

Cluster Big Data

Jesús Morán

Grupo de Investigación en Ingeniería del Software

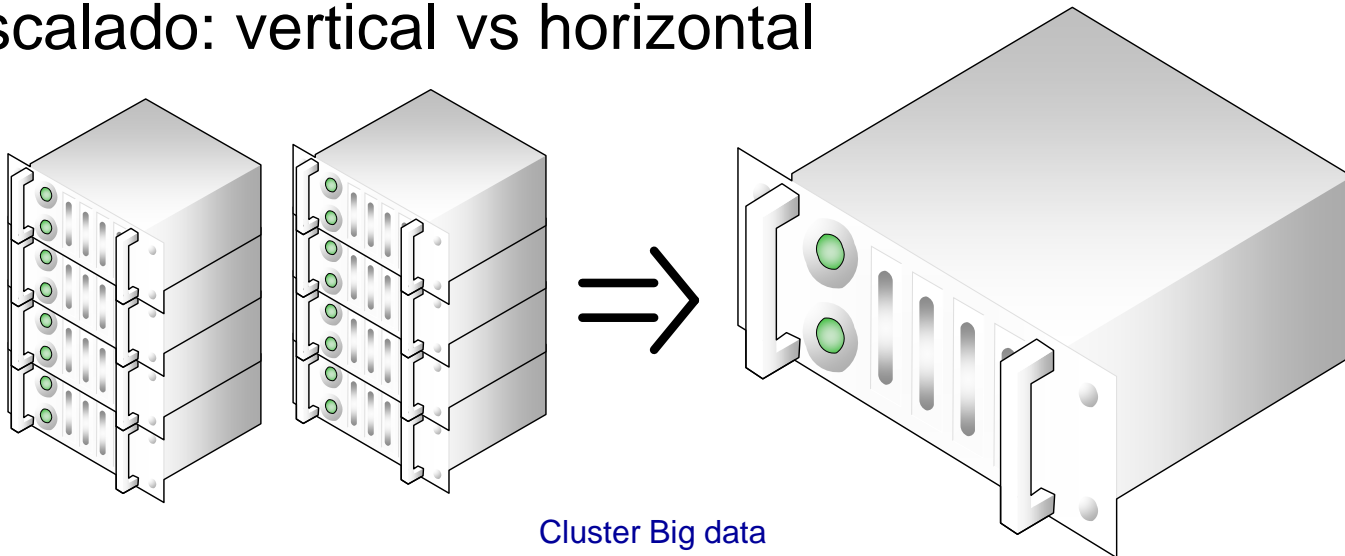
<http://giis.uniovi.es>

Universidad de Oviedo



Cluster

- Varios servidores que trabajan como si fuese uno
 - Rendimiento: tareas con alta memoria, CPU, etc
 - Disponibilidad: tolerancia a fallos
 - Eficiencia: gestión de recursos y tareas
 - Escalado: vertical vs horizontal



Administración y gestión

- Servidores y racks
- Red y conexiones
- Sistemas de archivos distribuidos
- Gestión de usuarios
- Permisos y seguridad
- Configuraciones y políticas de acceso
- Monitorización
- Fallos de servidores
- Recursos en caliente
- Interoperabilidad de servicios
- Balanceo de carga

Tipos de clusteres

- Altas prestaciones (HPC)
 - Requieren el análisis de muchos nodos
 - Paradigmas distribuidos
- Alta disponibilidad
 - Replicaciones
 - Sin puntos de fallo únicos
 - Servicio disponible (si falla se restaura)
- Balanceo de carga
 - Varios nodos corren en mismo servicio y se balancea el acceso

Aplicaciones de clusteres

- **Análisis y cálculo:**
 - Bases de datos: SQL, NoSQL, newSQL
 - Big Data
- **Renderizado**
- **Desencriptación**
- **Simulaciones:**
 - Mecánica
 - Química
 - ...

