

Supervisión y Mantenimiento de un clúster Hadoop

Iván González

Agenda

- Configuración inicial
- Ficheros de Log de Hadoop
- Administración de un clúster Hadoop
- Configuración avanzada de Hadoop
- Seguridad en Hadoop
- Administración y Planificación de trabajos
- Mantenimiento del clúster
- Monitorización del clúster

Parte del contenido de esta presentación ha sido extraído de la web de Cloudera <http://www.cloudera.com>

Configuración inicial

- Cada máquina del clúster Hadoop tiene su propio conjunto de ficheros de configuración
 - Los ficheros residen típicamente en `/etc/hadoop/conf`
- La mayoría de estos ficheros son XML
- Al iniciar el clúster, cada demonio accede a su ficheros de configuración
 - Es importante reiniciar el demonio si se ha cambiado algún parámetro

Configuración inicial

- Un ejemplo de fichero XML (`mapred-site.xml`)

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
 href="configuration.xsl">
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>localhost:8021</value>
  </property>
</configuration>
```

Configuración inicial – Ficheros de configuración

File	Type of Configuration
<code>core-site.xml</code>	Core
<code>hdfs-site.xml</code>	HDFS
<code>mapred-site.xml</code>	MapReduce
<code>hadoop-policy.xml</code>	Access control policies
<code>log4j.properties</code>	Logging
<code>hadoop-metrics.properties</code> , <code>hadoop-metrics2.properties</code>	Metrics
<code>include</code> , <code>exclude</code> (file names are configurable)	Host inclusion/exclusion in a cluster
<code>allocations.xml</code> (file name is configurable)	FairScheduler
<code>masters</code> , <code>slaves</code>	Scripted startup (not recommended)
<code>hadoop-env.sh</code>	Environment variables

Configuración inicial – hadoop-env.sh

- Define un conjunto de variables de entorno que son necesarias para Hadoop
 - HADOOP_CLASSPATH
 - HADOOP_HEAPSIZE
 - HADOOP_LOG_DIR
 - HADOOP_PID_DIR
 - JAVA_HOME
- Estos valores son utilizados por otros scripts de Hadoop, incluyendo los demonios
- Si es necesario cambiar los valores, hay que hacerlo en este fichero

Configuración inicial – hadoop-env.sh

- **HADOOP_HEAPSIZE**
 - Controla el tamaño del HEAP para todos los demonios de Hadoop
 - Por defecto 1GB
 - Recomendación: Definir el tamaño para cada demonio de manera independiente
- **HADOOP_NAMENODE_OPTS**
 - Opciones de Java para el NameNode
 - Al menos 4GB: -Xmx4g
- **HADOOP_JOBTRACKER_OPTS**
 - Opciones de Java para el JobTracker
 - Al menos 4GB: -Xmx4g
- **HADOOP_DATANODE_OPTS, HADOOP_TASKTRACKER_OPTS**
 - Establecer 1 GB cada uno: -Xmx1g

Configuración inicial – Precedencia de valor

- Los parámetros de configuración se pueden definir más de una vez
- El valor de mayor precedencia tiene prioridad
- Orden de precedencia (más baja a más alta)
 - *-site.xml en el nodo esclavo
 - *-site.xml en la máquina cliente
 - Valores establecidos en el objeto Job de un trabajo MapReduce
- Si un valor en un fichero de configuración se marca como *final* sobrescribe al resto

