livelessons
video instruction from technology experts

Lesson 4: Spark Internals

4.12 Making Spark Fly: Caching

Caching

Spark 1.4.1 Jobs Stages Storage Environment Executors **Performance Tuning** application UI

Storage

`airline_rows`

RDD Name	Storage Level	Cached Partitions	Fraction Cached	Size in Memory	Size in ExternalBlockStore	Size on Disk
PythonRDD	Memory Serialized 1x Replicated	11	100%	115.0 MB	0.0 B	0.0 B
PythonRDD	Memory Serialized 1x Replicated	2	100%	4.5 MB	0.0 B	0.0 B
PythonRDD	Memory Serialized 1x Replicated	2	100%	30.3 KB	0.0 B	0.0 B

NOTE: In Python stored objects always serialized with Pickle (so storage level doesn't matter)

RDD Storage Info for PythonRDD

Storage Level: Memory Serialized 1x Replicated
Cached Partitions: 11
Total Partitions: 11
Memory Size: 115.0 MB
Disk Size: 0.0 B

Data Distribution on 5 Executors

Host	Memory Usage	Disk Usage
10.25.111.149:60895	0.0 B (265.1 MB Remaining)	0.0 B
10.25.111.149:61548	68.5 MB (194.4 MB Remaining)	0.0 B
10.25.111.149:60907	0.0 B (265.1 MB Remaining)	0.0 B
10.25.111.149:61547	46.5 MB (216.3 MB Remaining)	0.0 B
10.25.111.149:60908	0.0 B (265.1 MB Remaining)	0.0 B

11 Partitions

Block Name	Storage Level	Size in Memory	Size on Disk	Executors
rdd_218_0	Memory Serialized 1x Replicated	11.0 MB	0.0 B	10.25.111.149:61547
rdd_218_1	Memory Serialized 1x Replicated	11.7 MB	0.0 B	10.25.111.149:61548
rdd_218_10	Memory Serialized 1x Replicated	11.0 MB	0.0 B	10.25.111.149:61548
rdd_218_2	Memory Serialized 1x Replicated	10.6 MB	0.0 B	10.25.111.149:61548
rdd_218_3	Memory Serialized 1x Replicated	26.0 B	0.0 B	10.25.111.149:61547
rdd_218_4	Memory Serialized 1x Replicated	10.9 MB	0.0 B	10.25.111.149:61547
rdd_218_5	Memory Serialized 1x Replicated	11.6 MB	0.0 B	10.25.111.149:61548

Spark Jobs (?)

Total Uptime: 3.1 h
Scheduling Mode: FIFO
Completed Jobs: 40
Failed Jobs: 2

▸ Event Timeline

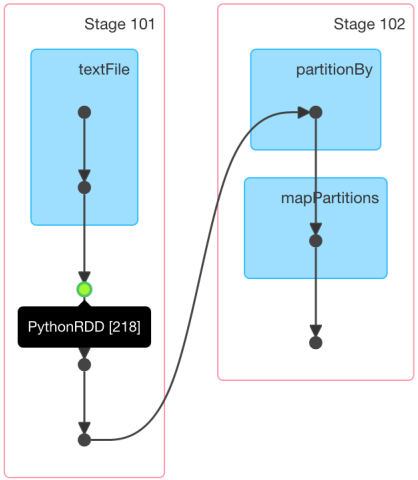
Completed Jobs (40)

