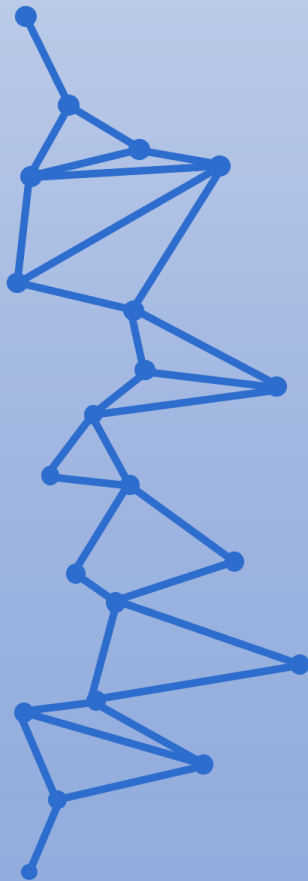




Curso de Especialización de Inteligencia Artificial y Big Data (IABD)



Big Data Aplicado

UD04. Administración y monitorización
de sistemas Hadoop.
Resumen.

JUAN ANTONIO GARCIA MUELAS

Big Data Aplicado UD04. Administración y monitorización de sistemas Hadoop

Para la **administración de Hadoop** distinguimos tres actividades: la **configuración**, **operación** de los **servicios** y la **monitorización**.

CONFIGURACIÓN.

Los **ficheros de configuración** suelen estar en el directorio `/etc/conf`.

Los principales ficheros de configuración de los servicios más importantes:

- ✓ **Hadoop core (HDFS, YARN y MapReduce):** para configurarlo, existen varios ficheros de configuración en los nodos master:
 - **core-site.xml:** con la **configuración** común de los **componentes core**, encontraremos propiedades de configuración como:
 - `fs.defaultFS`, que indica el **endpoint de HDFS** al que deben apuntar los clientes.
 - `hadoop.security.authentication`: tipo de **autenticación requerida** (Kerberos, simple).
 - **hdfs-site.xml:** con la **configuración específica de HDFS**, como por ejemplo:
 - `fs.namenode.name.dir`: **directorio local** en el **servidor** donde se aloja la información del **Namenode**.
 - `dfs.datanode.data.dir`: **directorio local** en el **servidor worker** donde se almacenan los **bloques de HDFS**.
 - `dfs.namenode.http-address`: **dirección** en la que se va a arrancar una **consola web de monitorización de HDFS**.
 - **yarn-site.xml:** con la **configuración específica de YARN**, como por ejemplo:
 - `yarn.resourcemanager.scheduler.class`: para la gestión de **prioridades**.
 - `yarn.resourcemanager.resource-tracker.address`: **dirección** del **ResourceManager** al que se conectarán los **NodeManager**.
 - `yarn.nodemanager.log-dirs`: **directorio** en los nodos **worker** en los que se escribirá el **log de la ejecución** de las aplicaciones.
 - `yarn.resourcemanager.webapp.address`: **dirección** en la que se va a arrancar una **consola web de monitorización** de YARN.
- ✓ **Apache Hive:** utiliza un fichero **hive-site.xml** para almacenar su **configuración principal**. Los principales parámetros de configuración son:
 - **hive.execution.engine:** indica qué **motor** se utilizará para la ejecución de consultas, teniendo 3 posibilidades:
 - MapReduce.
 - Tez.
 - Spark.
 - **hive.server2.enable.doAs:** permite **ejecutar las queries** con el usuario que hizo la petición, en lugar de con el usuario de sistema con el que se ejecuta Hive.
 - **hive.server2.thrift.port:** **puerto** en el que se levantará el **servidor Thrift** al que se conectarán los clientes de Hive.
- ✓ **Apache Spark:** utiliza el fichero **spark-defaults.conf**, que **no es de tipo XML** a diferencia de los ficheros de configuración que suelen tener las herramientas del ecosistema Hadoop. Sirve para **parametrizar los valores por defecto**, que se pueden modificar mediante código por las aplicaciones. Los parámetros más importantes son:
 - **spark.yarn.historyServer.address:** contiene la **ruta de YARN** donde se ejecuta Spark.

- **spark.history.ui.port:** puerto en el que se levantará un interfaz web de monitorización de Spark.
- **spark.eventLog.dir:** ruta en la que se almacenarán los ficheros de log con la ejecución de las aplicaciones.

Estos parámetros se pueden **modificar** de dos formas:

- ✓ Accediendo **por consola** a los ficheros y modificándolos con un editor tipo vi.
- ✓ Mediante una **herramienta de administración** como Apache Ambari o Cloudera Manager.

MONITORIZACIÓN.

Las principales **variables que se monitorizan** en un sistema **Linux** son:

- ✓ **Uso de CPU:** es normal encontrar picos de consumo de CPU, aunque no deberían ser constantes o durar más de unos segundos. Es importante revisar la media de consumo. El principal **comando para monitorizar el uso de CPU** es **top**.
- ✓ **Uso de la memoria:** la memoria es uno de los primeros lugares en los que se debe buscar cuando hay problemas de rendimiento. Los comandos que se utilizan principalmente para monitorizar la memoria son **vmstat**, **meminfo** y **free**.
- ✓ **Almacenamiento en disco:** primer lugar, verificar que no se esté quedando sin espacio. En segundo lugar, observar el rendimiento del disco: ¿hay cuellos de botella? Los **comandos para monitorizar el disco** son **iostat** y **sar**.
- ✓ **Tráfico de red:** la red es un componente importante de su sistema: si las conexiones de red son lentas, Hadoop funcionará lentamente. Las estadísticas de red simples, como la cantidad de bytes recibidos y enviados, ayudarán a identificar problemas de red. El principal **comando para monitorizar la red** es **dstat**.