```
<property>
    <name>alguna.propiedad.nombre</name>
    <value>algunvalor</value>
    <final>true</final>
</property>
```

Configuración inicial – Valores por defecto

- Existe muchos parámetros a configurar
- Es posible encontrar un listado con sus valores por defecto en <http://hadoop.apache.org/docs/current/>
 - Menú de la izquierda abajo, en Configuration
- En cualquier caso existen muchos factores que pueden afectar a la hora de definir la configuración más adecuada
 - Estos valores por defecto sirven como punto de partida inicial

Configuración inicial de HDFS – hdfs-site.xml

- **dfs.name.dir**
 - El valor más importante del clúster, y usado por el NameNode
 - Indica el lugar del sistema de ficheros local donde guarda el NameNode los metadatos. Se puede pasar una lista separada por coma (sin espacios). Por defecto: \${hadoop.tmp.dir}/dfs/name
 - La pérdida de los metadatos supone la pérdida efectiva de todos los datos, aunque los datos existen en los DataNodes, no es posible reconstruir los ficheros originales sin los metadatos
 - Es importante indicar al menos dos discos (o un RAID) en el NameNode
 - Es importante definir este valor correctamente

Configuración inicial de HDFS – hdfs-site.xml

- El NameNode escribe la información en todos los directorios de manera síncrona
- Si un directorio desaparece, el NameNode continua
 - Ignora el directorio hasta que aparezca de nuevo
- Ejemplo: `/disk1/dfs/nn,/disk2/dfs/nn`
- Se puede emplear un sistema NFS. La recomendación para el montaje es: `tcp,soft,intr,timeo=10,retrans=10`
 - *soft* evita que el NameNode se cuelgue si el punto de montaje desaparece
 - Se intenta retransmitir 10 veces, cada 1-10 segundos, antes de indicar un fallo

Configuración inicial de HDFS – hdfs-site.xml

- **dfs.block.size**
 - El tamaño de bloque para nuevos ficheros, en bytes
 - Por defecto 64 MB
 - Se recomiendan 128MB
- **dfs.data.dir**
 - Donde se almacenan los bloques de datos dentro del sistema de ficheros local del DataNode. Se puede proveer una lista separada por coma (sin espacio).
 - Los directorios se escriben en round-robin (sin redundancia)
 - Puede ser diferente en cada DataNode

Configuración inicial de HDFS – hdfs-site.xml

- **dfs.http.address**
 - Dirección y puerto usado por la IU Web del NameNode
 - Por ejemplo: <nombredelnodo>:50070
- **dfs.replication**
 - El número de veces que cada bloque debe ser replicado cuando se escribe un fichero
 - Por defecto 3 (recomendado)
- **dfs.datanode.du.reserved**
 - La cantidad de espacio de cada volumen que no puede ser usado por HDFS
 - Por defecto 0
 - Recomendado 25%, con un mínimo de 10GB

Configuración inicial de HDFS – core-site.xml

- **fs.default.name**
 - El nombre del sistema de ficheros por defecto
 - Normalmente incluye el nombre del sistema de ficheros, el nombre del NameNode y el puerto
 - Ejemplo: `hdfs://<nombredelnamenode>:8020/`
 - Lo emplea cada máquina con acceso al clúster
- **hadoop.tmp.dir**
 - Directorio temporal tanto en el sistema de ficheros local como en HDFS
 - Por defecto `/tmp/hadoop-${user.name}`
 - Lo emplean todos los nodos
 - Este parámetro afecta a otros parámetros como `dfs.data.dir`

Configuración inicial de MapReduce – mapred-site.xml

- **mapred.job.tracker**
 - Nombre del host y puerto del JobTracker
 - Ejemplo: *nombredeljobtracker:8021*
 - Usado por el JobTracker y TaskTrackers
- **mapred.child.java.opts**
 - Opciones de Java para los procesos “hijo” del TaskTracker.
 - Por defecto –Xmx200m
 - Se recomiendan 1GB - 4GB dependiendo de los requisitos de las aplicaciones
- **mapred.child.ulimit**
 - Tamaño máximo de memoria virtual en KB que puede solicitar un proceso “hijo” del TaskTracker
 - Si se especifica, debe ser 1,5 veces el valor de *mapred.java.opts*

Configuración inicial de MapReduce – mapred-site.xml

- **mapred.local.dir**
 - El directorio local donde MapReduce almacena los ficheros de datos temporales. Puede ser una lista separada por comas (sin espacios) en diferentes dispositivos.
 - Recomendación: incluir todos los discos, establecer `dfs.datanode.du.reserved` (en hdfs-site.xml) al 25%
- **mapred.system.dir**
 - El directorio HDFS donde MapReduce almacena los ficheros a compartir durante la ejecución de un trabajo
 - Ejemplo: `/mapred/system`

Configuración inicial de MapReduce – mapred-site.xml

- **mapreduce.jobtracker.staging.root.dir**
 - El directorio raíz en HDFS donde se almacenan los ficheros de usuario
 - Recomendado /user
- **mapred.tasktracker.map.tasks.maximum**
 - Número de tareas Map que pueden ser ejecutadas simultáneamente en el TaskTracker
- **mapred.trasktracker.reduce.tasks.maximum**
 - Número de tareas Reduce que pueden ser ejecutadas simultáneamente en el TaskTracker

Configuración inicial de MapReduce – mapred-site.xml

➤ Regla:

- N° total de tareas Map + Reduce aprox. 1,5 veces el número de cores
- Se asume que hay suficiente RAM
- Esto debe ser controlado
 - Si el nodo no está limitado por CPU o E/S, se puede incrementar el número de tareas
- Distribución típica: 60% Map, 40% Reduce o 70% Map, 30% Reduce

Ficheros de Log de Hadoop

Type of Log	Description
Daemon	Informational, warning, and error messages generated by Hadoop daemons. Each Hadoop daemon produces two log files: <ul style="list-style-type: none">• <code>.log</code> – First port of call when diagnosing problems• <code>.out</code> – Combination of <code>stdout</code> and <code>stderr</code> during daemon startup, doesn't usually contain much output
Task	<code>stdout</code> , <code>stderr</code> , and <code>syslog</code> output generated by MapReduce applications.
Job Configuration	Job configuration settings specified by the developer.
Job History	Job summary and counters, task summary and analysis, stack traces for any thrown exceptions, URLs to navigate to the task logs.

Ficheros de Log de Hadoop – Localización

Type of Daemon Log	Location
MRv1 (JobTracker, TaskTracker)	<p>Default directory: /var/log/hadoop-0.20-mapreduce (Set HADOOP_LOG_DIR in hadoop-env.sh to configure)</p> <p>Default log file names: hadoop-hadoop-<daemon>-<hostname>. {log out}</p> <p>Example: /var/log/hadoop-0.20-mapreduce/hadoop-hadoop-tasktracker-elephant.log</p>
HDFS and MRv2	<p>Default directory: /var/log/hadoop-<component> (Set HADOOP_LOG_DIR in hadoop-env.sh to configure)</p> <p>Default log file names: hadoop-<component>-<daemon>-<hostname>. {log out}</p> <p>Example: /var/log/hadoop-hdfs/hadoop-hdfs-datanode-tiger.log</p>

Ficheros de Log de Hadoop – Rotación

Type of Daemon Log	Default Retention
All .out Files	Rotated when daemon restarts, five files retained
MRv1 .log Files	Rotated daily Cannot limit file size or the number of files kept Provide your own scripts to compress, archive, delete logs
HDFS and MRv2 .log Files	Maximum size of generated log files: 256MB Number of files retained: 20 Maximum disk space for logs: 5GB

Ficheros de Log de Hadoop – Tareas

Type of Task Log	Location
MRv1	<p>Default directory: /var/log/hadoop-0.20-mapreduce/userlogs (Set HADOOP_LOG_DIR in hadoop-env.sh to configure)</p> <p>Contains symbolic links to paths under mapred.local.dir</p> <p>Default retention: 24 hours</p> <p>Configure retention with mapred.userlog.retain.hours</p>
MRv2	<p>Default directory: /var/log/hadoop-yarn/userlogs (Set HADOOP_LOG_DIR in hadoop-env.sh to configure)</p>

Ficheros de Log de Hadoop – Trabajos

Job Log File	Contents	Location and Retention
Job Configuration XML File	Job configuration settings specified by the developer	<code> \${hadoop.log.dir}/<job_id>_conf.xml</code> (Set HADOOP_LOG_DIR in hadoop-env.sh to configure) Retention: mapred.jobtracker.retirejob.interval milliseconds Default: 1 day (24 * 60 * 60 * 1000)
Job History on Local Disk	Job summary and counters Task summary and analysis Stack traces for any thrown exceptions URLs to navigate to task logs	<code>hadoop.job.history.location</code> Default location: <code> \${hadoop.log.dir}/history</code> Retention: 30 days
Job History in HDFS	Same as Job History on local disk	<code>hadoop.job.history.user.location</code> Default location: <code> <job_output_directory>/_logs/history</code> Retention: As long as the output directory exists

Ficheros de Log de Hadoop – Trabajos

- El JobTracker también almacena los logs de los trabajos en memoria por un tiempo limitado
- Se puede acceder al historial de trabajos mediante la línea de comandos