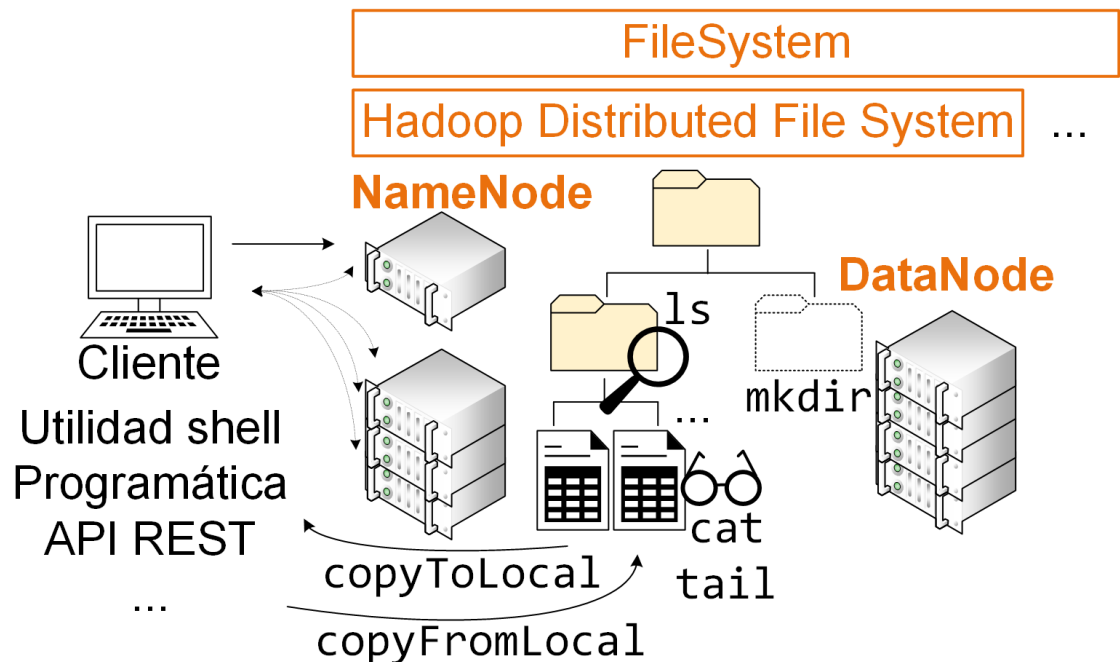
■ Framework Big Data

-
- The diagram illustrates the Hadoop ecosystem components and their interactions:
- Usuario (User):** Represented by a tree structure of folders and documents, indicating the source of data for processing.
 - MapReduce, Storm, Spark, ...:** These are the distributed processing frameworks that interact with the Hadoop ecosystem.
 - Hadoop:** The core distributed storage and processing system, enclosed in a dashed oval. It consists of:
 - Mapper:** The processing units that execute MapReduce jobs. Multiple mappers are shown on both sides of the Hadoop cluster.
 - Resource Manager:** Manages the cluster's resources and schedules tasks.
 - Datanode + NodeManager:** These components are responsible for storing data (Datanode) and managing the node's resources (NodeManager). Multiple instances are shown on both sides of the Hadoop cluster.
 - History Server:** Stores the execution history of MapReduce jobs.
 - Secondary NameNode:** Provides a backup of the NameNode's metadata.
 - Standby Namenode:** A standby instance of the NameNode.

Sistema de archivos distribuido

- Bloques de 128MB (por defecto)
- Replicación en 3 servidores (por defecto)
- Componentes:

- ☐ Namenode
- ☐ Datanode



Procesamiento

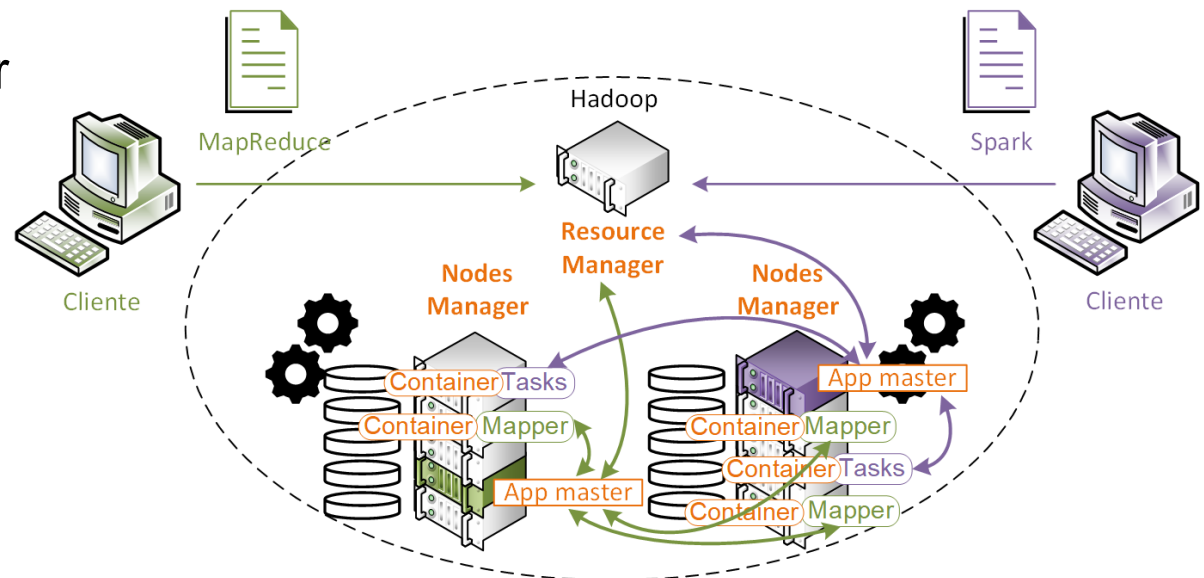
■ Componentes:

