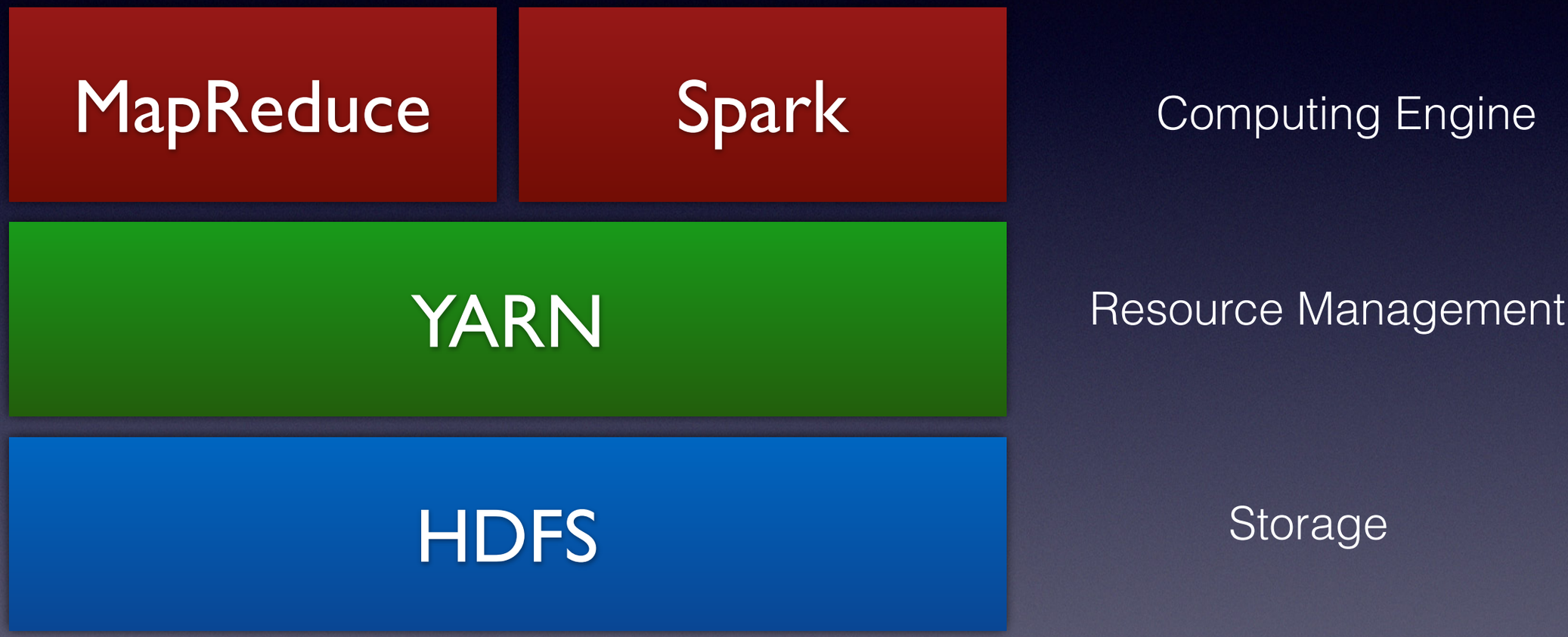
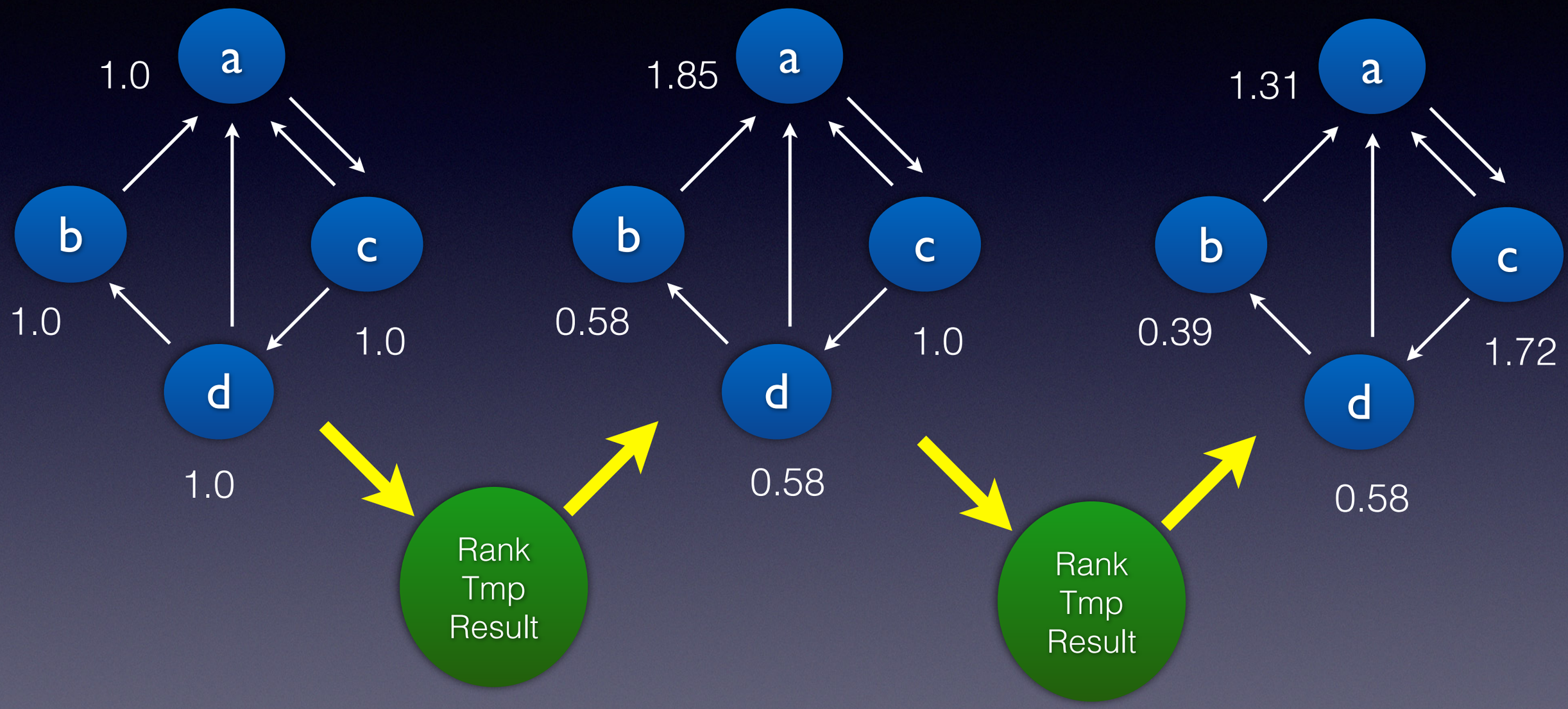
What is Spark?



