Data Engineering with Apache Spark

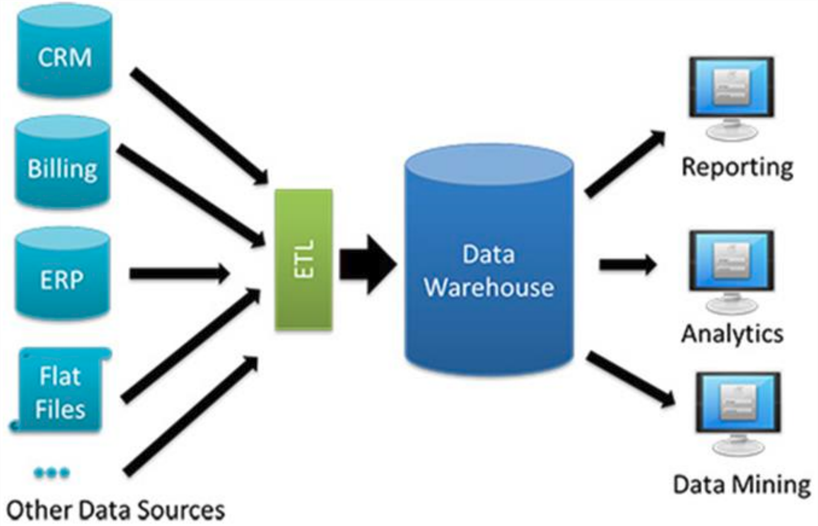
Markus Dale, medale@asymmetrik.com

May 2019

Intro, Slides And Code

- Slides: https://github.com/medale/
- Scala Spark Code Examples: https://github.com/medale/

Data engineering



Data engineering on small dataset

Figure 1: Laptop

Data engineering for larger dataset (Vertical Scaling)

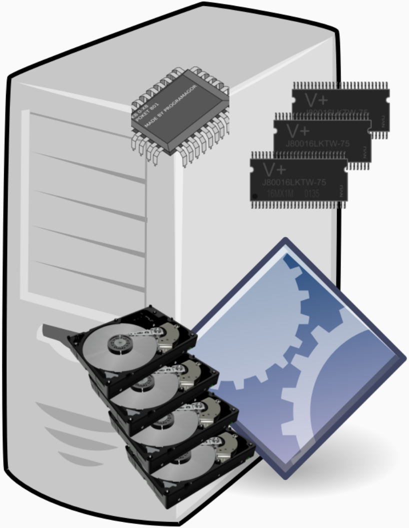
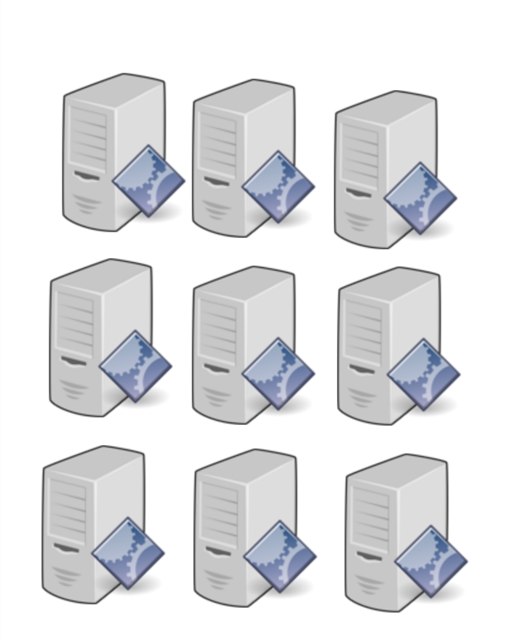
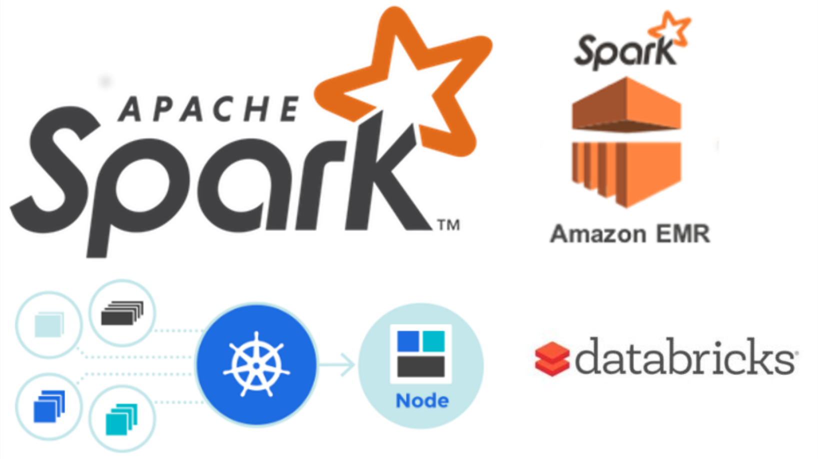


