Software Requirements

- Scala
- Java
- Spark
- Maven *
- Sbt *
- Python *
- R *
- Hadoop *

^{*} Depends on user preference.
Java, Python, R API availability lags behind Scala

API

- RDDs
- Resilient Distributed Datasets
 - Main abstraction in Spark
 - Distributed, fault-tolerant collection of partitioned records on multiple nodes
- Transformations
 - Create new RDDs from existing RDDs
- Actions
 - Kicks off execution

Spark Application

- Spark driver => coordinates spark processes processes
- Spark processes => run independently on cluster
- A Spark application is combination of driver and processes
- Note that actions like 'collect()' can bring back too much data to the driver

API Levels - RDD

- Low level, compile-time type-safe
- Expresses how instead of what
- Functional programming constructs
- No query optimization done for us
- Can build inefficient RDD transformation chains (ReduceByKey – counting hole dataset, before a filtering) rdd.reduceByKey().filter {etc}

API Levels - DataFrame

- Relational database feel
- Domain specific language to manipulate data
- Expresses what instead of how
- Schema is imposed (columns with names and types)
- Queries are optimized by Spark engine
- No type safety

API Levels - Dataset

- New abstraction, higher level API
- Merges with DataFrames API
- Adds compile-time errors
- Errors are raised before jobs kid off
- More Info :
 - https://www.youtube.com/watch? v=pZQsDloGB4w

Spark Application

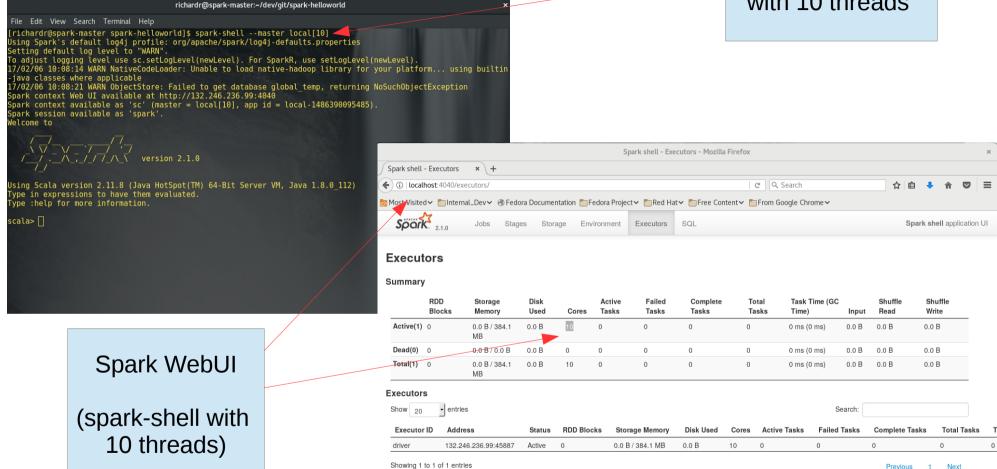
- Spark driver => coordinates spark processes processes
- Spark processes => run independently on cluster
- A Spark application is combination of driver and processes
- Note that actions like 'collect()' can bring back too much data to the driver

Spark Shell

- Modified Scala shell
- spark-shell --master local[*]
 - Master URL for distributed cluster OR local[N] to run locally with N threads
 - Ex:
 - spark-shell -- master spark://spark-master:7077
 - spark-shell -- master local[3]
 - http://localhost:4040 <- worker
 - http://spark-master:8080 <-cluster & workers

