

Introducció a NoSQL

Capacitació Tecnològica per a Professionals i
Empreses



Index

- 1. HBase**
- 2. Exercicis sobre HBase**



1. HBase

- 1.1 Introducció
- 1.2 Taula comparativa
- 1.3 Característiques
- 1.4 Quins projectes l'utilitzen?
- 1.5 Entorn
- 1.6 Propietats
- 1.7 Operacions
- 1.8 Distribució física, utilitats i APIs
- 1.9 Comparativa SQL
- 1.10 Instal·lació
- 1.11 Driver
- 1.12 Exemple



1.1 Introducció

És una implementació de BigTable: un mapa (diccionari) ordenat, persistent, multidimensional, distribuït i dispers.

La indexació es basa en una clau de fila, una clau de columna i una marca de temps.

{row, column, timestamp} → value
Column → ColumnFamily:Qualifier

Cada valor desat es una cadena de bytes sense interpretació.







1.1 Introducció

Com a definició, podem donar la següent:

Forma part de l'entorn Hadoop, i proporciona un entorn tolerant a errors per desar grans quantitats de dades disperses (sparse data).



1.2 Taula comparativa

Engine	MongoDB	Cassandra	Redis	HBase
Imatge				
Desenvolupador	10gen	Apache Software Foundation	Salvatore Sanfilippo	Apache Software Foundation
Data de llançament	2009	2008	2009	2010
Usuaris destacats	Foursquare, Craigslist	Twitter, Digg	StackOverflow	Facebook
Tipologia	Document	Columna	Key-value	Columna



1.2 Taula comparativa

Llicència	AGPL	Apache	BSD	Apache
Llenguatge	C++	Java	C/C++	Java
Exemple d'ús	CMS, sistemes de vots, desat de comentaris	Banca, logging	Stock prices, Analítics, dades en temps real	Missatgeria en temps real
Millor utilitzar en...	Consultes dinàmiques, escriptures freqüents, poques consultes estadístiques	Grans volums d'escriptures en front a les lectures	Canvis freqüents de dades, escriptura freqüent, poques consultes estadístiques	Lectures aleatòries en grans bases de dades



1.2 Taula comparativa

Punts clau	Manté algunes propietats d'SQL, com els índexs i les consultes	Alta disponibilitat: es tracta d'un encreuament entre les idees de BigTable i Dynamo	Molt ràpid	Modelat sota la direcció de BigTable. Dissenyat per desar ingents quantitats de dades.
Control de concurrència	Locks	Multiversion concurrency control (MVCC)	Locks	Locks
URL	mongodb.org	cassandra.apache.org	redis.io	hbase.apache.org
Procés de replicació	Asíncron	Asíncron	Asíncron	Asíncron



1.2 Taula comparativa

Sistemes operatius	Linux, Mac, Windows	Linux, Mac, Windows	Linux, Mac, Windows	Linux, Mac, Windows
Desat de dades	Disc	Disc	RAM	Hadoop
Característiques	Consistència, tolerància a particions, persistència	Alta disponibilitat, tolerància a particions, persistència	Consistència, tolerància a particions, persistència	Consistència, tolerància a particions, persistència



1.3 Característiques

Característiques de HBase:

- *No relacional*
- *BD distribuïda*
- *Orientada a columnes*
- *Multi dimensional*
- *Alta disponibilitat*
- *Alt rendiment*



1.4 Quins projectes l'utilitzen?

Alguns dels usuaris més destacats:

- *Facebook*
- *Twitter*
- *Mozilla*
- *Adobe*
- *Yahoo*



1.5 Entorn

Hbase surt d'un entorn més gran compost per:

- *Hadoop*
- *HDFS*
- *Hbase*
- *Altres eines per la gestió del sistema, com Zookeeper*

Està molt lligat a l'ecosistema Hadoop. Tot i que es pot utilitzar en solitari, només és recomana en entorns de proves.



1.6 Propietats

Es tracta d'una base de dades, o més aviat un data store, orientat a columnes

Està optimitzat per lectures

El model de consistència és estricte (Escriptures similars a estructura màster-slave). Les escriptures es transmeten en format pipeline.

Només suporta Ordered Partitioning: les column families son desades en l'ordre de les claus de fila.

Això permet fer un esquema de sharding en els anomenats Hfiles, els quals es distribueixen en els diferents data nodes.



1.6 Propietats

Gràcies al Ordered Partitioning, l'escalat horitzontal és senzill.

Les transaccions son suportades a nivell de fila.

Igual que Cassandra, està orientat a l'escalabilitat horitzontal amb tolerància a errors mitjançant rèpliques.

Hbase suporta triggers, sobre les operacions Get/Set/Delete



1.6 Propietats

No suporta índexs secundàris. Això significa que les operacions es fan a nivell de row key.