Figure 2: Beefed-up Server

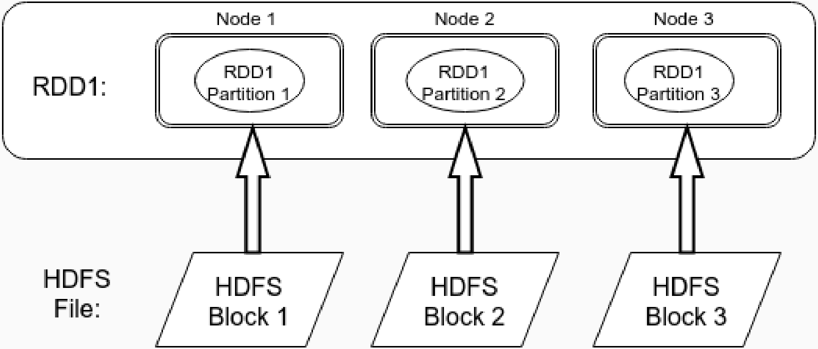
Data engineering for large datasets (Horizontal Scaling)



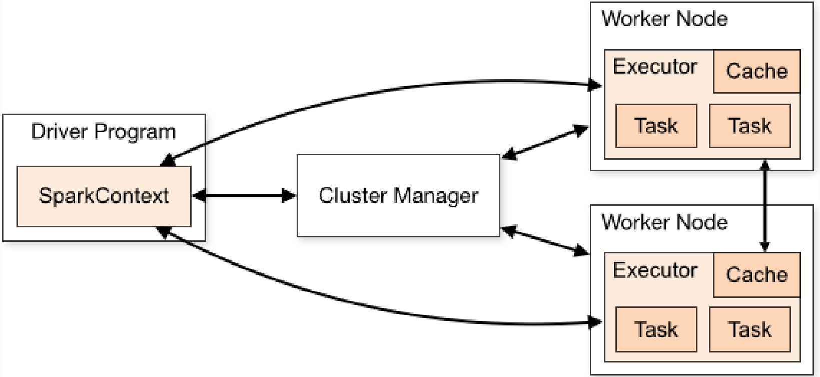
Cluster Manager



Resilient Distributed Datasets (RDDs)

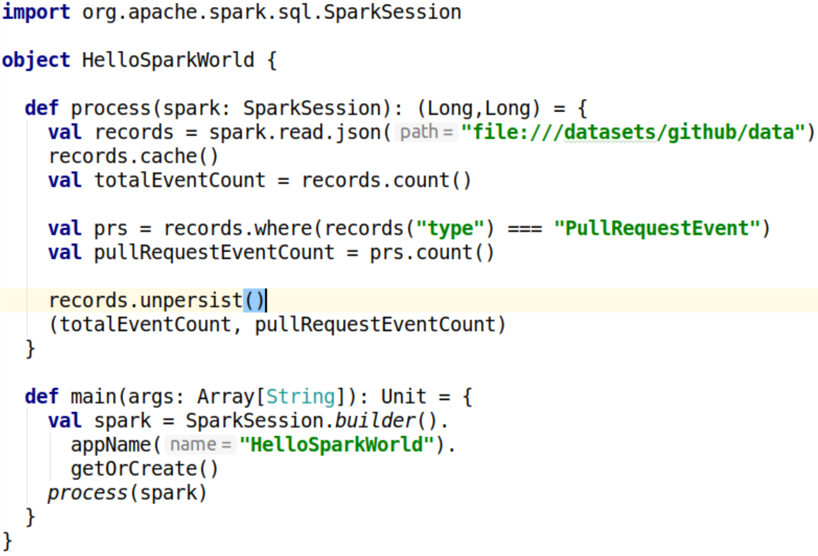


Anatomy of a Spark Application



Source: Apache Spark website

Hello, Spark World!

