Bases de données documentaires et distribuées
Cours NFE04
Cassandra en distribuÃľ

Auteur : Philippe Rigaux

Département d'informatique
Conservatoire National des Arts & Métiers, Paris, France

Distribution

Cassandra s'appuie sur le hachage cohérent, très inspiré du système Dynamo (Amazon)

- Anneau sur l'espace $[-2^{63}, 2^{63}]$
- Les serveurs ont une position (*token*) calculée ou attribuée explicitement.
- Le découpage peut être lissé grâce à un système de nœuds virtuels (256 par nœud physique).

Des serveurs peuvent être ajoutés ou supprimés à tout moment.

Routage des requêtes

Chaque nœud connaît la topologie complète de l'anneau.

- Chaque mise à jour (ajout suppression de serveur) est propagée à l'ensemble des participants.
- La table peut devenir assez volumineuse (surtout avec nœuds virtuels) :

Cassandra fonctionne en **multi-nœuds**.

- Un requête peut être prise en charge par n'importe quel nœud, le **coordinateur**
- Le coordinateur identifie les nœuds responsables de la clé du nouveau document

Réplication

La réplication est fixée au niveau du *keyspace* : 3 par défaut.

On indique également une *replication strategy*.

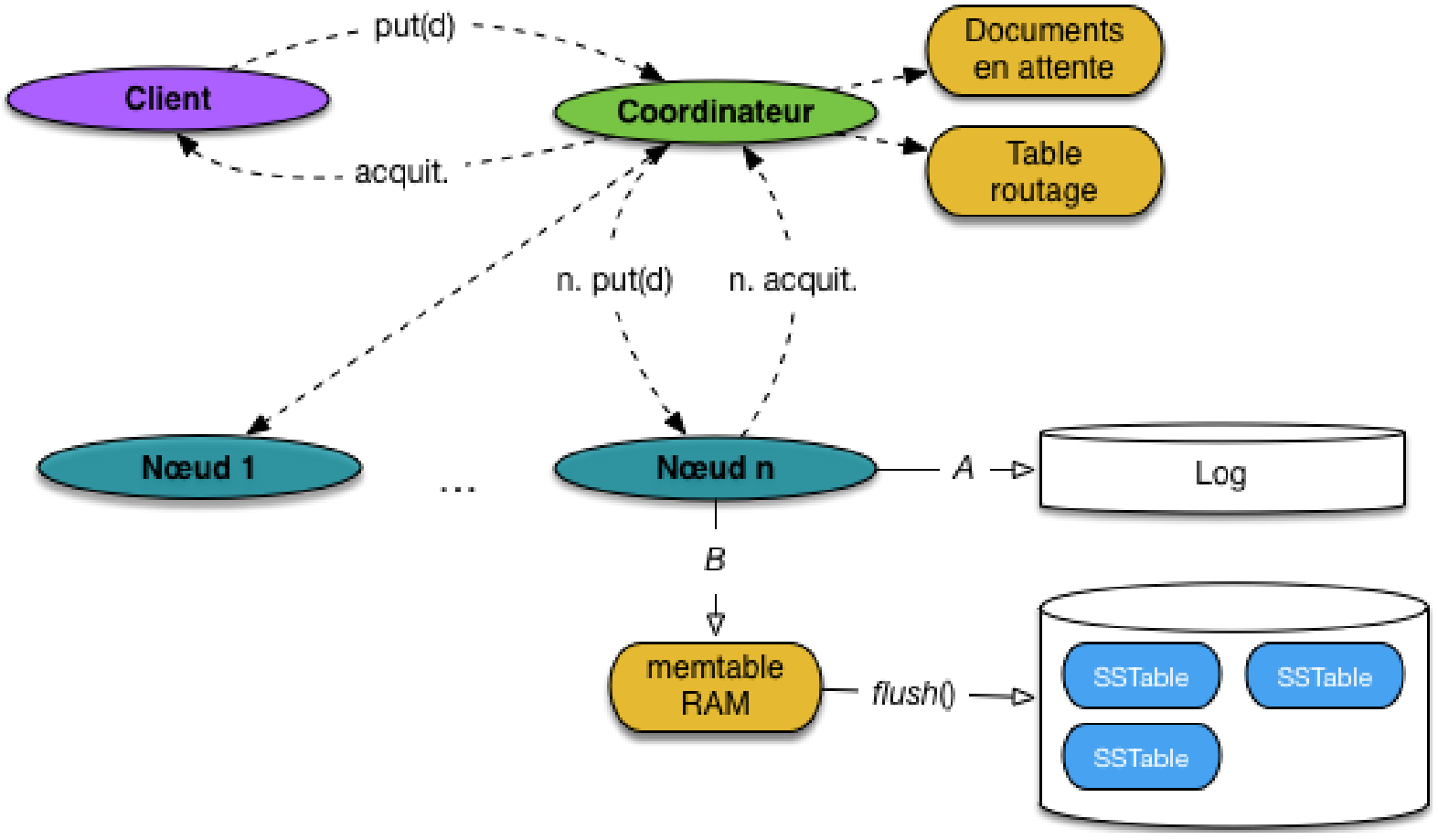
```
CREATE keyspace repli
        with replication = {'class':'SimpleStrategy',
                            'replication_factor':3};
```

La stratégie de réplication "simple" copie sur les successeurs du nœud-cible de l'insertion.

Une stratégie plus complexe tient compte de la topologie du réseau (niveaux *racks* et *data centers*).

Ecriture et cohérence des données

Cassandra propose un mécanisme avancé de gestion du compromis entre cohérence et latence.



Paramétrage de la cohérence en écriture

Le niveau indique le nombre d'acquittements que le coordinateur doit rececoir des nœuds de stockage avant d'acquitter à son tour le client.

- `ANY` : on accepte 0 acquittements ! Le document est conservé dans une zone temporaire, en attente d'une nouvelle tentative d'écriture...
- `ONE` : Au moins un acquittement
- `QUORUM` : Au moins $\lfloor replication/2 \rfloor + 1$ acquittements, soit 2 pour un facteur de réplication 3.
- `ALL` : La réponse au client sera assurée lorsque la ressource aura été écrite dans tous les réplicas.

Rappel : plus on demande d'acquittements, plus on privilégie la cohérence sur la latence.

Cassandra, système réputé très rapide en insertions.

Les stratégies en lecture

Même type de paramétrage en lecture. La version avec l'estampille la plus récente est renvoyée au client.

- `ONE` : le coordinateur transmet la première réponse reçue.
- `QUORUM` : Le coordinateur reçoit la réponse de au moins $\lfloor replication/2 \rfloor + 1$ réplicas, et renvoie au client la ressource avec l'estampille la plus récente.
- `ALL` : Le coordinateur attend d'avoir reçu la réponse de tous les réplicas.

Même remarque : compromis latence/cohérence

Comment régler les paramètres ?

Soit

- **N** : Taux de réplication
- **W** : Nb minimum d'écritures devant accuser de réception
- **R** : Nb copies d'une donnée à consulter pour une requête

Alors, si $W + R > N$, on assure la cohérence des lectures.

Exemple

- Soit N=3, W=ALL, R=1 : toute lecture lit un document à jour
- Soit N=3, W=1, R=ALL : une des lectures lit le document à jour, et c'est celui-ci qui est renvoyé
- Soit N=3, W=QUORUM, R=QUORUM : une des lectures lit forcément lune des versions à jour, et c'est celui-ci qui est renvoyé

Simple et élégant !