```
# mapred job -history all <directorio_salida_trabajo_HDFS>
```

Administración de un clúster Hadoop

- Hadoop es un sistema complejo
 - Un pequeño clúster es fácil de configurar
- Administrar un clúster grande es mucho más difícil
 - Desplegar los cambios en todos los nodos
 - Verificar la configuración en todos los nodos
 - Configuración Hadoop
 - Versiones del software
- Existen muchas opciones para optimizar un clúster
 - Muchas no están bien documentadas
 - Las mejoras configuraciones todavía no están claras y van apareciendo a medida que el uso de Hadoop aumenta

Administración de un clúster Hadoop

- La configuración de un clúster se considera “dark art”
- Los ficheros de configuración son complejos y es fácil romper algo
- Un clúster tiene un gran número de equipos
 - Múltiples servicios ejecutando en cada equipo
 - Difícil conocer el estado de cada uno
- Es inevitable que surjan problemas con el hardware o el software
- A medida que el clúster crece, es necesario modificar los valores de configuración y monitorizar su efecto
 - Por ejemplo, para comparar el rendimiento de trabajos similares antes y después del cambio

Administración de un clúster Hadoop

- Mantener el seguimiento del rendimiento del clúster es difícil
 - Existen muchos elementos que monitorizar
 - Mantener la información durante un tiempo es a veces un reto
- El rendimiento puede degradarse
 - Quizás cierto equipo se está ralentizando
 - Un trabajo puede estar ralentizado
- Es posible que solo queramos que ciertos usuarios tengan acceso al clúster y definir qué puede hacer cada usuario
 - Algunos solo acceden a HDFS
 - Otros pueden lanzar trabajos
 - Otros son administradores y pueden crear usuarios

Configuración avanzada de Hadoop

- Hemos visto las propiedades básicas para configurar un clúster Hadoop
- A continuación discutimos propiedades adicionales
 - Optimización y mejora del rendimiento
 - Manejo de recursos
 - Control de acceso
- Las configuraciones que vamos a tratar son solo un punto de partida

Configuración avanzada de Hadoop – hdfs-site.xml

- **dfs.namenode.handler.count**
 - Número de threads del NameNode usados para manejar peticiones RPC desde los DataNodes.
 - Por defecto 10
 - Recomendado $\ln(n^{\circ} \text{ de nodos}) * 20$
 - Síntomas de que este valor es bajo: mensajes “connection refused” en los logs del DataNode
- **dfs.datanode.failed.volumes.tolerated**
 - Número de volúmenes que se permite que fallen antes de que el DataNode se ponga *offline*, lo que puede resultar en que todos sus bloques se repliquen.
 - Defecto 0, pero a menudo se incrementa en maquina con varios discos

Configuración avanzada de Hadoop – core-site.xml

- **fs.trash.interval**
 - Cuando se borra un fichero se almacena en el directorio `.Trash` del home del usuario, en vez de ser eliminado directamente. Se elimina de HDFS tras los minutos indicados en este parámetro.
 - Por defecto 0 (deshabilitado)
 - Recomendado 1440 (un día)
- **io.file.buffer.size**
 - Se indica cuantos datos se almacenan en el buffer para operaciones de escritura y lectura. Debe ser una potencia de 2 del tamaño de página del hardware
 - Por defecto 4096
 - Recomendado 65536 (64KB)

Configuración avanzada de Hadoop – core-site.xml

➤ io.compression.codecs

- Lista de codecs para compresión que Hadoop puede usar para comprimir ficheros
- Por defecto :
 - org.apache.hadoop.io.compress.DefaultCodec,
 - org.apache.hadoop.io.compress.GzipCodec,
 - org.apache.hadoop.io.compress.BZip2Codec,
 - org.apache.hadoop.io.compress.DeflateCodec,
 - org.apache.hadoop.io.compress.SnappyCodec

Configuración avanzada de Hadoop – mapred-site.xml

- **mapred.job.tracker.handler.count**
 - Número de threads usados por el JobTracker para responder a los *heartbeats* de los TaskTrackers
 - Por defecto 10
 - Recomendado $\ln(n^{\circ} \text{ de nodos}) * 20$
- **mapred.reduce.parallel.copies**
 - Número de TaskTrackers que un Reducer puede conectar en paralelo para transferir sus datos
 - Por defecto 5
 - Recomendación $\ln(n^{\circ} \text{ de nodos}) * 4$ con un *floor* de 10

Configuración avanzada de Hadoop – mapred-site.xml

➤ **tasktracker.http.threads**

- N° de threads HTTP en el TaskTracker que son usados por los Reducers para obtener datos
- Por defecto 40
- Recomendado 80

➤ **mapred.reduce.slowstart.completed.maps**

- % de las tareas Map que debe estar completado antes de que el JobTracker ejecute Reducers en el clúster
- Por defecto 0,05 (5%)
- Recomendado 0,8 (80%)

Configuración avanzada de Hadoop – mapred-site.xml

- **mapred.map.tasks.speculative.execution**
 - Permite la ejecución especulativa de tareas Map
 - Por defecto true (recomendado)
- **mapred.reduce.tasks.speculative.execution**
 - Permite la ejecución especulativa de tareas Reduce
 - Por defecto true
 - Recomendado false
- Si una tarea se está ejecutando significativamente más lenta que la media del trabajo, puede que esté ocurriendo ejecución especulativa
 - Otro intento de ejecución de la misma tarea es instanciada en otro nodo
 - Se usan los resultados de la primera tarea completada
 - La tarea más lenta se elimina

Configuración avanzada de Hadoop – mapred-site.xml

- **mapred.compress.map.output**
 - Determina si los datos de los Mappers deben ser comprimidos antes de transferirlos por la red
 - Por defecto false
 - Recomendación true
- **mapred.map.output.compression.codec**
 - El códec de compresión de usados para comprimir los datos intermedios de los Mappers
 - Por defecto org.apache.hadoop.io.compress.DefaultCodec
 - Recomendado org.apache.hadoop.io.compress.SnappyCodec

Configuración avanzada de Hadoop – mapred-site.xml

➤ io.sort.mb

- Tamaño del buffer en RAM del Mapper en el cual el Mapper almacena sus pares de clave/valor antes de escribirlos a disco
- Por defecto 100 MB
- Recomendado: Para un proceso “hijo” con 1GB de Heap, 128 MB (sale del Heap Java)

➤ io.sort.factor

- N° de streams para combinar cuando se ordenan ficheros
- Por defecto 10
- Recomendado 64

Configuración avanzada de Hadoop – Puertos Hadoop

- Cada demonio de Hadoop dispone de su propia interface de usuario Web
 - Útil para usuarios y administradores
- Se muestra información en diferentes puertos
 - Los números de los puertos son configurables, aunque hay por defecto en la mayoría de los casos
- Hadoop también usa puertos para que los diferentes componentes del sistema se comuniquen

Configuración avanzada de Hadoop – Puertos Hadoop

	Daemon	Default Port	Configuration parameter
HDFS	NameNode	50070	dfs.http.address
	DataNode	50075	dfs.datanode.http.address
	Secondary NameNode	50090	dfs.secondary.http.address
MR	JobTracker	50030	mapred.job.tracker.http.address
	TaskTracker	50060	mapred.task.tracker.http.address

Daemon	Default Port	Configuration Parameter	Used for
NameNode	8020	fs.default.name	Filesystem metadata operations
DataNode	50010	dfs.datanode.address	DFS data transfer
DataNode	50020	dfs.datanode.ipc.address	Block metadata operations and recovery
JobTracker	Usually 8021, 9001, or 8012	mapred.job.tracker	Job submission, TaskTracker heartbeats
TaskTracker	Usually 8021, 9001, or 8012	mapred.task.tracker.report.address	Communicating with child tasks

Configuración avanzada de Hadoop – Incluir/Excluir nodos

- La propiedad `dfs.hosts` en `hdfs-site.xml` permite indicar un fichero con la lista de nodos que tienen permitido conectarse al NameNode y que puedan actuar como DataNodes
 - Idéntico caso `mapred.hosts`, indica un ficheros con las lista de nodos que pueden conectar al JobTracker como TaskTrackers
- Ambos ficheros son opcionales
 - Si se omiten, cualquier equipo puede conectarse y actuar como DataNode/TaskTracker
 - Un posible problema de seguridad
- El NameNode puede ser forzado a releer el fichero indicado por `dfs.hosts` con `hadoop dfsadmin -refreshNodes`
 - JobTracker igual con `hadoop madmin -refreshNodes`

Configuración avanzada de Hadoop – Incluir/Excluir nodos

- Es posible prevenir explícitamente que uno o más equipos puedan actuar como DataNodes
 - Crear la propiedad `dfs.hosts.exclude`, e indicar un nombre de fichero
 - Incluir en el fichero la lista de nombres de equipos a excluir
 - No se permitirá a estos equipos conectar al NameNode
 - Esto se emplea a menudo cuando se quiere eliminar nodos
 - Ejecutar `hadoop dfsadmin -refreshNodes` para que el NameNode relea el fichero
- Algo similar se puede hacer con `mapred.hosts.exclude` para especificar la lista de nodos que no pueden conectar al JobTracker
 - Ejecuta `hadoop mradmin -refreshNodes` para que el JobTracker relea el fichero

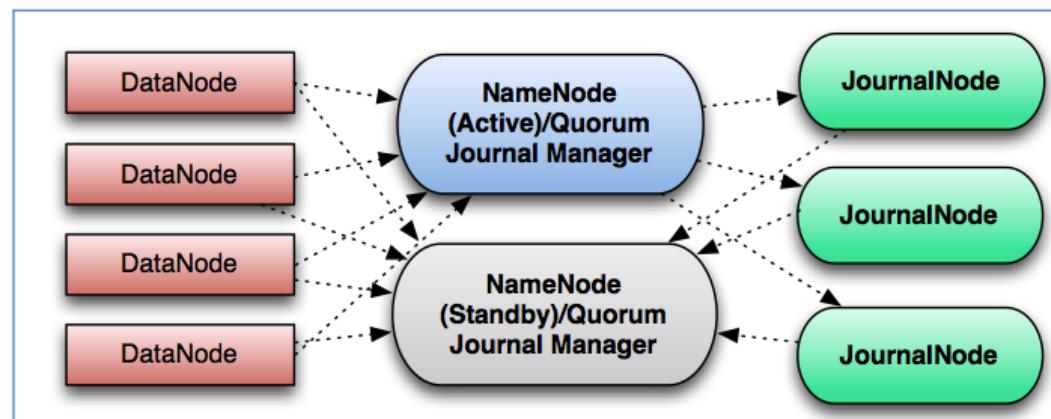
Configuración avanzada de Hadoop – HDFS Rack Awareness

- Recordar que HDFS es “rack aware”
 - Distribuye los bloques basándose en la localización de los nodos
 - Fichero `/etc/hadoop/conf/host-rack.map`