Exemple d'ús: Trobar tots els jugadors d'Holanda

- Scan de tota la taula

 - Utilitzar range queries si el row index està ben el·laborat

- Manualment, elaborar un índex secundari

 - Explotar el fet de l'ordenació lexicogràfica de les claus

Suporta nativament agregacions simples: SUM, MIN, MAX, AVG, STD



1.6 Propietats

L'esquema de replicació està basat en el sistema de fitxers que el suporta.

Els sistemes de fitxers que suporten Hbase (en esquema de distribució) son HDFS i S3

L'ordenació de les claus és estrictament alfabètica.

aaa – aab – xyz – zzz

El concepte de Column Families en Hbase és lleugerament diferent a Cassandra: es poden prendre com namespaces per definir el conjunt de columnes de cada taula.



1.6 Propietats

Un exemple ens ho aclarirà: analogia amb JSON

```
{
  'aaa': {
    'A': {
      'foo': {
        15: "test1",
        19: "test3"
      },
      'bar': {
        8: "this is a test"
      }
    },
    'B': {
      "": {
        3: "test6"
      }
    }
  },
  'aab': {
    'A': {
      'othercolumn': {
        3: "test2",
      }
    },
    'B': {
    }
  }
}
```

Com veiem, cada registre (o fila) ha de tenir un nombre de column families marcat, però les columnes incloses son arbitràries.



1.7 Operacions

Podeu aprofundir en la definició de les operacions disponibles a <http://wiki.apache.org/hadoop/Hbase/Shell>

Arquitectura

- **alter**: canvia l'esquema de les Column Families, afegint-les o eliminant-les en la taula.
- **count**: compta les files dins una taula.
- **create**: crea una nova taula i defineix les Column Families que contindrà.
- **describe**: dona una imatge de l'estructura d'una taula.
- **disable**: desactiva una taula. Fase essencial en el procés d'eliminació d'una taula.
- **drop**: elimina la taula.
- **enable**: activa una taula.



1.7 Operacions

- **exists**: diu si una taula existeix o no.
- **exit**: surt de la shell.
- **list**: llista les taules de la base de dades.
- **tools**: llista les eines disponibles pel tractament d'Hbase.
- **status**: mostra l'estat del cluster. Permet opcions que filtren el contingut a mostrar.
- **shutdown**: apaga el cluster.
- **truncate**: trunca una taula, recreant-la completament.
- **version**: dona la versió d'Hbase.



1.7 Operacions

Sobre dades

- **get**: obté un conjunt de dades. Pot ser a nivell de fila, o més segmentat, a nivell de columna.
- **put**: desa un valor en una posició de la base de dades.
- **scan**: genera un scan complet de la taula. Admet alguns modificadors per acotar el conjunt de sortida.
- **delete**: elimina un contingut determinat (columna).
- **deleteall**: elimina tots els continguts d'una fila.



1.7 Operacions

Joins: No estan implementades dins el motor. Per assolir aquesta funcionalitat, s'ha d'utilitzar map-reduce.

Tipus de dades

Només te noció de byte-arrays. Les dades, per tant, son interpretades pel client.

També te suport per comptadors, i per tant, per operacions incrementals (atòmiques).



1.8 Distribució física, utilitats i APIs

Distribució física

L'entorn es distribueix per regions (contigües de files, ordenades).

Les dimensions es poden regular en longitud per configuració.

L'estructura de cada regió es divideix en:

- **Memstore**: desa en memòria els canvis aplicats a la regió.
- **Hlog**: registre únic de log.
- **Hfile**: registres de dades.



1.8 Distribució física, utilitats i APIs

Utilitats

Existeixen múltiples utilitats per millorar l'usabilitat de Hbase

- *Lily*: és una llibreria d'indexació
- **Full text indexing**
 - * *Lucene*
 - * *Elasticsearch*
 - * *Solr*



1.8 Distribució física, utilitats i APIs

APIs

- **Java**: natiu. Ofereix les instruccions *Get/Put/Delete/Scan*, *IncrementColumnValue* i les relacionades amb *map-reduce*.
- **Rest** (reemplaçat per *stargate*)
- **Thrift**
- **Scala**
- **Jython**
- **Groovy DSL**
- **Ruby Shell**
- **Pig**
- **Hive**



1.9 Comparativa SQL

Esquema ACID

- **Atomicity**: *totes les mutacions son atòmiques si ocorren dins la fila*

Tanmateix, les apis que operen sobre conjunts de files, no garanteixen l'atomicitat

- **Consistency and isolation**: *totes les files retornades per una operació en l'API son files completes que han existit en algun punt de la història de la taula.*

Tanmateix, els scans no mostren una vista consistent de la taula, ja que no ofereixen aïllament.