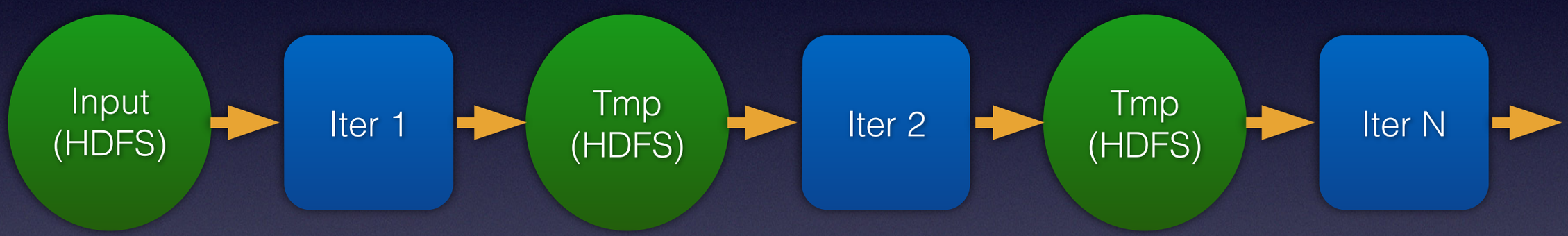
Most machine learning algorithms need iterative computing

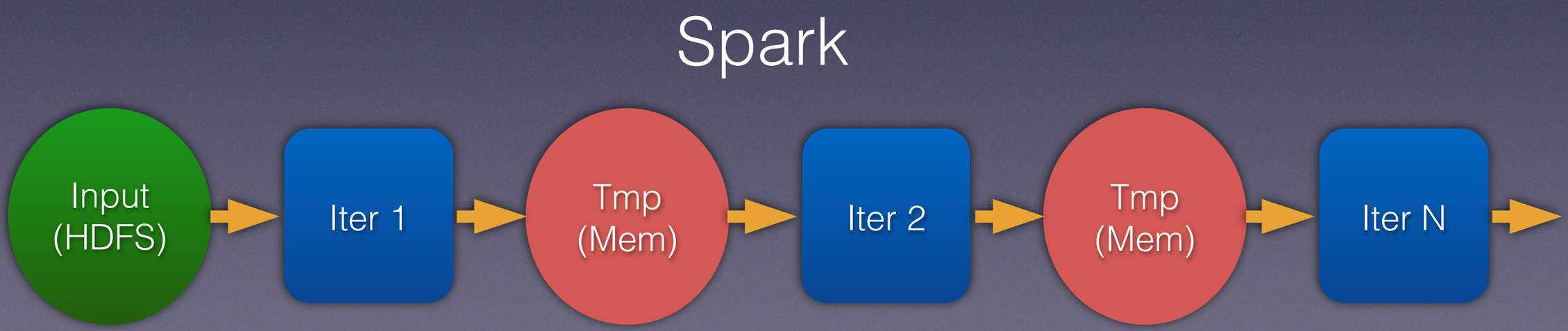
PageRank



1st Iter 2nd Iter 3rd Iter

HDFS is 100x slower than memory





scription		Duration
e at <console>:20	2014/04/09 08:10:49	207 ms
nt at <console>:18	2014/04/09 08:10:49	279 ms
uceByKey at <console>:20		7.7 s
e at <console>:19		220 ms
nt at <console>:19		287 ms
uceByKey at <console>:20	2014/04/09 08:10:01	7.4 s
nt at <console>:19	2014/04/09 08:09:43	321 ms
uceByKey at <console>:18	2014/04/09 08:09:23	19.9 s
inct at <console>:16		49.4 s
inct at <console>:16		1.2 m

