Debugging PySpark

Or why is there a JVM stack trace and what does it mean?

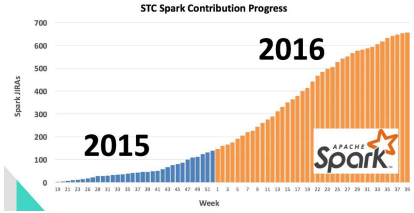
Holden Karau

IBM - Spark Technology Center

Who am I?

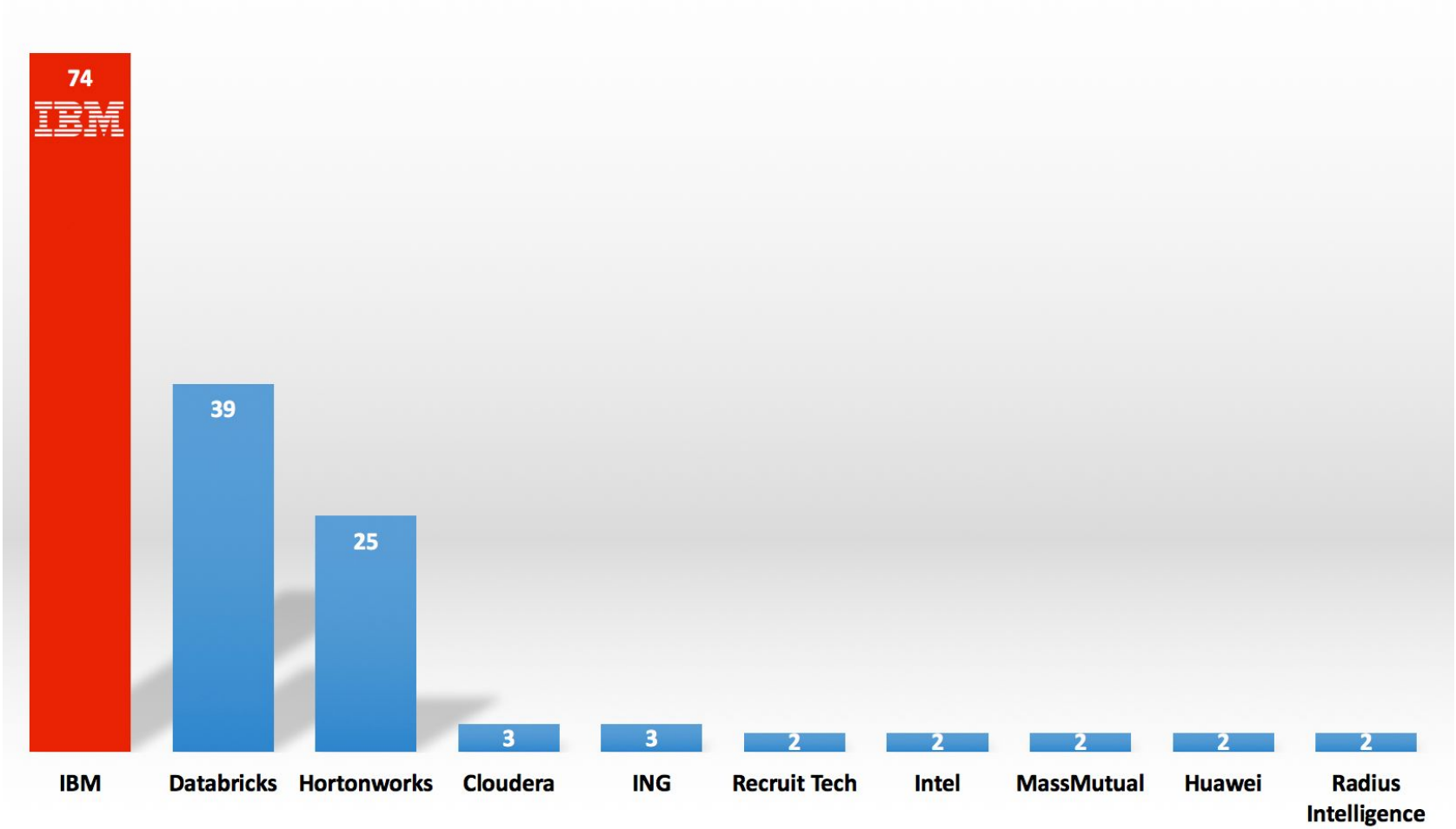
- My name is Holden Karau
- Prefered pronouns are she/her
- I'm a Principal Software Engineer at IBM's Spark Technology Center
- Apache Spark committer (as of last month!) :)
- previously Alpine, Databricks, Google, Foursquare & Amazon
- co-author of Learning Spark & Fast Data processing with Spark
  - co-author of a new book focused on Spark performance coming this year*
- @holdenkarau
- Slide share http://www.slideshare.net/hkarau
- Linkedin https://www.linkedin.com/in/holdenkarau
- Github https://github.com/holdenk
- Spark Videos http://bit.ly/holdenSparkVideos

What is the Spark Technology Center?



