Máster en Ingeniería del Software: Cloud, Datos y Gestión TI.

Big Data

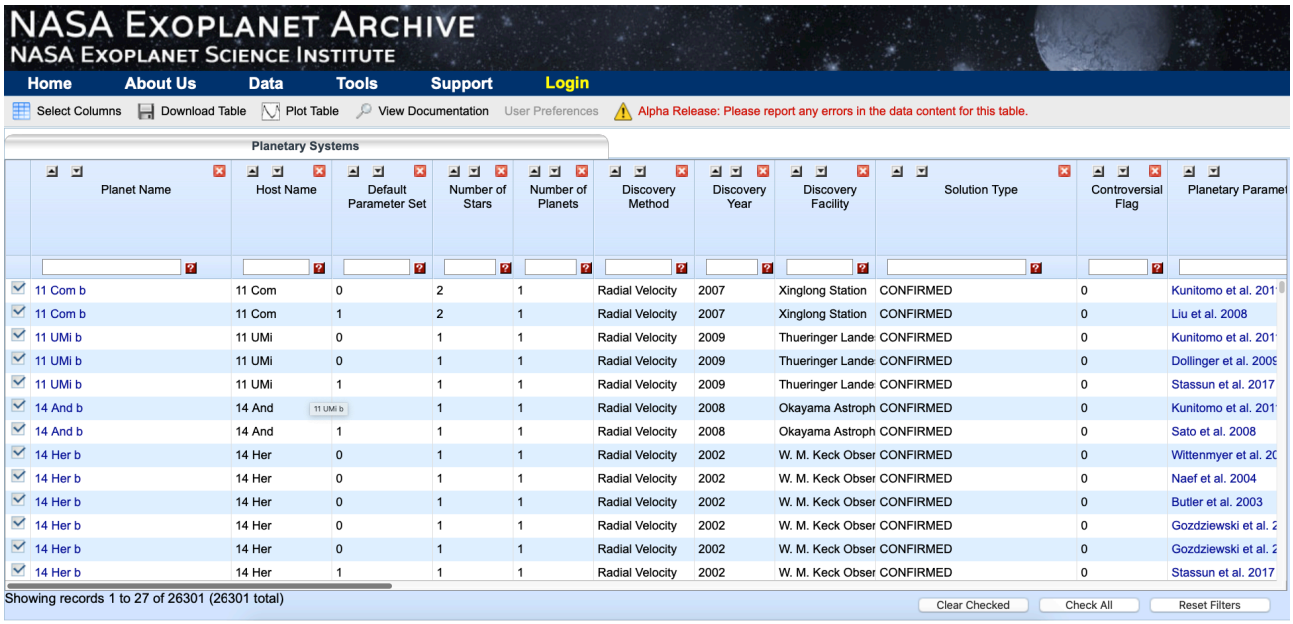
Práctica 4: Spark Core

Alumno: Agustín Núñez Arenas

Propuesta del alumno

El conjunto de datos que se ha tratado en esta práctica y que utilizaremos durante esta segunda parte de la asignatura, serán datos de **exoplanetas** en un archivo que dispone el Centro de Análisis y Procesamiento Infrarrojo de la NASA. Forma parte del Instituto Tecnológico de California [1].

El conjunto de datos presenta unas 82 columnas y 26.301 filas. Cada fila representa un exoplaneta descubierto, es decir, un planeta con órbita distinta a la del Sol, y contiene datos de su radio, su flujo estelar, la distancia a la tierra, entre otros más.



