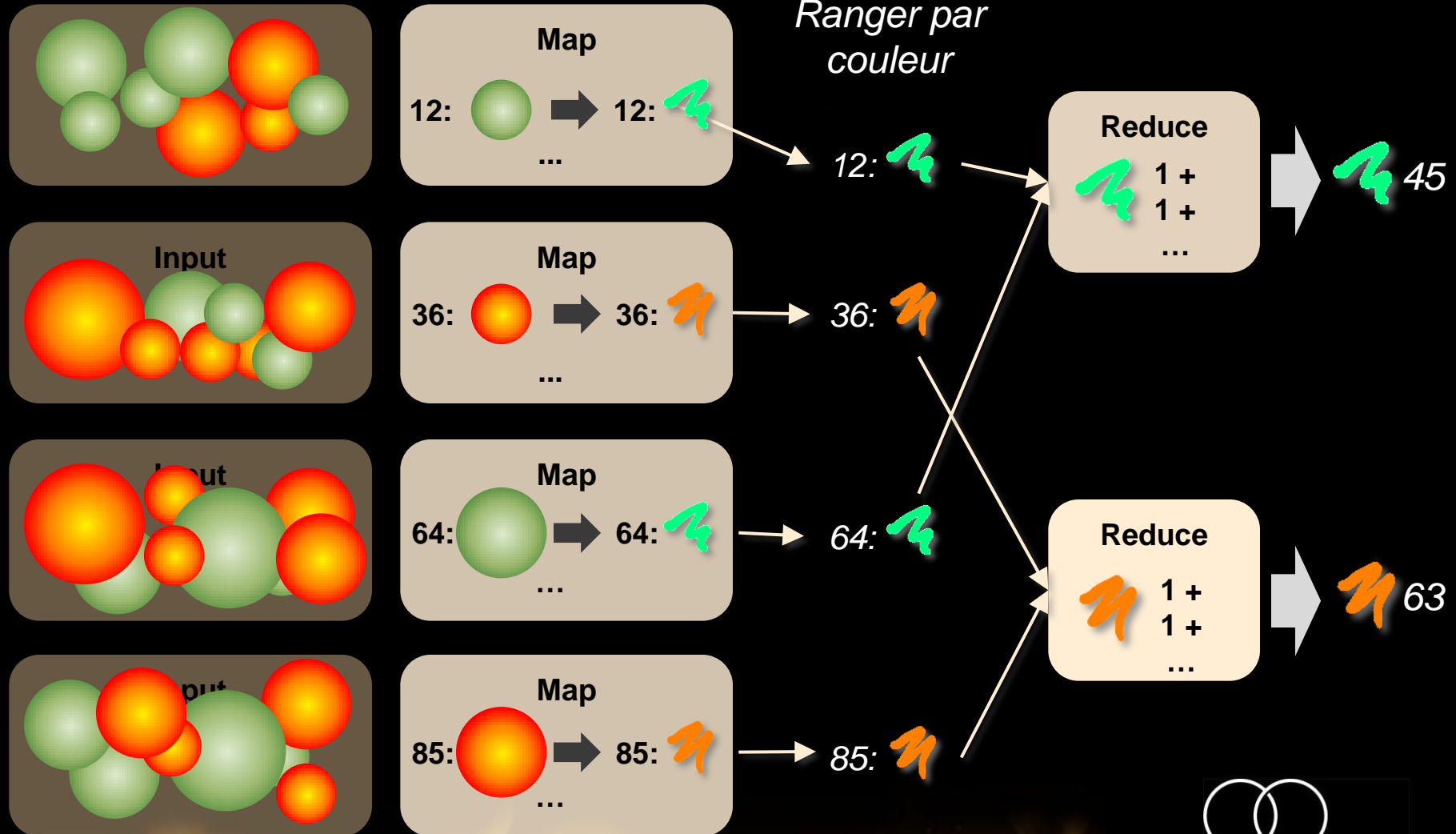
Map function :
couleur



Reduce function :
Compter par couleur



Map Reduce : Scalabilité



Map Reduce : Combien de tweets par user_id ?