□ Resource Manager

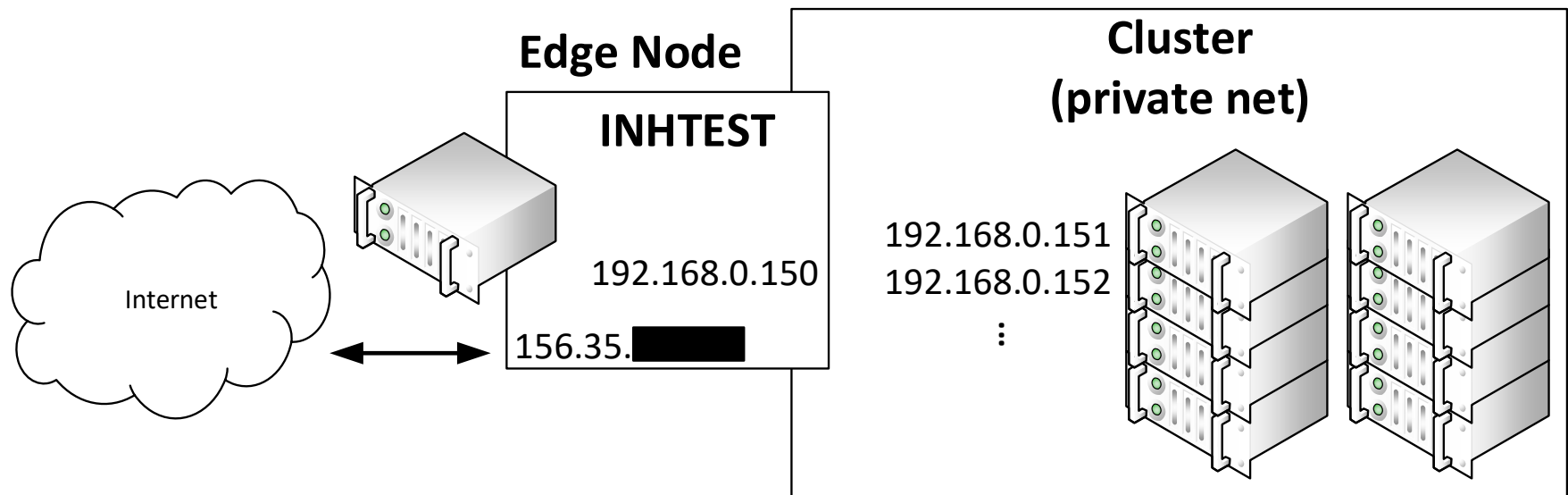
□ NodeManager

■ AppMaster

■ Container



Cluster a desplegar



Instalaciones de Hadoop

- Operativo: Almalinux
- Asignar IP (ej. DHCP)
- Instalar Java
 - Variable de entorno \$JAVA_HOME
 - Enlace simbólico:

```
alternatives --install /usr/bin/java java /usr/...
```
- Crear usuario hadmin y grupo hadoop
- Descargar Hadoop en /usr/local (ej. wget)
- Asignar a hadmin y hadoop como propietarios
- Crear carpeta de archivos temporales en /hadoopTemp

Instalaciones en cada nodo (red)

■ Nombre de hosts y red

■ Abrir puertos [maestro = INHTEST]