El objetivo es calcular, de entre todos esos exoplanetas, cuál se parece más a la Tierra. Para ello se va a calcular el Índice de Similitud con la Tierra de todos los exoplanetas: ESI, del inglés *Earth Similarity Index*. [2]

$$ESI(S,R) = 1 - \sqrt{\frac{1}{2} * \left[\left(\frac{S - S_\oplus}{S + S_\oplus} \right)^2 + \left(\frac{R - R_\oplus}{R + R_\oplus} \right)^2 \right]}$$

S = Flujo estelar del exoplaneta $\quad S_\oplus$ = Flujo estelar de la Tierra

R = Radio del exoplaneta $\quad R_\oplus$ = Radio de la Tierra

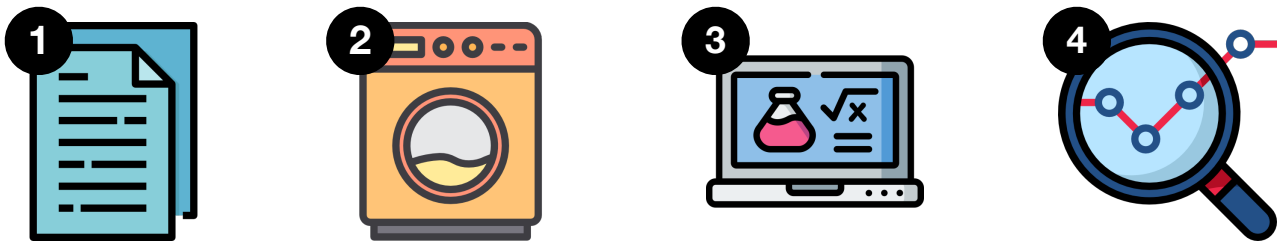
Este índice no es una medida de habitabilidad, pero da una medida de probabilidad de que sea un planeta rocoso y disponga de una temperatura superficial moderada.

Para dar un ejemplo, Marte tiene un ESI de 0.73

Un índice de 1 equivale a que ese planeta es idéntico a la Tierra.

Esquema del proceso

Para este cómputo vamos a seguir el siguiente esquema en el tratamiento con RDDs en Spark.



1. **Lectura de los datos en crudo.** No hay complejidad aquí, tan solo hacemos uso de los métodos para leer ficheros y los separamos por el delimitado de la coma **“,”**.

2. **Limpieza de los datos.** Hay filas que no presentan las columnas que guardan los datos del radio o del flujo, esas las vamos a descartar, ya que no se puede calcular el índice de ninguna otra forma. Además las medidas del radio están en unidades con respecto a Júpiter y no con la tierra. Para hacer la conversión, multiplicamos por el ratio de Júpiter y Tierra con respecto a su radio. [3]

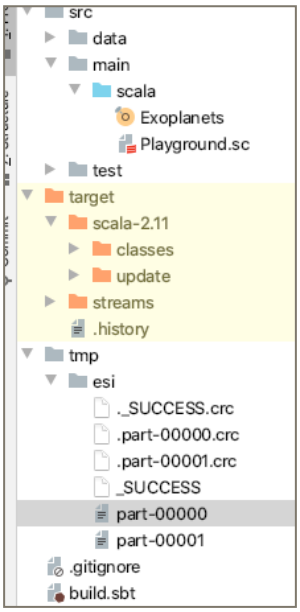
3. **Cálculo del índice ESI.** El cálculo se hace con una transformación map. Devuelve el nombre del exoplaneta y su ESI. Luego se hace otra operación de map-reduce para obtener la media de los ESI en caso de que haya varios ESI repetidos por planeta. Esto es porque el dataset contiene datos del mismo exoplaneta pero con diferentes estrellas de referencia. Los exoplanetas se miden con estrellas estáticas fijas en el cielo como referencia. Es una práctica común medir el mismo exoplaneta con distintas estrellas de referencia, para una mejor precisión de la medición.

4. **Filtrado de los mejores índices.** Cogemos sólo los datos de los planetas que tienen ESI > 0.6