- **Durability**: *totes les dades retornades, dins d'un “success”, son durables (o dit d'una altre manera, una lectura no retornarà mai dades que no han sigut desades a disc)*



1.9 Comparativa SQL

Finalment, a mode de comparativa unilateral, podem destacar els següents aspectes:

- ***Desnormalització de dades***: *no és necessari, ni aconsellable per el model obert proposat per Hbase, generar una normalització. Per tant, els següents punts son ignorats:*

- * *Evitar redundancia de dades*
- * *Protegir la integritat de les dades*
- * *Unicitat de les files*
- * *Tipat fort de les dades en les columnes*



1.9 Comparativa SQL

- **Escalabilitat horitzontal:** degut a l'esquema d'arquitectura, és senzill programar una escalabilitat horitzontal en aquest sistema. En canvi, el sharding en sistemes RDBMS genera una sèrie de problemes que no són fàcilment solucionables, ni transparents
- **Consistència:** en tots dos sistemes queda garantida la consistència de dades
- **Transaccions:** no tenim concepte fort de transacció a nivell arbitrària, com en els RDBMS
- **Estructura de dades:** mentre els RDBMS són adequats per els sistemes de dades ben estructurats, Hbase es comporta igual de bé tant en sistemes ben estructurats com en sistemes semi estructurats.



1.10 Instal·lació

```
adduser hduser
addgroup hadoop
wget http://apache.rediris.es/hbase/stable/hbase-<VERSION>
tar xzvf hbase-<VERSION>
chown -R hduser:hadoop hbase-<VERSION>
cd hbase-<VERSION>
su hduser
set up JAVA_HOME in .bashrc
export JAVA_HOME=$(readlink -f /usr/bin/java | sed "s:bin/java::")
* Assegureu-vos que l'entrada dins /etc/hostname pren el valor
127.0.0.1 en /etc/hosts
./bin/start-hbase.sh
```



1.11 Driver

Tenim diverses opcions per utilitzar aplicacions en conjunt amb HBase. Per homogeneïtat, en aquest curs comptarem amb drivers (o interfícies) per Python.

Happybase

<https://github.com/wbolster/happybase>

* Podeu trobar implementacions de drivers per HBase per molts llenguatges actuals



1.12 Exemple

```
– Iniciar el servidor –  
./bin/start-hbase.sh  
– Comandes –  
./bin/hbase shell  
create 'test', 'cf'  
put 'test', 'row1', 'cf:a', 'value1'  
put 'test', 'row2', 'cf:b', 'value2'  
put 'test', 'row3', 'cf:c', 'value3'  
scan 'test'  
get 'test', 'row1'  
disable 'test'  
drop 'test'  
– Aturar el servidor –  
./bin/stop-hbase.sh
```



2. Exercicis sobre HBase

- 2.1 Implementació via shell
- 2.2 Implementació per aplicació
- 2.3 Benchmarking



2.1 Implementació via shell

En aquesta primera part pràctica, generarem una serie de projectes a partir de la consola del sistema que estem tractant.

- **Twitter:** generarem un clon del software conegut com Twitter. Com a definició, prendrem l'esquema d'usuari, tweet i seguidor.

- **Registre d'hotel:** necessitem un sistema d'informació per el registre d'un hotel. S'ha de contindre informació sobre totes les habitacions, així com un historial tant d'usuaris com d'ús.

- **Log enriquit:** volem tenir una peça de software que mantingui informació de log, concretament el còdi, una marca de temps, i un objecte, que contindrà informació arbitrària (fins i tot, altres objectes)



2.1 Implementació via shell

- **Item vault:** un banc d'informació sobre objectes varis dins diverses col·leccions. L'objectiu és donar la versatilitat a un negoci d'objectes estranys per poder tenir un sistema d'informació amb totes les característiques d'aquests.

- **Enfermatim:** base de dades d'informació sobre malalties. Aquest ha de contenir informació sobre geografia, causes, públic afectat, i estadístiques.

- **Workouts:** tenim una gran cadena de gyms, i volem mantenir una base de coneixement sobre les màquines que tenim, així com dels clients i el seu creixement personal (muscular, dietes, salut, dies d'entrenament...)

- **Job box:** es tracta d'un directòri d'ofertes de treball sobre tot el món. Volem informació sobre el treball, així com localització, empresa, perfils necessitats, i tota la informació de valor sota aquest concepte.



2.2 Implementació per aplicació

En la segona part, generarem els mateixos projectes anteriors, a partir del llenguatge de programació que nosaltres escollim i els seus bindings sobre el sistema que tractem.



2.3 Benchmarking

Finalment, en aquesta tercera part, generarem scripts que omplirán les nostres bases de dades, i farem benchmarking sobre els processos d'escriptura i de lectura.

Barcelona **a**ctiva



Ajuntament
de Barcelona

bcn.cat/barcelonactiva
bcn.cat/cibernarium