- An IBM technology center focused around Spark
- We work on open source Apache Spark to make it more awesome
  - Python, SQL, ML, and more! :)
- Related components as well:
  - Apache Toree [Incubating] (Notebook solution for Spark with Jupyter)
  - spark-testing-base (testing utilites on top of Spark)
  - Apache Bahir
  - Apache System ML Incubating - Machine Learning
- Partner with the Scala Foundation and other important players
- Multiple Spark Committers (Nick Pentreath, Xiao (Sean) Li, Prashant Sharma, Holden Karau (me!))
- Lots of contributions in Spark 2.0 & beyond :)

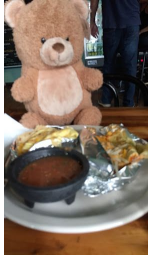
Top 10 Contributing Companies to PySpark 2.0.0



Who I think you wonderful humans are?

- Friendly people (this is a Python focused talk after all)
- Don’t mind pictures of cats or stuffed animals
- Know some Python
- Know some Spark
- Want to debug your Spark applications
- Ok with things getting a little bit silly

What will be covered?



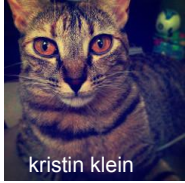
- A quick overview of PySpark architecture to understand how it can impact our debugging
- Getting at Spark's logs & persisting them
- What your options for logging are
- Attempting to understand Spark error messages
- My some what subtle attempts to get you to use spark-testing-base or similar
- My even less subtle attempts to get you to buy my new book
- Pictures of cats & stuffed animals

Aka: Building our Monster Identification Guide

First: a detour into PySpark's internals

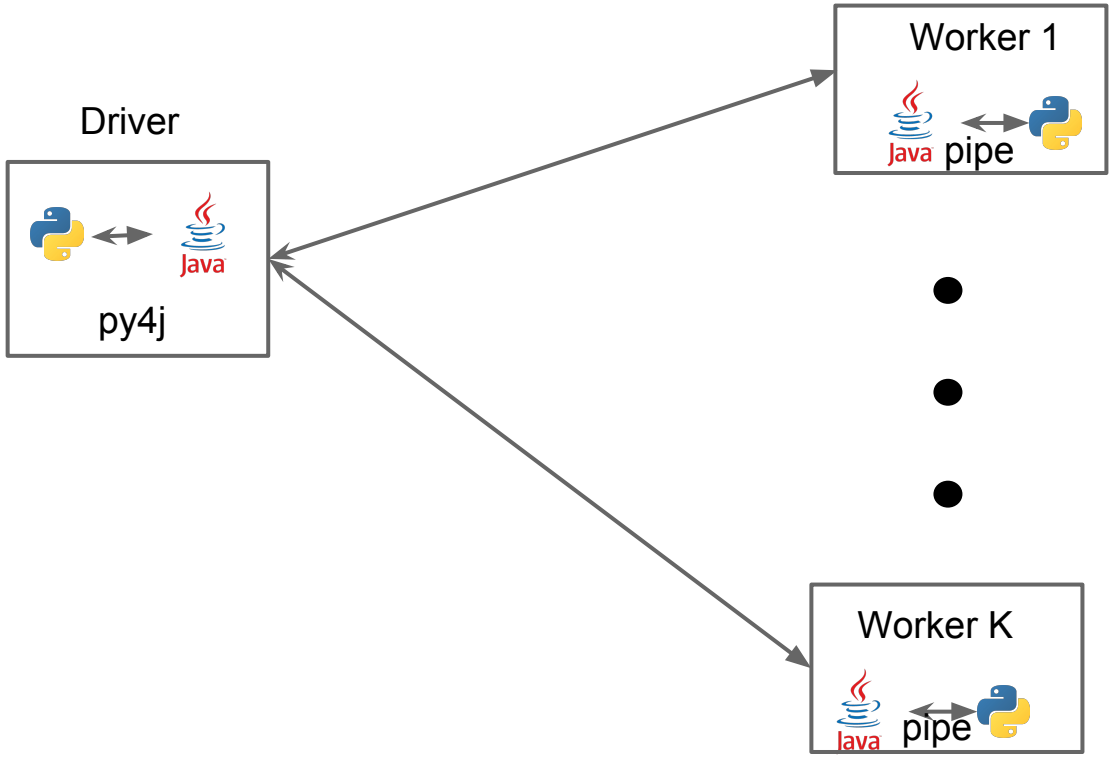
Photo by Bill Ward

Spark in Scala, how does PySpark work?