Pasos seguidos

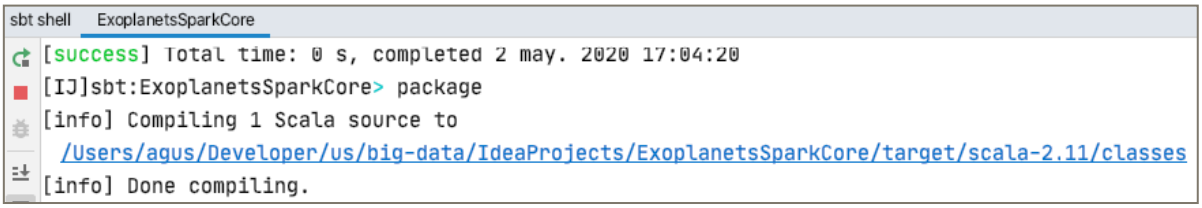
1. Programación en local

Se ha elegido el IDE IntelliJ IDEA para programar la clase que se ejecutará para calcular los índices ESI. Una vez el programa corre sin problemas en local y arroja los resultados esperados es momento de cambiar los parámetros de lectura y escritura para que funcionen en la máquina virtual de Hortonworks



2. Compilación del proyecto en un ejecutable

Con la ejecución del comando `sbt package` obtenemos el ejecutable en formato jar.



Ahora aprovechamos que tenemos un repositorio donde guardamos todo el código y el dataset para añadir también este fichero jar y clonarlo en la máquina virtual.

En la máquina virtual tan solo ejecutamos:

```
$ > git clone https://github.com/Agusnez/ExoplanetsSparkCore
$ > cd ExoplanetsSparkCore
```

El dataset lo importamos a HDFS mediante el panel de control de Ambari.

3. Ejecución del programa

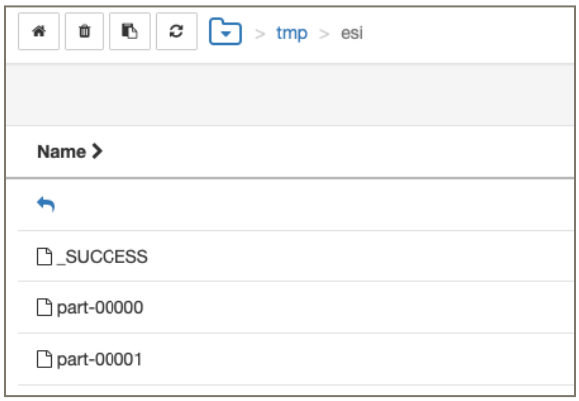
Tenemos todo listo para ejecutar el programa en nuestra máquina. Con el siguiente comando lo ejecutaremos en Hortonworks:

```
$ > spark-submit --class Exoplanets --master local ./
exoplanetssparkcore_2.11-0.1.jar
```

Esto genera una traza así:

```
[maria_dev@sandbox-hdp ~]$ cd ExoplanetsSparkCore/
[maria_dev@sandbox-hdp ExoplanetsSparkCore]$ spark-submit --class Exoplanets --master local ./exoplanetssparkcore_2.11-0.1.jar
SPARK_MAJOR_VERSION is set to 2, using Spark2
20/05/02 15:29:54 INFO SparkContext: Running Spark version 2.3.0.2.6.5.0-292
20/05/02 15:29:55 INFO SparkContext: Submitted application: ESIExoplanets
20/05/02 15:29:55 INFO SecurityManager: Changing view acls to: maria_dev
20/05/02 15:29:55 INFO SecurityManager: Changing modify acls to: maria_dev
20/05/02 15:29:55 INFO SecurityManager: Changing view acls groups to:
20/05/02 15:29:55 INFO SecurityManager: Changing modify acls groups to:
20/05/02 15:29:55 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permissions
with modify permissions: Set(maria_dev); groups with modify permissions: Set()
20/05/02 15:29:55 INFO Utils: Successfully started service 'sparkDriver' on port 39427.
20/05/02 15:29:55 INFO SparkEnv: Registering MapOutputTracker
20/05/02 15:29:55 INFO SparkEnv: Registering BlockManagerMaster
20/05/02 15:29:55 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology inf
20/05/02 15:29:55 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
20/05/02 15:29:55 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-d6a579ac-60d9-463b-a371-8749348f5836
20/05/02 15:29:55 INFO MemoryStore: MemoryStore started with capacity 366.3 MB
20/05/02 15:29:55 INFO SparkEnv: Registering OutputCommitCoordinator
20/05/02 15:29:55 INFO log: Logging initialized @1851ms
20/05/02 15:29:55 INFO Server: jetty-9.3.z-SNAPSHOT
20/05/02 15:29:55 INFO Server: Started @1936ms
```

