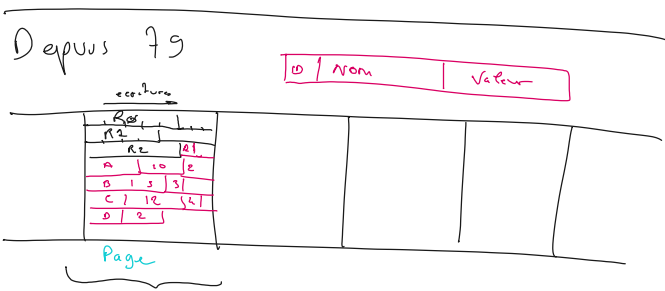
Relationnels → New Sql

SQL SRV
Oracle

NoSql
→ Cassandra
→ Hbase
Mongo
Redis
- - - - - - - -

Depuis 79



Page de données → Stockage en ligne

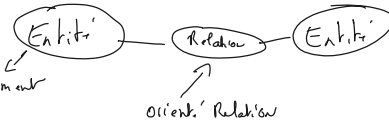
ROW STORE

Catégorie NoSql (global, non exhaustif)

→ Clef / Valeur → a 1 clef (numérique, chaine, binaire, Composite)
= 1 valeur non exploitable par le Serveur
ex: Redis
ex: cache

→ SRV type document
ex: MongoDB
→ 1 clef = 1 Document (XML, JSON, ...)
→ Doc indexable, modifiable, etc...

Graphe :



ex : Qui a liké quel post de qui ?

Column Store

ex : HBase, Hive (Hadoop)

Row store → bon pour de nombreuses lectures - modifications

Data Warehouse → - Ecriture unique
- Lectures d'analyse multiples -

- Sybase IQ / TeraData
- MySql Infinidb → MariaDb Columnstore

} Serveur ACID traditionnels

No Sql : HBase,

Insert	1, A	Multiple Version Concurrency check.
Insert	2, B	
Update	1, C	
Delete	2	

---

3 Colonnes → 2 colonnes du schéma + version

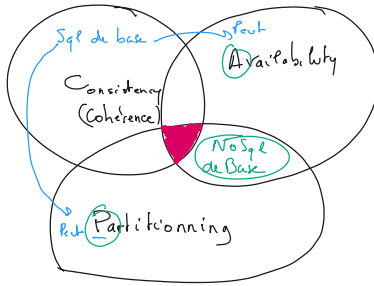
Version	ID	Valeur
1	1	A
1	2	B
2	1	C
2	2	✗

↓

Compactage → | 2 | 1 | C |

Timestamps
Numéro (pas conseillé)

Théorème de CAP



Pas Possible

NoSql de Base

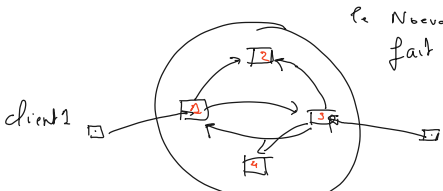
Basically Available, Soft state, Eventual Consistency.

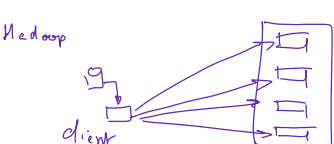
Création d'un schéma de données :

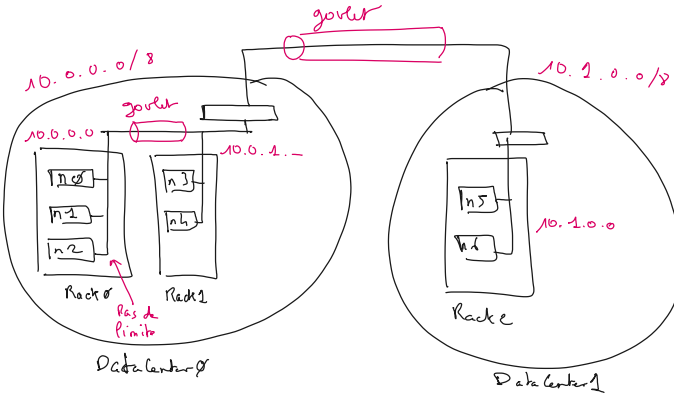
- Dénormaliser ! (les données sont redondées, précalculées, multiples)

- BDD Relationnelles = on établit un schéma de données selon une méthode (Merise) et tout type de requête peuvent avoir lieu (+/- lentes)