- ☐ NameNode: 9000 y 50070 (sólo maestro)
- ☐ Secondary NameNode: 50090 (sólo maestro)
- ☐ DataNode: 50020, 50010 y 50075
- ☐ ResourceManager: 8088, 8031, 8032 y 8030 (sólo maestro)
- ☐ NodeManager: 8042, 8040 y 10200
- ☐ HistoryServer: 19888 y 10020
- ☐ Shuffle: 13562
- ☐ Hadoop ephemeral ports: 50200:51200
- ☐ Otros puertos de Spark, Flink, etc. (generalmente sólo maestro)

Instalaciones en cada nodo (seguridad)

- Instalamos ssh
- Creamos una clave pública/privada
`ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa`
- La ponemos como clave autorizada:
`cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`
- Las copiamos a otros nodos:
`ssh-copy-id -i ~/.ssh/id_rsa.pub hadmin@INHTEST2`

Instalaciones en cada nodo (env)

- Crear variables de entorno de Hadoop:
 - `HADOOP_HOME=/usr/local/hadoop/hadoop-x.y.0`
 - `HADOOP_MAPRED_HOME=$HADOOP_HOME`
 - `HADOOP_COMMON_HOME=$HADOOP_HOME`
 - `HADOOP_HDFS_HOME=$HADOOP_HOME`
 - `HADOOP_YARN_HOME=$HADOOP_HOME`
 - `HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop`
 - `HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native`
 - `export HADOOP_OPTS="-Djava.library.path=/usr/local/hadoop/hadoop-2.2.0/lib"`
- Añadir `$HADOOP_HOME/bin` al path