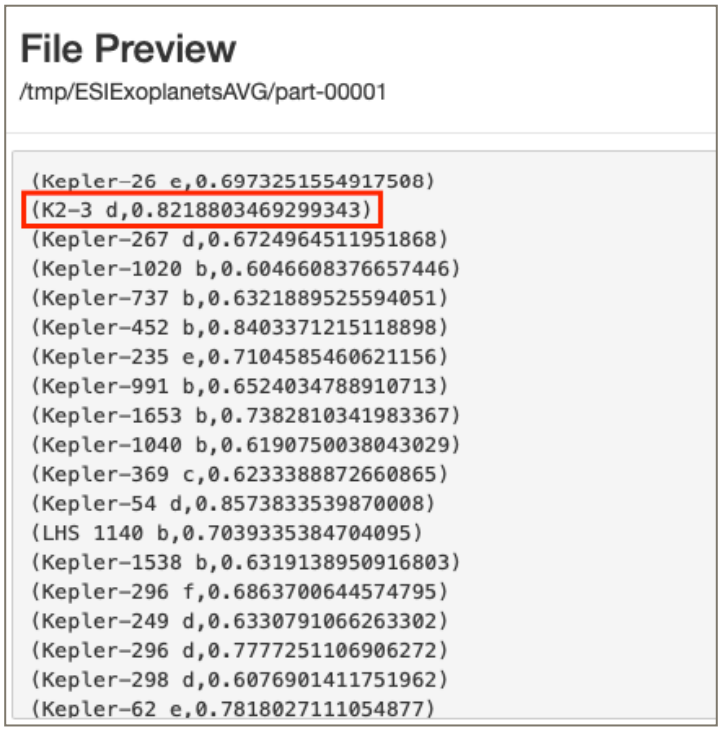
Y al no haber fallos, si nos dirigimos con la interfaz de Ambari a la ruta donde habíamos indicado en el programa */tmp/ESIExoplanets*



Con esto podemos observar los resultados si hacemos click en los ficheros `part-x`

4. Resultados

Vamos a echar un ojo a los resultados obtenidos:



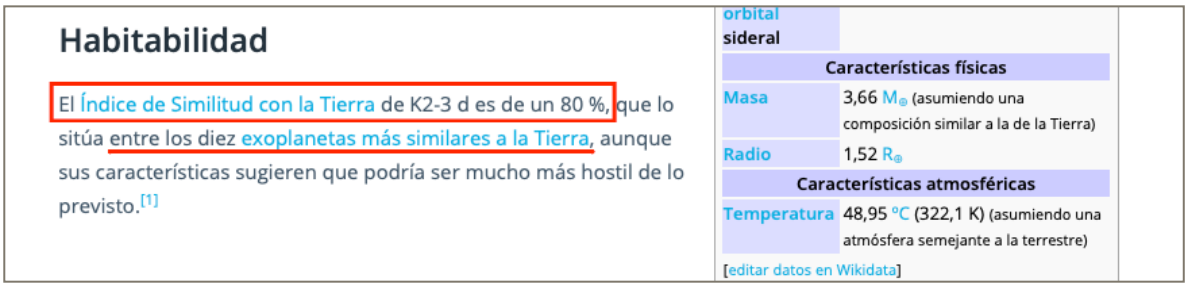
File Preview

/tmp/ESIExoplanetsAVG/part-00001

```
(Kepler-26 e,0.6973251554917508)
(K2-3 d,0.8218803469299343)
(Kepler-267 d,0.6724964511951868)
(Kepler-1020 b,0.6046608376657446)
(Kepler-737 b,0.6321889525594051)
(Kepler-452 b,0.8403371215118898)
(Kepler-235 e,0.7104585460621156)
(Kepler-991 b,0.6524034788910713)
(Kepler-1653 b,0.7382810341983367)
(Kepler-1040 b,0.6190750038043029)
(Kepler-369 c,0.6233388872660865)
(Kepler-54 d,0.8573833539870008)
(LHS 1140 b,0.7039335384704095)
(Kepler-1538 b,0.6319138950916803)
(Kepler-296 f,0.6863700644574795)
(Kepler-249 d,0.6330791066263302)
(Kepler-296 d,0.7777251106906272)
(Kepler-298 d,0.6076901411751962)
(Kepler-62 e,0.7818027111054877)
```

