Lorem Ipsum Dolor

Spark入门分享

程怡

2015-05-29

目录

What is Spark?

Fast and expressive cluster computing system compatible with Apache Hadoop

» Works with any Hadoop-supported storage system and data format (HDFS, S3, SequenceFile, …)

Improves efficiency through:

» In-memory computing primitives

» General computation graphs

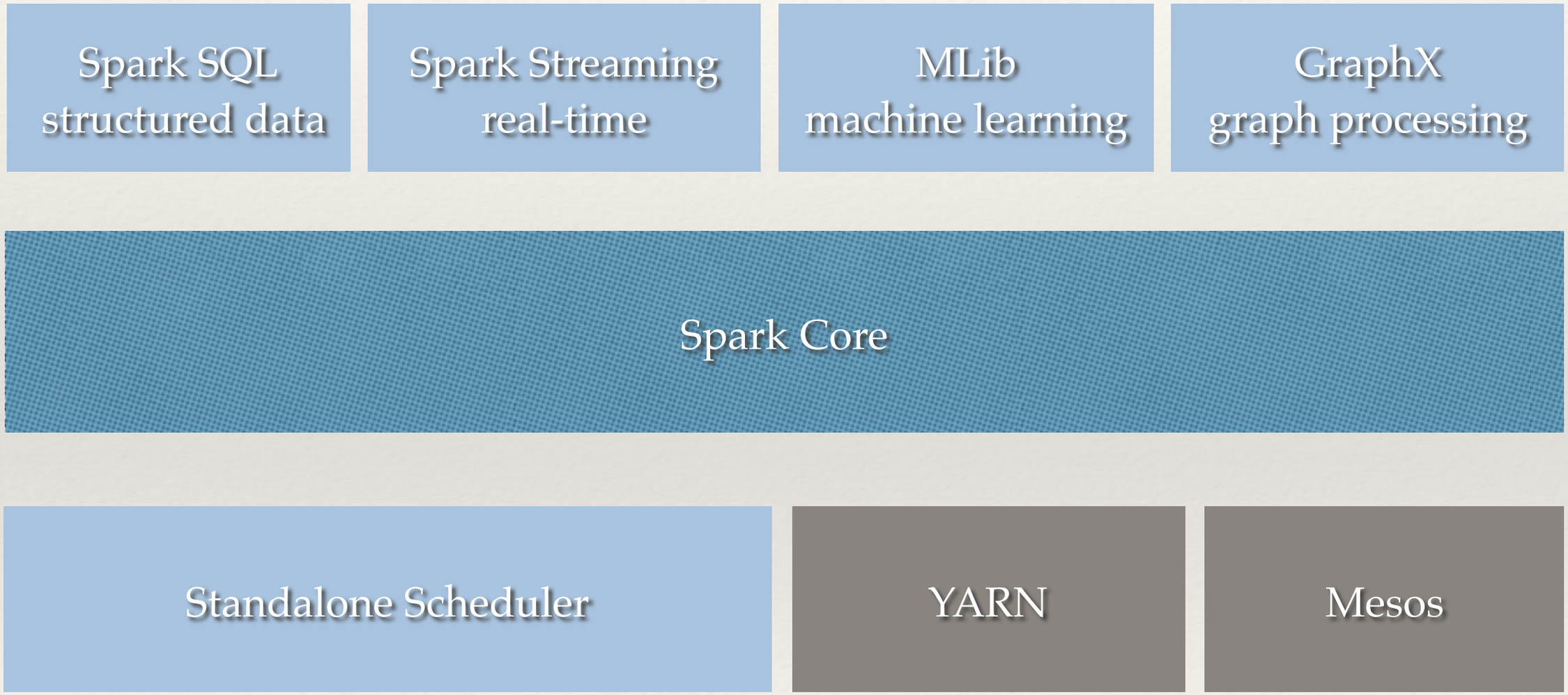
As much as 30× faster

Improves usability through rich Scala and Java APIs and interactive shell

Often 2-10× less code

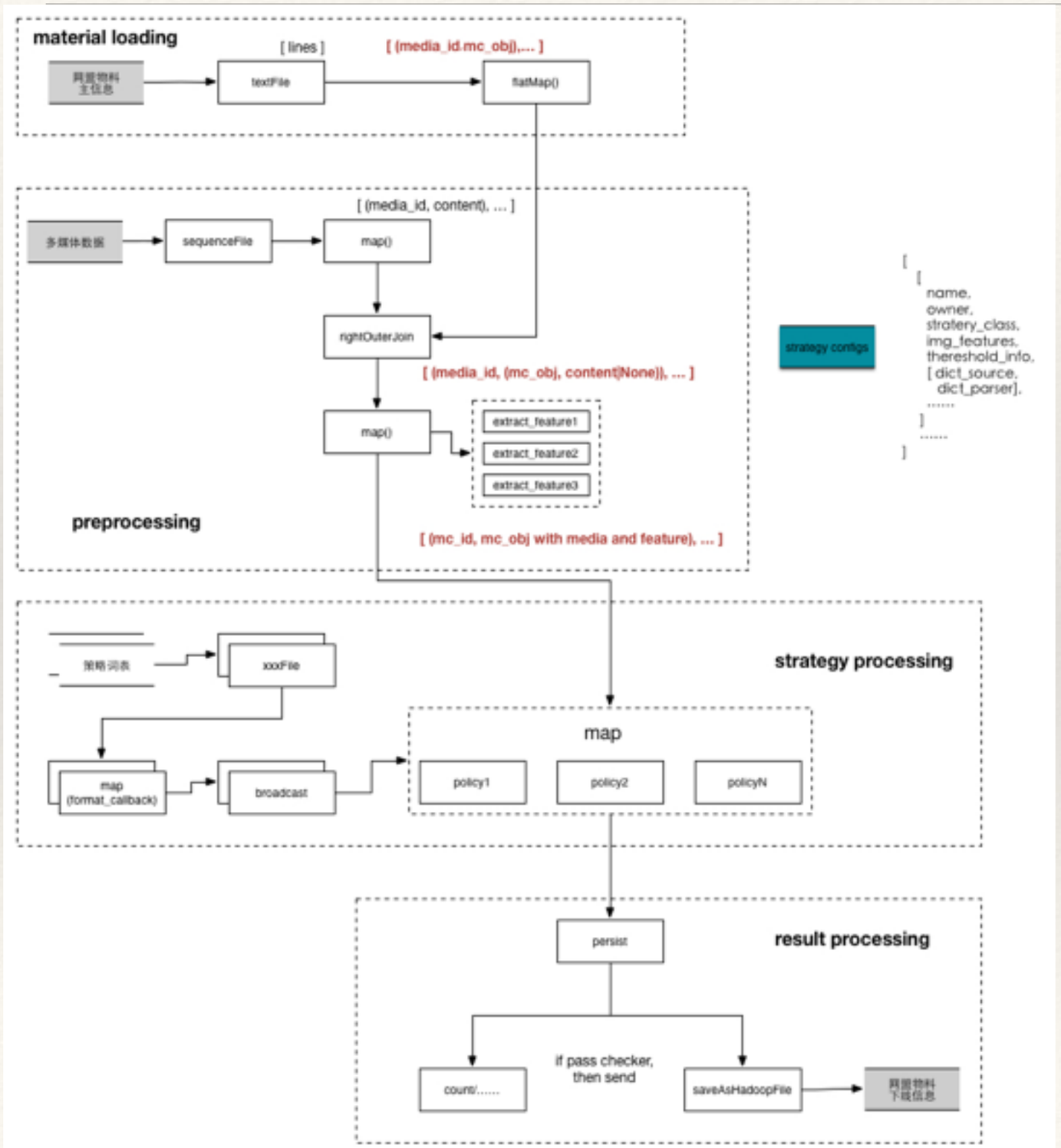
matei-zaharia-part-1-amp-camp-2012-spark-intro

