



# Administration des SGBD

Bases de données NoSQL : ACID & CAP

M1 Informatique

Damien Ploix

# No ? SQL ?

- Problématique :
  - Traitement d'un volume « très » important de données
  - Traitement de données « non structurées »
- Obtenu par la réduction des contraintes des 4 "no" :
  - NO SCHEMA
    - Le schéma n'est pas figé
  - NO JOIN
    - L'extraction des données est réalisée sans jointure
  - NO DATA FORMAT (graph, document, row/column, key/val)
    - Pas de typage des données (graphes, documents, lignes/colonnes, clés/valeur)
  - NO ACID Transactions
    - Pas de respect de l'ensemble des contraintes ACID

# *Fil conducteur*

- Administration du SGBD d'un système de réservation de salles de réunion RoomFree permettant le partage de toutes les salles de réunions disponibles dans toutes les tours de la Défense.
- SGBD Relationnel : Oracle
- SGBD « BigData » : MongoDB
- Traitement principal effectué : recherche et réservation d'une salle (localisation (bâtiment, étage, numéro de salle), créneau (heure début, heure fin), participants (abonné, invités, dont le nom et le prénom))
- Réflexion (commune) : SLA ?
  - D
  - I
  - C
  - T

# No ACID ?

- Définition ACID
  - Atomic : soit toute la transaction succède, soit l'intégralité est rejetée.
  - Consistant : une transaction ne peut pas laisser la base de donnée dans un état incohérent : les dépendances fonctionnelles sont vérifiées via la vérification constante des CIF (cf les formes normales).
  - Isolée : chaque transaction est isolée des autres : mise en place de verrous et d'attentes (pb de l'interblocage)
  - Durable : la transaction persiste même après le redémarrage du serveur (redo log).

***Les contraintes "ACID" s'appliquent sur les données applicatives gérées par la base de donnée.***

# ACID

- A pour "Atomic" :
  - Une modification "atomique" se définit par une série de modifications (SQL) qui doivent soit être toutes traitées, soit toutes rejetées.
  - En PL/SQL, le début d'une modification atomique est indiquée par "begin trans" (début de transaction). Sa validation est indiquée par "commit", son rejet par "rollback".
- Exemples :
  - La modification du type de contrat va de pair avec la modification du montant maximum et de la date de fin de validité,
  - L'emprunt d'un livre entraîne l'enregistrement de l'emprunteur et de la date de l'emprunt.

# PL/SQL exemples de transactions

```
BEGIN
  SELECT employee_id INTO e_id FROM emp WHERE empname='Forbs ross';
  SAVEPOINT maj_sal;
  INSERT INTO job_history (employee_id, job_id) VALUES (e_id, new_job_id);
  UPDATE emp SET SALARY = sal
    WHERE employee_id = e_id;
  UPDATE emp SET commission_PCT = sal / nb_parts
    WHERE employee_id = e_id;
  COMMIT;

EXCEPTION
  WHEN ZERO_DIVIDE THEN
    ROLLBACK TO maj_sal;
END;
/
```

# DBMS / (A)CID

- Correspond aux moyens en œuvre par le DBMS pour garantir l'atomicité d'une transaction.
  - Pour Oracle, ce sont :
    - Les segments Rollback
    - Le tablespace TMP
  - Le dimensionnement de ces éléments permet de garantir l'atomicité par le SGBD en fonction du besoin exprimé par l'application.
- Point d'attention sur les traitements en masse via la mise en place de « lots »

*Les bases NoSQL garantissent une atomicité unitaire (enregistrement d'une « donnée » dans la base) mais n'incluent pas (toujours) la possibilité de construire des « atomes complexes » via des « transactions ». Certaines l'intègrent (type MongoDB via `start_transaction`, `commit_transaction`).*