Configuración en todos los nodos

■ Configuramos Hadoop

- \$HADOOP_HOME/etc/hadoop/core-site.xml

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://INHTEST1:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/hadoopTemp</value>
  </property>
</configuration>
```

Configuración en todos los nodos

■ Configuramos Hadoop

- \$HADOOP_HOME/etc/hadoop/core-site.xml
- \$HADOOP_HOME /etc/hadoop/yarn-site.xml
 - yarn.resourcemanager.hostname: INHTEST1
 - yarn.resourcemanager.address: INHTEST1:8032
 - yarn.resourcemanager.resource-tracker.address: INHTEST1:8031
 - yarn.resourcemanager.scheduler.address: INHTEST1:8030
 - yarn.log.server.url: http://INHTEST1:19888/jobhistory/logs/
 - yarn.nodemanager.aux-services: mapreduce_shuffle
 - yarn.nodemanager.aux-services.mapreduce_shuffle.class: org.apache.hadoop.mapred.ShuffleHandler
 - yarn.log-aggregation-enable: true

Configuración en todos los nodos

■ Configuramos Hadoop

- \$HADOOP_HOME/etc/hadoop/core-site.xml
- \$HADOOP_HOME /etc/hadoop/yarn-site.xml
- \$HADOOP_HOME /etc/hadoop/hdfs-site.xml
 - dfs.data.dir: file:///usr/local/hadoop/Data/Datanode
 - dfs.name.dir: file:///usr/local/hadoop/Data/Namenode
 - dfs.namenode.secondary.http-address: INHTEST1:50090
 - dfs.namenode.http-address: INHTEST1:50070
 - dfs.namenode.secondary.http-address: INHTEST1:50090
 - dfs.permissions.superusergroup: hadoop
 - dfs.replication: 3
 - dfs.permissions.enabled: true

Configuración en todos los nodos

■ Configuramos Hadoop

- \$HADOOP_HOME/etc/hadoop/core-site.xml
- \$HADOOP_HOME /etc/hadoop/yarn-site.xml
- \$HADOOP_HOME /etc/hadoop/hdfs-site.xml
- \$HADOOP_HOME /etc/hadoop/mapred-site.xml
 - mapreduce.jobtracker.http.address: INHTEST1:50030
 - mapreduce.jobhistory.address: INHTEST1:10020
 - mapreduce.jobhistory.webapp.address: INHTEST1:19888
 - yarn.app.mapreduce.am.job.client.port-range: 50200-51200
 - mapreduce.framework.name: yarn
 - mapreduce.jobtracker.staging.root.dir: /user

Configuración del maestro

- Indicar los esclavos:
`$HADOOP_HOME/etc/hadoop/slaves`
- Formateamos el sistema de archivos distribuido:
`hadoop namenode -format`
 - ☐ Formatea los metadatos del namenode ~ formatear sistema de archivos
- Arrancar los servicios:
`/usr/local/hadoop/hadoop-2.2.0/sbin/start-all.sh`
- Crear las carpetas de usuario:
`hadoop fs -mkdir /user`
`hadoop fs -mkdir /user/hadmin`

Crear un cliente

- Tiene que ser un Edge node
- Instalar:
 - ☐ Java
 - ☐ Hadoop (configuración de Hadoop)
 - ☐ ssh
- Crear usuario
 - ☐ Creamos usuario local con grupo hadoop
 - ☐ Crear carpeta en sistema de archivos distribuido
`hadoop fs -mkdir /user/cristian/`
 - ☐ Ponerlo como propietario
`hadoop fs -chown -R cristian:hadoop /user/cristian/`

Servicios instalados

- Sistema de archivos distribuido: HDFS (maestro y esclavos)
- Gestor de recursos: YARN (maestros y esclavos)
- Motores de procesamiento: Hadoop, Spark,...
- Otros:
 - ☐ Servidores web del cluster
 - ☐ Historial
 - ☐ Logs distribuidos

Otros servicios

- Ganglia: monitorizar el uso de recursos
- Nagios: alertar de problemas en el cluster
- Kerberos: autenticación integrada con Hadoop
- Squid:
 - Proxy transparente para actualizar los operativos
 - SSL BUMP man-in-the-middle para otras instalaciones
- Otros frameworks/librerías de cálculo:
 - R (para utilizar SparkR)
 - Zeppelin: notebook Big Data
 - ...

Tutorial SparkR Uniovi



- 3.2 Mostrar primeros datos
- 3.3 Dimensiones del dataset (fi
- 3.4 Seleccionar columnas
- 3.5 Filtrar columnas
- 3.6 Resumen de las columnas
- 4 Operaciones básicas de Spark
 - 4.1 Agregaciones de columnas
 - 4.2 Resúmenes estadísticos de
 - 4.3 Resúmenes estadísticos er
 - 4.4 Ordenación
 - 4.5 Añadir columnas
 - 4.6 Eliminar columnas
 - 4.7 Añadir filas
 - 4.8 Obtener datos únicos
 - 4.9 Muestreo
 - 4.10 Eliminar valores nulos
 - 4.11 Filtrar valores nulos
- 5 Funciones definidas por el usu:
 - 5.1 dapply
 - 5.2 gapply
 - 5.3 lapply
- 6 Operaciones sobre varios Data
 - 6.1 Unión de columnas (join)
 - 6.2 Unión de filas (rbind)
 - 6.3 Intersección
 - 6.4 Resta de filas (except)
- 7 Operaciones de programación
 - 7.1 SQL
 - 7.2 Sustituir cada dato por otro
- 8 Funciones específicas de Spar
 - 8.1 Guardar DataFrame
 - 8.2 Cacheo de información
 - 8.3 Crear un dataframe de R
- 9 Visualización de datos
- 10 Machine learning en Big Data

```
%spark.r
nrow(myDF)
```

FINISHED    

[1] 15542579

Took 31 sec. Last updated by admin at February 27 2019, 6:51:16 PM.

```
%spark.r
ncol(myDF)
```

FINISHED

[1] 24

Took 0 sec. Last updated by admin at February 16 2019, 7:32:08 PM.

3.4 Seleccionar columnas

READY

select

Notar que \$ es una operación de R y no de SparkR, aunque a veces la podemos utilizar para indicar la columna a la que nos referimos

