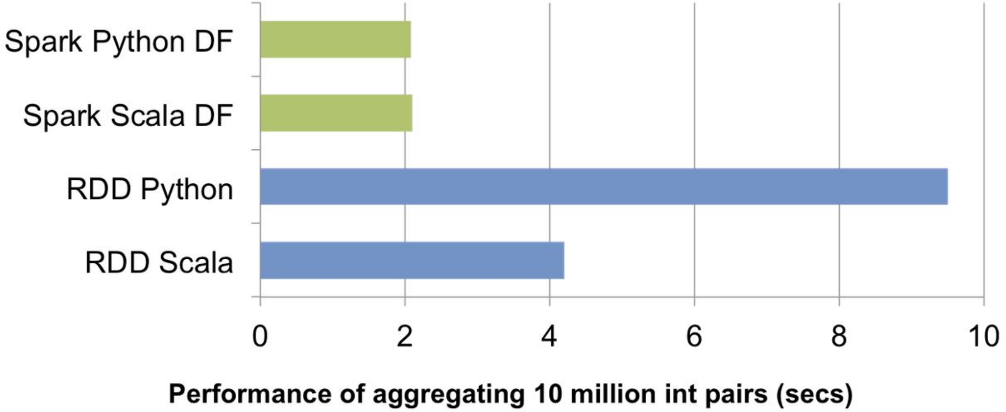
Final Frontier

December 5th, 2017

RDDs and Dataframes

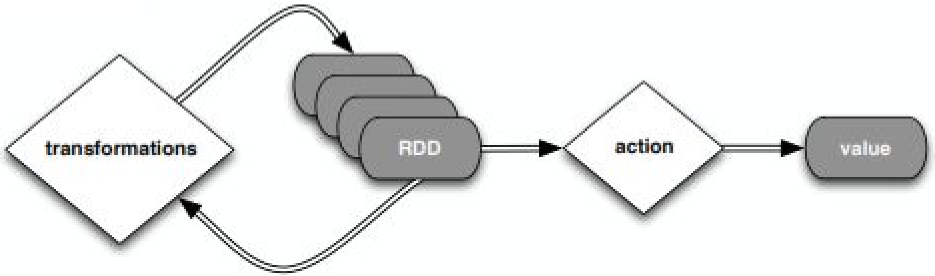
- Resilient Distributed Dataset
  - Fault tolerant
  - Distributed across nodes
  - Good for unstructured data
- Dataframes
  - Subclass of RDD
  - More query optimization
  - Structure data



source

Lazy Evaluation

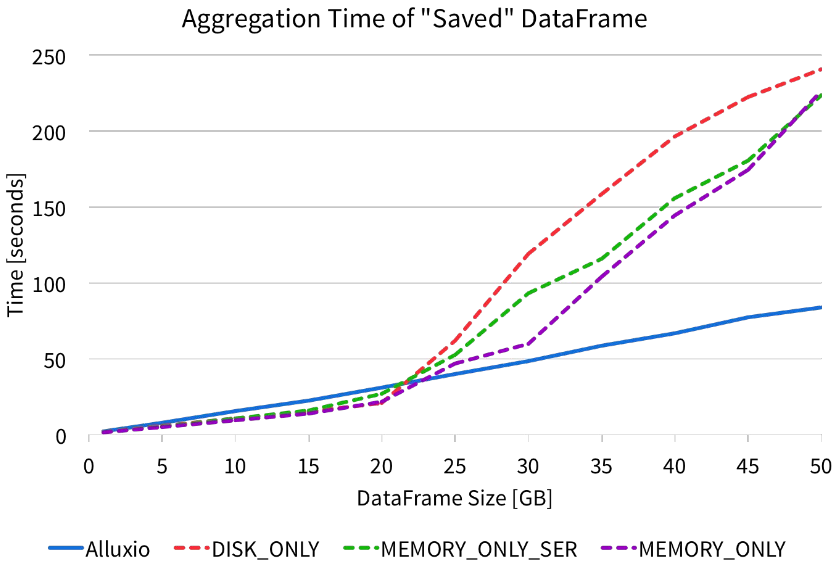
- Everything in Spark
- Each new transformation set to the side
- New dataset not computed until an action requests the results
- Common actions:
  - Count
  - Take
  - Collect



source

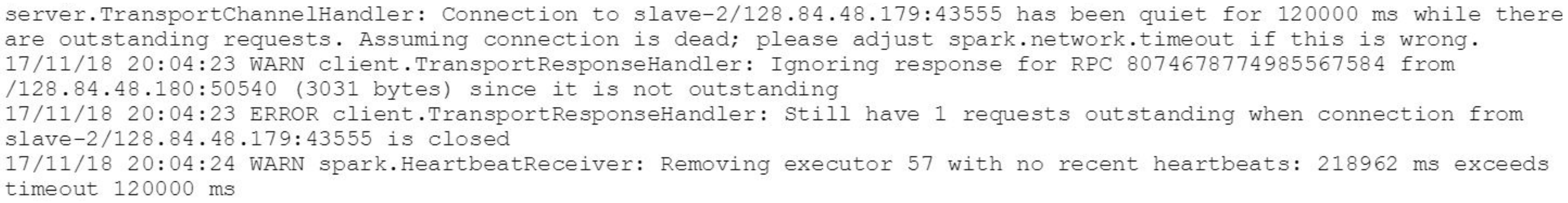
Caching and Persisting

- Store the resulting dataframe for future use
- Persisting options:
  - Memory only
  - Memory and disk
  - Seralized
  - Replicated
- When to persist:
  - Any repetitive access to the data frame
  - Access the same data frame multiple times