# Commit à 2 phases / pattern Saga

## Modifications impliquants plusieurs nœuds/systèmes

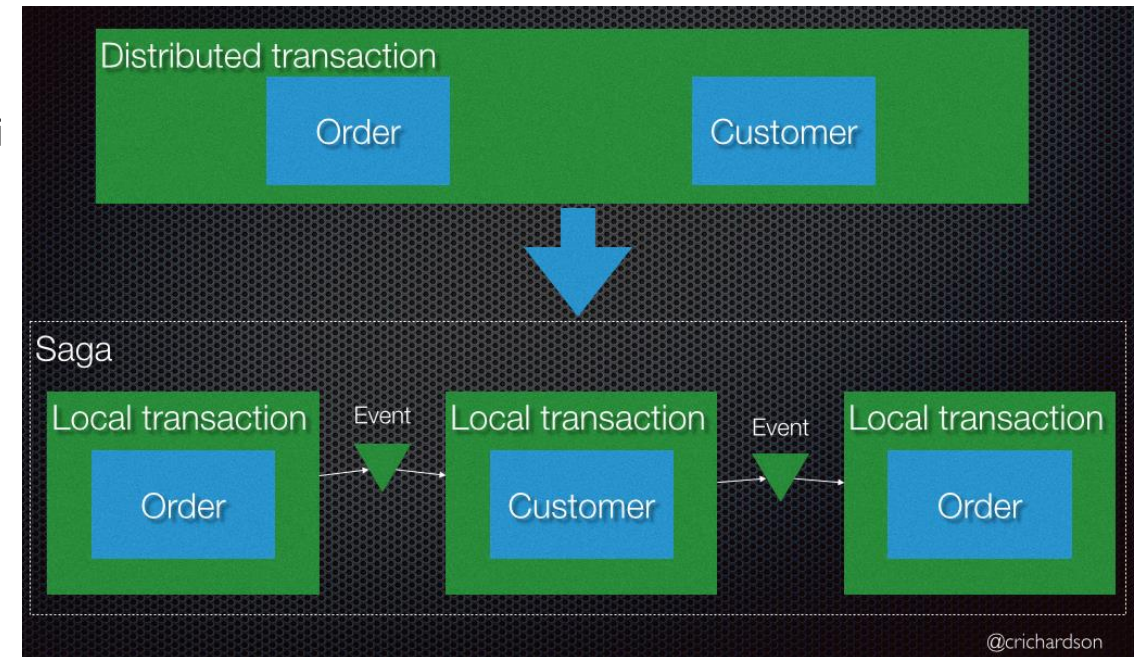
### 2PC : Commit à 2 phases

- Phase 1 : **Prepare** : L'ensemble des nœuds reçoivent les modifications à appliquer sans le faire.
- Phase 2 : **Commit/Forget** : L'ensemble des nœuds appliquent (commit) ou rollback (forget) les changements.

### Pattern Saga :

l'application gère l'ensemble des opérations à réaliser ainsi que, pour chaque, le moyen de l'annuler.

*Pour les bases de données qui ne permettent pas la gestion de 2PC pour des données complexes. L'utilisation du pattern Saga permet de le gérer via l'application (hors SGBD).*





## *Reprise du fil conducteur*

- Questions :
  - Sur quelle dimension (DICT) s'applique la propriété A ?
  - Par rapport aux hypothèses posées de niveau de SLA, pour notre système de réservation de salles, quel est l'impact :
    - Sur le codage de l'application
    - Sur le mode de déploiement du SGBD
    - Sur l'administration du SGBD

# A(C)ID

- C pour "Consistant" :
  - Toute modification doit vérifier les contraintes d'intégrité fonctionnelles.
  - Les contraintes d'intégrité fonctionnelles font parties de la définition des tables via les clés étrangères (Foreign key), les vérifications de valeurs (CHECK), les triggers.
  - Les cohérences des modifications réalisées par plusieurs systèmes travaillant sur les mêmes données sont assurées par leur coordination constante (compteur pour RAC) et/ou via le commit à 2 phases.
- Exemples :
  - Définition de valeurs booléennes (0/1, 'Y'/'N', 'O'/'N', True/False, ...), de liste de valeur,
  - Tout message doit avoir un expéditeur.
  - Utilisation de compteur Oracle pour les clés primaires.

