

STIC-B-415 : La mouvance NoSQL

Laurent Contzen

9 décembre 2011

Historique

Le modèle relationnel

ACID et CAP

Différents cas d'utilisation

Les différents types de bases de données NoSQL

Key-Value

Document-oriented

Columns-oriented

Graph-oriented

Conclusion

Historique

Historique

- Les bases de données : un besoin fondamental

Historique

- ▶ Les bases de données : un besoin fondamental
- ▶ Première standardisation : Codasyl Approach

Historique

- ▶ Les bases de données : un besoin fondamental
- ▶ Première standardisation : Codasyl Approach
- ▶ Le modèle relationnel

Historique

- ▶ Les bases de données : un besoin fondamental
- ▶ Première standardisation : Codasyl Approach
- ▶ Le modèle relationnel
- ▶ Apparition de la mouvance NoSQL

Le modèle relationnel

Le modèle relationnel : Structure

- Données organisées sous forme de tables

Le modèle relationnel : Structure

- ▶ Données organisées sous forme de tables
- ▶ Schéma défini à l'avance

Le modèle relationnel : Structure

- ▶ Données organisées sous forme de tables
- ▶ Schéma défini à l'avance
- ▶ Requêtes ou transactions en langage SQL

Le modèle relationnel : Structure

- ▶ Données organisées sous forme de tables
- ▶ Schéma défini à l'avance
- ▶ Requêtes ou transactions en langage SQL
- ▶ Très formalisé

Le modèle relationnel : Exemple de table

Characters			
<u>Id</u>	Name	Show	Actor
1	Jackson Teller	Sons of Anarchy	Charlie Hunnam
2	John Dorian	Scrubs	Zach Braff
3	Bill Adama	Battlestar Galactica	Edward J. Olmos
4	Kara Thrace	Battlestar Galactica	Katee Sackhoff
5	Tyrion Lannister	Game of Thrones	Peter Dinklage
6	Ted Mosby	How I Met Your Mother	Josh Radnor
7	Seth Bullock	Deadwood	Timothy Oliphant
8	Tobias Beecher	Oz	Lee Tergesen
9	Emily Sullivan	Jericho	Ashley Scott
10	Jimmy McNulty	The Wire	Dominic West

Le modèle relationnel : Exemples de requêtes SQL

► `SELECT * FROM Characters ;`

Le modèle relationnel : Exemples de requêtes SQL

- ▶ `SELECT * FROM Characters ;`
- ▶ `SELECT * FROM Characters WHERE Show="Battlestar Galactica" ;`

Le modèle relationnel : Exemples de requêtes SQL

- ▶ `SELECT * FROM Characters ;`
- ▶ `SELECT * FROM Characters WHERE Show="Battlestar Galactica" ;`
- ▶ `SELECT Actor FROM Characters WHERE Name="Tyrion Lannister" AND Show="Game of Thrones" ;`

Le modèle relationnel : Caractéristiques

- Nécessite beaucoup de rigueur

Le modèle relationnel : Caractéristiques

- ▶ Nécessite beaucoup de rigueur
- ▶ Difficile de changer la structure une fois en utilisation

Le modèle relationnel : Caractéristiques

- ▶ Nécessite beaucoup de rigueur
- ▶ Difficile de changer la structure une fois en utilisation
- ▶ Difficile à distribuer

Le modèle relationnel : Caractéristiques

- ▶ Nécessite beaucoup de rigueur
- ▶ Difficile de changer la structure une fois en utilisation
- ▶ Difficile à distribuer
- ▶ Très peu de possibilités de redimensionnement

ACID et CAP

ACID

- ▶ Atomicity : Réussite ou échec pour une transaction

ACID

- ▶ Atomicity : Réussite ou échec pour une transaction
- ▶ Consistency : Bases de données toujours dans un état correct

ACID

- ▶ Atomicity : Réussite ou échec pour une transaction
- ▶ Consistency : Bases de données toujours dans un état correct
- ▶ Isolation : Transactions indépendantes et non simultanées

ACID

- ▶ Atomicity : Réussite ou échec pour une transaction
- ▶ Consistency : Bases de données toujours dans un état correct
- ▶ Isolation : Transactions indépendantes et non simultanées
- ▶ Durability : Pérenité des modifications

CAP

- ▶ Théorème CAP : Impossibilité de garantir les trois principes suivants pour un système à données partagées

CAP

- ▶ Théorème CAP : Impossibilité de garantir les trois principes suivants pour un système à données partagées
- ▶ Consistency : Toujours accès à la dernière version de l'information

CAP

- ▶ Théorème CAP : Impossibilité de garantir les trois principes suivants pour un système à données partagées
- ▶ Consistency : Toujours accès à la dernière version de l'information
- ▶ Availability : La base de données est toujours en service et réponds toujours aux requêtes.

CAP

- ▶ Théorème CAP : Impossibilité de garantir les trois principes suivants pour un système à données partagées
- ▶ Consistency : Toujours accès à la dernière version de l'information
- ▶ Availability : La base de données est toujours en service et réponds toujours aux requêtes.
- ▶ Partition tolerance : Entièrement des données accessible même si une partie des serveurs tombe.