3rd iteration(mem)
take 7.7 sec

2nd iteration(mem)
take 7.4 sec

First iteration(HDFS)
take 200 sec

Page Rank algorithm in 1 billion record url

What is PySpark?

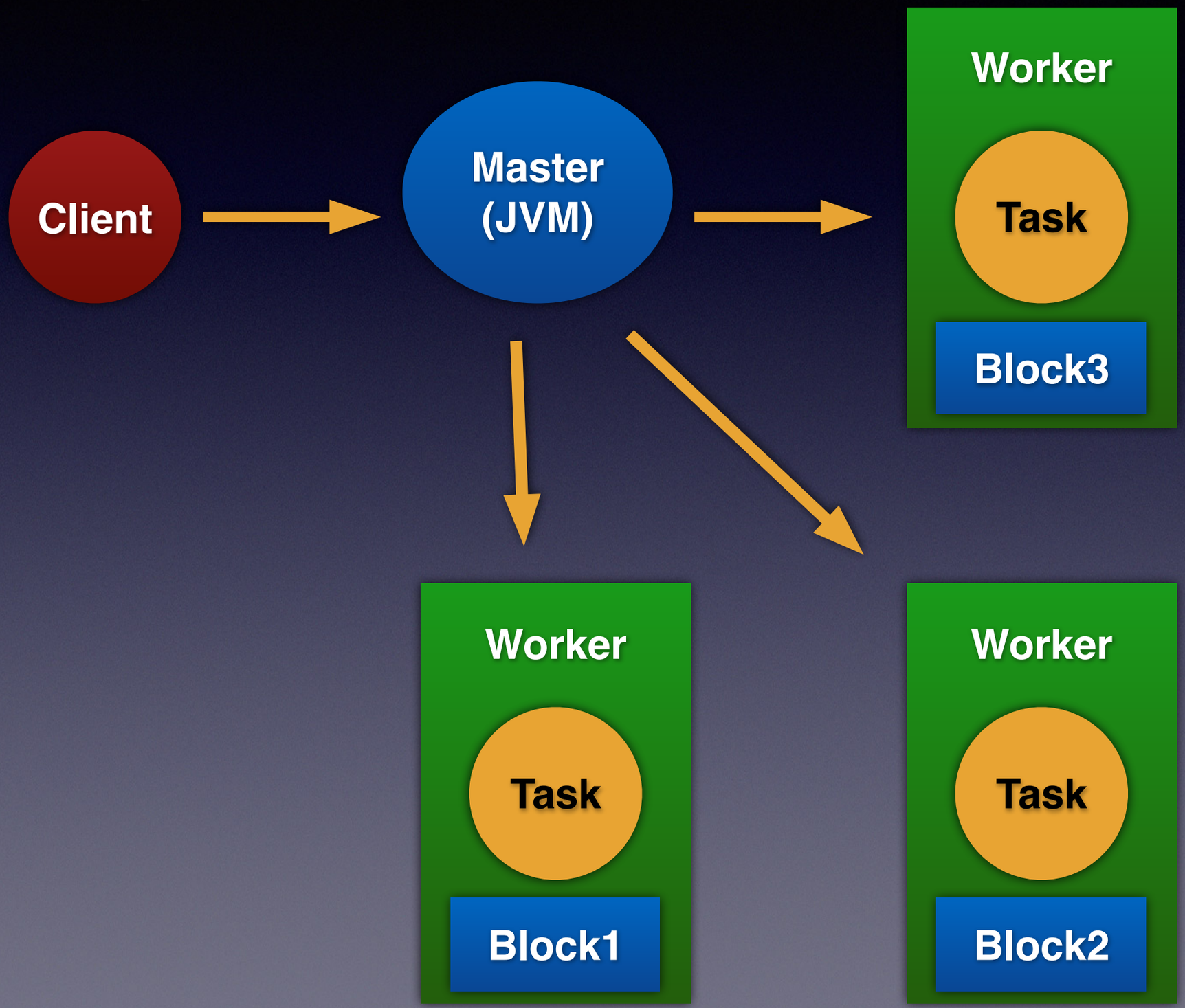
Spark API

- Multi Language API
  - JVM: Scala, JAVA
