

NoSQL

Module: 165

Utiliser des bases de données NoSQL

Base de données NoSQL

Objectifs

- 1.2 Connaître le théorème CAP
- 1.3 Connaître le modèle de cohérence BASE et ses différences par rapport au modèle ACID

Base de données

Modèle de cohérence

ACID & BASE

Base de données

ACID

Quand utilise-t-on le **SQL**?

- Les données doivent être structurées. L'organisation est connue (ou pourrait être connue) d'avance.
- **L'intégrité des données doit-être respectée.**
- Les transactions sont importantes. **Le principe ACID est important.**
- Faire des requêtes complexes avec un langage de haut niveau (SQL)

A - Atomicité

C - Cohérence

I - Isolation

D - Durabilité

Base de données

ACID

- | | |
|-----------------------|--|
| A - Atomicité | Une transaction s'effectue entièrement ou pas du tout |
| C - Cohérence | Le contenu d'une base doit être cohérent au début et à la fin d'une transaction. |
| I - Isolation | Les modifications d'une transaction ne sont visibles/modifiables que quand celle-ci a été validée |
| D - Durabilité | Une fois la transaction validée, l'état de la base est permanent (non affecté par les pannes ou autre) |

Base de données

BASE

Quand utilise-t-on le **NoSQL**?

- La structure de données n'est pas importante. Évolutive et pas (ou peu) connue d'avance.
- Gestion de beaucoup de données structurées, et non structurée.

BA - Basic Available

S - Soft-state

E - Eventually consistent

Base de données

BASE

BA - Basic Available

Quelle que soit la charge de la base de données, le système garantit la disponibilité des données.

S - Soft-state

La base peut changer lors des mises à jour ou lors d'ajout/suppression de serveurs. La base NoSQL n'a pas à être cohérente à tout instant.

E - Eventually consistent

À terme, la base atteindra un état cohérent.

Base de données

Le théorème de CAP

Base de données

Théorème de CAP

Théorème de Brewer dit « **théorème de CAP, 2000** »: Indique qu'il est impossible, pour un système distribué, de garantir en même temps les trois contraintes suivantes:

Cohérence (**C**onsistency): Tous les noeuds du système voient les mêmes données au même moment.

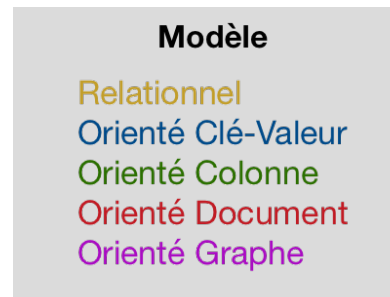
Disponibilité (**A**vailability): Toutes les requêtes reçoivent une réponse.

Tolérance au partitionnement (**P**artition Tolerance) : Aucune panne ne doit empêcher le système de répondre correctement (sauf une coupure complète du réseau).