Différents cas d'utilisation

Différents cas d'utilisation

- Système bancaire

Différents cas d'utilisation

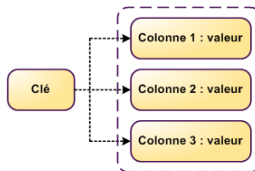
- ▶ Système bancaire
- ▶ Réseaux Sociaux

Les différents types de bases de données NoSQL

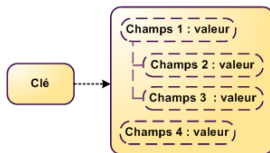
Les différents types de bases de données NoSQL



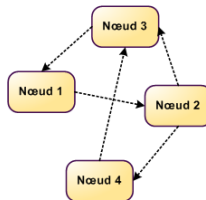
BDD Clé-Valeur



BDD Orientée colonnes



BDD Orientée document



BDD Orientée graphe

Key-Value



Les bases de données de type Key-Value

- ▶ Grandes tables de hashage

Les bases de données de type Key-Value

- ▶ Grandes tables de hashage
- ▶ Lectures ou écritures à partir d'un identifiant

Les bases de données de type Key-Value

- ▶ Grandes tables de hashage
- ▶ Lectures ou écritures à partir d'un identifiant
- ▶ Données en bloc binaire

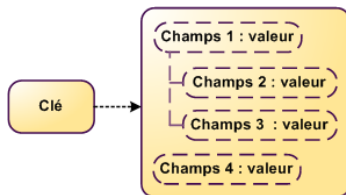
Les bases de données de type Key-Value

- ▶ Grandes tables de hashage
- ▶ Lectures ou écritures à partir d'un identifiant
- ▶ Données en bloc binaire
- ▶ Table partitionnée répliquée

Les bases de données de type Key-Value

- ▶ Grandes tables de hashage
- ▶ Lectures ou écritures à partir d'un identifiant
- ▶ Données en bloc binaire
- ▶ Table partitionnée répliquée
- ▶ Taux de consistance souhaité

Document-oriented



Les bases de données orientées document

- Extension du modèle Key-Value

Les bases de données orientées document

- ▶ Extension du modèle Key-Value
- ▶ Données sous forme de document structuré

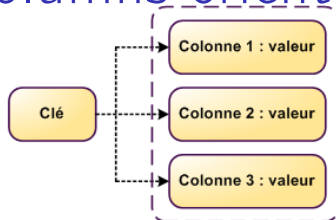
Les bases de données orientées document

- ▶ Extension du modèle Key-Value
- ▶ Données sous forme de document structuré
- ▶ Connaissance du contenu du document

Les bases de données orientées document

- ▶ Extension du modèle Key-Value
- ▶ Données sous forme de document structuré
- ▶ Connaissance du contenu du document
- ▶ Possibilité de requêtes plus élaborées

Columns-oriented



Les bases de données orientées colonnes

- Extension du modèle Key-Value avec des principes du modèle relationnel

Les bases de données orientées colonnes

- ▶ Extension du modèle Key-Value avec des principes du modèle relationnel
- ▶ Pour chaque clé, données sous forme de colonnes contenant les informations

Les bases de données orientées colonnes

- ▶ Extension du modèle Key-Value avec des principes du modèle relationnel
- ▶ Pour chaque clé, données sous forme de colonnes contenant les informations
- ▶ Uniquement les colonnes utiles pour chaque clé

	A	B	C	D	E
1	foo	bar	hello		
2		Tom			
3			java	scala	cobol

Organisation d'une table dans
une BDD relationnelle

1	A foo	B bar	C hello
2	B Tom		
3	C java	D scala	E cobol

Organisation d'une table dans
une BDD orientée colonnes

Les bases de données orientées colonnes

- ▶ Extension du modèle Key-Value avec des principes du modèle relationnel
- ▶ Pour chaque clé, données sous forme de colonnes contenant les informations
- ▶ Uniquement les colonnes utiles pour chaque clé

	A	B	C	D	E
1	foo	bar	hello		
2		Tom			
3			java	scala	cobol

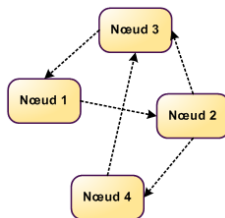
Organisation d'une table dans
une BDD relationnelle

1	A foo	B bar	C hello
2	B Tom		
3	C java	D scala	E cobol

Organisation d'une table dans
une BDD orientée colonnes

- ▶ Méta-colonnes dans Cassandra

Graph-oriented



Les bases de données orientées graphes

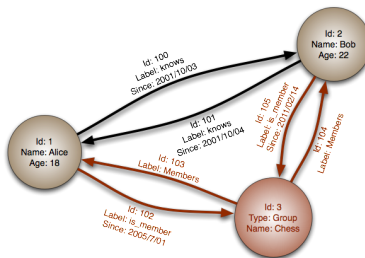
- Différentes des autres modèles NoSQL

Les bases de données orientées graphes

- ▶ Différentes des autres modèles NoSQL
- ▶ Basées sur la théorie des graphes

Les bases de données orientées graphes

- ▶ Différentes des autres modèles NoSQL
- ▶ Basées sur la théorie des graphes
- ▶ Noeuds contenant des propriétés et arcs labellisés



Conclusion