- Sous Cassandra → c'est l'inverse !
  → Vous devez connaître vos requêtes -
  → pour déterminer le meilleur schéma (pas de méthode)






gateway
10.0.0.0/8
gateway
10.0.0.0
10.0.1.-
n0
n1
n2
n3
n4
Rack 0
Rack 1
Pas de limite
Data Center 0
10.1.0.0/8
n5
n6
10.1.0.0
Rack e
Data Center 1

Hash (V1) → V2 (numérique)

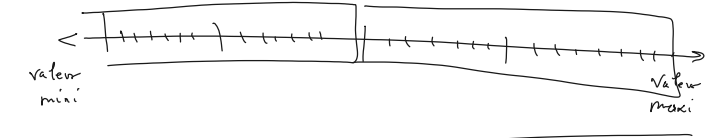
↑ ⌊ impossible

fct hash CRC

OA OB

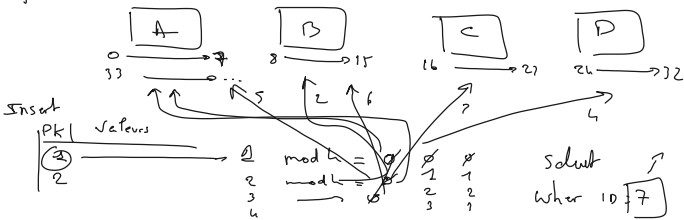
OB OA

md5 → extrêmement sécure / lente



Exemple

fct partition = mod 4 pour 4 noeuds



Insertion des colonnes d'une ligne :

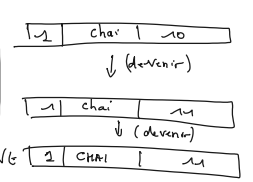
ID Nom, Valeur

Traditionnel :

Insert : ID = 1
Nom = Chai (a)
V = 10

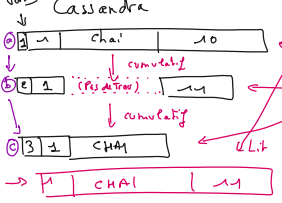
Update ID = 1, V = V + 1 (b)

Update Nom = Upper(Nom) (c)



Version

Cassandra



Lecture

Index

L.I.t

Lancer Cassandra :

- Télécharger depuis cassandra.apache.org
- créer un utilisateur cassandra : useradd cassandra
- extraire : tar xf apache-cassandra...
- Lancer en tant que user cassandra
  bin / cassandra
- Vérifier : bin / nodetool status
  UN = Up / Normal

$CASSANDRA_HOME → dossier cassandra

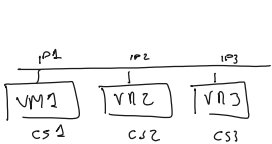
$PATH → $CASSANDRA_HOME / bin

$CASSANDRA_CONF → dossier conf | cassandra.yaml

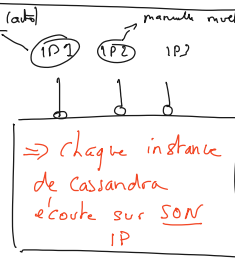
$JAVA_HOME

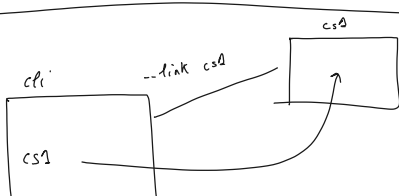
python → bon interpréteur (cqlsh)

Scénarios



→ modifier l'ip d'écoute
→ seeds, etc...





Plusieurs instances CASSANDRA dans une VM :

$C_HOME
$C_CONF → fichier de conf
→ IP
→ dossier data

~/ cs1 / conf, data
cs2
cs3
cs4

~/cs1/conf
~/cs1/data
~/cs2/conf
.....

binaire → toujours le même

Préparation du cluster :

/var/cassandra/cs1/ cs1.sh → environnement

mon instance
↙ /conf/ ... → config
/data/ ... → data

Copier la conf depuis de dossier de Cassandra (qui est donc un template)

chmod +x cs1.sh
lancer : source cs1.sh

Fichier de conf :
export CASSANDRA_HOME=/usr/local/cassandra/apache-cassandra-3.11.4
export PATH=$PATH:$CASSANDRA_HOME/bin
export CASSANDRA_CONF=/var/cassandra/cs1/conf