```
host1 /datacenter1/rack1
host2 /datacenter1/rack1
host3 /datacenter1/rack1
host4 /datacenter1/rack1
host5 /datacenter1/rack2
host6 /datacenter1/rack2
host7 /datacenter1/rack2
host8 /datacenter1/rack2
...
.
```

Configuración Avanzada de Hadoop – HDFS en Alta Disponibilidad

- HDFS en Alta disponibilidad usa un par de NameNodes
 - Una activo el otro en espera
 - Los clientes solo conectan con el activo
 - DataNodes *heartbeat* ambos
 - El NameNode activo escribe los metadatos a un conjunto de JournalNodes
 - El NameNode en espera lee de los JournalNodes para mantener el sincronismo con el NameNode activo



Seguridad en Hadoop

- **La privacidad de los datos es importante**
 - Existen leyes que lo regulan, particularmente en salud y finanzas
 - Existen regulaciones para la exportación de datos
 - Es importante proteger los datos de investigaciones
- **En una misma empresa existen diferentes políticas respecto a los datos**
 - Diferentes equipos tienen diferentes necesidades
- **Disponer de diferentes clústeres es una solución muy común**
 - Un clúster para datos protegidos, otro para datos sin proteger

Seguridad en Hadoop

➤ Seguridad

- La seguridad en equipos de computación es un área muy amplia
- El control de acceso es lo que importa en cuanto al uso de Hadoop
- Debemos centrarnos en autenticación y autorización

➤ Autenticación

- Confirmar la identidad del usuario
- Típicamente se solicitan credenciales (usuario/contraseña)

➤ Autorización

- Determinar si el usuario puede realizar una acción determinada
- Típicamente chequeando una lista de control de acceso

Seguridad en Hadoop

- Seguridad en HDFS: propiedad y permisos
 - Protección modesta
 - La autenticación usuario/grupo se puede evitar fácilmente (lado del cliente)
 - Principalmente enfocada a evitar el borrado/sobre-escritura de datos accidentalmente
- Mejorar la seguridad con Kerberos
 - Provee un mecanismo de autenticación fuerte en el lado del cliente y servidor
 - Aplica seguridad a las llamadas del API Hadoop (SASL)
 - Las tareas se pueden ejecutar bajo una cuenta específica para enviar trabajos
 - Esta seguridad es opcional (deshabilitada por defecto)

Seguridad en Hadoop

- Es posible cifrar la transferencia de datos en HDFS
- Es posible cifrar el tráfico HTTP
 - La interface Web
 - Los datos intermedios transferidos durante las etapas de shuffle y sort
- No existe en Hadoop posibilidad de cifrar los datos almacenados en disco
- La seguridad del clúster se puede mejorar por “aislamiento”
 - Usar su propia red
 - Solo permitir el acceso de usuarios de confianza

Administración de trabajos

- Para ver todos los trabajos ejecutados en el clúster