# PL/SQL exemples Cohérence

```
CREATE or REPLACE TRIGGER trg1
AFTER DELETE ON emp1 FOR EACH ROW
BEGIN
    IF :old.eno = 1 THEN
        raise_application_error(-20015, 'You can't delete this row');
    END IF;
END;
```

```
CREATE TABLE suppliers
(
    supplier_id numeric(4),
    supplier_name varchar2(50),
    CONSTRAINT check_supplier_id
        CHECK (supplier_id BETWEEN 100 and 9999)
    CONSTRAINT check_supplier_name
        CHECK (supplier_name = upper(supplier_name))
);
```

## BDMS A(C)ID

Au-delà des cohérences fonctionnelles issues de l'analyse, la cohérence s'applique pour l'administration à identifier l'ensemble des informations nécessaire à garantir la cohérence des données à un instant T.

- Données réparties dans plusieurs SGBD,
- Données présentes dans des fichiers,
- Code applicatif correspondant à une version du SGBD, ...

C'est l'ensemble de ces informations qui permettent de garantir le redémarrage de l'application en cas de problème.

*Les SGBD NoSQL ne proposent pas (toujours) de moyen de gestion de la cohérence fonctionnelle des données. Certains (type documents) peuvent proposer des mécanismes de validation via des schémas de documents qui intègrent des validations de type/de complétude.*

## *Reprise du fil conducteur*

- Questions :
  - Sur quelle dimension (DICT) s'applique la propriété C ?
  - Par rapport aux hypothèses posées de niveau de SLA, pour notre système de réservation de salles, quel est l'impact :
    - Sur le codage de l'application
    - Sur le mode déploiement du SGBD
    - Sur l'administration du SGBD

# AC(I)D

- I pour "Isolée" :
  - Chaque transaction est isolée des autres,
  - Dans les SGBD Transactionnels, cette isolation résulte de mise en place de verrous sur les objets/enregistrements concernés par les modifications réalisées par une transactions. Ces verrous bloquent l'accès en lecture/écriture pour les autres transactions.

# Niveaux d'isolation en norme SQL

- Niveaux d'Isolation :
  - *Dirty read / lecture sale* :
    - toute lecture est possible mais sans garantie : pas d'intégrité des données, les clés étrangères sont pas respectées, contraintes d'unicité pas respectées.
  - *Lecture pas renouvelable* :
    - La donnée renvoyée à la lecture faite à T0 ne sera pas forcément la même que celle faite à T1, le contenu de l'enregistrement peut avoir été changées voir même il peut avoir disparu
  - *Lecture fantôme* :
    - De nouveaux enregistrements peuvent vérifier les clauses de sélection entre T1 et T0 (mais les données à T0 sont présentes et n'ont pas changées).
  - **Mode standard 'serializable'** :
    - Cas usuel : pas sale, renouvelable, pas fantôme.

Niveau d'isolation SQL ANSI	Lecture sale	Non répétable	fantôme
READ UNCOMMITTED	permise	permise	permise
READ COMMITTED		permise	permise
REPETABLE READ			permise
SERIALIZABLE			

## Exemple : consultation de comptes pendant une série de changements

Réalisation d'une requête de somme du total de tous les comptes.

Row	Numero compte	Total compte
1	123	500.00
2	456	240.25
...	...	...
342023	987	100.00

Et modification en parallèle du solde des comptes :

Le compte 123 passe de 500 à 100 et le compte 987 passe de 100 à 500.