Además de en Linux, **Hadoop** dispone de varias **herramientas de monitorización o administración** que veremos a continuación:

- ✓ Dos **interfaces** sencillos que proporcionan el **ResourceManager** y el **Namenode**.
- ✓ Apache **Ambari** y **Cloudera Manager** como herramientas de administración y monitorización más complejas.
- ✓ **Ganglia** como herramienta de monitorización general para clústers de servidores.

Interfaz de HDFS: Namenode UI.

HDFS **ofrece por defecto** una web de administración denominada **Namenode UI**, o **Web UI**.

El servicio Namenode UI permite ver cuánto espacio está ocupando HDFS.

Permite navegar por el sistema de ficheros de HDFS.

Es un **servicio web** que se arranca en el servidor donde se ejecuta este (por defecto, el puerto **50070**, aunque la podemos configurar en **hdfs-site.xml** con el parámetro **dfs.http.address**).

En la pestaña **Utilities**, se puede **acceder a los logs** de HDFS en el **Datanode**.

En la pantalla principal, se puede encontrar otras pestañas:

- ✓ **Datanode Volume Failures:** muestra los **errores en disco** que se han detectado en HDFS.
- ✓ **Snapshot:** en caso de haberse realizado una copia de la estructura de HDFS para poder **restaurarlo**, en esta pantalla aparecen todos los puntos generados.
- ✓ **Startup progress:** muestra cómo ha sido el **proceso de arranque de HDFS y el estado** en el que se encuentra ahora.

Interfaz de YARN: ResourceManager UI.

Permite **ver el estado de ejecución** de las aplicaciones, el estado de **recursos** del sistema, o ver el **detalle** de las aplicaciones **y los logs** que están generando.

En la opción **Applications**, se puede ver las aplicaciones que se han ejecutado o las que se están ejecutando en este momento.

La opción “**Scheduler**”, muestra el **estado de las colas de ejecución** que se han configurado en YARN, mostrando la capacidad de cada cola, su ocupación, etc

Apache Ambari.

Apache Ambari tiene como **objetivo simplificar la administración de Hadoop** para el aprovisionamiento, la administración y el monitoreo de clústeres.

Permite a los administradores del sistema:

- ✓ **Instalar** un clúster de Hadoop:
 - Ambari proporciona un **asistente paso a paso** para instalar los servicios de Hadoop en cualquier número de hosts.
Permite iniciar una instalación de Hadoop en un conjunto de servidores
 - Ambari maneja la **configuración de los servicios** de Hadoop para el clúster.
- ✓ **Administrar** un clúster Hadoop, ya que proporciona funcionalidades para **iniciar, detener y reconfigurar los servicios** de Hadoop en todo el clúster, así como los diferentes nodos o servidores que lo componen.
Ambari ofrece la posibilidad de que aplicaciones externas se comuniquen con ella para arrancar o parar servicios de forma automática.
- ✓ **Monitorizar** un clúster Hadoop, ya que:
 - Ambari proporciona **cuadros de mando a medida** para monitorear la salud y el estado del clúster de Hadoop.
 - Ambari permite **definir alertas** y notificará cuando se cumpla la condición de la alerta (por ejemplo, un nodo deja de funcionar, el espacio restante en el disco es bajo, etc.).

Cloudera Manager.

Fue la **primera herramienta** de administración de Hadoop.

La organización de opciones del menú de Cloudera Manager es muy parecida a la de Apache Ambari, separando los servicios (denominados Clústeres en Cloudera Manager) de los servidores donde está desplegado Hadoop (denominado Hosts).

Cloudera Manager tiene más o menos la misma funcionalidad que Ambari.

Podría simular la caída de un servidor para hacer pruebas de tolerancia a fallos.

Para saber más

Si deseas obtener más información de Cloudera Manager, puedes acceder a la web oficial en [este enlace](#).

Asimismo, en la página de Cloudera tienes muchos vídeos de demostración de diferentes funcionalidades de Cloudera Manager. Puedes verlos en [este enlace](#).

Ganglia.

Ganglia es una **herramienta opensource** que permite la **recogida de métricas** de un sistema y su **monitorización**. No es una herramienta específica (**No pertenece al ecosistema**) de Hadoop, ya que su propósito es ayudar en la monitorización de cualquier tipo de clúster.

Hay cuatro componentes principales en un sistema de monitoreo de Ganglia:

- ✓ **gmond:** es un **demonio que se ejecuta en cada nodo** del clúster, cuyo trabajo es recopilar los datos de métricas de cada nodo. Cada nodo ejecuta el demonio gmond y el nodo recibirá métricas del resto de los nodos del clúster (todos los nodos se comunican con todos los nodos). Esto significa que el proceso de recogida de métricas (gmetad, que se explica a continuación) necesita solo un nodo para obtener las métricas del clúster y también en caso de un fallo de un nodo, Ganglia sigue proporcionando servicio.
- ✓ **gmetad:** este es el **demonio que sondea los nodos** en busca de datos de las métricas. Puede obtener un volcado de métricas para todo el clúster desde cualquier nodo del clúster. El demonio gmetad **crea tablas denominadas RRD** para almacenar los datos de métricas.
- ✓ **RRDtool:** este componente **almacena los datos** de las métricas recogidas por el demonio gmetad en cada nodo.
- ✓ **gweb:** esta es la **interfaz web para visualizar las métricas recopiladas** por el sistema de monitoreo de Ganglia a través de los datos almacenados en las bases de datos de RRD. Puede ver métricas específicas mediante gráficos y también crear gráficos personalizados profundizando en los detalles de una métrica o host específico. El proceso gweb es en realidad un programa PHP que se ejecuta en un servidor web Apache.