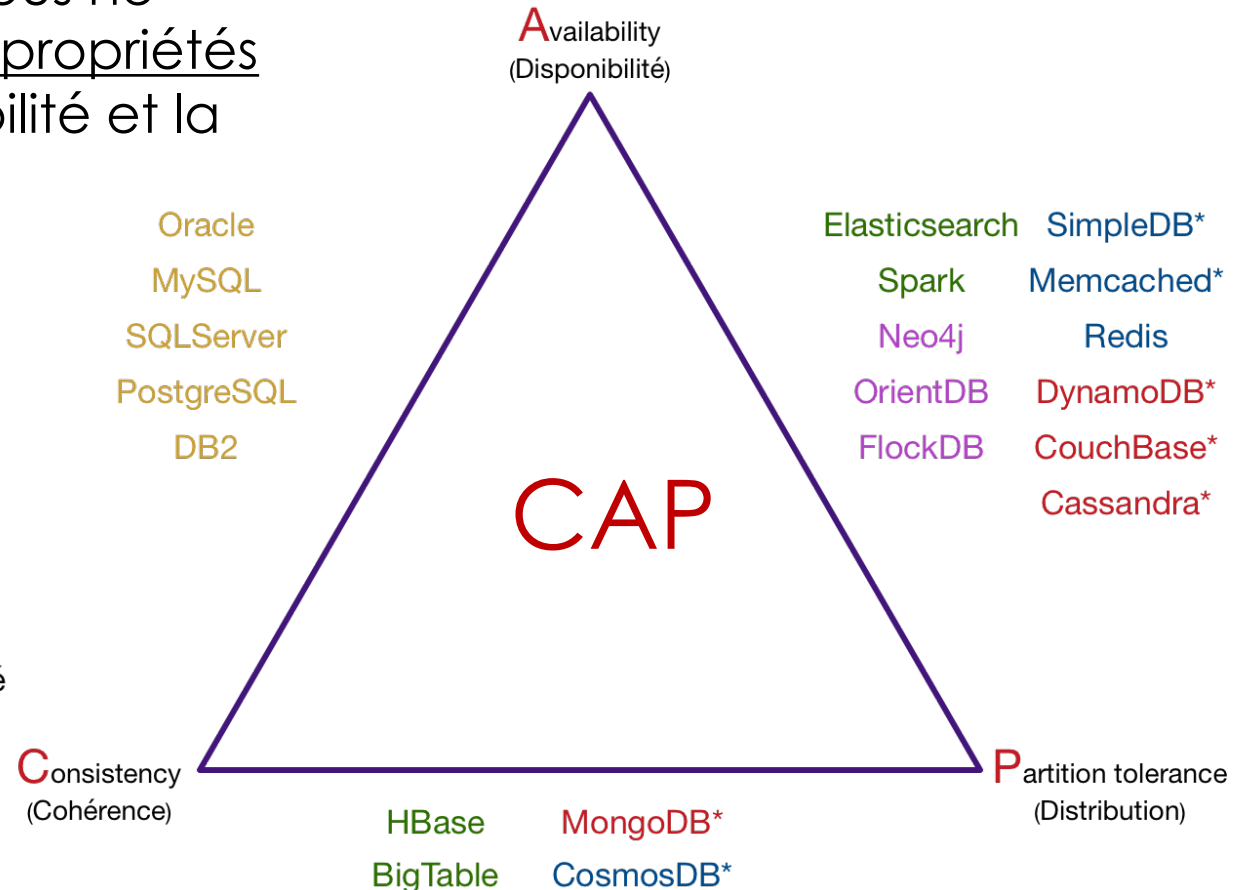
Base de données

Théorème de CAP

Dans toute base de données, vous ne pouvez respecter au plus que 2 propriétés parmi la **C**ohérence, la **D**isponibilité et la **D**istribution.



* Possibilité de changer
le mode de cohérence
Cohérence <-> Disponibilité



Base de données

SQL ou NoSQL ?

Base de données

SQL ou NoSQL ?

Les bases de données relationnelles privilégient la disponibilité des données et leur cohérence, mais de un LA propriété ACID est très importante.

Les bases de données NoSQL tendent à privilégier la disponibilité et la tolérance au partitionnement.

Les systèmes NoSQL fonctionnent sur ce principe : Les données et l'état de la base de données seront **éventuellement cohérents** et consistants. L'important est que l'accès soit toujours permis.

Base de données

SQL ou NoSQL ?

Finalement, la différence qui existe entre une base de données relationnelle et une base de données non relationnelle est la façon de stocker.

L'une stocke les données dans des tables tandis que l'autre les stockent au format clé-valeur de manière à stocker davantage en termes de quantité.

Les bases de données NoSQL ne sont pas en train de supplanter les bases relationnelles mais viennent les compléter, dans un contexte de croissance exponentielle de données. (Big Data).

Base de données

Avantage et inconvénients NoSQL

Avantages:

- Permet de gérer rapidement des tonnes de données (grand volume à une vitesse rapide).
- Schémas dynamiques pour les données non structurées (évolutifs, n'a pas à être connu d'avance).
- Plusieurs façons de stocker des données.
- Moins coûteux (ajout de serveurs).

Inconvénients:

- La cohérence des données n'est pas garantie.
- Pas de langage de requête abstrait partagé, donc un travail de programmation spécifique plus important.

Base de données NoSQL

En résumé