```
# mapred job -list
```

Total jobs:1

JobId RsvdContainers	State UsedMem	StartTime RsvdMem	UserName NeededMem	Queue AM info	Priority	UsedContainers
job_1445426559562_6167 NORMAL 94 0	RUNNING 192512M	1455197949204 0M	ivangm 192512M	http://hadoop- slave2:8088/proxy/application_1445426559562_6167/	root.ivangm	

Administración de trabajos

- Para ver todos los trabajos, incluidos los finalizados

```
# mapred job -list all
```

Total jobs:6162

JobId RsvdMem	State NeededMem	StartTime AM info	UserName	Queue	Priority	UsedContainers	RsvdContainers	UsedMem
job_1445426559562_3507 N/A	SUCCEEDED N/A	1447395836267 N/A	ivangm https://hadoop- slave2:8088/proxy/application_1445426559562_3507/jobhistory/job/job_1445426559562_3507	ivangm http://hadoop- slave2:8088/proxy/application_1445426559562_3507/jobhistory/job/job_1445426559562_3507	root.ivangm	NORMAL	N/A	N/A
job_1445426559562_0343 N/A	SUCCEEDED N/A	1445448206864 N/A	ivangm http://hadoop- slave2:8088/proxy/application_1445426559562_0343/jobhistory/job/job_1445426559562_0343	ivangm http://hadoop- slave2:8088/proxy/application_1445426559562_0343/jobhistory/job/job_1445426559562_0343	root.ivangm	NORMAL	N/A	N/A
job_1445426559562_3200 N/A	SUCCEEDED N/A	1447376870759 N/A	ivangm http://hadoop- slave2:8088/proxy/application_1445426559562_3200/jobhistory/job/job_1445426559562_3200	ivangm http://hadoop- slave2:8088/proxy/application_1445426559562_3200/jobhistory/job/job_1445426559562_3200	root.ivangm	NORMAL	N/A	N/A
job_1445426559562_0585 N/A	SUCCEEDED N/A	1445462654585 N/A	ivangm http://hadoop- slave2:8088/proxy/application_1445426559562_0585/jobhistory/job/job_1445426559562_0585	ivangm http://hadoop- slave2:8088/proxy/application_1445426559562_0585/jobhistory/job/job_1445426559562_0585	root.ivangm	NORMAL	N/A	N/A
job_1445426559562_0244 N/A	SUCCEEDED N/A	1445443127954 N/A	ivangm http://hadoop- slave2:8088/proxy/application_1445426559562_0244/jobhistory/job/job_1445426559562_0244	ivangm http://hadoop- slave2:8088/proxy/application_1445426559562_0244/jobhistory/job/job_1445426559562_0244	root.ivangm	NORMAL	N/A	N/A
job_1445426559562_2891 N/A	SUCCEEDED N/A	1447358353854 N/A	ivangm http://hadoop- slave2:8088/proxy/application_1445426559562_2891/jobhistory/job/job_1445426559562_2891	ivangm http://hadoop- slave2:8088/proxy/application_1445426559562_2891/jobhistory/job/job_1445426559562_2891	root.ivangm	NORMAL	N/A	N/A

Administración de trabajos

- **Estados de un trabajo**
 - RUNNING
 - SUCCEEDED
 - FAILED
 - IN PREPARATION
 - KILLED
- **Es fácil crear un tarea en cron para mostrar periódicamente una lista de los trabajos fallidos**

```
# mapred job -list all | grep FAILED
```

Administración de trabajos

➤ Para mostrar el estado de un determinado trabajo

```
# mapred job -status <job_id>
```

- % completado
- Valores de los contadores de sistema y definidos por el usuario
- No se muestra el nombre del trabajo
 - En este caso es mejor usar la interface Web

➤ Eliminar un trabajo

- Un trabajo no puede eliminarse con Ctrl-C

➤ Esto para la salida del trabajo por la consola, pero no el trabajo

```
# mapred job -kill <job_id>
```

Administración de trabajos

- Desde la interface Web no es posible detener trabajos
 - Es read-only, solo muestra información
 - Es posible activar la posibilidad de eliminar trabajos o tareas Map o Reduce individualmente
 - Añadir la siguiente propiedad a `core-site.xml`