```
%spark.r
head(select(myDF, "Bottle Volume (ml)"))
```

FINISHED

Bottle Volume (ml)	
1	600


Cluster

History Server - Mozilla Firefox

Universidad de Oviedo - Inicio X History Server X +

inhtest1:18080 90% ... ✓ ☆ Buscar

Más visitados JobHistory

 **History Server**

Event log directory: hdfs://inhtest1:9000/tmp/spark-logs

Last updated: 26/2/2019, 18:02:36

Show 20 entries Search:

App ID	App Name	Attempt ID	Started	Completed	Duration	Spark User	Last Updated	Event Log
application_1523705380604_0242	Zeppelin		2019-02-18 22:31:46	2019-02-21 19:21:23	68.8 h	hadmin	2019-02-21 19:21:23	Download
application_1523705380604_0236	Zeppelin		2019-02-17 09:23:06	2019-02-18 22:12:05	36.8 h	hadmin	2019-02-18 22:12:05	Download
application_1523705380604_0241	example dapply		2019-02-18 17:03:06	2019-02-18 17:06:31	3.4 min	hadmin	2019-02-18 17:06:31	Download
application_1523705380604_0240	example dapply		2019-02-18 16:29:49	2019-02-18 16:36:09	6.3 min	hadmin	2019-02-18 16:36:09	Download
application_1523705380604_0238	my SparkR example		2019-02-17 20:33:00	2019-02-17 20:56:52	24 min	hadmin	2019-02-17 20:56:53	Download
application_1523705380604_0235	Zeppelin		2019-02-16 18:20:11	2019-02-17 09:22:18	15.0 h	hadmin	2019-02-17 09:22:18	Download
application_1523705380604_0234	my SparkR example		2019-02-14 23:07:46	2019-02-14 23:08:27	41 s	hadmin	2019-02-14 23:08:27	Download
application_1523705380604_0233	Zeppelin		2019-02-12 16:39:05	2019-02-12 16:42:02	3.0 min	hadmin	2019-02-12 16:42:02	Download
application_1523705380604_0232	Zeppelin		2019-02-12 16:00:33	2019-02-12 16:05:11	4.6 min	hadmin	2019-02-12 16:05:11	Download
local-1549987136877	Zeppelin		2019-02-12 15:58:54	2019-02-12 16:00:12	1.3 min	hadmin	2019-02-12 16:00:12	Download
application_1523705380604_0231	my SparkR example		2019-02-12 15:40:55	2019-02-12 15:41:35	40 s	hadmin	2019-02-12 15:41:36	Download
application_1523705380604_0230	Zeppelin		2019-02-12 15:19:10	2019-02-12 15:24:39	5.5 min	hadmin	2019-02-12 15:24:39	Download
application_1523705380604_0229	Zeppelin		2019-02-12 13:16:14	2019-02-12 15:18:50	2.0 h	hadmin	2019-02-12 15:18:50	Download
application_1523705380604_0227	Zeppelin		2019-02-12 12:32:18	2019-02-12 13:14:59	43 min	hadmin	2019-02-12 13:14:59	Download
application_1523705380604_0228	my SparkR example		2019-02-12 12:33:32	2019-02-12 12:34:59	1.4 min	hadmin	2019-02-12 12:34:59	Download

Universidad de Oviedo - Inicio Zeppelin - Details for Stage 20 (Attempt 0) +

inhtest1:18080/history/application_15 90% ... ✓ ☆ Buscar

Más visitados JobHistory

APACHE **spark** 2.1.2 Jobs Stages Storage Environment Executors SQL Zeppelin application UI

Details for Stage 20 (Attempt 0)

Total Time Across All Tasks: 19 min

Locality Level Summary: Node local: 32

Input Size / Records: 3.9 GB / 15542579

Shuffle Write: 38.3 KB / 32

► [DAG Visualization](#)

► [Show Additional Metrics](#)

► [Event Timeline](#)

Job Id (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
18 (zeppelin-2E3PDKKPK-20190209-135527_150275013)	+Started by: admin describe at NativeMethodAccessorImpl.java:0	2019/02/16 19:34:18	9,7 min	2/2	33/33

Summary Metrics for 32 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	17 s	36 s	36 s	37 s	40 s
GC Time	0,2 s	0,3 s	0,3 s	0,3 s	0,3 s
Input Size / Records	59.2 MB / 228060	128.0 MB / 492773	128.0 MB / 492818	128.0 MB / 494118	128.0 MB / 500576
Shuffle Write Size / Records	1189.0 B / 1	1215.0 B / 1	1227.0 B / 1	1238.0 B / 1	1262.0 B / 1

Aggregated Metrics by Executor

Executor ID	Address	Task Time	Total Tasks	Failed Tasks	Killed Tasks	Succeeded Tasks	Input Size / Records	Shuffle Write Size / Records
1	INHTEST2:50595	9,7 min	16	0	0	16	2.0 GB / 7902264	19.1 KB / 16
2	INHTEST3:50616	9,5 min	16	0	0	16	1979.2 MB / 7640315	19.2 KB / 16

Tasks (32)



Jesús Morán

Grupo de Investigación en Ingeniería del Software

<http://giis.uniovi.es>

Universidad de Oviedo