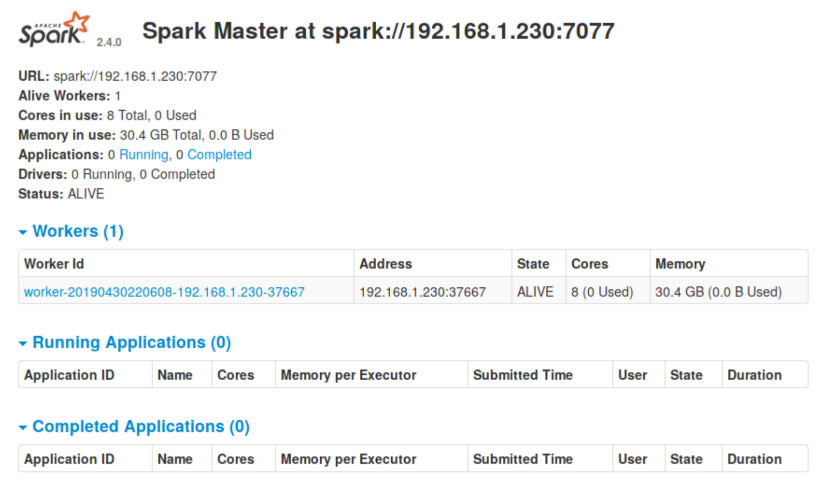


```scala
import org.apache.spark.sql.SparkSession

object HelloSparkWorld {

  def process(spark: SparkSession): (Long,Long) = {
    val records = spark.read.json( path = "file:///datasets/github/data")
    records.cache()
    val totalEventCount = records.count()

    val prs = records.where(records("type") === "PullRequestEvent")
    val pullRequestEventCount = prs.count()

    records.unpersist()
    (totalEventCount, pullRequestEventCount)
  }

  def main(args: Array[String]): Unit = {
    val spark = SparkSession.builder().
      appName( name = "HelloSparkWorld").
      getOrCreate()
    process(spark)
  }
}
```

Starting Spark Standalone Cluster Manager

```
# Start on master
$SPARK_HOME/sbin/start-master.sh --host 192.168.1.230

# Start one or more workers
$SPARK_HOME/sbin/start-slave.sh spark://192.168.1.230:7077
```

Spark Standalone Cluster Manager UI - idle



Spark Master at spark://192.168.1.230:7077

URL: spark://192.168.1.230:7077
Alive Workers: 1
Cores in use: 8 Total, 0 Used
Memory in use: 30.4 GB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (1)

Worker Id	Address	State	Cores	Memory
worker-20190430220608-192.168.1.230-37667	192.168.1.230:37667	ALIVE	8 (0 Used)	30.4 GB (0.0 B Used)

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

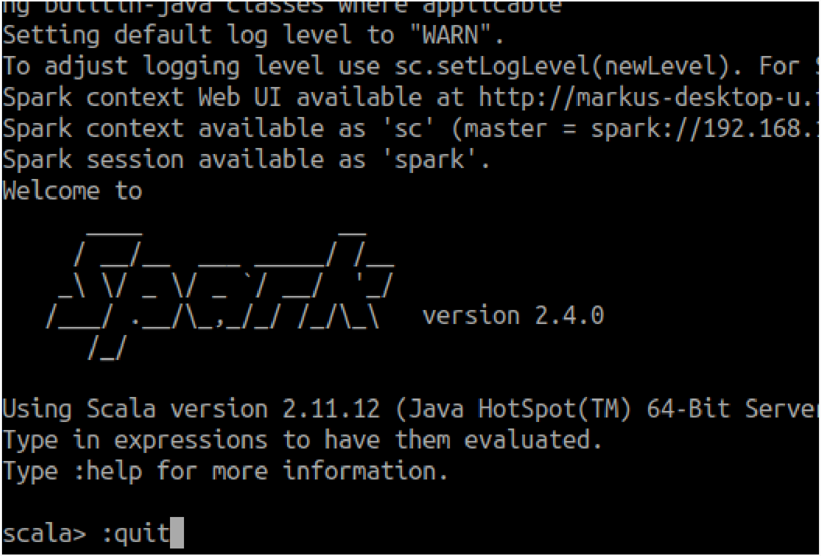
Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Running spark-shell in cluster