Spark Shell

Start Spark locally with 10 threads C Q Search



http://spark.apache.org/docs/latest/programming-guide.html#using-the-shell

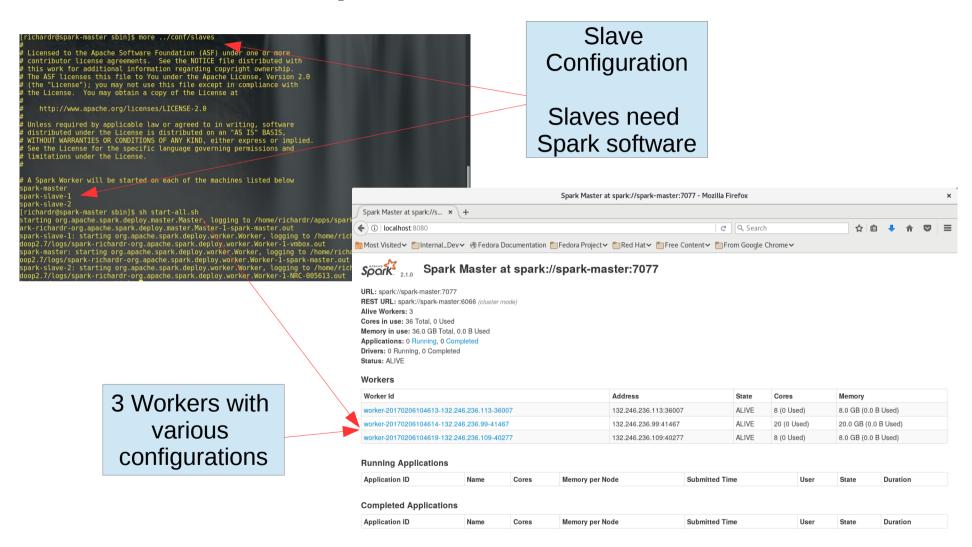
Example – Scala, RDD, Spark Shell

```
//Read in a text file :
val lines = sc.textFile("/home/richardr/Documents/data/inputdata.txt")
//Actions : Show first element of the lines RDD and print all lines
lines.first()
lines.foreach(println)
//Split each lines on blan space
val words = lines.flatMap(ln => ln.split(" "))
//Actions : Show first element of words RDD and print all words
words.first()
words.foreach(println) //prints array of strings
//Create a pairs RDD, first element of pair is the word, second is
//the number 1
val pairs = words.map(w \Rightarrow (w, 1))
pairs.foreach(println)
//Count words (grouped by word). Key is word, count is
//second element of pair
val counts = pairs.reduceByKey( + )
//Create RDD of string representations for pair RDD
val strPairs = counts.map(tuple => tuple.productIterator.mkString(","))
//Print each word and count
```

Spark Cluster

- Local[*] <- Great for development
- Spark://[a spark master]:8080 <- larger jobs
- Starting a cluster
 - conf/slaves
 - conf/spark-env.sh
 - sbin/start-all.sh
 - sbin/start-slave.sh spark://[a spark master]:7077

Spark Cluster



Submitting a Job

- For self-contained applications
- deploy-mode -- local <- driver runs locally (default)
- deploy-mode -- cluster <- driver runs on cluster (fire and forget)

spark-submit

```
#!/bin/bash
spark-submit \
--class ca.redsofa.jobs.HelloWorld \
--master local[*] \
./target/spark-helloworld-1.0-SNAPSHOT.jar
```

Submit locally
With:
threads = # cores
Driver memory 1G default

Submit to cluster

```
#!/bin/bash
spark-submit --class ca.nrc.jobs.App \
--master spark://spark-master:7077 \
--driver-memory 2G \|
./target/batch-convert-json-to-parquet-0.1-SNAPSHOT.jar \
file:///home/richardr/data/qcr_data/Olympics_Data/Olympics_corenlp_annotated_excerpt.json \
file:///home/richardr/data/qcr_data/Olympics_Data/Olympics_corenlp_annotated_excerpt_field_subset.json
```

```
#!/bin/bash
spark-submit --class ca.nrc.jobs.App \
--master spark://spark-master:7077 \
--deploy-mode cluster \
--driver-memory 2G \
--conf 'spark.executor.memory=8g' \
hdfs://hadoop-master:9000/home/richardr/jars/batch-corenlp-annotation-0.1-SNAPSHOT.jar \
hdfs://hadoop-master:9000/home/richardr/data/Ottawa_Shooting/ottawa_shooting_tweets.json \
hdfs://hadoop-master:9000/home/richardr/data//Ottawa_Shooting/OS_cluster_batch_corenlp_annotated.json
```