```
<property>
    <name>webinterface.private.actions</name>
    <value>true</value>
</property>
```

- Reiniciar el JobTracker

Administración de trabajos

- Aparecerán nuevos botones para eliminar tareas
- Cualquiera con acceso a la web puede eliminar trabajos
 - Mejor usar la línea de comandos

horse Hadoop Map/Reduce Administration

State: RUNNING
Started: Thu Oct 10 20:08:37 EDT 2013
Version: 2.0.0-mr1-cdh4.4.0, Unknown
Compiled: Tue Sep 3 19:45:53 PDT 2013 by jenkins from Unknown
Identifier: 201310102008

Cluster Summary (Heap Size is 81.06 MB/253.94 MB)

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Bl
0	0	1	4	0	0	0	0	4	4	2.00	0

Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

<input type="checkbox"/> Select All	<input type="button" value="Kill Selected Jobs"/>	<input type="button" value="NORMAL"/>	<input type="button" value="Change"/>	Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total
<input type="checkbox"/>	<input type="button" value="job_201310102008_0002"/>	<input type="button" value="NORMAL"/>	<input type="button" value="Change"/>	job_201310102008_0002	NORMAL	training	word count	0.00%	1	0	0.00%	2

Planificación de trabajos

- Un trabajo Hadoop está compuesto por
 - Un conjunto desordenado de tareas Map con preferencias locales
 - Un conjunto desordenado de tareas Reduce
- Las tareas son planificadas por el JobTracker
 - Y son lanzadas/planificadas por los TaskTrackers
 - Un TaskTracker por nodo
 - Cada TaskTracker tiene un número fijo de slots para tareas Map y Reduce
 - Puede variar por nodo
 - Los TaskTracker reportan la disponibilidad de slots al JobTracker
- Planificar un trabajo requiere asignar tareas Map y Reduce a los slots de tareas disponibles

Planificación de trabajos

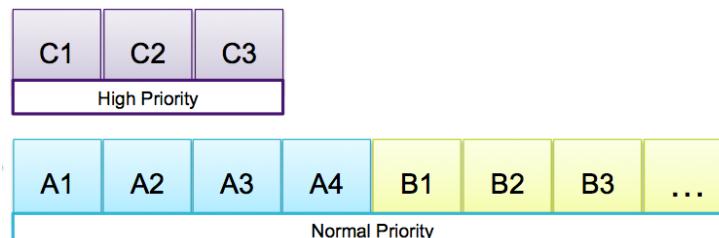
- El planificador por defecto de Hadoop es FIFO (First In, First Out)
 - Dados dos trabajos A y B, enviados en orden, todas las tareas Map del trabajo A son planificadas antes que cualquier tarea Map del trabajo B sea considerada
 - El orden de ejecución de tareas dentro de un trabajo pueden ser alterado



Planificación de trabajos

- El planificador FIFO soporta la asignación de prioridades a los trabajos
 - VERY_HIGH, HIGH, NORMAL, LOW, VERY_LOW
 - Se establecen en la prioridad *mapred.job.priority*
 - Se puede cambiar también desde la línea de comandos cuando el trabajo está ejecutándose

```
# mapred job -set-priority <job_id> <priority>
```
 - Hasta que no se termina el trabajo de la cola, no se procesa el siguiente



Planificación de trabajos

- El planificador FIFO puede presentar algún problema
 - Asumamos que el trabajo A tiene 2000 tareas y el trabajo B tiene 20
 - Incluso si el trabajo B tiene prioridad más alta, el trabajo B no avanzará hasta que casi haya finalizado el trabajo A
 - El tiempo de finalización debería ser proporcional al tamaño del trabajo
- Usuarios con poco conocimiento del sistema podrían indicar alta prioridad para todos sus trabajos afectando a otros trabajos
- La naturaleza “todo o nada” del planificador podría hacer que compartir los recursos del clúster sea complicado

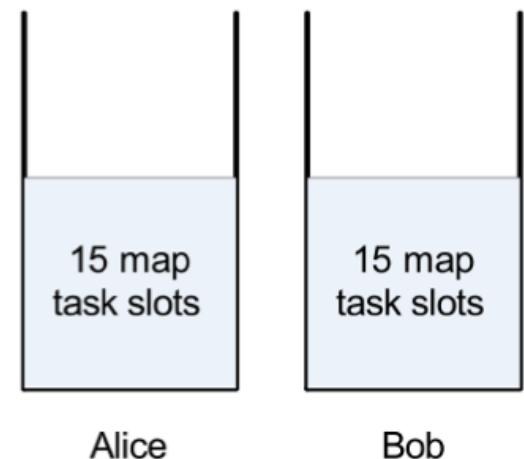
Planificación de trabajos

- La alternativa más justa es el planificador Fair, diseñado para permitir a múltiples usuarios compartir el clúster simultáneamente
 - Debería permitir a trabajos interactivos de corta duración coexistir con trabajos de producción de mayor duración
 - Debería permitir asignar los recursos proporcionalmente
 - Debería permitir utilizar el clúster de manera eficiente
 - Suele ser recomendado para entornos de producción

Planificación de trabajos

- El planificador Fair asigna a cada trabajo un banco de recursos (pool)
 - Por defecto un banco por usuario (se crea al lanzar la primera tarea)
 - Los trabajos podrían ser asignados a un banco determinado
 - Por ejemplo, “producción”
- Los slots no están asignados a ningún banco
 - Cada banco recibe una porción de los slots disponibles

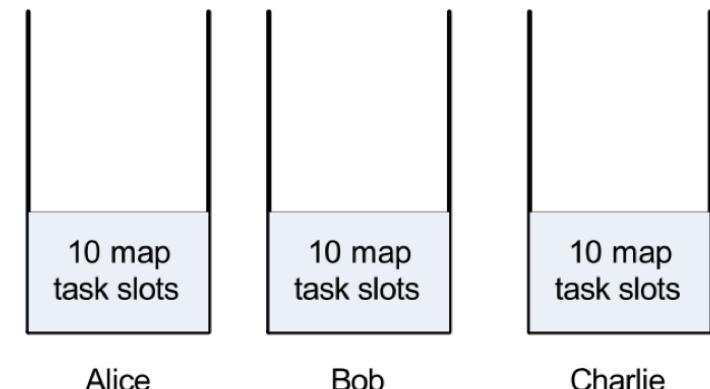
Total: 30 task slots



Planificación de trabajos

- Por defecto los bancos son creados dinámicamente a partir del nombre del usuario que envía las tareas
 - No es necesario configurar nada
- Los trabajos también se pueden enviar a bancos designados (por ejemplo, “producción”)
 - Se deben definir en un fichero de configuración
 - Deben definir un mínimo número de slots
- Si se crea un nuevo usuario se reasignarían los recursos del clúster

Total: 30 task slots



Planificación de trabajos

- El reparto *justo* de slots asignados a cada banco se basa en
 - El actual número de slots de tareas disponibles en el clúster
 - La demanda de cada banco (nº tareas que deben ejecutarse)
 - El mínimo configurado
- La asignación de slots a un banco nunca será superior que la actual demanda
- Los bancos se llenan desde su mínimo, asumiendo la capacidad del clúster
- Un exceso de capacidad en el clúster se reparte entre todos los bancos
 - El objetivo es mantener la mayor carga posible

Planificación de trabajos

- **Cuando hay múltiples trabajos en el mismo banco**
 - Se emplea la misma filosofía, los recursos dentro de cada banco se asignan mediante otro planificador Fair
 - Se podría forzar un planificador FIFO
 - El banco solo puede mantener un número determinado de trabajos ejecutándose al mismo tiempo
 - La prioridad de un trabajo dentro de un banco determina su “peso” en el banco

Planificación de trabajos

- Existe un tercer planificador, el planificador Capacity
 - Cada trabajo se asigna a una cola
 - Similar a Fair
 - A cada cola se le asigna un % de los recursos del clúster
 - Similar al *mínimo* del planificador Fair
 - Los recursos de una cola no usada no se asignan a otras
 - Al contrario que Fair, donde los bancos con pocos slots en uso pierden los slots
 - Los trabajos dentro de una misma cola se planifican en modo FIFO
 - Se pueden usar prioridades
 - La asignación de tareas puede ser basada en el uso de memoria de la tarea

Planificación de trabajos

Requirement	FIFO Scheduler	Fair Scheduler	Capacity Scheduler
Learning tool or proof of concept	✓		
Pool utilization varies, so it is desirable that pools give away resources when they are not in use		✓	
Jobs within a pool need to make equal progress		✓	
Data locality makes a significant difference in job run-time performance		✓	
Pool utilization has little fluctuation			✓
Jobs have a high degree of variance in memory utilization			✓

Mantenimiento del clúster - HDFS

- ***hdfs fsck* se puede emplear para comprobar si existen bloques de datos corruptos o perdidos**
 - Al contrario que *fsck* de Linux, no repara los errores
 - Se puede configurar para listar todos los ficheros
 - También los bloques de cada fichero, la localización de cada bloque, los racks
 - Ejemplos