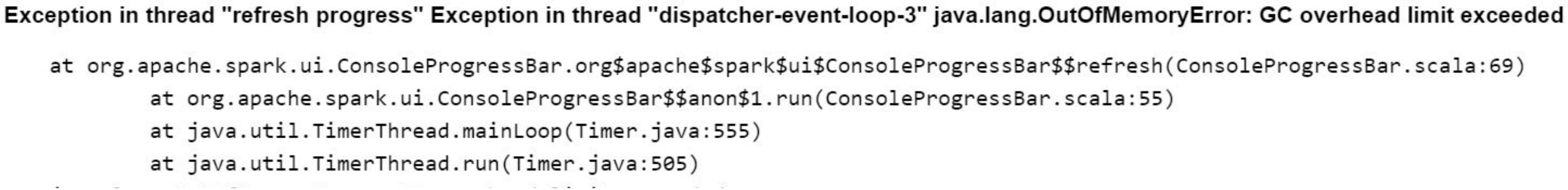


source

In practice...

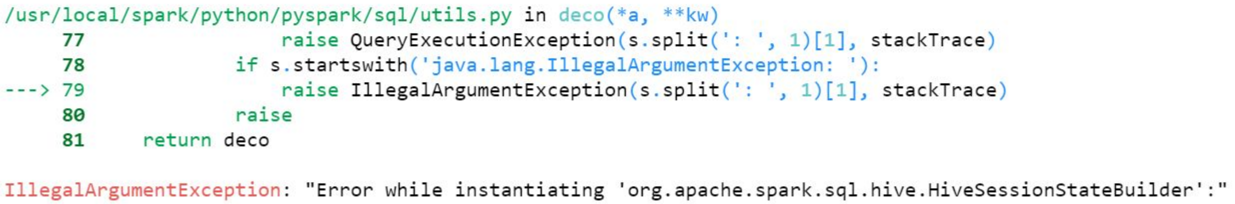


```
server.TransportChannelHandler: Connection to slave-2/128.84.48.179:43555 has been quiet for 120000 ms while there
are outstanding requests. Assuming connection is dead; please adjust spark.network.timeout if this is wrong.
17/11/18 20:04:23 WARN client.TransportResponseHandler: Ignoring response for RPC 8074678774985567584 from
/128.84.48.180:50540 (3031 bytes) since it is not outstanding
17/11/18 20:04:23 ERROR client.TransportResponseHandler: Still have 1 requests outstanding when connection from
slave-2/128.84.48.179:43555 is closed
17/11/18 20:04:24 WARN spark.HeartbeatReceiver: Removing executor 57 with no recent heartbeats: 218962 ms exceeds
timeout 120000 ms
```



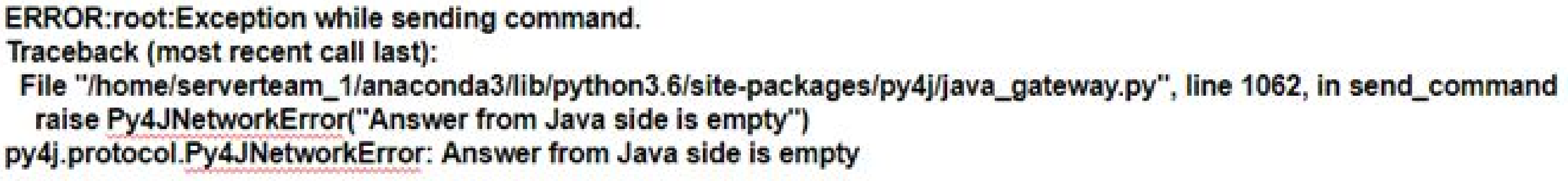
```
Exception in thread "refresh progress" Exception in thread "dispatcher-event-loop-3" java.lang.OutOfMemoryError: GC overhead limit exceeded

  at org.apache.spark.ui.ConsoleProgressBar.org$apache$spark$ui$ConsoleProgressBar$$refresh(ConsoleProgressBar.scala:69)
          at org.apache.spark.ui.ConsoleProgressBar$$anon$1.run(ConsoleProgressBar.scala:55)
          at java.util.TimerThread.mainLoop(Timer.java:555)
          at java.util.TimerThread.run(Timer.java:505)
```



```
/usr/local/spark/python/pyspark/sql/utils.py in deco(*a, **kw)
     77                 raise QueryExecutionException(s.split(': ', 1)[1], stackTrace)
     78             if s.startswith('java.lang.IllegalArgumentException: '):
---> 79                 raise IllegalArgumentException(s.split(': ', 1)[1], stackTrace)
     80             raise
     81     return deco

IllegalArgumentException: "Error while instantiating 'org.apache.spark.sql.hive.HiveSessionStateBuilder':"
```



```
ERROR:root:Exception while sending command.
Traceback (most recent call last):
  File "/home/serverteam_1/anaconda3/lib/python3.6/site-packages/py4j/java_gateway.py", line 1062, in send_command
    raise Py4JNetworkError("Answer from Java side is empty")
py4j.protocol.Py4JNetworkError: Answer from Java side is empty
```