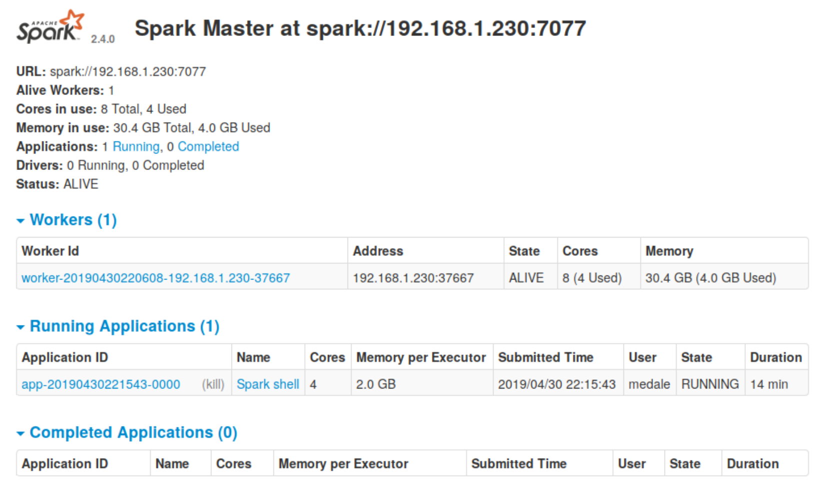
```
spark-shell --master spark://192.168.1.230:7077 \
 --driver-memory 1g \
 --executor-memory 2g \
 --total-executor-cores 4 \
 --executor-cores 2 \
 --jars /tmp/dataset-0.9.0-SNAPSHOT-fat.jar
```

Spark Shell Startup



```
ng builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For S
Spark context Web UI available at http://markus-desktop-u.
Spark context available as 'sc' (master = spark://192.168.
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.0
      /_/

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server
Type in expressions to have them evaluated.
Type :help for more information.

scala> :quit
```

Spark Standalone Cluster Manager - 1 running application



Spark Master at spark://192.168.1.230:7077

URL: spark://192.168.1.230:7077
Alive Workers: 1
Cores in use: 8 Total, 4 Used
Memory in use: 30.4 GB Total, 4.0 GB Used
Applications: 1 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (1)

Worker Id	Address	State	Cores	Memory
worker-20190430220608-192.168.1.230-37667	192.168.1.230:37667	ALIVE	8 (4 Used)	30.4 GB (4.0 GB Used)

Running Applications (1)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
app-20190430221543-0000 (kill)	Spark shell	4	2.0 GB	2019/04/30 22:15:43	medale	RUNNING	14 min

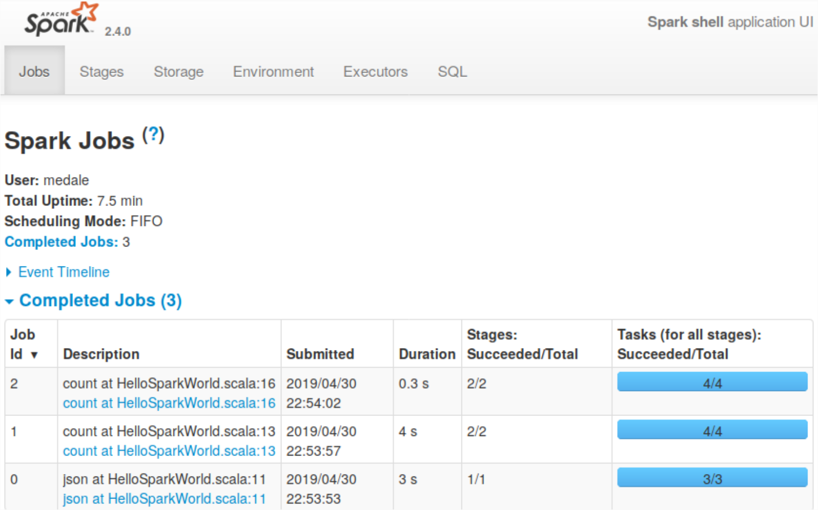
Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

HelloSparkWorld in spark-shell

```
scala> import com.uebercomputing.HelloSparkWorld
import com.uebercomputing.HelloSparkWorld

scala> HelloSparkWorld.process(spark)
res0: (Long, Long) = (147374,6699)
```

Spark Application UI - Jobs, stages, tasks



Spark Jobs (?)