Job Id (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
41 (Airline Data -- filtered)	cache second run runJob at PythonRDD.scala:366	2015/08/13 18:43:40	0.1 s	2/2 (1 skipped)	12/12 (11 skipped)
40 (Airline Data -- filtered)	cache second run sortBy at <ipython-input-45-c5ef31608b74>:9	2015/08/13 18:43:40	68 ms	1/1 (1 skipped)	11/11 (11 skipped)
39 (Airline Data -- filtered)	cache second run sortBy at <ipython-input-45-c5ef31608b74>:9	2015/08/13 18:43:29	11 s	2/2	22/22
38 (Airline Data -- filtered)	cache first run runJob at PythonRDD.scala:366	2015/08/13 18:43:29	0.1 s	2/2 (1 skipped)	12/12 (11 skipped)
37 (Airline Data -- filtered)	cache first run sortBy at <ipython-input-44-a8444b85785e>:9	2015/08/13 18:43:28	79 ms	1/1 (1 skipped)	11/11 (11 skipped)
36 (Airline Data -- filtered)	cache first run sortBy at <ipython-input-44-a8444b85785e>:9	2015/08/13 18:31:48	12 min	2/2	22/22
35 (Airline Data -- filtered)	reduceByKey + filtered runJob at PythonRDD.scala:366	2015/08/13 18:31:48	0.2 s	2/2 (1 skipped)	12/12 (11 skipped)
34 (Airline Data -- filtered)	reduceByKey + filtered sortBy at <ipython-input-41-4f121cf53178>:9	2015/08/13 18:31:47	99 ms	1/1 (1 skipped)	11/11 (11 skipped)
33 (Airline Data -- filtered)	reduceByKey + filtered sortBy at <ipython-input-41-4f121cf53178>:9	2015/08/13 18:17:58	14 min	2/2	22/22

Details for Job 39

Status: SUCCEEDED
Job Group: Airline Data -- filtered
Completed Stages: 2

▸ Event Timeline
▾ DAG Visualization



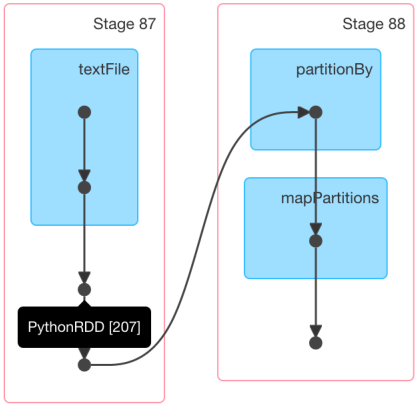
Completed Stages (2)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
102	cache second run sortBy at <ipython-input-45-c5ef31608b74>:9	2015/08/13 18:43:40	61 ms	11/11			78.0 KB	

Details for Job 33

Status: SUCCEEDED
Job Group: Airline Data -- filtered
Completed Stages: 2

▸ Event Timeline
▾ DAG Visualization



Completed Stages (2)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
88	reduceByKey + filtered sortBy at <ipython-input-41-4f121cf53178>:9	2015/08/13 18:31:47	98 ms	11/11			78.0 KB	
87	reduceByKey + filtered reduceByKey at <ipython-input-41-4f121cf53178>:7	2015/08/13 18:17:58	14 min	11/11	256.3 MB			78.0 KB

Persist Levels

Mode	*Advantage*
MEMORY_ONLY (default level)	*Store RDD as deserialized Java object in JVM*
MEMORY_AND_DISK	*Store RDD as deserialized Java object in JVM (but spill to disk for what doesn't fit in memory)*
MEMORY_ONLY_SER	*Store RDD as serialized Java object (more space efficient but CPU-intensive on reads)*
MEMORY_AND_DISK_SER	*Same as above but spill serialized partitions to disk*
DISK_ONLY	*Store RDD partitions only on disk*
MEMORY_ONLY_2, MEMORY_AND_DISK_2, etc.	*Same as above but replicate each partition on two cluster nodes*

Serialization/Deserialization

No silver bullet, but…

```
conf.set(‘spark.serializer”,
          ‘org.apache.spark.serializer.KryoSerializer’)
```

Performance Triage

- Task distribution (straggler tasks) due to:
  - Slow nodes/machines
  - Uneven data distribution/partitioning

- Individual task performance on single node
  - Memory allocation
  - Computationally expensive operations
  - Garbage collection processes

Review

- If you can't measure it, you can't improve it
- Rewriting functions can result in more efficient DAG scheduling
- Data locality can be optimized with partition functions
- Shuffles are an expensive operation (usually)

Next Up: Advanced Applications