Podemos ver que el exoplaneta **K2-3 d** tiene un ESI de 0.82, un valor bastante alto. Eso quiere decir que se parecería a nuestro planeta Tierra.

Si queremos, a modo de *fact-check,* comprobar nuestro "descubrimiento", podemos localizar este planeta en Google y ver, que efectivamente, se encuentra como uno de los exoplanetas más estudiados y similares a la tierra. [4]



Habitabilidad

El Índice de Similitud con la Tierra de K2-3 d es de un 80 %, que lo sitúa entre los diez exoplanetas más similares a la Tierra, aunque sus características sugieren que podría ser mucho más hostil de lo previsto.[1]

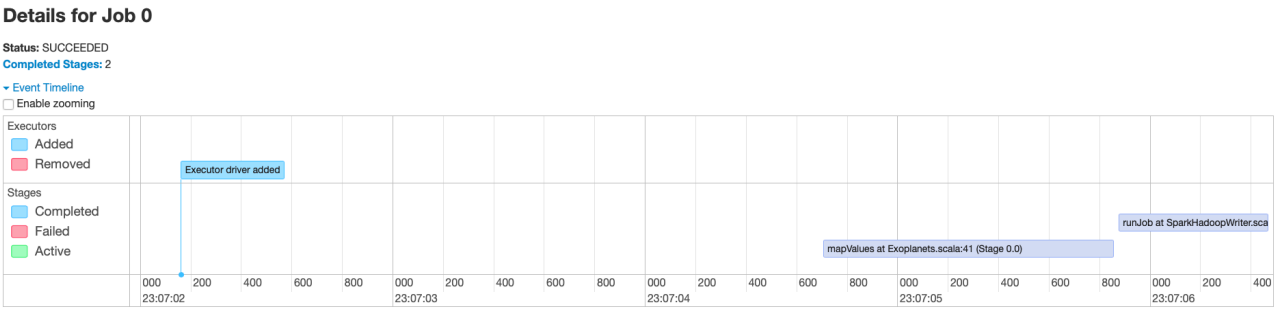
orbital sideral	
Características físicas	
Masa	3,66 $M_\oplus$ (asumiendo una composición similar a la de la Tierra)
Radio	1,52 $R_\oplus$
Características atmosféricas	
Temperatura	48,95 °C (322,1 K) (asumiendo una atmósfera semejante a la terrestre)

[editar datos en Wikidata]

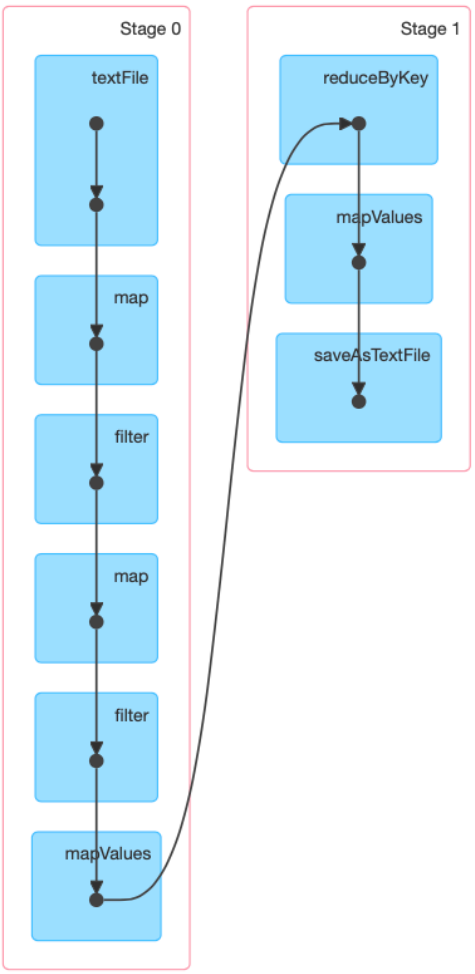
Monitorización en Spark UI

Para la monitorización de este trabajo hemos usado el cuadro de mandos que nos ofrece la máquina virtual de Hortonworks.