Row	Numéro compte	Total compte	Locked ?
1	123	(500.00) changé en 100.00	X
2	456	240.25	--
...	...	...	--
342023	987	(100.00) changé en 500.00	X



*BD en READ COMMITTED (non Oracle)  
non répétable, fantôme.*

temps	requête de somme	Transaction de transfert sur les comptes
T1	Lecture row 1. Somme = 500.00	--
T2	Lecture row 2. Somme = 740.25.	--
T3	--	Changement row 1 mise d'un lock exclusif sur row 1, empêchant tout changement ou lecture. Row 1 = 100.00.
T4	Lecture row N. Somme = . . .	--
T5	--	Changement row 342023 mise d'un lock exclusif sur row 342023, empêchant tout changement ou lecture. Row 342023 = 500.00.
T6	Tentative de lecture row 342,023 mais locked... Le calcul se met en attente de la fin du lock sur l'enregistrement.	--
T7	--	Commits transaction des changements sur 1 et 342023
T8	Lecture row 342023, retourne 500.00. La somme totale retournée inclus les 400.00 en trop de l'enregistrement 1.	

## *BD en REPEATABLE READ répétable, fantôme*

temps	requête de somme	Transaction de transfert sur les comptes
T1	Lecture row 1. Somme = 500.00	--
T2	Lecture row 2. Somme = 740.25.	--
T3	--	Tentative de changement 1 mais bloqué. Transaction est suspendue jusqu'à l'obtention d'un lock exclusif.
T4	Lecture row N. Somme = . . .	--
T5	Lecture row 342023, retourne 100.00 et présente le résultat final.	
T6	commit transaction	--
T7		Change le row 1 et met 100.00, bloque l'enregistrement en exclusif
T8		Change le row 342023, met 500. Commit la transaction.

## *BD non Oracle REPETABLE READ (situation de deadlock)*

Time	Query	Account Transfer Transaction
T1	Lecture row 1. Sum = 500.00 . Block 1 a un lock lecture partagée.	--
T2	Lecture row 2. Sum = \$740.25 . Block 2 a un lock de lecture partagé (shared read).	--
T3	--	MAJ row 342023 et met un lock exclusif sur le block 342023, empechant toute MAJ et locks de type shared read. Valeur = 50.00.
T4	Lecture row N. Sum = . . .	--
T5	--	Essaye de MAJ row 1 mais bloqué. Transaction est mise en attente jusqu'à obtention d'un lock exclusif.
T6	Tentative de lect row 342023 mais ne peut pas, un lock exclusif est en place.	--

## *Transactions / conclusion*

- Dans Oracle les changements sont enregistrés dans les Rollback segments tant que la transaction n'est pas validée
  - la table n'est *pas* modifiée tant que la transactions n'est pas commitée et oracle maintient les valeurs du début d'une transaction en lecture.
  - read committed avec, au sein d'une transaction, les propriétés d'un repetable read.

## *Reprise du fil conducteur*

- Questions :
  - Sur quelle dimension (DICT) s'applique la propriété I ?
  - Par rapport aux hypothèses posée de niveau de SLA, pour notre systèmes de réservation de salles, quel est l'impact :
    - Sur le codage de l'application
    - Sur le mode déploiement du SGBD
    - Sur l'administration du SGBD

# ACI(D)

- D pour "Durable"
  - la transaction persiste même après le redémarrage du serveur
  - La durabilité est assurée via
    - les mécanismes de sécurisation de l'enregistrement des modifications (redolog) garantissant leur prise en compte même en cas de crash avant l'écriture effective dans les fichiers de données,
    - Les mécanismes de réplication via le commit à 2 phases (phase de préparation – phase de validation) en œuvre avec le dataguard mode protection maximale.

## *Reprise du fil conducteur*

- Questions :
  - Sur quelle dimension (DICT) s'applique la propriété D ?
  - Par rapport aux hypothèses posées de niveau de SLA, pour notre système de réservation de salles, quel est l'impact :
    - Sur le codage de l'application
    - Sur le mode déploiement du SGBD
    - Sur l'administration du SGBD

# No ACID ?

- Pb : (très) grand volume de données...
- Cas exemple :
  - Pour une librairie standard, l'ACID est OK. Pour Amazon, du fait du nombre d'ouvrage, une dérogation du "C/I" peut être tolérée car permet d'éviter des interblocages... risque :
    - toto et titi achètent le dernier exemplaire d'un livre et qu'Amazon doit s'excuser à l'un d'entre les deux...
  - impact sur I et C :
    - I permet l'isolation : tant que toto n'a pas terminé son achat (ou timeout), titi ne peut pas choisir le livre (ex. Fnac).
    - C garanti la cohérence : l'achat de toto va vérifier que le livre est disponible au moment du paiement final.
- Pour FaceBook, l'important est d'afficher une information même si pas la plus récente via des systèmes de cache repartis et de stockage des données des utilisateurs dans une base unique.