  - PySpark: Python

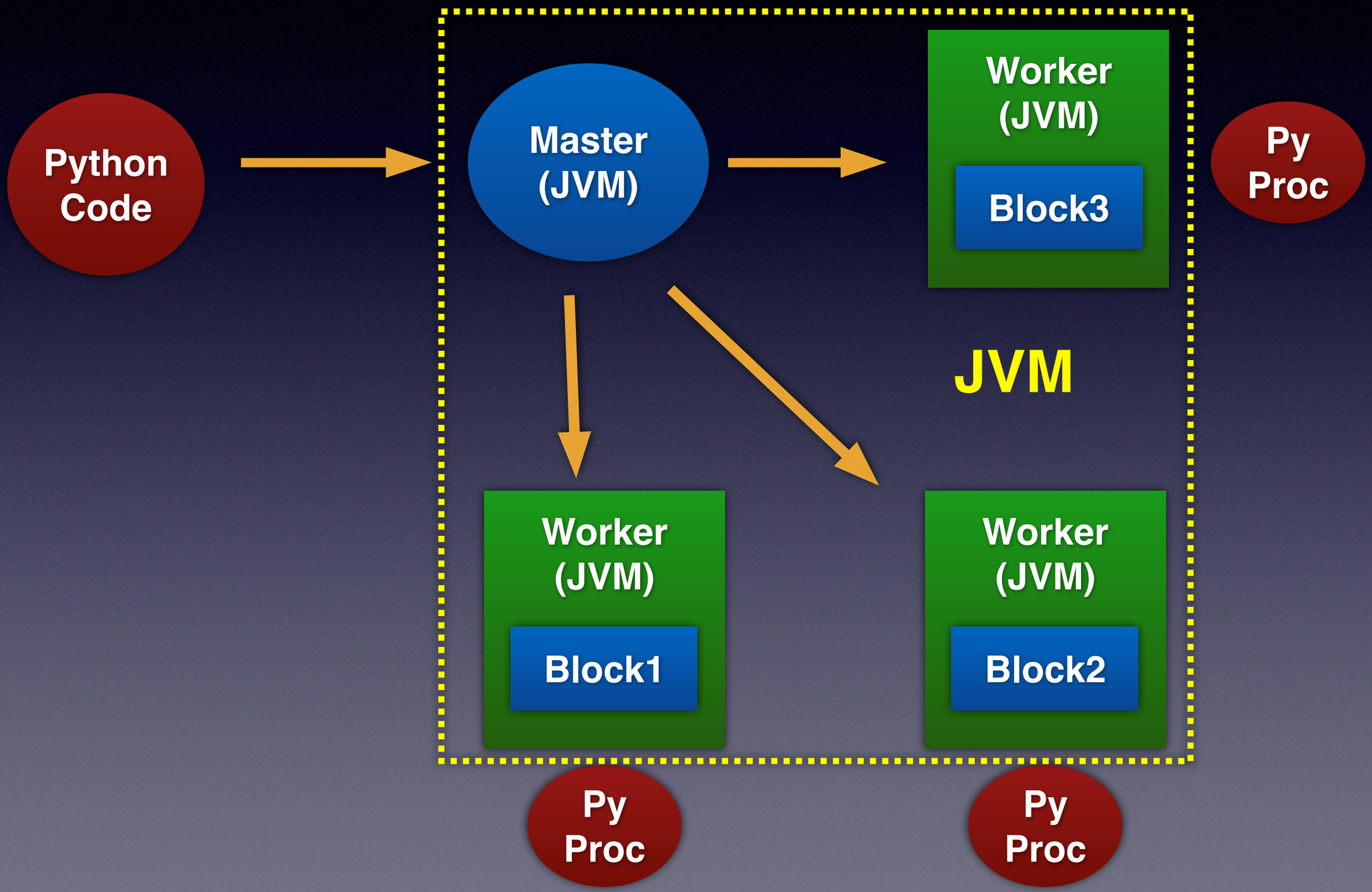
PySpark

- Process via Python
  - CPython
    - Python lib (NumPy, Scipy…)
- Storage and transfer data in Spark
  - HDFS access/Networking/Fault-recovery
  - scheduling/broadcast/checkpointing/