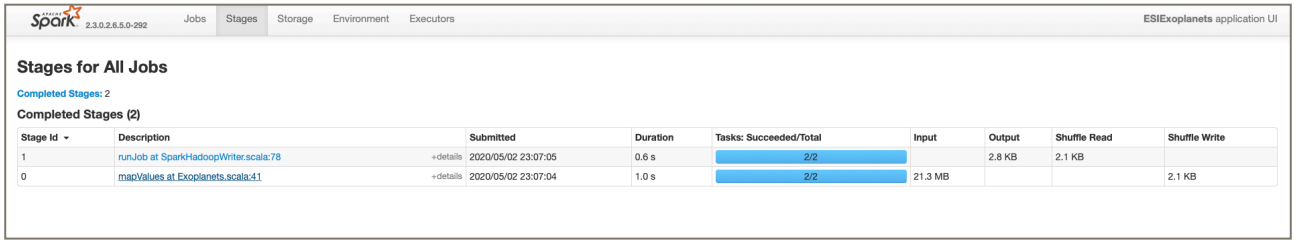
Nos conectamos a la máquina con un navegador al puerto 18081 y podemos ver las distintas a*pps* que se han lanzado en el servidor. Elegimos la nuestra, que se llama ESIExoplanets y comprobamos si se ha lanzado correctamente:



Así es como se ve la dependencia de los RDD en forma de grafo:



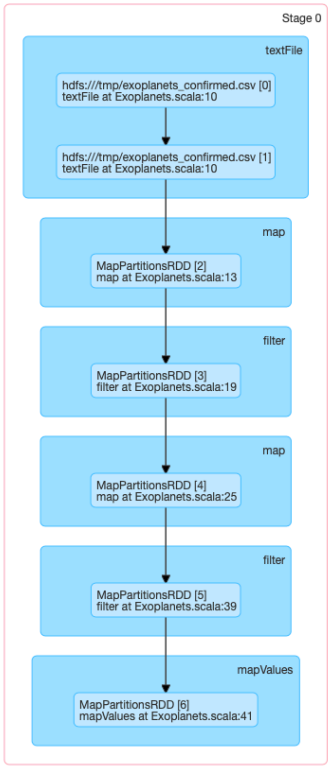
Ahora vamos a ver en cuántos *stage* se ha lanzado nuestro Job:



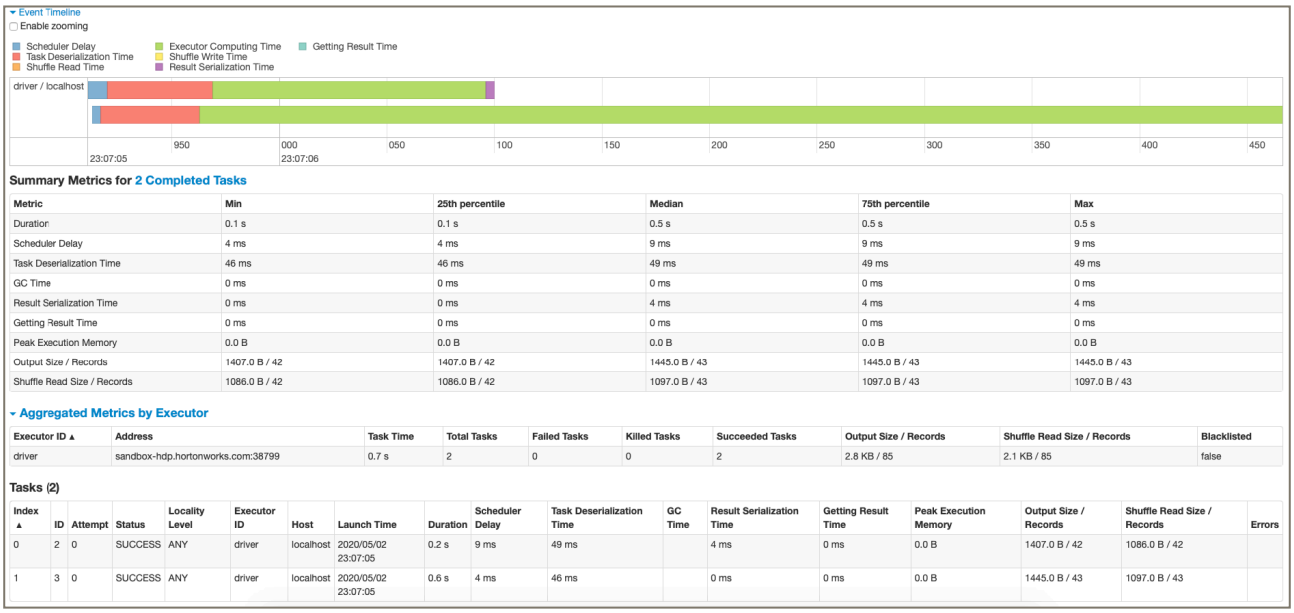
Vemos que han sido en dos *stages*: `runJob` y `mapValues`.

El *stage* de `mapValues` (Stage 0) se ve así:

Aquí es donde ocurre todo el proceso de lectura, cálculo del ESI y posterior mapeo para la media:



En cuanto al otro *Stage 1,* `mapValues`. Destacar que es el más rápido en ejecutar, debido a que en el anterior *stage*, se ha filtrado bastantes datos que no necesitábamos:



Event Timeline

Enable zooming

Scheduler Delay · Task Deserialization Time · Shuffle Read Time · Executor Computing Time · Shuffle Write Time · Result Serialization Time · Getting Result Time

driver / localhost

950 · 000 · 050 · 100 · 150 · 200 · 250 · 300 · 350 · 400 · 450

23:07:05 · 23:07:06

Summary Metrics for 2 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	0.1 s	0.1 s	0.5 s	0.5 s	0.5 s
Scheduler Delay	4 ms	4 ms	9 ms	9 ms	9 ms
Task Deserialization Time	46 ms	46 ms	49 ms	49 ms	49 ms
GC Time	0 ms	0 ms	0 ms	0 ms	0 ms
Result Serialization Time	0 ms	0 ms	4 ms	4 ms	4 ms
Getting Result Time	0 ms	0 ms	0 ms	0 ms	0 ms
Peak Execution Memory	0.0 B	0.0 B	0.0 B	0.0 B	0.0 B
Output Size / Records	1407.0 B / 42	1407.0 B / 42	1445.0 B / 43	1445.0 B / 43	1445.0 B / 43
Shuffle Read Size / Records	1086.0 B / 42	1086.0 B / 42	1097.0 B / 43	1097.0 B / 43	1097.0 B / 43

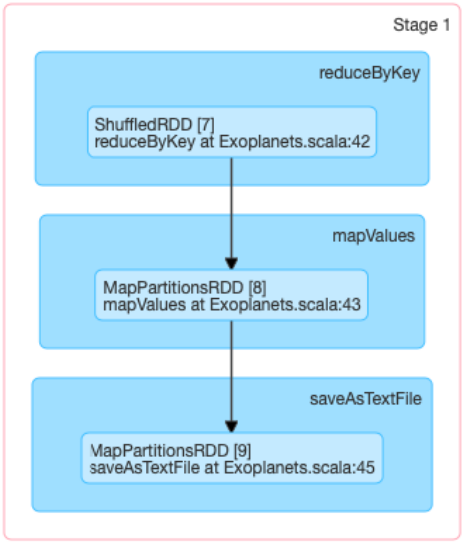
Aggregated Metrics by Executor

Executor ID ▲	Address	Task Time	Total Tasks	Failed Tasks	Killed Tasks	Succeeded Tasks	Output Size / Records	Shuffle Read Size / Records	Blacklisted
driver	sandbox-hdp.hortonworks.com:38799	0.7 s	2	0	0	2	2.8 KB / 85	2.1 KB / 85	false