User: medale
Total Uptime: 7.5 min
Scheduling Mode: FIFO
Completed Jobs: 3

▸ Event Timeline

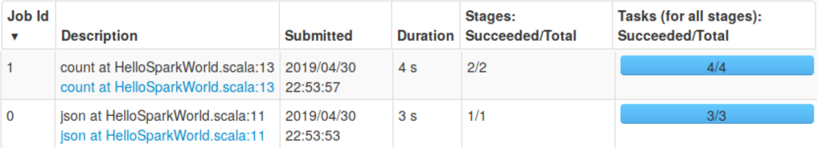
▾ **Completed Jobs (3)**

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
2	count at HelloSparkWorld.scala:16 count at HelloSparkWorld.scala:16	2019/04/30 22:54:02	0.3 s	2/2	4/4
1	count at HelloSparkWorld.scala:13 count at HelloSparkWorld.scala:13	2019/04/30 22:53:57	4 s	2/2	4/4
0	json at HelloSparkWorld.scala:11 json at HelloSparkWorld.scala:11	2019/04/30 22:53:53	3 s	1/1	3/3

Job - n lazy transformations, 1 action

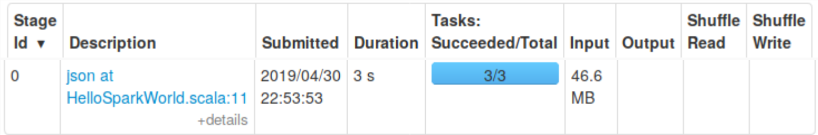
```
//job 0 - list files, infer schema
val records = spark.read.json(”file:///datasets/github/data”)
//transformation
records.cache()
//action - job 1
val totalEventCount = records.count()

//transformation - datasets are immutable!
val prs = records.where(records(”type”) === ”PullRequestEvent”)
//action - job 2
val pullRequestEventCount = prs.count()
```

Job 0 - Stages, tasks, partitions



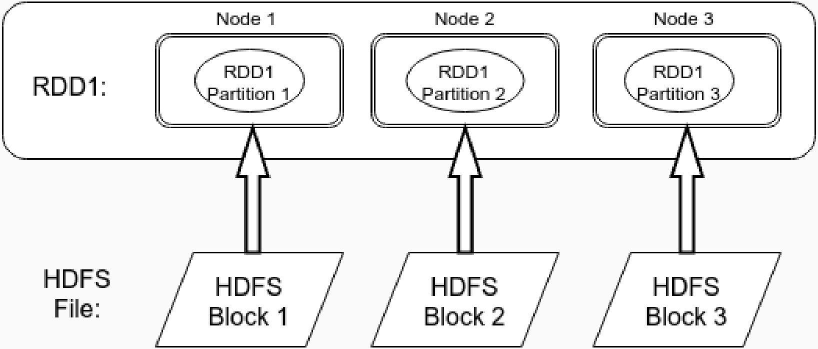
Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
1	count at HelloSparkWorld.scala:13 count at HelloSparkWorld.scala:13	2019/04/30 22:53:57	4 s	2/2	4/4
0	json at HelloSparkWorld.scala:11 json at HelloSparkWorld.scala:11	2019/04/30 22:53:53	3 s	1/1	3/3

Job 0 - Stage 0, tasks, partitions

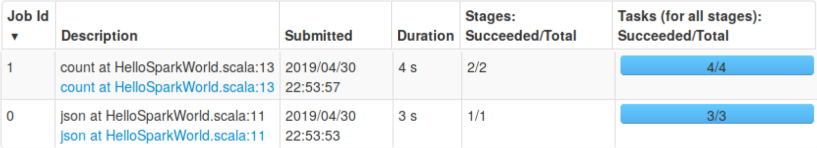


Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
0	json at HelloSparkWorld.scala:11 +details	2019/04/30 22:53:53	3 s	3/3	46.6 MB			

Input partitions - splittable file?

