Python API for Spark

Josh Rosen
UC Berkeley

www.spark-project.org
PySpark at a Glance

Write Spark jobs in Python
   » Supports Python C extensions; not Jython

Interactive use through the Python REPL

Under development, but a beta should be available soon
Examples: Word Count

from pyspark.context import SparkContext

sc = SparkContext(...)  
lines = sc.textFile(sys.argv[2], 1)

counts = lines.flatMap(lambda x: x.split(' '))  
    .map(lambda x: (x, 1))  
    .reduceByKey(lambda x, y: x + y)

for (word, count) in counts.collect():
    print "%s : %i" % (word, count)
Interactive Console

Demo
Implementation
Implementation

Built on top of the Java API
  » Communicates with a local Java process using Py4J.

Python RDDs are stored in Spark as RDD[Array[Byte]] of serialized Python objects.
Implementation cont’d

Functions are executed in Python worker processes that communicate with Spark Worker
  » Communicate with Spark over local pipes

Python functions are serialized and shipped to workers
  » Can serialize closures
Pipelining

To minimize serialization and communication costs, transformations are pipelined

E.g: `lines.flatMap(lambda x: x.split(' ')) \ .map(lambda x: (x, 1))`
Scala and Java Integration

RDDs produced by Java and Scala Spark jobs can be further transformed using Python

Planned support for processing Python-created RDDs in Java / Scala
Coming Soon

PySpark will be released once its performance improves and additional Spark features are added.

Planned support for

» Accumulators
» Broadcast variables
» Python-friendly input and output formats