HBase. Tecnologías “on-line”, “real-time” y sistemas híbridos

Programa experto en Big Data

Profesor: Ignacio Marrero Hervás

Ampliación arquitectura, 1º Trimestre
Curso 2014-2015

ÍNDICE

Administración básica

- WebUI
  - Proporciona información básica de Regions, Region Servers y Tablas
  - Permite hacer split y compaction
  - Visualiza Thread dump
  - Visualiza dump de Zookeeper
  - Visualiza logs
  - Visualiza la configuración
- Comandos de shell
  - Permiten operar sobre Regiones, Region Servers, replicación, seguridad, snapshot, logs, balanceo
- Comandos de hadoop
  - Permiten examinar el sistema de ficheros, importar y exportar datos

Administración básica

Utilidad para imprimir la configuración

```
$ hbase org.apache.hadoop.hbase.HBaseConfiguration
<?xml version="1.0" encoding="UTF-8" standalone="no"?><configuration>
<property><name>hbase.online.schema.update.enable</name><value>false</
value><source>hbase-default.xml</source></property>
<property><name>hbase.master.port</name><value>60000</value><source>hbase-
default.xml</source></property>
<property><name>hbase.hstore.compaction.max</name><value>10</
value><source>hbase-default.xml</source></property>
<property><name>hbase.regionserver.port</name><value>60020</value><source>hbase-
default.xml</source></property>
<property><name>ha.failover-controller.cli-check.rpc-timeout.ms</
name><value>20000</value><source>core-default.xml</source></property>
<property><name>ipc.client.connect.max.retries.on.timeouts</name><value>45</
value><source>core-default.xml</source></property>
<property><name>hbase.zookeeper.dns.nameserver</name><value>default</
value><source>hbase-default.xml</source></property>
<property><name>hadoop.tmp.dir</name><value>/tmp/hadoop-${user.name}</
value><source>core-default.xml</source></property>
```

Administración básica

Verificación de la consistencia del cluster

```
$ hbase hbck -?
```

Unión de regiones

El cluster de HBase tiene que estar offline para poder hacer merge

Recomendación:

Operación muy crítica. No se recomienda hacerla

```
$ hbase org.apache.hadoop.hbase.util.Merge
For hadoop 0.20,  Usage: bin/hbase org.apache.hadoop.hbase.util.Merge [-
Dfs.default.name=hdfs://nn:port] <table-name> <region-1> <region-2>

For hadoop 0.21+, Usage: bin/hbase org.apache.hadoop.hbase.util.Merge [-
Dfs.defaultFS=hdfs://nn:port] <table-name> <region-1> <region-2>
```

Administración básica

División avanzada de regiones

```
$ hbase org.apache.hadoop.hbase.util.RegionSplitter
usage: RegionSplitter <TABLE> <SPLITALGORITHM>
                      SPLITALGORITHM is a java class name of a class
                      implementing SplitAlgorithm. HexStringSplit treats
                      keys as hexadecimal ASCII, and UniformSplit treats
                      keys as arbitrary bytes.
 -c <region count>         Create a new table with a pre-split number of
                           regions
 -D <property=value>       Override HBase Configuration Settings
 -f <family:family:...>    Column Families to create with new table.
                           Required with -c
    --firstrow <arg>       First Row in Table for Split Algorithm
 -h                        Print this usage help
    --lastrow <arg>        Last Row in Table for Split Algorithm
 -o <count>                Max outstanding splits that have unfinished
                           major compactions
 -r                        Perform a rolling split of an existing region
    --risky                Skip verification steps to complete
                           quickly.STRONGLY DISCOURAGED for production
                           systems.
```

Administración básica

Examinar ficheros de almacenamiento

```
$ hbase hfile
usage: HFile [-a] [-b] [-e] [-f <arg>] [-k] [-m] [-p] [-r <arg>] [-s] [-v]
       [-w <arg>]
 -a,--checkfamily       Enable family check
 -b,--printblocks       Print block index meta data
 -e,--printkey          Print keys
 -f,--file <arg>        File to scan. Pass full-path; e.g.
                        hdfs://a:9000/hbase/.META./12/34
 -k,--checkrow          Enable row order check; looks for out-of-order
                        keys
 -m,--printmeta         Print meta data of file
 -p,--printkv           Print key/value pairs
 -r,--region <arg>      Region to scan. Pass region name; e.g. '.META.,,1'
 -s,--stats             Print statistics
 -v,--verbose           Verbose output; emits file and meta data
                        delimiters
 -w,--seekToRow <arg>   Seek to this row and print all the kvs for this
                        row only
```