Tasks (2)

Index ▲	ID	Attempt	Status	Locality Level	Executor ID	Host	Launch Time	Duration	Scheduler Delay	Task Deserialization Time	GC Time	Result Serialization Time	Getting Result Time	Peak Execution Memory	Output Size / Records	Shuffle Read Size / Records	Errors
0	2	0	SUCCESS	ANY	driver	localhost	2020/05/02 23:07:05	0.2 s	9 ms	49 ms		4 ms	0 ms	0.0 B	1407.0 B / 42	1086.0 B / 42	
1	3	0	SUCCESS	ANY	driver	localhost	2020/05/02 23:07:05	0.6 s	4 ms	46 ms		0 ms	0 ms	0.0 B	1445.0 B / 43	1097.0 B / 43	

Aquí es donde ocurre la reducción de los elementos que tienen más de una medición. Se coge la media de todas las mediciones y finalmente se guarda en un fichero de salida el resultado final.



Código desarrollado

```scala
import org.apache.spark.{SparkContext, SparkConf}
object Exoplanets {
  def main(args: Array[String]): Unit = {

    // Configuración Spark. Note que el maestro está configurado en local
    // porque pretendemos correr el fichero compilado en la misma máquina
    // que Spark. El [*] hace que utilice todos los hilos disponibles.
    val conf = new SparkConf().setAppName("ESIExoplanets")
      .setMaster("local[*]")
    val sc = new SparkContext(conf)
    val inputfile = sc.textFile("hdfs:///tmp/exoplanets_confirmed.csv")

    // Lectura de datos
    val rdd = inputfile.map(f=>{
      f.split(",")
    })

    // Limpieza de datos: Nos quedamos sólo con exoplanetas que contienen
    // datos en las columnas que nos interesa. Quitamos la fila de cabecera
    val cleanRdd = rdd.filter(line => {
      line(19) != "" && line(19) != "pl_radj" && line(32) != "pl_insol"
            && line(32) != ""
      })

    // Map: Calcula el índice ESI
    // Planet name , ESI
    var esiPlanets = cleanRdd.map(p => {

      val s = p(32).toDouble
      val se = 1.0

      val r = p(19).toDouble * 11.209
      val re = 1.0

      val esi = 1.0 - scala.math.sqrt(
        0.5 *
          (scala.math.pow((s-se)/(s+se),2) + scala.math.pow((r-re)/(r+re),2))
      )

      (p(0),esi)
    }).filter(l => l._2 > 0.6)

    // Map-reduce: Calcula la media de los planetas duplicados
    val esiPlanetsReduced = esiPlanets.mapValues((_, 1))
      .reduceByKey((x, y) => (x._1 + y._1, x._2 + y._2))
      .mapValues{ case (sum, count) => (1.0 * sum) / count }

    esiPlanetsReduced.saveAsTextFile("hdfs:///tmp/ESIExoplanetsAVG")

  }
}
```

Bibliografía

[1] Exoplanets Archive. NASA Exoplanets Science Institute. https://exoplanetarchive.ipac.caltech.edu/docs/data.html

[2] Earth Similarity Index from Planetary Habitability Laboratory. University of Puerto Rico at Arecibo. http://phl.upr.edu/projects/earth-similarity-index-esi

[3] Jupiter Fact Sheet. Jupiter/Earth Comparison. Dr. David R. Williams. NASA. https://nssdc.gsfc.nasa.gov/planetary/factsheet/jupiterfact.html

[4] K2-3 d WikiZero Article. Based on a Wikipedia Article. https://www.wikizero.com/es/K2-3_d