- Py4J + pickling + magic
  - This can be kind of slow sometimes
- RDDs are generally RDDs of pickled objects
- Spark SQL (and DataFrames) avoid some of this

So what does that look like?



So how does that impact PySpark?

- Data from Spark worker serialized and piped to Python worker
  - Multiple iterator-to-iterator transformations are still pipelined :)
- Double serialization cost makes everything more expensive
- Python worker startup takes a bit of extra time
- Python memory isn't controlled by the JVM - easy to go over container limits if deploying on YARN or similar
- Error messages make ~0 sense
- etc.

So where are the logs/errors?

(e.g. before we can identify a monster we have to find it)

- Error messages reported to the console*
- Log messages reported to the console*
- Log messages on the workers - access through the Spark Web UI or Spark History Server :)

(*When running in client mode)

Working in Jupyter?

PythonClassifierAdventure - Google Chrome

localhost:8888/notebooks/PythonClassifierAdventure.ipynb

jupyter PythonClassifierAdventure Last Checkpoint: 16 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Help

Create a Python2Vec model using PySpark source code as the input, we do this using the sc + wholeTextFiles so we can get access on a per-file basis

In [22]:
```
from pyspark.find_spark_home import _find_spark_home
spark_home = _find_spark_home()
```

In [30]:
```
spark_files_rdd = sc.wholeTextFiles(spark_home)
pyspark_files_rdd = spark_files_rdd.filter(lambda x: x[0].endsWith(".py"))
```

In [31]:
```
pyspark_files_rdd.take(1)
```

```
---------------------------------------------------------------------------
Py4JJavaError                             Traceback (most recent call last)
<ipython-input-31-fdc848153a99> in <module>()
----> 1 pyspark_files_rdd.take(1)

/home/holden/spark-venv/local/lib/python2.7/site-packages/pyspark/rdd.pyc in take(self, num)
   1341
   1342             p = range(partsScanned, min(partsScanned + numPartsToTry, totalParts))
-> 1343             res = self.context.runJob(self, takeUpToNumLeft, p)
   1344
   1345             items += res
```

"The error messages were so useless - I looked up how to disabled error reporting in Jupyter"

(paraphrased from PyData DC)

SPARK SUMMIT EAST 2017

Working in Jupyter - try your terminal for help


```
17/01/03 14:03:48 WARN TaskSetManager: Lost task 0.0 in stage 3.0 (TID 3, localhost, executor driver): org.apache.spark.api.python.PythonException: Traceback (most recen
t call last):
  File "/home/holden/spark-venv/lib/python2.7/site-packages/pyspark/python/lib/pyspark.zip/pyspark/worker.py", line 174, in main
    process()
  File "/home/holden/spark-venv/lib/python2.7/site-packages/pyspark/python/lib/pyspark.zip/pyspark/worker.py", line 169, in process
    serializer.dump_stream(func(split_index, iterator), outfile)
  File "/home/holden/spark-venv/lib/python2.7/site-packages/pyspark/python/lib/pyspark.zip/pyspark/serializers.py", line 268, in dump_stream
    vs = list(itertools.islice(iterator, batch))
  File "/home/holden/spark-venv/local/lib/python2.7/site-packages/pyspark/rdd.py", line 1339, in takeUpToNumLeft
    yield next(iterator)
  File "<ipython-input-30-52b70b27ce31>", line 2, in <lambda>
AttributeError: 'unicode' object has no attribute 'endsWith'
```

Possibly fix by https://issues.apache.org/jira/browse/SPARK-19094 but may not get in

Working in YARN?

(e.g. before we can identify a monster we have to find it)

- Use yarn logs to get logs after log collection
- Or set up the Spark history server
- Or yarn.nodemanager.delete.debug-delay-sec :)

Spark is pretty verbose by default

- Most of the time it tells you things you already know
- Or don’t need to know
- You can dynamically control the log level with sc.setLogLevel
- This is especially useful to increase logging near the point of error in your code

But what about when we get an error?