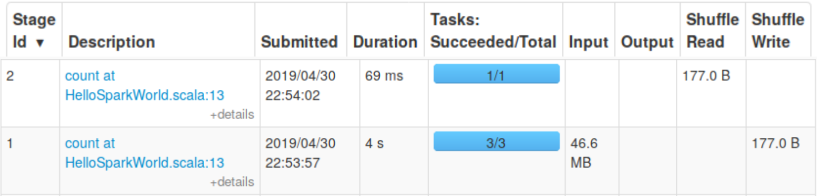


Job 1 - count



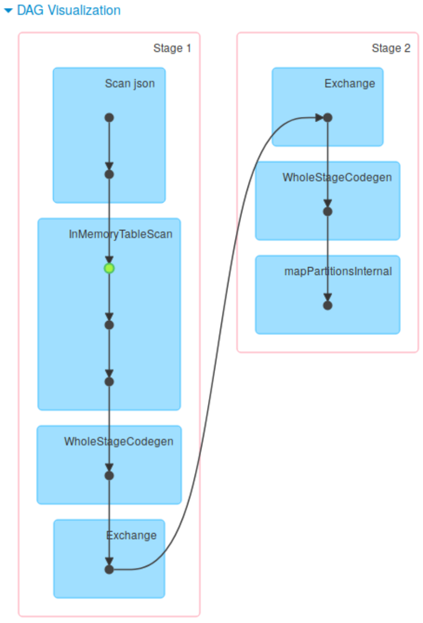
Job Id ▼	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
1	count at HelloSparkWorld.scala:13 count at HelloSparkWorld.scala:13	2019/04/30 22:53:57	4 s	2/2	4/4
0	json at HelloSparkWorld.scala:11 json at HelloSparkWorld.scala:11	2019/04/30 22:53:53	3 s	1/1	3/3

Job 1 - count Stages 1 and 2

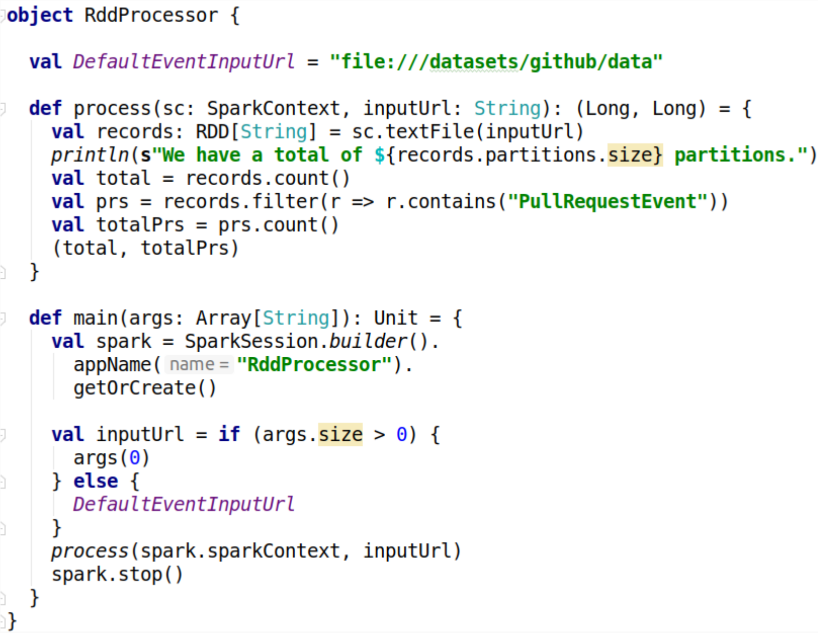


Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
2	count at HelloSparkWorld.scala:13 +details	2019/04/30 22:54:02	69 ms	1/1			177.0 B	
1	count at HelloSparkWorld.scala:13 +details	2019/04/30 22:53:57	4 s	3/3	46.6 MB			177.0 B

Job 1 - Stages 1 and 2 DAG



RDDs - Not deprecated!



```
object RddProcessor {

  val DefaultEventInputUrl = "file:///datasets/github/data"

  def process(sc: SparkContext, inputUrl: String): (Long, Long) = {
    val records: RDD[String] = sc.textFile(inputUrl)
    println(s"We have a total of ${records.partitions.size} partitions.")
    val total = records.count()
    val prs = records.filter(r => r.contains("PullRequestEvent"))
    val totalPrs = prs.count()
    (total, totalPrs)
  }

  def main(args: Array[String]): Unit = {
    val spark = SparkSession.builder().
      appName( name = "RddProcessor").
      getOrCreate()

    val inputUrl = if (args.size > 0) {
      args(0)
    } else {
      DefaultEventInputUrl
    }
    process(spark.sparkContext, inputUrl)
    spark.stop()
  }
}
```

And now for something completely different: Colon Cancer

- Screening saves lives!
    - Colonoscopy - talk to your doc
    - Dave Barry: A journey into my colon — and yours
- Colorectal Cancer Alliance

Questions?

- medale@asymmetrik.com
- Infrequent blog/past presentations http://uebercomputing.com/