Administración básica

Utilidad para importar datos masivamente

```
$ hbase org.apache.hadoop.hbase.mapreduce.ImportTsv
Usage: importtsv -Dimporttsv.columns=a,b,c <tablename> <inputdir>
Imports the given input directory of TSV data into the specified table.
[. . . . . .]
By default importtsv will load data directly into HBase. To instead generate
HFiles of data to prepare for a bulk data load, pass the option:
  -Dimporttsv.bulk.output=/path/for/output
  Note: if you do not use this option, then the target table must already exist
in HBase
Other options that may be specified with -D include:
  -Dimporttsv.skip.bad.lines=false - fail if encountering an invalid line
  '-Dimporttsv.separator=|' - eg separate on pipes instead of tabs
  -Dimporttsv.timestamp=currentTimeAsLong - use the specified timestamp for the
import
  -Dimporttsv.mapper.class=my.Mapper - A user-defined Mapper to use instead of
org.apache.hadoop.hbase.mapreduce.TsvImporterMapper
For performance consider the following options:
  -Dmapred.map.tasks.speculative.execution=false
  -Dmapred.reduce.tasks.speculative.execution=false
```

Administración básica

Utilidad para importar tablas

```
$ hbase org.apache.hadoop.hbase.mapreduce.Import
ERROR: Wrong number of arguments: 0
Usage: Import [options] <tablename> <inputdir>
By default Import will load data directly into HBase. To instead generate
HFiles of data to prepare for a bulk data load, pass the option:
  -Dimport.bulk.output=/path/for/output
 To apply a generic org.apache.hadoop.hbase.filter.Filter to the input, use
  -Dimport.filter.class=<name of filter class>
  -Dimport.filter.args=<comma separated list of args for filter
 NOTE: The filter will be applied BEFORE doing key renames via the
HBASE_IMPORTER_RENAME_CFS property. Futher, filters will only use
theFilter#filterKeyValue(KeyValue) method to determine if the KeyValue should be
added; Filter.ReturnCode#INCLUDE and #INCLUDE_AND_NEXT_COL will be considered as
including the KeyValue.
For performance consider the following options:
  -Dmapred.map.tasks.speculative.execution=false
  -Dmapred.reduce.tasks.speculative.execution=false
  -Dimport.wal.durability=<Used while writing data to hbase. Allowed values are
the supported durability values like SKIP_WAL/ASYNC_WAL/SYNC_WAL/...>
```

Administración básica

Utilidad para exportar tablas

```
$ hbase org.apache.hadoop.hbase.mapreduce.Export
ERROR: Wrong number of arguments: 0
Usage: Export [-D <property=value>]* <tablename> <outputdir> [<versions>
[<starttime> [<endtime>]] [^[regex pattern] or [Prefix] to filter]]

  Note: -D properties will be applied to the conf used.
  For example:
   -D mapred.output.compress=true
   -D mapred.output.compression.codec=org.apache.hadoop.io.compress.GzipCodec
   -D mapred.output.compression.type=BLOCK
  Additionally, the following SCAN properties can be specified
  to control/limit what is exported..
   -D hbase.mapreduce.scan.column.family=<familyName>
   -D hbase.mapreduce.include.deleted.rows=true
For performance consider the following properties:
   -Dhbase.client.scanner.caching=100
   -Dmapred.map.tasks.speculative.execution=false
   -Dmapred.reduce.tasks.speculative.execution=false
For tables with very wide rows consider setting the batch size as below:
   -Dhbase.export.scanner.batch=10
```

Administración básica

Parametro	Descripción
versions	Número de versiones por columna (defecto=1)
starttime	Tiempo de inicio para limitar las versiones
endtime	Tiempo de fin para limitar las versiones
regex/prefix	Prefijo o expresión regular para la `rowkey`