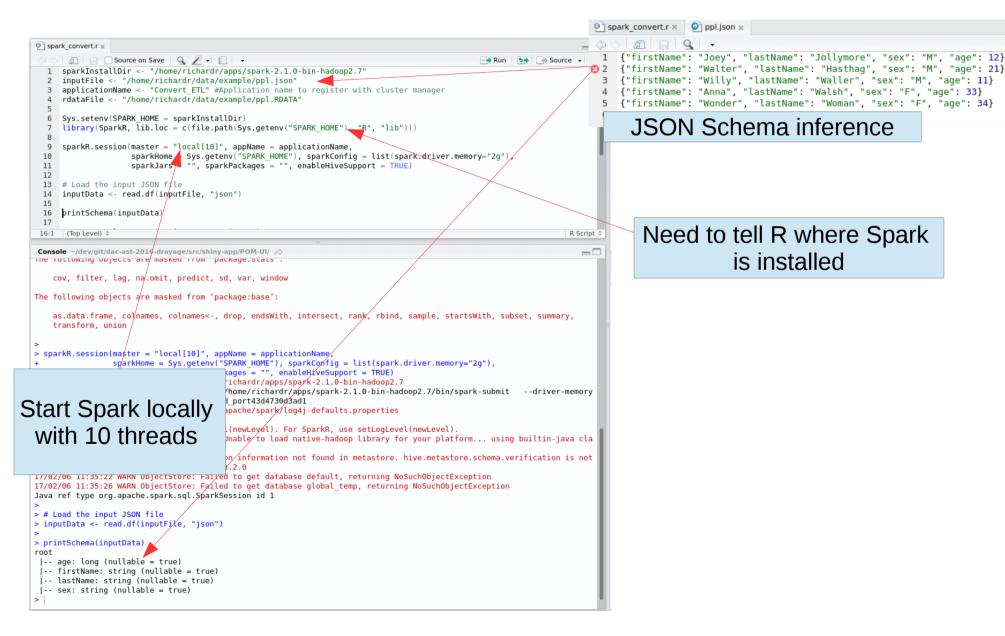
Self-Contained Applications Java, Dataset, spark-submit

```
Reading a text file
SparkSession spark = SparkSession
                                                                                                      and splitting lines on
       .builder()
       .appName("Simple Batch Job")
       .config("spark.driver.memory", "2g")
                                                                                                                  Spaces
       .enableHiveSupport()
       .get0rCreate();
Dataset<String> inputDataDs = spark.read().text(INPUT FILE).as(Encoders.STRING());
                                                                                                                   Note:
                                                                                            If reading JSON, the schema is
Dataset<String> words = inputDataDs.flatMap(s -> {
                         return Arrays.asList(s.toLowerCase().split(" ")).iterator();
                                                                                                     automatically inferred
                      }, Encoders.STRING())
                      .filter(s -> !s.isEmpty());
                                                          [richardr@spark-master spark-helloworld] sh submit job.sh
words.printSchema();
                                                          Stating Job ...
                                                          17/02/06 10:22:53 WARN NativeCodeLoader: Unable to load native-hadoop library
                                                          -java classes where applicable
words.createOrReplaceTempView("words");
                                                          17/02/06 10:22:59 WARN ObjectStore: Failed to get database global temp, retu
                                                           |-- value: string (nullable = true)
String sql = "SELECT " +
               "value as word, " +
              "COUNT(value) as word count " +
                                                            word word count
             "words " -
            "GROUP BY " +
                                                           again
            "ORDER BY "
                                                           hello
               "value "
                                                             is
                                                             one
Dataset<Row> wordCount = spark.sql(sql);
                                                            test
                                                            this
                                                           three
                                                             two
                                                           world
```

User Defined Functions

```
public static void registerStringLengthUdf(SparkSession spark){
    spark.udf().register("stringLengthUdf", new UDF1<String, Long>() {
        @Override
        public Long call(String str) {
            if(str != null && !str.isEmpty()){
                return new Long(str.length());
        }else{
                return 0L;
        }
        }, DataTypes.LongType);
}
Any Java code
```