Map :

- Déterminer le user_id d'un tweet
- Réparti sur les nœuds où sont stockés les tweets

Reduce :

- Totaliser par user_id
- Centralisé ou réparti par range de user_id

Très adapté aux langages fonctionnels :

- Support natif du Map/Reduce
- `collect { it.getArea() }`
- `groupBy { it.color }.sum()`

Use Case : Facilité d'utilisation



User history chez bit.ly

Bit.ly

- 50 Millions d'utilisateurs
- 10 000 d'accès simultanés
- 1,25 milliards de raccourcis par mois (1000/s pic)

Stockage des user history dans MongoDB

- url raccourcies, leurs stats, les accès

Les bases Documents

Facile à utiliser

- Manipulation de collections d'objets
- Bien intégré aux langages fonctionnels
- Accès en REST

Rapide

- Pas de jointure pour retrouver les données dépendantes
- Stocker une portion de la base sur le client

Flexible

- Evolutivité du schéma

Use case : utiliser le modèle de données adapté



Twitter : activité des amis

12 millions de tweets

- Autojointure sur n millions de lignes ?
- Compter les tweets avec un count(*) sur des millions de lignes ?

FlockDB pour reconstruire le réseau social

Les bases Graphe

Représentation native des réseaux de personnes, d'équipement, de concepts ...

Manipulation d'objets

- Les noeuds et les relations sont des objets

Langages des requêtes : SparQL, Gremlin

Souvent transactionnelles

Réplication mais montent un peu moins en charge

- Jusqu'à 1 milliard de noeuds/relation/propriété

Sommaire

Partie 1

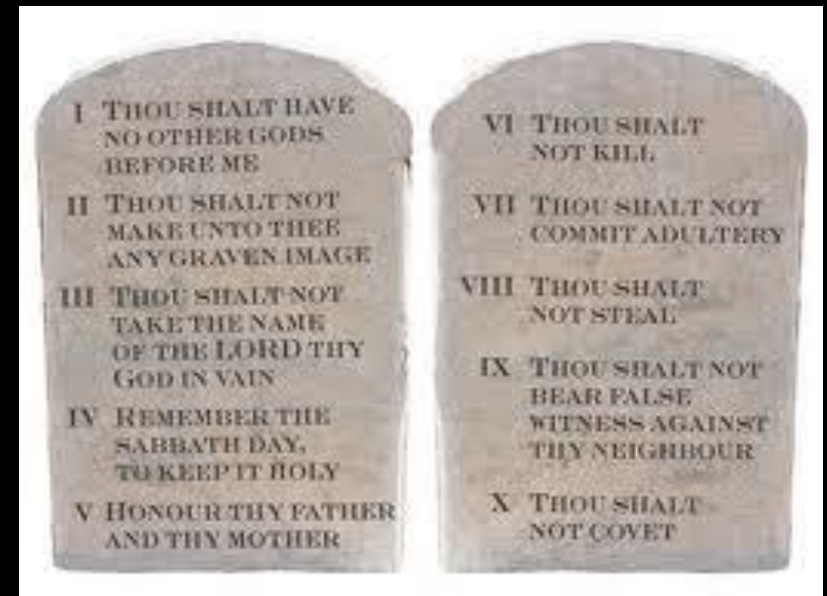
Contexte et enjeux

Partie 2

Le tour du monde

Partie 3

Les 10 commandements



Les dix commandements

- 1 - Solution très bonnes pour les gros volumes et la haute disponibilité
Cassandra, Google Tables, Amazon Tables , Azure Tables,...
- 2 - Solution très bonnes pour la distribution de données
Map Reduce, Cloud DB, Web 2.0
- 3 - Plus de souplesse des schémas
Solution Orienté Colonne