Ficheros de log del cluster relacionados con HBase

1. `hbase-<id>-MASTER-<hostname>.log`
2. `hbase-<id>-REGIONSERVER-<hostname>.log`
3. `hadoop-<id>-NAMENODE-<hostname>.log`
4. `hadoop-<id>-DATANODE-<hostname>.log`
5. `hadoop-<id>-RESOURCEMANAGER-<hostname>.log`
6. `hadoop-<id>-NODEMANAGER-<hostname>.log`
7. `zookeeper-<id>-SERVER-<hostname>.log`

Shell

Sintaxis de los comandos de shell

```
<comand name> <diccionario>, <diccionario>, …
```

Los parámetros <diccionario> pueden ser de la forma

<keyword> => <value>

y

{<keyword> => <value>, <keyword> => <value>, …}

donde <value> irá entre símbolos ‘’ si es un string

Shell

Catálogo de los comandos más usados

scan

Permite buscar varias filas y devolver sus celdas

Uso:

```
scan '<table name>', <scan dictionary>
```

Dictionary:

```
TIMERANGE, FILTER, LIMIT, STARTROW, STOPROW, TIMESTAMP,
MAXLENGTH, COLUMNS, CACHE, CACHE_BLOCKS, RAW, VERSIONS
```

get

Permite buscar una fila y devolver sus celdas

Uso:

```
get'<table name>', '<rowkey>', <get dictionary>
```

Dictionary:

```
TIMERANGE, FILTER, LIMIT, TIMESTAMP, COLUMN, VERSIONS
```

Shell

Catálogo de los comandos más usados

create

Permite crear tablas y sus column families

Uso:

```
create '<table name>', <table dictionary>, <CF dictionary>, …
<CF dictionary> debe incluir la clave NAME con el nombre de la
column family
```

Dictionary:

```
SPLITS, SPLITS_FILE, SPLITALGO, NUMREGIONS
Constantes de HColumnDescriptor
Constantes de HtableDescriptor
```

Shell

Catálogo de los comandos más usados

alter

Permite cambiar esquemas de tablas y sus column families

Uso:

```
alter '<table name>', <table dictionary>, <CF dictionary>, …
<CF dictionary> debe incluir la clave NAME con el nombre de la column family
```

Dictionary:

```
MAX_FILESIZE, MEMSTORE_FLUSHSIZE, READONLY, DEFERRED_LOG_FLUSH, METHOD
Constantes de HColumnDescriptor
Constantes de HtableDescriptor
```

Shell. Catálogo

Comandos generales

status

Devuelve información sobre el cluster

version

Devuelve la versión del software, repositorio, compilación

whoami

Muestra el usuario actual

Shell. Catálogo

Comandos de definición de datos

alter
: Modifica el esquema de una tabla

alter_async
: Cambia el esquema de una tabla. No espera a que el cambio de haya propagado a todos los RS