SparkR



Need to tell R where Spark is installed

Closure Serialization

- "Task not serializable" exceptions
- Need to be mindful about executing code across a cluster
- Closures are serialized and sent to each cluster executor
- See Understanding Closures

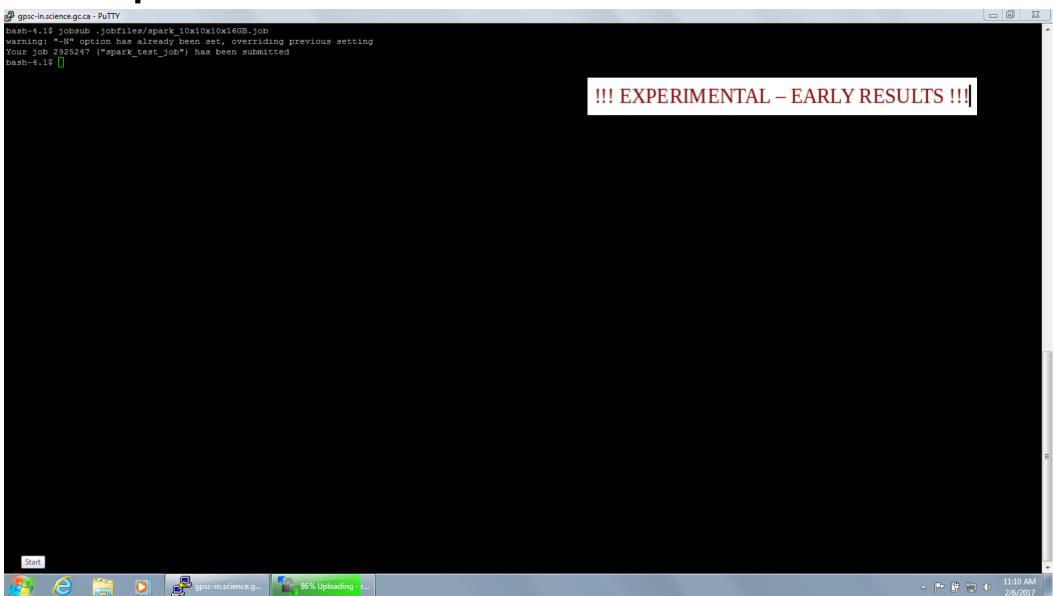
Spark on Dorval Cluster

!!! EXPERIMENTAL – EARLY RESULTS !!!