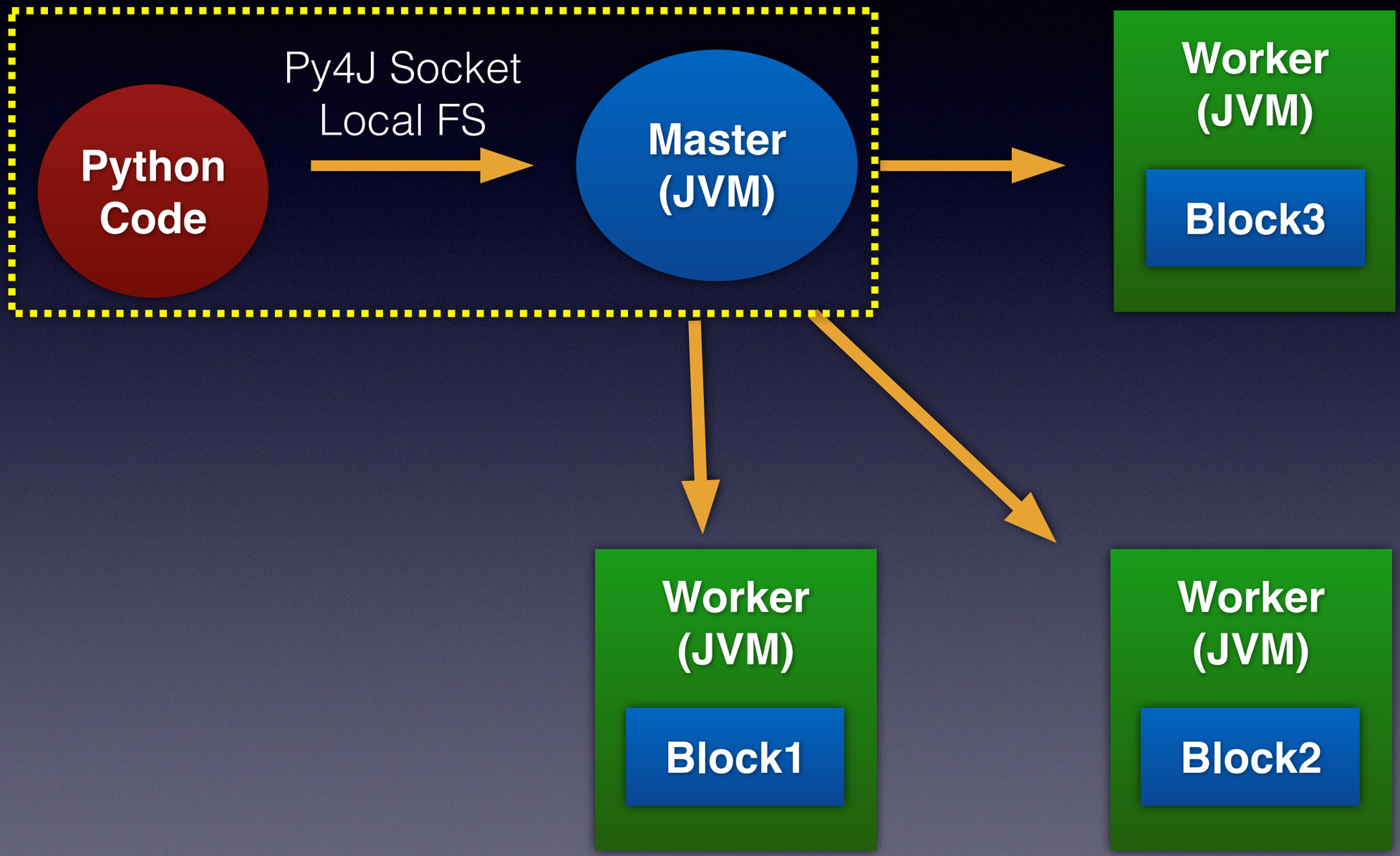
Spark Architecture



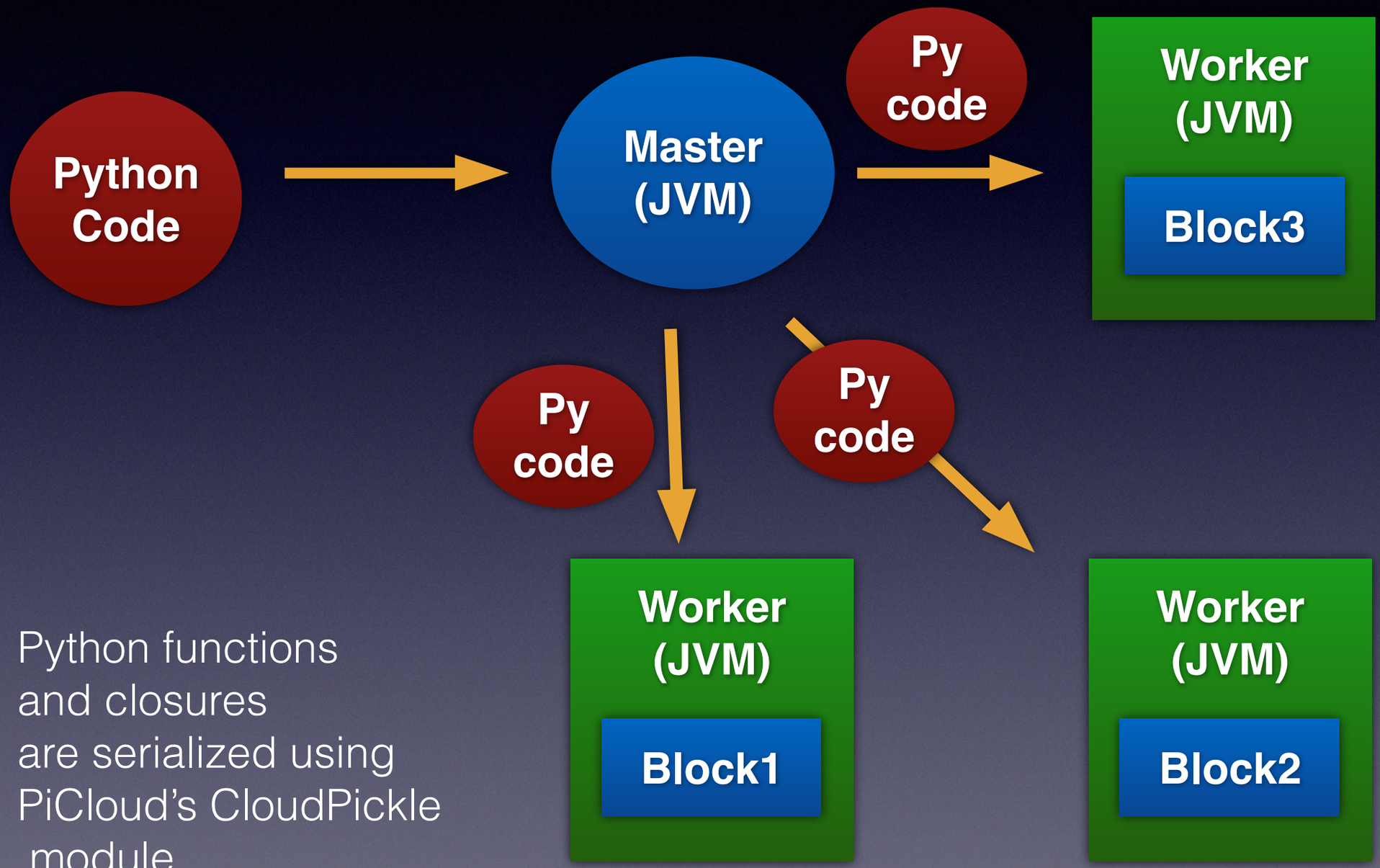
PySpark Architecture



PySpark Architecture

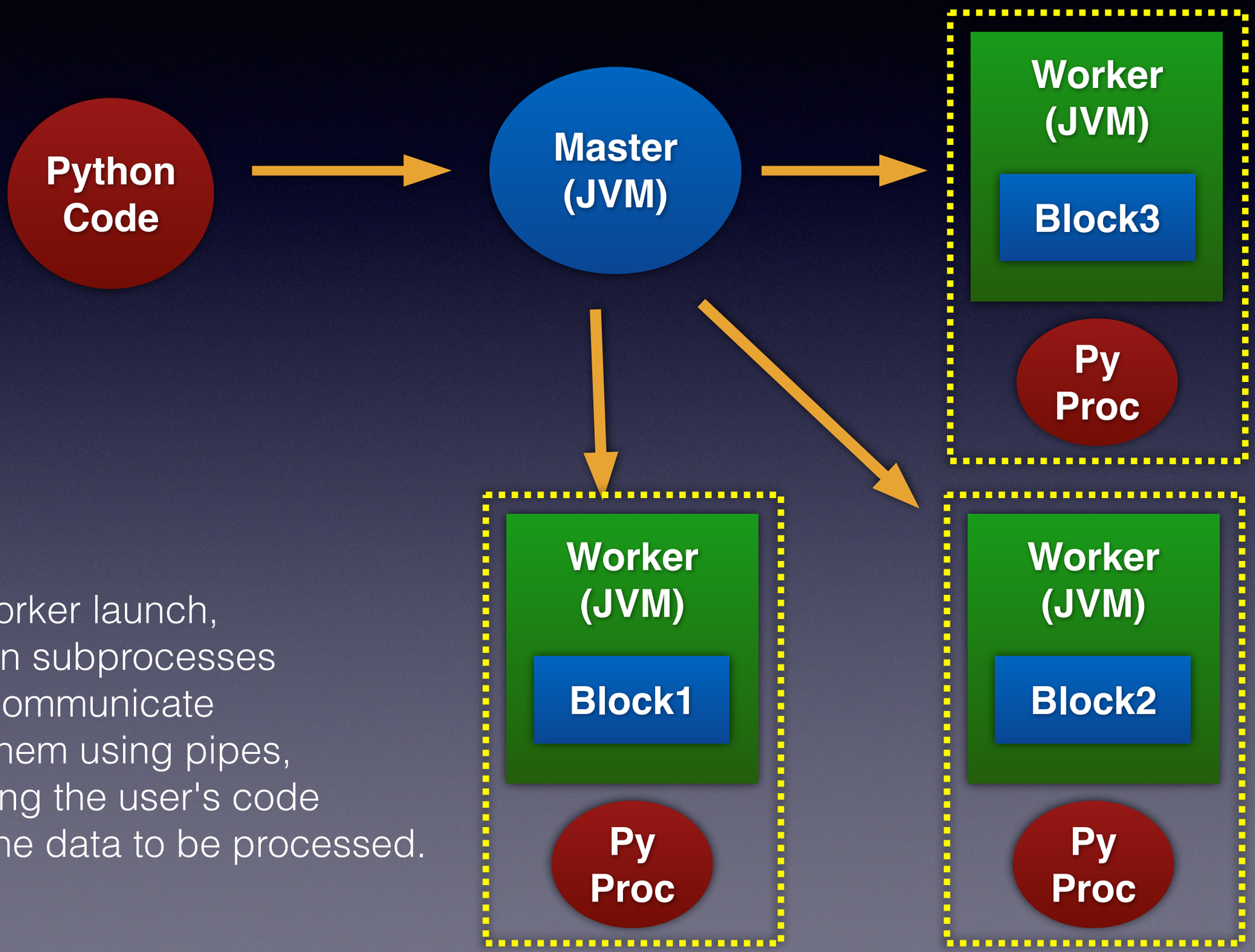


PySpark Architecture



Python functions
and closures
are serialized using
PiCloud's CloudPickle
module

PySpark Architecture



On worker launch,
Python subprocesses
and communicate
with them using pipes,
sending the user's code
and the data to be processed.

```
Processes: 216 total, 4 running, 2 stuck, 210 sleeping, 1219 threads
Load Avg: 1.84, 1.91, 1.89  CPU usage: 14.54% user, 3.13% sys, 82.32% idle  SharedLibs:
MemRegions: 62813 total, 2926M resident, 65M private, 1158M shared. PhysMem: 7398M used
VM: 523G vsize, 1026M framework vsize, 2188651(0) swapins, 2465680(0) swapouts.
Networks: packets: 2442860/1480M in, 2554811/1366M out. Disks: 3821317/67G read, 176313

PID    COMMAND      %CPU  TIME     #TH   #WQ  #PORT #MREG MEM    RPRVT  PURG   CMPRS  VI
92727  mdflagwriter 0.0   00:00.37 2     1    24    91    4096B  0B     0B     0B     94