1 Dans ce fichier de conf, router les datadir dans /var/cassandra/cs1/data ... :

- hints_directory : /var/cassandra/cs1/data/hints
- data_file_directories :
  - /var/cassandra/cs1/data/data ⚠
- commitlog_directory : /var/cassandra/cs1/data/commitLog
- cdc_raw_directory : /var/cassandra/cs1/data/cdc_raw
  (si cdc activé)
- saved_caches_directory :

Définir des ip statiques sous CentOS v7 :
[root@localhost ~]#
nmcli c modify eth0 ipv4.addresses 10.0.0.30/24,10.0.0.31/24

Spécification d'IP :

Seed provider : 1 ou 2 seed par data center
Tous les noeuds ont la même conf

Port 7000 → port inter-noeuds

listen_address ou listen_interface
Port 9042 → CQL / clients

RPC_address RPC_port : 9160 (Thrift) - Coordinateur
- ancien protocole natif remplacé par CQL

Utilisateur.

a) Create Keyspace <nom> ... ;

b) cqlsh connector: Use <nom du Keyspace> ;

c) CREATE TABLE ...

-

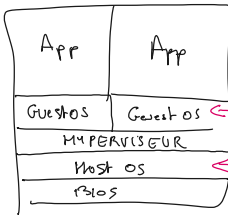
Fichier de conf intégrant un autre port/iip pour le JMX :
export CASSANDRA_HOME=/usr/local/cassandra/apache-cassandra-3.11.3
export PATH=$PATH:$CASSANDRA_HOME/bin
export CASSANDRA_CONF=/var/cassandra/cs1/conf

export JMX_PORT="7199"
export JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.host=10.0.0.1 -Djava.rmi.server.hostname=10.0.0.1"
export JVM_OPTS="$JVM_OPTS -Dcassandra.jmx.remote.port=$JMX_PORT -
Dcom.sun.management.jmxremote.rmi.port=$JMX_PORT"

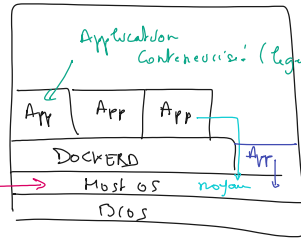
Retirer un noeud :
nodetool decommission (le noeud doit être up)
si pas up : nodetool removenode depuis un autre noeud

Réintégrer le cluster : purger le dossier data, et rejoindre à nouveau

Virtualisation :



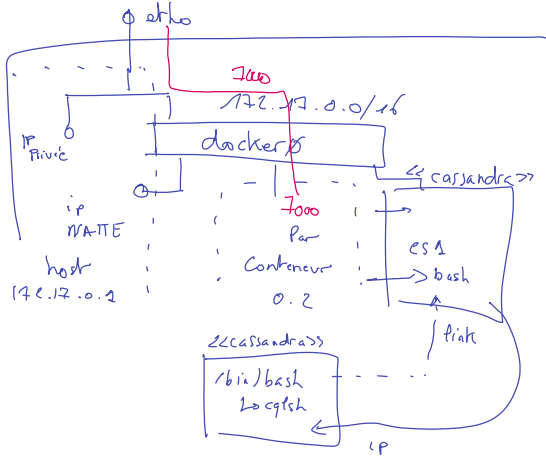
Conteneurisation

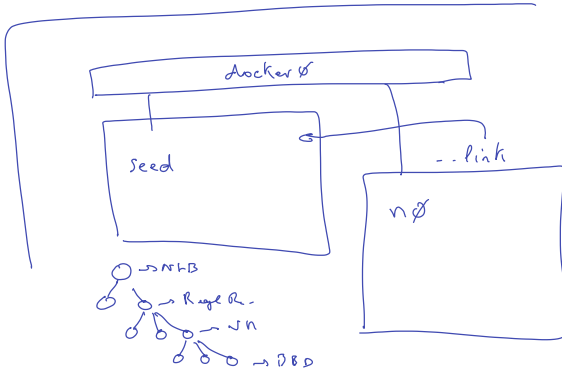


Boot2Docker = distrib Linux officielle host OS Docker

Image : "OS" de base → lecture seule

Conteneur = instance d'image qui a un état




docker ø
seed
--link
n ø
→ NHB
→ Region R.
→ JVM
→ DBD