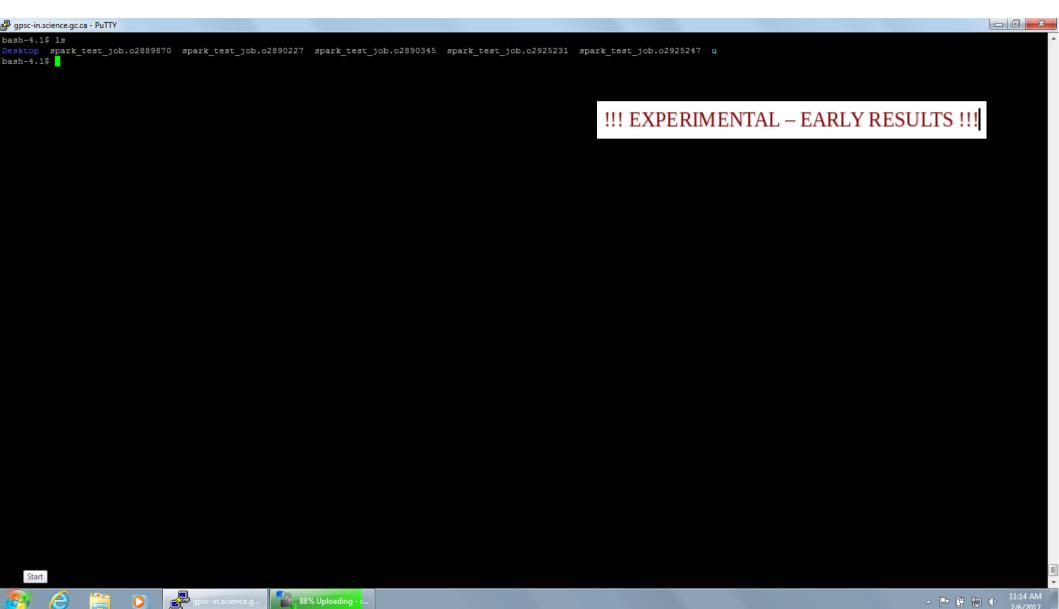
Spark on Dorval Cluster - jobfile

```
gpsc-in.science.gc.ca - PuTTY
                                                                                                                  !!! EXPERIMENTAL – EARLY RESULTS !!!
 at $GECOSHEP PNET HOSTFILE | grep -v $(hostname)
  -f $HOME/u/apps/spark-2.1.0-bin-hadoop2.7/logs/*
  -f $HOME/u/apps/spark-2.1.0-bin-hadoop2.7/conf/slaves
 for HOST in $(cat $GECOSHEP PNET HOSTFILE | grep -v $(hostname));
  echo $HOST >> $HOME/u/apps/spark-2.1.0-bin-hadoop2.7/conf/slaves
echo "Starting Spark Cluster ..."
ssh $(hostname) 'sh $HOME/u/apps/spark-2.1.0-bin-hadoop2.7/sbi
 m -f $HOME/u/apps/spark-2.1.0-bin-hadoop2.7/conf/slaves
       10x10x10x16GB.job" 39L, 1074C
                                                   85% Uploading - r
                                                                                                                                                                            △ 🗠 🛱 🖫 🕩
```

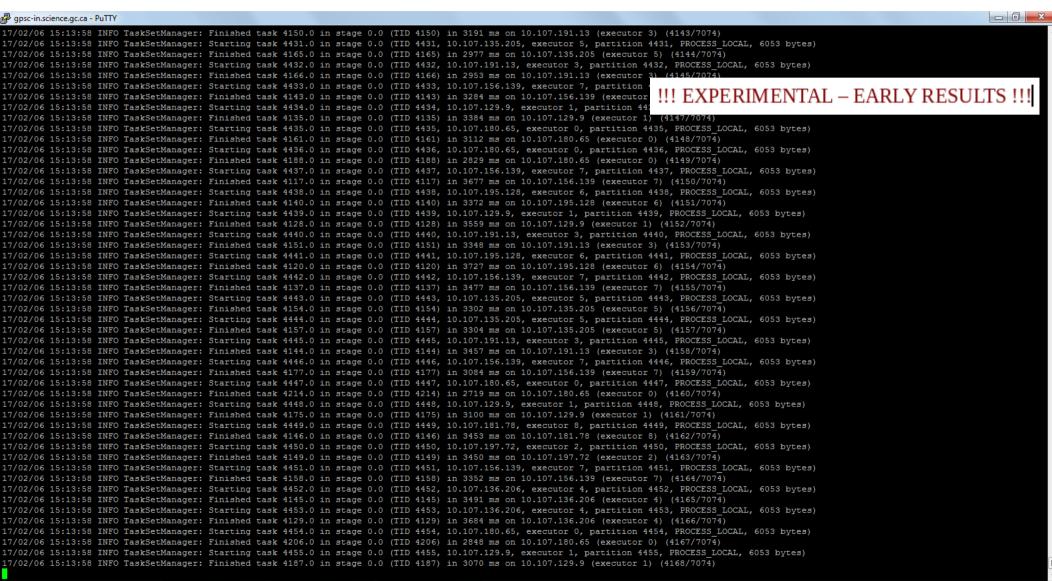
Spark on Dorval Cluster – submit



Spark on Dorval Cluster - log



Spark on Dorval Cluster - log



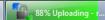














Spark on Dorval Cluster - webui