The Spark Stack



OReilly.Learning.Spark.Lightning-Fast.Big.Data.Analysis

Spark here



基于SparkCore，多媒体物料审核框架

- 扩展性
- 性能
- 工程与策略的隔离
- ……

and SparkSQL@luofei

Spark here

1.5TB 输入数据，1次outjoin，
2组特征抽取，2组策略，
运行时间11分钟

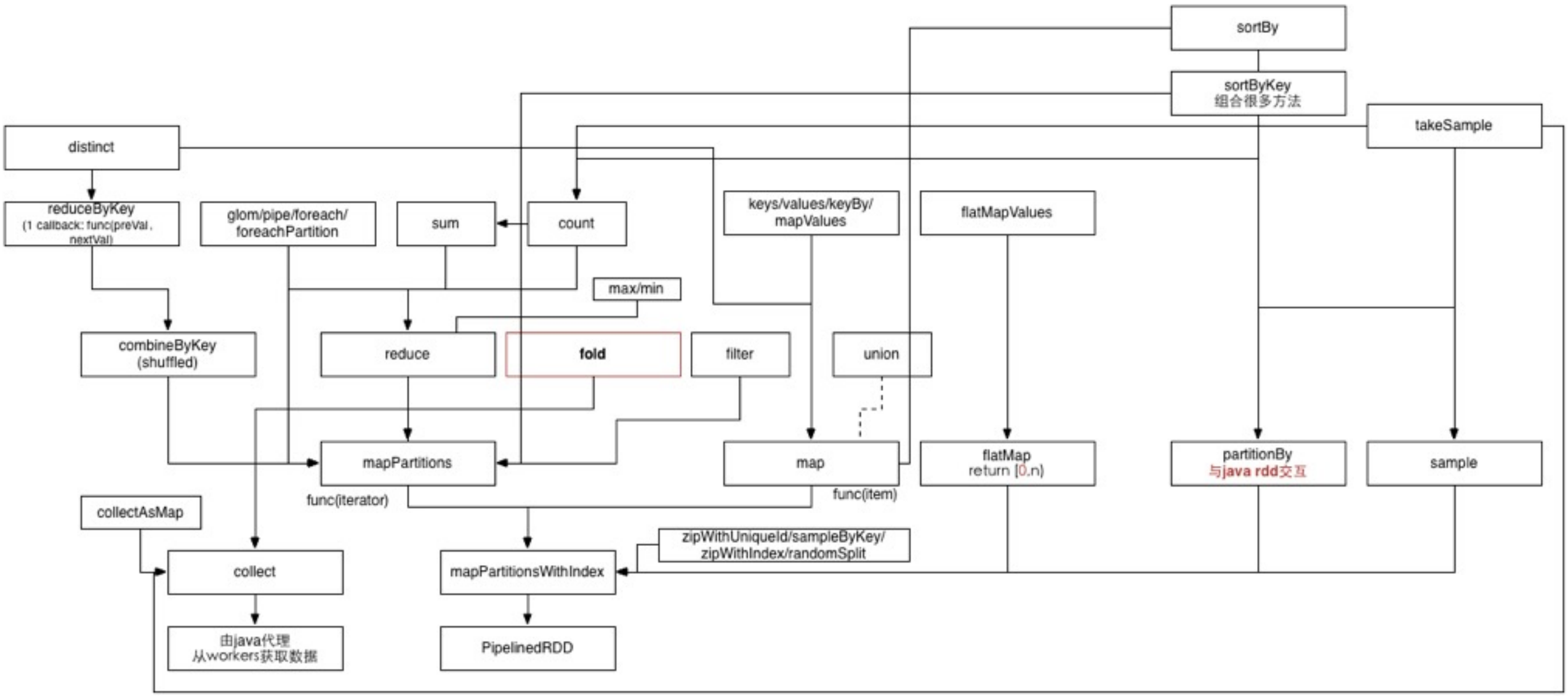
app-20150527140357-5025	davinci_beidou_media_review	1869	100.0 GB	2015/05/27 14:03:57	chengyi02	FINISHED	11 min