Les dix commandements

4 - NoSQL ne remplace pas les systèmes classiques

Oracle, MySQL, DB2, SQL Server,

5 - Pas de standards

Pas de langage commun , Pas d'outils de type ORM,
Outil d'administration faible

6 - Attention à la sécurité sur les bases NoSQL

Pas d'accès restreint



Les dix commandements

7 - Pas de transaction

Des problématiques qui remontent au niveau Applicatif , Perte d'acidité ,

8 – Performance très bonne

N'implémentent que les mécanismes dont elles ont besoin

9 - Coûts très attractifs et très bonne réputation en production

Cassandra, BigTable, Dynamo,



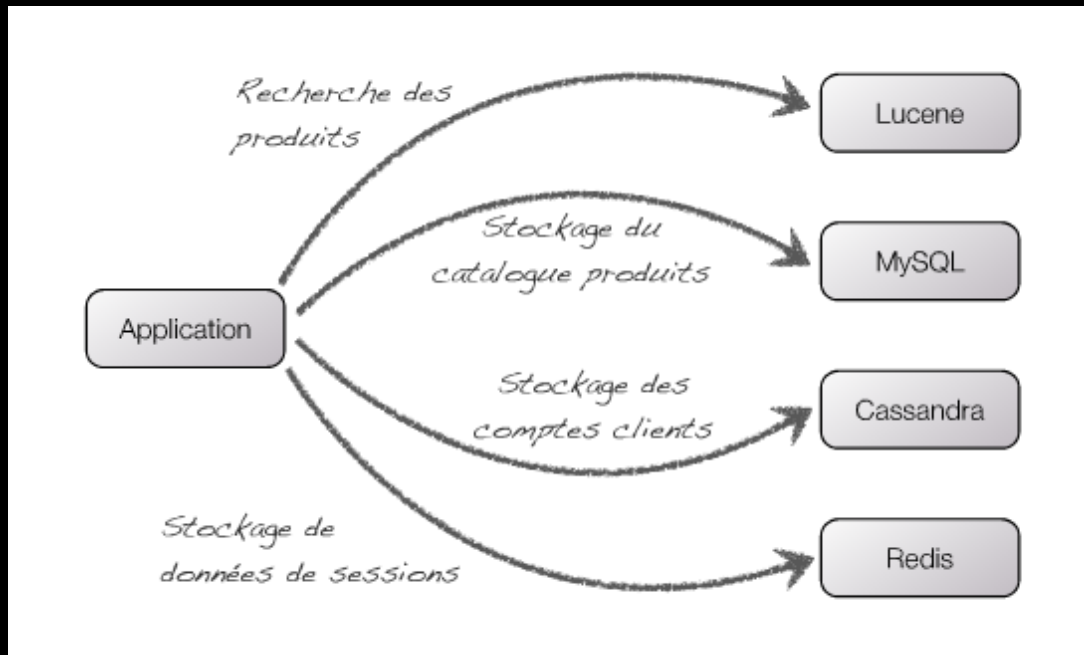
Les dix commandements

10 - Avenir incertain du mouvement NoSQL

Apparition d'ORM, prise en charge de la sécurité, Architecture Hybride, CQRS (« Command and Query Responsibility Segregation ».) , Sharding with SQL Azure

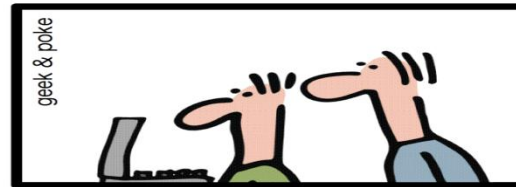
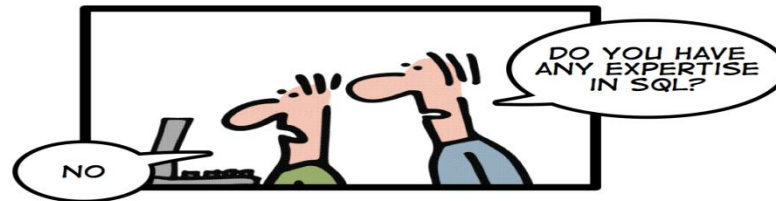


Solution Architecture Hybride



No SQL, No Relationnel

HOW TO WRITE A CV



Leverage the NoSQL boom

SGBD No Relationnel



Droits de reproduction

- Vous êtes libres de :

- Partager : reproduire, distribuer et communiquer cette présentation
- Remixer : modifier cette présentation

- Selon la condition de « Paternité » :

Vous devez impérativement citer le(s) auteur(s) ou le(s) titulaire(s) des droits (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation du contenu).

- Plus d'informations : <http://fr.creativecommons.org>