alter_status
: Devuelve el estado del comando alter_async en curso

create
: Crea una tabla

describe
: Muestra el descriptor de una tabla

disable
: Deshabilita una tabla

Shell. Catálogo

Comandos de definición de datos

disable_all

Deshabilita todas las tablas dada una expresión regular

drop

Borra una tabla

drop_all

Borra todas las tablas dada una expresión regular

enable

Habilita una tabla

enable_all

Habilita todas las tablas dada una expresión regular

Shell. Catálogo

Comandos de definición de datos

exists

Comprueba si existe una tabla

get_table

Devuelve un objeto de tipo tabla para referenciarlo en otros comandos

is_disabled

Comprueba si una tabla está deshabilitada

is_enabled

Comprueba si una tabla está habilitada

list

Devuelve la lista de tablas

show_filters

muestra todos los filtros disponibles

Shell. Catálogo

Comandos de gestión de namespaces

alter_namespace

Modifica las propiedades de un namespace

create_namespace

Crea un namespace

describe_namespace

Muestra las propiedades de un namespace

drop_namespace

Borra un namespace y todas sus tablas

list_namespace

Lista todos los namespaces

list_namespace_tables

Lista las tablas de un namespace

Shell. Catálogo

Comandos de gestión de datos

append
: Añade bytes al contenido de una celda

count
: Cuenta las filas de una tabla

delete
: Borra una celda

deleteall
: Borra un conjunto de columnas: column family, fila

get
: Devuelve una fila

get_counter
: Devuelve un contador

incr
: Incrementa un contador

Shell. Catálogo

Comandos de gestión de datos

put

Guarda una celda

scan

Escanea un rango de celdas

truncate

disable + drop + create

Shell. Catálogo

Comandos de herramientas

assign
: Asigna una región a un servidor

balance_switch
: Cambia de estado el balanceador

balancer
: Arranca el balanceador

close_region
: Cierra una región

compact
: Inicia la compactación de una tabla o región

flush
: Inicia el “flush” de una tabla o región

Shell. Catálogo

Comandos de herramientas

hlog_roll
: Rota el log de un region server

major_compact
: Inicia la compactación total de una tabla o región

merge_region
: Hace merge de dos regiones

move
: Mueve una región a un Region Server diferente

split
: Divide una región o tabla

unassign
: Desasigna una región

zk_dump
: Devuelve los detalles de los zNodes de Zookeeper

Shell. Catálogo

Comandos de replicación de la base de datos

add_peer

Añade un cluster de replicación

disable_peer

Deshabilita el stream al cluster de replicación pero conserva las actualizaciones

enable_peer

Habilita de nuevo el stream al cluster de replicación

list_peers

Lista de nuevo todos los clusteres de replicación

remove_peer

Elimina un cluster de replicación

Shell. Catálogo

Comandos de snapshot de la base de datos

clone_snapshot

Crea una tabla apartir de un snapshot

delete_snapshot

Borra un snapshot

list_snapshots

Muestra todos los snapshots disponibles

restore_snapshot

Restaura un snapshot a su tabla original sobreescribiendo el contenido

snapshot

Crea un snapshot de una tabla

Shell. Catálogo

Comandos de autorización de la base de datos

grant

Da permisos Read,Create,Exec,Create,Admin a usuarios concretos con la granularidad Tabla:Column Family:Qualifier

revoke

Quita permisos Read,Create,Exec,Create,Admin a usuarios concretos con la granularidad Tabla:Column Family:Qualifier

user_permission

Muestra todos los permisos de un usuario, globales o por tabla

Shell. Constantes

Enlaces relacionados con las constantes de shell

Código principal de la shell de HBase

http://svn.apache.org/viewvc/hbase/branches/0.98/hbase-shell/src/main/ruby/hbase.rb?view=markup

Tabla de constantes con sus valores por defecto

http://archive.cloudera.com/cdh5/cdh/5/hbase/apidocs/constant-values.html

Javadoc de HConstants

http://archive.cloudera.com/cdh5/cdh/5/hbase/apidocs/org/apache/hadoop/hbase/HConstants.html

Javadoc de HColumnDescriptor

http://archive.cloudera.com/cdh5/cdh/5/hbase/apidocs/org/apache/hadoop/hbase/HColumnDescriptor.html

Javadoc de HTableDescriptor

http://archive.cloudera.com/cdh5/cdh/5/hbase/apidocs/org/apache/hadoop/hbase/HTableDescriptor.html

Optimización

Propiedades relacionadas con los scanners

hbase.regionserver.lease.period (ms)

- tiempo de lock de los scanners sobre las filas
- cualquier lock no liberado tardará este tiempo en eliminarse

hbase.client.scanner.caching

- numero de filas cacheadas por un scanner

Actividad del master sobre los logs

hbase.master.logcleaner.ttl (ms)

- tiempo que tarda el master en borrar los ficheros de log viejos

Optimización

Splits y Compactions

hbase.hregion.max.filesize (bytes)

tamaño en bytes que tiene que tener un store file para generar un split de región

hbase.hstore.compaction.min

numero mínimo de store files en un minor compaction

hbase.hstore.compaction.max

numero máximo de store files en un minor compaction

hbase.hstore.compaction.min.size (bytes)

todos los store files menores no entraran en un minor compaction

hbase.hstore.compaction.max.size (bytes)

todos los store files mayores no entraran en un minor compaction

Optimización

Splits y Compactions

hbase.hstore.compaction.ratio (ratio)