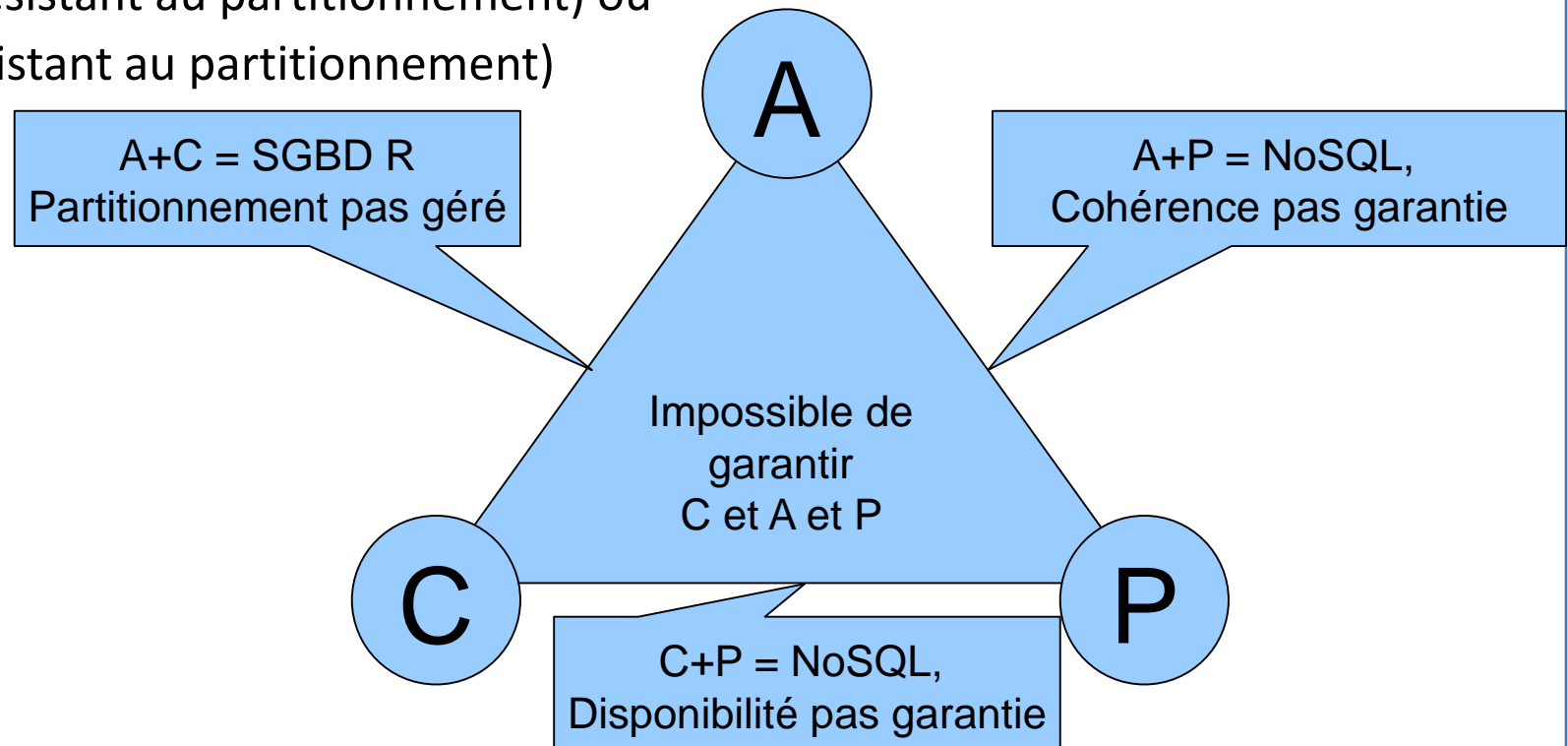


# De ACID à CAP

- Le théorème CAP aborde la problématique de systèmes distribués via trois propriétés :
  - Cohérence :
    - Les données sont cohérentes (au sens ACID du terme) dans toutes les répliquions des données.
      - Une modification faite dans une instance doit être répercutées dans toutes les répliques pour être validée.
  - La Disponibilité (Availability) :
    - Le système continue de répondre même si toutes les instances ne sont pas disponibles.
      - Il y a toujours au moins une instance joignable.
  - Le Partitionnement :
    - En cas d'interruption des communications des instances elles peuvent continuer à répondre de manière indépendantes.
      - les systèmes peuvent (sur)vivre de manière indépendante.

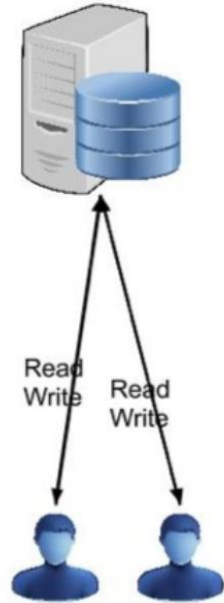
# Théorème CAP

- Les SGBDR assurent les propriétés de Consistance et de Disponibilité (Availability) => Système AC
- Les SGBD « NoSQL » sont des systèmes :
  - AP (Disponible et Résistant au partitionnement) ou
  - CP (Cohérent et Résistant au partitionnement)