```
hdfs fsck /
```

```
hdfs fsck / -files
```

```
hdfs fsck / -files -blocks
```

```
hdfs fsck / -files -blocks -locations
```

```
hdfs fsck / -files -blocks -locations -racks
```

Mantenimiento del clúster - HDFS

- Es buena idea ejecutar `hdfs fsck` como una tarea regular en un nodo del clúster, y que envíe un email al administrador
 - Eligiendo la hora adecuada, con bajo uso
- La opción `–move` mueve los ficheros corruptos a `/lost+found`
 - Un fichero corrupto es aquel donde todas las replicas de un bloque se han perdido
- La opción `–delete` elimina ficheros corruptos

Mantenimiento del clúster - HDFS

- El comando *dfsadmin* incluye una serie de herramientas de administración interesantes

- Listar la información de HDFS por nodo

```
# hdfs dfsadmin -report
```

- Releer los ficheros *dfs.hosts* and *dfs.hosts.exclude*

```
# hdfs dfsadmin -refreshNodes
```

- Guarda los metadatos del NameNode en disco y resetea el log

- Es necesario activa el “modo seguro” (a continuación)

```
# hdfs dfsadmin -saveNamespace
```

Mantenimiento del clúster - HDFS

- El comando *dfsadmin* permite poner manualmente el sistema de ficheros en “modo seguro”
 - Se inicia el NameNode en “modo seguro”
 - Solo lectura
 - No se replican o borran bloques
 - Se deja el “modo seguro” cuando el % mínimo (configurado) de bloques satisface la condición mínima de replicación
 - # hdfs dfsadmin -safemode enter
 - # hdfs dfsadmin -safemode leave
- Se puede bloquear hasta salir del modo seguro
 - # hdfs dfsadmin -safemode wait

Mantenimiento del clúster - HDFS

➤ Copiar datos entre clúster

- Muy útil para realizar backups
- Presenta un problema cuando se manejan grandes cantidades de datos
- *distcp* permite copiar datos dentro del clúster o entre clústeres
 - Perfecto para copiar grandes cantidades de datos
 - Utiliza MapReduce para realizar la copia
 - Se pueden copiar ficheros o directorios completos
 - Los ficheros previamente copiados se descartan, ya que detecta duplicados si coincide el nombre y el checksum

Mantenimiento del clúster - HDFS

- **Copiar datos de un clúster a otro**

```
# hadoop distcp hdfs://nn1:8020/path_to_src \
    hdfs://nn2:8020/path_to_dst
```

- **Copia de datos dentro del clúster**