Stage Id	Description		Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
13	saveAsTextFile at NativeMethodAccessorImpl.java:-2	+details	2015/05/27 14:15:07	3 s	100/100	1034.2 MB			
10	count at /home/users/chengyi02/svn-root/app/ecom/darwin/dr-davinci/src/beidou/schedule_frame.py:84		2015/05/27 14:15:06	1.0 s	100/100	1034.2 MB			
7	saveAsTextFile at NativeMethodAccessorImpl.java:-2	+details	2015/05/27 14:14:56	4 s	100/100			366.9 MB	
6	reduceByKey at build/bdist.linux-x86_64/egg/src/schedule_frame.py:444		2015/05/27 14:08:04	6.9 min	1000/1000			2.1 TB	372.9 MB
5	rightOuterJoin at build/bdist.linux-x86_64/egg/src/schedule_frame.py:307		2015/05/27 14:05:27	2.6 min	7747/7747	1546.9 GB			2.2 TB
4	collect at build/bdist.linux-x86_64/egg/src/schedule_frame.py:277		2015/05/27 14:04:35	50 s	1/1	39.3 KB			
3	collect at build/bdist.linux-x86_64/egg/src/schedule_frame.py:277		2015/05/27 14:04:21	5 s	49/49	10.8 MB			
2	first at SerDeUtil.scala:202	+details	2015/05/27 14:04:18	2 s	1/1	256.0 MB			
1	collect at build/bdist.linux-x86_64/egg/src/schedule_frame.py:467		2015/05/27 14:04:11	2 s	1/1	188.0 B			
0	collect at build/bdist.linux-x86_64/egg/src/schedule_frame.py:467		2015/05/27 14:04:02	2 s	1/1	219.0 B			

Spark Quick Start

- WordCount:
  - sc.**textFile**(*file_path*).**flatMap**(*lambda line: line.split(" ")*).**count**()
- Intersection:
  - rdd1 = sc.textFile(file_path1).flatMap(split_to_word)
  - rdd2 = sc.textFile(file_path2).flatMap(split_to_word)
  - intersection = rdd1.**intersection**(rdd2).collect()
- Broadcast:
  - b_list = sc.**broadcast**(load_dict(dict_local_path))
  - rdd1.intersection(rdd2).filter(lambda word: word in **b_list.value**).collect()

Spark transformations and actions



Spark Debug

- UnitTest
- Local mode: spark-submit —master local[*]
- Cluster mode: Worker Log (debug level)

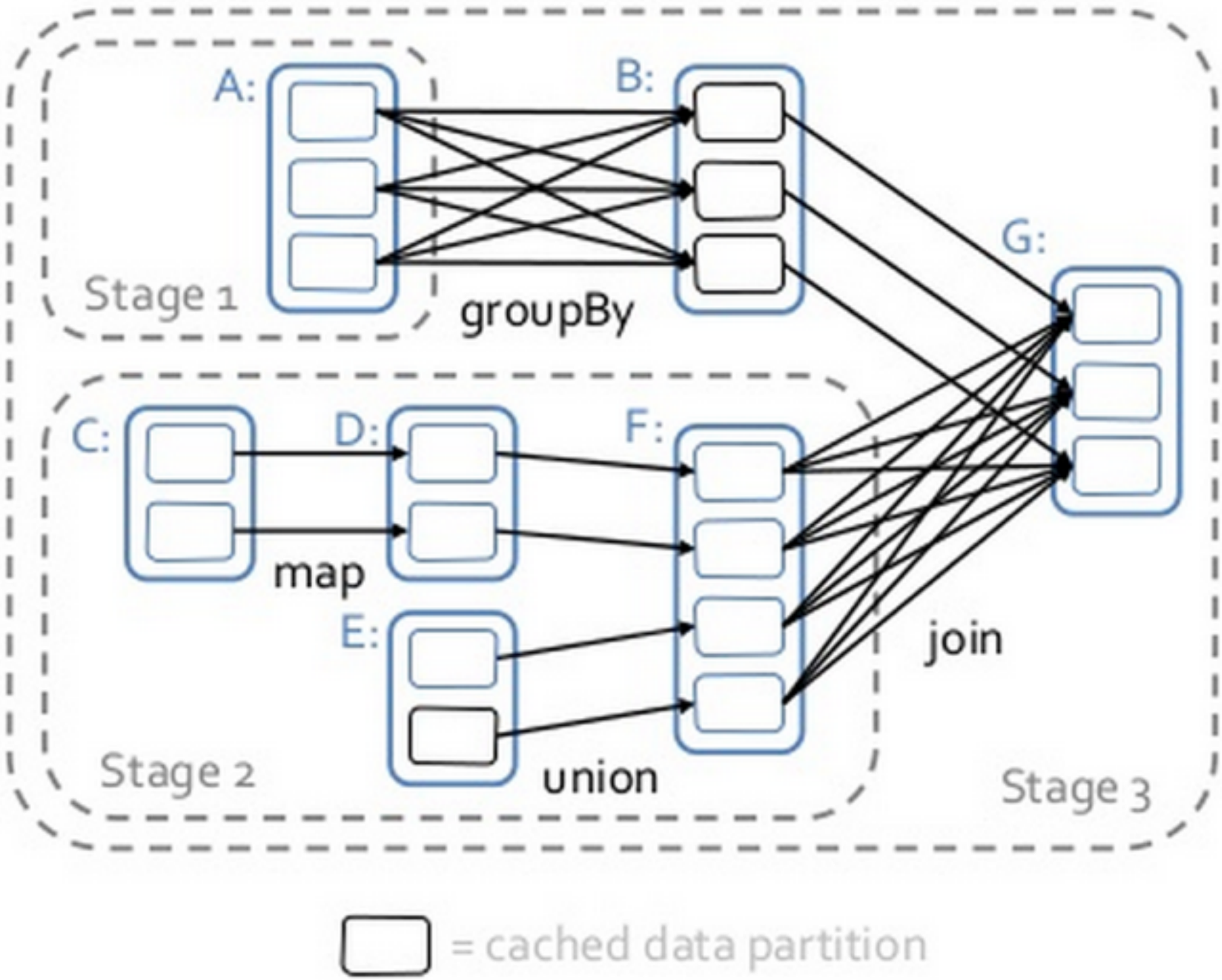
- -Dlog4j.configuration=ftp://cq01-rdqa-dev006.cq01.baidu.com/tmp/chengyi02
- log4j.propertieslog4j.rootCategory=**DEBUG**, console
- log4j.appender.console=org.apache.log4j.ConsoleAppender
- log4j.appender.console.target=System.err
- log4j.appender.console.layout=org.apache.log4j.PatternLayout
- log4j.appender.console.layout.ConversionPattern=%d{yy/MM/dd HH:mm:ss} %p %c{1}: %m%n