- en una segunda ronda incluirá los ficheros de tamaño = ratio * suma de los tamaños de todos los ficheros ya incluidos

hbase.server.thread.wakefrequency (ms)

- cada rs ejecuta una instancia de CompactionChecker para monitorizar que tipo de compaction debe realizarse

hbase.hregion.majorcompaction (ms)

- tiempo entre dos major compactions

hbase.hregion.majorcompaction.jitter (ratio)

- ratio de dispersión para la ejecución de major compactions entre los distintos stores

Optimización

Memstore

hbase.hregion.memstore.mslab.enabled (boolean)

habilita la funcionalidad MSLAB de buffers de tamaño fijo para objetos KeyValue

hbase.hregion.memstore.mslab.chunksize (bytes)

tamaño máximo del buffer MSLAB

Balanceador

hbase.balancer.period (ms)

tiempo entre pasadas del proceso balancer ejecutado por el master

hbase.balancer.max.balancing (ms)

tiempo máximo que puede estar el balanceador ejecutándose

Optimización

Parámetros más críticos

zookeeper.session.timeout (ms)

- timeout de la sesión de zookeeper. Si un RS se cae el master tarda este tiempo en notar la caída
- En procesos de alta carga como importaciones, disminuirlo es un problema porque puede introducir inestabilidad

hbase.regionserver.handler.count

- Número de threads que puede servir peticiones a los RS simultáneamente
- Si los payloads son pequeños se puede aumentar el número
- Un numero demasiado alto podría generar problemas de OutOfMemory o pausas por Garbage Collection

HBASE_HEAPSIZE (proceso master)

HBASE_REGIONSERVER_OPTS (procesos de RS)

- tamaño del heap de los procesos de HBase (variables en hbase-env.sh)

Optimización

Parámetros más críticos

hbase.hregion.max.filesize (bytes)

- tamaño en bytes que tiene que tener un store file para generar un split de región
- en lo posible hay que minimizar el número de regiones

hbase.hstore.blockingStoreFiles

- si un RS supera este máximo de store files bloqueará las escrituras para que el compactador tenga tiempo de actuar
- Debe aumentarse en procesos de alta carga de escritura

hbase.regionserver.maxlogs

- número máximo de logs en disco
- hay que aumentarlo para alta carga de escritura
- hay que disminuirlo si queremos flush frecuentes

Optimización

Parámetros más críticos

hfile.block.cache.size (ratio)

- ratio de heap destinado al cache de bloques
- se debe incrementar cuando tenemos cargas de lectura

hbase.regionserver.global.memstore.upperLimit (ratio)

hbase.regionserver.global.memstore.lowerLimit (ratio)

- ratio máximo y mínimo del heap destinado al memstore
- es mejor mantener los límites cercanos para disminuir el flush
- en procesos alta carga de lectura se disminuye para aumentar el block cache

hbase.hregion.memstore.flush.size (bytes)

- tamaño que tiene que tener un memstore para hacer flush a disco

hbase.hregion.memstore.block.multiplier

- bloquea insert/update cuando memstore > multiplier * flush.size
- Si tenemos mucha memoria es mejor aumentarlo

Filtros

Expresión para un filtro:

FilterName (arg1, arg2,...)

Donde arg:

- Si es un string va entre símbolos ''
- Si es un booleano, entero u operador no lleva ''

Se pueden construir expresiones mas complejas con operadores:

- AND, OR: son los operadores lógicos habituales
- WHILE: para cada fila emite los KeyValues hasta que uno de ellos no cumple la condicion
- SKIP: si cualquier de los posibles KeyValues no cumple la condición del filtro lo salta

Ejemplo:

"(Filter1 AND Filter2) OR (Filter3 AND Filter4) AND SKIP Filter5"

Filtros

Operadores de comparación

- LESS (el cliente usa el símbolo <)
- LESS_OR_EQUAL (el cliente usa el símbolo <=)
- EQUAL (el cliente usa el símbolo =)
- NOT_EQUAL (el cliente usa el símbolo !=)
- GREATER_OR_EQUAL (el cliente usa el símbolo >=)
- GREATER (el cliente usa el símbolo >)

