Intro to Apache Spark

http://databricks.com/

download slides:

training.databricks.com/workshop/itas_workshop.pdf



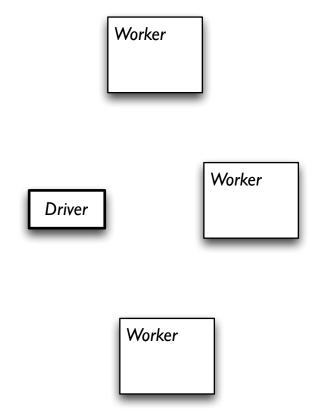
Licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License





```
// load error messages from a log into memory
// then interactively search for various patterns
// https://gist.github.com/ceteri/8ae5b9509a08c08a1132
// base RDD
val lines = sc.textFile("hdfs://...")
// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map( .split("\t")).map(r => r(1))
messages.cache()
// action 1
messages.filter( .contains("mysql")).count()
// action 2
messages.filter(_.contains("php")).count()
```

We start with Spark running on a cluster... submitting code to be evaluated on it:



```
// base RDD
val lines = sc.textFile("hdfs://...")

// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()

// action 1
messages.filter(_.contains("mysql")).count()

// Ciscussing the other part
messages.filter(_.contains("php")).count()
```

At this point, take a look at the transformed RDD operator graph:

```
scala> messages.toDebugString
res5: String =
MappedRDD[4] at map at <console>:16 (3 partitions)
   MappedRDD[3] at map at <console>:16 (3 partitions)
   FilteredRDD[2] at filter at <console>:14 (3 partitions)
   MappedRDD[1] at textFile at <console>:12 (3 partitions)
   HadoopRDD[0] at textFile at <console>:12 (3 partitions)
```

```
// base RDD
val lines = sc.textFile("hdfs://...")
// transformed RDDs
val errors = lines.filter( .startsWith("ERROR"))
val messages = errors.map(\_.split("\t")).map(r => r(1))
messages.cache()
// action 1
messages.filter(_.contains("mysql")).count()
                                                        Worker
mediscussing the other part
                                                                 Worker
                                                 Driver
                                                         Worker
```

```
// base RDD
val lines = sc.textFile("hdfs://...")
// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(\_.split("\t")).map(r => r(1))
messages.cache()
// action 1
messages.filter(_.contains("mysql")).count()
                                                         Worker
                                                          block 1
mediscussing the other part
                                                                  Worker
                                                  Driver
                                                                   block 2
```

Worker

block 3

```
// base RDD
val lines = sc.textFile("hdfs://...")
// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(\_.split("\t")).map(r => r(1))
messages.cache()
// action 1
messages.filter(_.contains("mysql")).count()
                                                         Worker
                                                          block 1
mediscussing the other part
                                                                  Worker
                                                   Driver
                                                                   block 2
                                                          Worker
                                                            block 3
```

```
// base RDD
val lines = sc.textFile("hdfs://...")
// transformed RDDs
val errors = lines.filter( .startsWith("ERROR"))
val messages = errors.map(\_.split("\t")).map(r => r(1))
messages.cache()
// action 1
messages.filter(_.contains("mysql")).count()
                                                           Worker
                                                                       read
                                                                      HDFS
                                                            block
                                                                       block
mediscussing the other part
                                                                     Worker
                                                                                read
                                                                                HDFS
                                                    Driver
                                                                                block
                                                                      block 2
                                                            Worker
                                                                        read
                                                                        HDFS
                                                                        block
                                                             block 3
```

```
// base RDD
val lines = sc.textFile("hdfs://...")
// transformed RDDs
val errors = lines.filter( .startsWith("ERROR"))
val messages = errors.map(\_.split("\t")).map(r => r(1))
messages.cache()
// action 1
                                                                 cache 1
                                                                            process.
messages.filter(_.contains("mysql")).count()
                                                                           cache data
                                                            Worker
                                                             block 1
mediscussing the other part
                                                                          cache 2
                                                                                     process.
                                                                                     cache data
                                                                      Worker
                                                     Driver
                                                                       block 2
                                                                  cache 3
                                                                              process,
                                                                             cache data
                                                              Worker
                                                               block 3
```

```
// base RDD
val lines = sc.textFile("hdfs://...")
// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(\_.split("\t")).map(r => r(1))
messages.cache()
// action 1
                                                               cache 1
messages.filter(_.contains("mysql")).count()
                                                          Worker
                                                           block 1
mediscussing the other part
                                                                       cache 2
                                                                   Worker
                                                   Driver
                                                                    block 2
                                                               cache 3
                                                           Worker
                                                            block 3
```

```
val eiscussien giter et o sun a PRROPA PT
val messages = errors.map( .split("\t")).map(r => r(1)
                                                                  cache 1
                                                             Worker
// action 2
                                                              block 1
messages.filter(_.contains("php")).count()
                                                                           cache 2
                                                                       Worker
                                                      Driver
                                                                        block 2
                                                                   cache 3
                                                              Worker
                                                               block 3
```

```
van eiscussien outernet on evan eische School of the control of th
val messages = errors.map( .split("\t")).map(r => r(1)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                cache 1
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  process
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         from cache
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       Worker
// action 2
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                block 1
messages.filter(_.contains("php")).count()
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              cache 2
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              process
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      from cache
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         Worker
                                                                                                                                                                                                                                                                                                                                                                                                                                                                          Driver
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   block 2
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          cache 3
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       process
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             from cache
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  Worker
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           block 3
```

```
vad iscussing tenetothe Proart
val messages = errors.map( .split("\t")).map(r => r(1)
                                                              cache 1
                                                          Worker
// action 2
                                                           block 1
messages.filter(_.contains("php")).count()
                                                                       cache 2
                                                                   Worker
                                                   Driver
                                                                    block 2
                                                               cache 3
                                                           Worker
                                                            block 3
```

Looking at the RDD transformations and actions from another perspective...

```
// load error messages from a log into memory
// then interactively search for various patterns
// https://gist.github.com/ceteri/8ae5b9509a08c08a1132
// base RDD
val lines = sc.textFile("hdfs://...")
                                                    transformations
                                                                                         action
// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()
// action 1
messages.filter(_.contains("mysql")).count()
// action 2
messages.filter(_.contains("php")).count()
```