92480  cfprefsd     0.0   00:00.27 2     1    29    47    608K   416K   0B     4096B  89
92479  distnoted    0.0   00:00.49 2     0    38    44    768K   596K   0B     0B     89
92464  installd     0.0   00:04.37 3     1    56    196   1120K  932K   0B     20K    1
92448  launchd      0.0   00:12.24 2     0    68    47    428K   476K   0B     0B     91
92417  hdiejectd    0.0   00:00.11 3     1    33    49    348K   256K   0B     0B     97
92412  diskimages-h 0.0   00:00.52 4     1    87    80    1016K  776K   0B     16K    10
61944  com.apple.ap 0.0   00:00.03 2     1    34    54    400K   240K   0B     16K    89
52926  Python       100.0 00:02.45 1/1   0    7     274+  63M+   62M+   0B     0B     80
52925  screencaptur 1.1   00:00.   2     1    29    103+  1992K+ 992K+  16K    0B     42
52923  Python       0.0   00:00.   1     0    7     135   976K   564K   0B     0B     10
52922  Python       0.0   00:00.   1     0    7     135   996K   584K   0B     0B     11
52921  Python       0.0   00:00.00 1     0    7     135   984K   576K   0B     0B     11
52920  Python       0.0   00:00.00 1     0    7     136   1000K  588K   0B     0B     21
52919  Python       0.0   00:00.00 1     0    7     134   972K   564K   0B     0B     68
52918  Python       0.0   00:00.00 1     0    7     135   976K   568K   0B     0B     10
52917  Python       0.0   00:00.00 1     0    7     135   1040K  324K   0B     0B     44
52916  Python       0.0   00:00.00 1     0    7     134   1176K  600K   0B     0B     72
52914  Python       0.0   00:00.09 1     0    16    134   9360K  452K   0B     0B     57
52901  top          10.1  00:00.99 1/1   0    33    44    2720K  2488K  0B     0B     72
52895  bash         0.0   00:00.00 1     0    19    34    932K   780K   0B     0B     40
52892  sh           0.0   00:00.00 1     0    19    31    544K   384K   0B     0B     30
```

A lot of python processes

How to write PySpark application?

Python Word Count

- file = spark.textFile("hdfs://...")

Access data via Spark API

- counts = file.flatMap(lambda line: line.split(" ")) \
- .map(lambda word: (word, 1)) \
- .reduceByKey(lambda a, b: a + b)
- counts.saveAsTextFile("hdfs://...")