Comparadores

- BinaryComparator (el cliente usa el string ‘binary’)
- BinaryPrefixComparator (el cliente usa el string ‘binaryprefix’)
- RegexStringComparator (el cliente usa el string ‘regexstring’)
- SubStringComparator (el cliente usa el string ‘substring’)

Filtros

Ejemplo:

Para seleccionar un column family:qualifier de un rowkey concreto este puede ser un filtro

"(PrefixFilter ('DXPE') AND (FamilyFilter (=, 'regexstring:open')) AND (QualifierFilter (=, 'regexstring:price')))”

El catálogo de filtros lo podemos consultar en el documento adjunto

Filtros

Filtro	Descripción
InclusiveStopFilter	Delvuelve todas las columnas de las filas hasta (inclusive) la rowkey <stop_row_key>
TimeStampsFilter	Devuelve todos los KeyValues cuyo timestamp está en la lista especificada <timestamp>
RowFilter	Devuelve todos los KeyValues cuyo Rowkey devuelva "true" en la comparación especificada
FamilyFilter	Devuelve todos los KeyValues cuyo Column Family devuelva "true" en la comparación especificada
QualifierFilter	Devuelve todos los KeyValues cuyo Qualifier devuelva "true" en la comparación especificada
ValueFilter	Devuelve todos los KeyValues cuyo Value devuelva "true" en la comparación especificada

Filtros

Filtro	Descripción
DependentColumnFilter	Si recibe <family> y <qualifier> devuelve todos los KeyValues de esa columna que tengan el mismo timestamp. Si recibe el argumento <dropDependentColumn> a true la columna no se devolverá. Si incluimos el operador de comparación <compare operator> y el comparador <value comparator> un KeyValue deberá pasar una condición adicional: su Value deberá devolver true en la comparación
SingleColumnValueFilter	Si recibe <compare operator>, '<comparator>', '<family>', '<qualifier>', cuando una Rowkey no tiene la columna especificada se emiten todos sus KeyValues. Si la Rowkey tiene esa columna se evalúa el operador de comparación y si devuelve true se emitirán todos los KeyValue Si aparece el argumento <filterIfColumnMissing_boolean> y es true los KeyValue no se emitirán si la Rowkey no tiene esa columna (por defecto, false). Si aparece el argumento <latest_version_boolean> y es true sólo se analizará la última versión (por defecto, true)
SingleColumnValueExcludeFilter	Funciona como el anterior salvo que si la Rowkey tiene esa columna y la comparación devuelve true se emitirán todos los KeyValue menos el de la columna especificada
ColumnRangeFilter	Devuelve la lista de KeyValues cuyas columnas estén entre <minColumn> y <maxColumn>

Historia. Big Table

El predecesor de HBase es Big Table, desarrollado por Google en 2006

Estas son sus diferencias principales (HBase vs Big Table)

Timestamps en milisegundos

Timestamps en microsegundos

Compresión LZO, Snappy, gz

Compresión en dos fases

Tiene filtros en servidor y coprocesors

Tiene su propia versión de coprocesors algo menos flexibles

Basado en HDFS

Basado en GFS

No mapea Store Files en memoria

Si mapea Store Files en memoria reduciendo el IO

Historia. Big Table

No agrupa Column Families para aplicar reglas comunes

Si agrupa Column Families

Implementa cache de bloques

Implementa además cache KeyValue

HBase puede desactivar el commit log por temas de performance

Big Table puede configurar un commit log de 2 fases

HBase no usa extensivamente la tabla. .META.

Big Table también usa .META. para información secundaria

Historia. Versiones

Las versiones mas relevantes de HBase son

0.20: primera versión estable basada en Hadoop 0.20

0.92: seguridad, coprocesors, nuevo formato Store File, splitting distribuido

0.94: mejora de la performance, compatible Hadoop 2.0

0.98: elimina ROOT table, compatible Hadoop 2.2

Actualmente la versión más estable de HBase es 0.98