Spark Inner

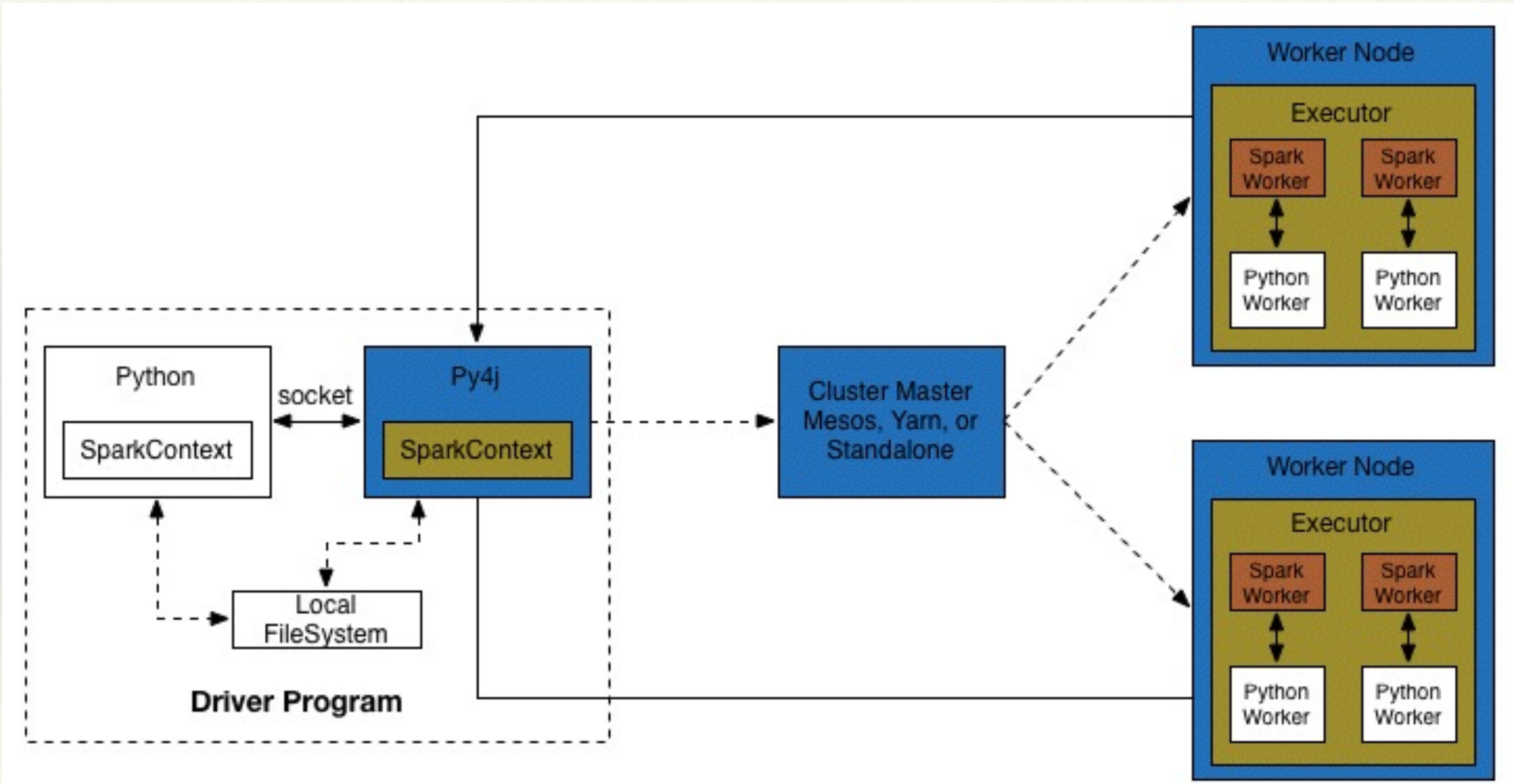
Dryad-like DAGs
Pipelines functions
within a stage