Process via Python

Python Word Count

- counts = file.flatMap(lambda line: line.split(" ")) \

Original text

You can find the latest Spark documentation, including the guide

List

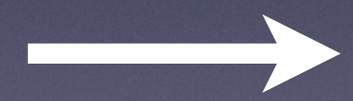
['You', 'can', 'find', 'the', 'latest', 'Spark', 'documentation,', 'including', 'the', ‘guide’]

Python Word Count

- .map(lambda word: (word, 1))

List

['You', 'can', 'find', 'the',
'latest', 'Spark',
'documentation,',
'including', 'the', ‘guide’]



Tuple List

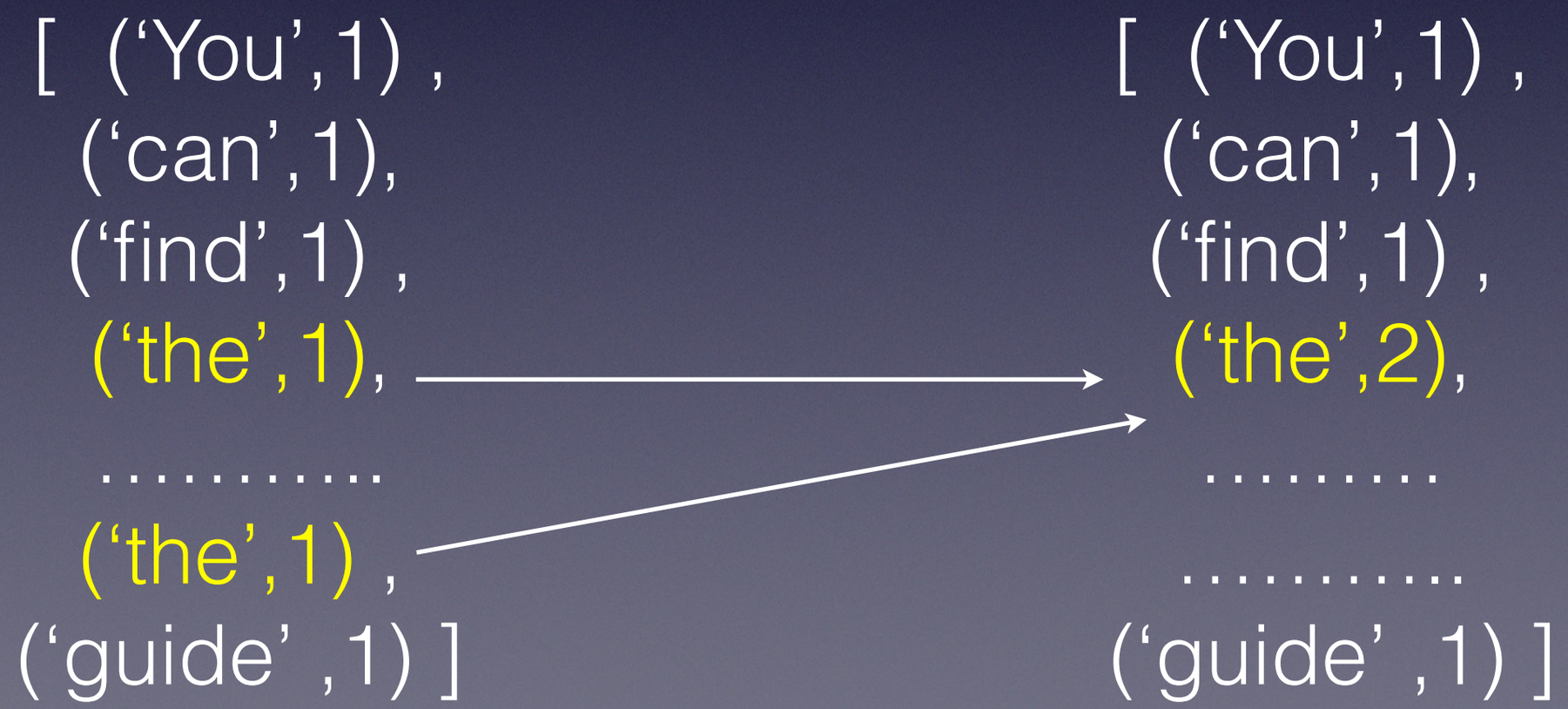
[(‘You’,1) , (‘can’,1),
(‘find’,1) , (‘the’,1) ….,
…….
(‘the’,1) , (‘guide’ ,1)]

Python Word Count

- .reduceByKey(lambda a, b: a + b)

Tuple List

Reduce Tuple List



Can I use ML python lib on PySpark?

PySpark + scikit-learn

- sgd = lm.SGDClassifier(loss='log')
- for ii in range(ITERATIONS):
- sgd = sc.parallelize(…) \
- .mapPartitions(lambda x:…) \
- .reduce(lambda x, y: merge(x, y))

Use scikit-learn in Single mode(master)

Use scikit-learn function in cluster mode , deal with partial data

Cluster operation

Source Code is From : http://0rz.tw/o2CHT

PySpark support MLlib

- MLlib is spark version machine learning lib
- Example: KMeans.train(parsedData, 2, maxIter=10, runs=30, "random")
- Check it out on http://0rz.tw/M35Rz