```
# hadoop distcp /path_to_src /path_to_dst
```

- **En general no se suelen copiar datos entre clústeres, se suelen importar los datos a ambos clúster al mismo tiempo**

- Es más eficiente
- No es necesario ejecutar las tareas MapReduce en el clúster de backup
- Disponiendo de los datos iniciales es posible generar los datos finales más tarde

Mantenimiento del clúster - Añadir o eliminar nodos

➤ Añadir nodos

1. Añadir el nombre de los nodos a los ficheros de “inclusión” *dfs.hosts* y *mapred.hosts*
2. Actualizar el fichero con información del rack
3. Actualizar el NameNode con la nueva información
hdfs dfsadmin -refreshNodes
4. Actualizar el JobTracker con la nueva información
hdfs mradmin -refreshNodes
5. Iniciar las nuevas instancias DataNode y TaskTracker
6. Comprobar que los nuevos DataNode y TaskTracker aparecen en la Web

Mantenimiento del clúster - Añadir o eliminar nodos

- Por diseño, un NameNode no tendrá preferencia por los nuevos nodos a la hora de escribir nuevos bloques
 - Se asume que los nuevos datos se quieren procesar mediante trabajos MapReduce
 - Si todos los nuevos bloques se escriben en los nuevos nodos, esto afecta a la localización de los trabajos MapReduce
 - Crearía un “punto caliente” cuando se escriben nuevos datos

Mantenimiento del clúster - Añadir o eliminar nodos

➤ Eliminar nodos

1. Añadir los nombres de los nodos a los ficheros de “exclusión”
`dfs.host.exclude` y `mapred.hosts.exclude`
2. Actualizar el JobTracker con la nueva lista de nodos

```
# hdfs mradmin -refreshNodes
```
3. Actualizar el JobTracker con la nueva lista de nodos

```
# hdfs dfsadmin -refreshNodes
```

 - La web del NameNode mostrará el estado “Decommission in Progress” para esos nodos
 - Cuando todos DataNodes reporten “Decommissioned”, todos los bloques estarán replicados en otros nodos
4. Apagar los nodos eliminados
5. Eliminar los nodos de los ficheros de “inclusión” y “exclusión” y actualizar el NameNode de nuevo

Mantenimiento del clúster - Rebalanceo del clúster

- Un clúster HDFS puede estar “desbalanceado”
 - Algunos nodos tienen más datos que otros
 - Por ejemplo: al añadir un nuevo nodo
 - Incluso después de añadir nuevos datos al clúster, este nodo tendrá bastantes menos datos
 - Durante la ejecución de tareas MapReduce, este nodo usará mucho más ancho de banda de red para obtener datos de otros nodos
- Se puede balancear el clúster usando la utilidad `hdfs balancer`
 - Ajusta los bloques para asegurar que todos los nodos están dentro de un $x\%$ de utilización respecto al resto
 - x es un threshold

Mantenimiento del clúster - Rebalanceo del clúster

- Comando: `hdfs balancer –threshold x`
 - El `–threshold` es opcional. Se utiliza 10 por defecto
 - Se puede cancelar mediante Ctrl-C
 - Se puede controlar el ancho de banda empleado por el `balancer` mediante la propiedad `dfs.balance.bandwidthPerSec` en `hdfs-site.xml` (por defecto 1MB/s, recomendado 0,1 x velocidad de la red)
- Un nodo está infrautilizado si su utilización es menor que `utilización media – threshold`
- Un nodo es sobre-utilizado si su utilización es mayor que `utilización media + threshold`
- `hdfs balancer` la utilización por disco de cada nodo, solo la utilización del nodo como “un todo”

Mantenimiento del clúster - Actualizar el clúster

- Pasos generales
 - Parar MapReduce
 - Parar HDFS
 - Instalar la nueva versión de Hadoop
 - Iniciar el NameNode con la opción `--upgrade`
 - Monitorizar HDFS hasta que reporte que la actualización se ha completado
 - Iniciar MapReduce
- Si el clúster no presenta problemas durante varios días, finalizar la actualización mediante `hdfs dfsadmin --finalizeUpgrade`
 - Los DataNodes eliminarán los directorios de trabajo de la versión anterior, y también el NameNode

Mantenimiento del clúster - Actualizar el clúster

- Si aparecen problemas, se puede volver a la versión anterior (roll back) parando el clúster, e iniciando la versión anterior de HDFS con la opción `-rollback`
- Este procedimiento de actualización solo es necesario cuando cambian las estructuras de datos de HDFS o el formato de comunicación RPC
 - No es probable en versiones menores de Hadoop
 - Chequear la documentación

Monitorización del clúster

- Es necesario emplear una herramienta de monitorización que avise de problemas potenciales o que ya están ocurriendo en las máquinas del clúster
- Hadoop provee información que permite integración con diferentes herramientas de monitorización
 - JMX broadcast
 - Metrics sinks

Monitorización del clúster

- ¿Qué monitorizar?
 - Los demonios de Hadoop
 - Avisar si un demonio “ha caído”
 - Discos y particiones
 - Avisar si falla un disco
 - Enviar un aviso si un disco llega al 80%
 - Enviar un aviso crítico si un disco llega al 90%
 - Uso de CPU en nodos maestro
 - Enviar alerta si existe un uso excesivo
 - Los nodos esclavo pueden alcanzar el 100%. No es problema

Monitorización del clúster

- El swap de todos los nodos
 - Avisar si se usar la partición de swap → Problemas de memoria
- La red
- HDFS
- Logs de demonios MapReduce
 - Posibles problemas de falta de espacio en disco
 - Asegurar que “rotan” adecuadamente
 - Aplicaciones con “mucho salida” pueden generar logs muy grandes
 - Pueden dejar los nodos esclavo sin disco

Monitorización del clúster

- **Fuentes comunes de problemas**
 - Fallos en la configuración
 - Fallos de hardware
 - Uso excesivo de recursos
 - No suficiente disco
 - No suficiente memoria
 - No suficiente ancho de banda de red
 - Imposibilidad de llegar a otros nodos a través de la red
 - Problemas con el nombre de los nodos
 - Problemas con el hardware de red
 - Retardos en la red

Monitorización del clúster

- **Buenas prácticas**
 - Fallos en la configuración
 - Emplear los valores recomendados inicialmente
 - No suponer que los valores por defecto son correctos
 - Entender la precedencia de las propiedades
 - Controlar la capacidad de los usuarios de hacer cambios en la configuración
 - Comprobar que los cambios son buenos antes de pasarlos a producción
 - Comprobar los cambios de las nuevas versiones de Hadoop
 - Automatizar la administración de la configuración

Monitorización del clúster

- **Buenas prácticas**
 - Fallos de hardware y uso excesivo de recursos
 - Monitorizar los equipos
 - Medir el rendimiento de los equipos para entender su impacto en el clúster
 - Resolución de nombres
 - Chequear la resolución directa e inversa de DNS