Cache-aware work
reuse & locality

Partitioning-aware
to avoid shuffles



Spark Inner


Worker Node
Executor
Spark Worker
Spark Worker
Python Worker
Python Worker
Python
SparkContext
socket
Py4j
SparkContext
Cluster Master
Mesos, Yarn, or
Standalone
Local FileSystem
Driver Program
Worker Node
Executor
Spark Worker
Spark Worker
Python Worker
Python Worker

Spark Inner

- **Questions**:
  - What's RDD? Spark如何容灾?
  - narrow vs wide dependency
  - 串行与并行的执行顺序
  - Driver、JVM Worker、Python Worker内存使用情况?
  - Workers、Driver间如何通信?
  - Python和Scala间如何通信
  - 如何复用RDD?
  - ……

Spark Inner

- core/src/main/scala/org/apache/spark/
- python/pyspark/
- streaming/
- mllib/
- graphx/

Tuning Spark

- 提高并发度：
  - 输入分片、spark.cores.max、numPartitions
- 减少磁盘IO：
  - 尽量使用内存，shuffle操作存在于scala和python里
- GC Tuning：spark.executor.extraJavaOptions
- 避免数据倾斜
- ……

https://spark.apache.org/docs/latest/tuning.html

Tuning Spark

Summary Metrics for 7764 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	0.2 s	15 s	17 s	19 s	1.7 min
Scheduler Delay	14 ms	26 ms	44 ms	63 ms	1 s
Task Deserialization Time	0 ms	2 ms	2 ms	3 ms	1 s
GC Time	0 ms	0.4 s	0.9 s	1 s	7 s
Result Serialization Time	0 ms	0 ms	0 ms	0 ms	14 ms
Getting Result Time	0 ms	0 ms	0 ms	0 ms	0 ms
Input	2.0 MB	254.8 MB	256.0 MB	256.0 MB	280.9 MB
Shuffle Write	2.8 MB	360.3 MB	362.2 MB	364.2 MB	419.0 MB

```
         5297434252 function calls (5297427252 primitive calls) in 353250.282 seconds

   Ordered by: internal time

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
62603148 313203.187    0.005 313203.187    0.005 {method 'read' of 'file' objects}
31301074 22194.833    0.001 22194.833    0.001 {cPickle.loads}
16958730 7980.783    0.000 7980.783    0.000 {_basic_image_lib.flash_decode}
55610843 1700.863    0.000 1793.400    0.000 feature_murmur_sign.py:33(run)
55610843 1071.213    0.000 9820.618    0.000 feature_flash_decoder.py:37(run)
```

Spark Resource

- http://spark.apache.org/docs/latest/
- https://databricks.com/spark/about
- http://wiki.baidu.com/pages/viewpage.action?pageId=38012601
- http://wiki.babel.baidu.com/twiki/bin/view/Com/Ecom/Aka/Internal/Spark%E8%B5%84%E6%96%99

Thanks~