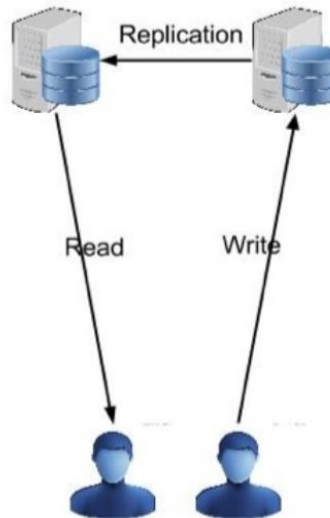
# Illustration CAP

Consistent  
Available  
Partition Tolerant

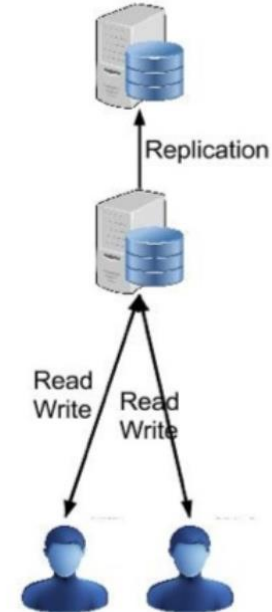


## Cap Theorem

Consistent  
Available  
Partition Tolerant



Consistent  
Available  
Partition Tolerant



## *CAP : solutions ?*

- C+A : SGBD traditionnels Oracle, MySQL, ...
  - Oracle RAC...
- A+P : Riak :
  - Choix de réponse par un nœud même si les données n'ont pas été complètement répliquées
    - Méthodes de résolution des incohérences par défaut (heure, ...) ou customisable.
- C+P : MongoDB :
  - Choix de la garantie de consistance en cas de perte du nœud principal via la réplication des données en cours et mise en attente de toute nouvelle écriture/lecture.

## *Reprise du fil conducteur*

- Questions :
  - Sur quelle dimension (DICT) s'applique les propriétés CAP ?
  - Par rapport aux SGBD Choisis et à leur orientation CAP, pour notre systèmes de réservation de salles, quel est l'impact :
    - Sur le codage de l'application
    - Sur le mode déploiement du SGBD
    - Sur l'administration du SGBD