- Python Spark errors come in two-ish-parts often
- JVM Stack Trace (Friend Monster - comes most errors)
- Python Stack Trace (Boo - has information)
- Buddy - Often used to report the information from Friend Monster and Boo

So what is that JVM stack trace?

- Doesn’t want your error messages to get lonely
- Often not very informative
  - Except if the error happens purely in the JVM - like asking Spark to load a file which doesn’t exist

Let’s make some mistakes & debug :)

Image by: Tomomi

- Error in transformation
- Run out of memory in the workers

Bad outer transformation:


```
data = sc.parallelize(range(10))
transform1 = data.map(lambda x: x + 1)
transform2 = transform1.map(lambda x: x / 0)
transform2.count()
```

Let’s look at the error messages for it:

```
[Stage 0:>                                                  (0 + 0) / 4]17/02/01 09:52:07 ERROR Executor: Exception in task 0.0 in stage 0.0 (TID 0)
org.apache.spark.api.python.PythonException: Traceback (most recent call last):
  File "/home/holden/repos/spark/python/lib/pyspark.zip/pyspark/worker.py", line 180, in main
    process()
  File "/home/holden/repos/spark/python/lib/pyspark.zip/pyspark/worker.py", line 175, in process
    serializer.dump_stream(func(split_index, iterator), outfile)
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 2406, in pipeline_func
    return func(split, prev_func(split, iterator))
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 2406, in pipeline_func
    return func(split, prev_func(split, iterator))
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 2406, in pipeline_func
    return func(split, prev_func(split, iterator))
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 345, in func
    return f(iterator)
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 1040, in <lambda>
    return self.mapPartitions(lambda i: [sum(1 for _ in i)]).sum()
```

Continued for ~400 lines

```
  File "high_performance_pyspark/bad_pyspark.py", line 32, in <lambda>
```

SPARK SUMMIT EAST 2017

Ok maybe the web UI is easier?

PySparkShell - Deta

localhost:4040/jobs/job/?id=0

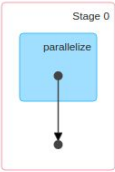
Apps | Bookmarks | Getting Started | Imported From F | Call and Respons | Conferences in S | stuff to watch | dockermagics | Valerie Halla | Validation Comic | Other bookmarks

Spark 2.2.0-SNAPSHOT | Jobs | Stages | Storage | Environment | Executors | SQL | PySparkShell application UI

Details for Job 0

Status: FAILED
Failed Stages: 1

- Event Timeline
- DAG Visualization



Failed Stages (1)

Stage Id ▼	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write	Failure Reason
0	count at high_performance_pyspark/bad_pyspark.py:33 +details	2017/02/01 09:52:07	0.5 s	0/4 (8 failed)					Job aborted due to stage failure: Task 2 in stage 0.0 failed 2 times, most recent failure: Lost task 2.1 in stage 0.0 (TID 6, localhost, executor driver): org.apache.spark.api.python.PythonException: Traceback (most recent call last): +details

```
Job aborted due to stage failure: Task 2 in stage 0.0 failed 2 times, most recent failure: Lost task 2.1 in stage 0.0 (TID 6, localhost, executor driver):
org.apache.spark.api.python.PythonException: Traceback (most recent call last):
  File "/home/holden/repos/spark/python/lib/pyspark.zip/pyspark/worker.py", line 180, in main
    process()
```

SPARK SUMMIT EAST 2017

And click through...

PySparkShell - Deta

localhost:4040/stages/stage/?id=0&attempt=0

Apps | Bookmarks | Getting Started | Imported From F | Call and Respons | Conferences in S | stuff to watch | dockermagics | Valerie Halla | Validation Comic | Other bookmarks

parallelize at PythonRDD.scala:480

PythonRDD [1]
count at high_performance_pyspark/bad_pyspark.py:33

Show Additional Metrics

Event Timeline

Summary Metrics for 0 Completed Tasks

No tasks have reported metrics yet.

Aggregated Metrics by Executor

Executor ID ▲	Address	Task Time	Total Tasks	Failed Tasks	Killed Tasks	Succeeded Tasks	Blacklisted
driver	172.17.42.1:37813	2 s	8	8	0	0	0

Tasks (8)

Index ▲	ID	Attempt	Status	Locality Level	Executor ID / Host	Launch Time	Duration	GC Time	Errors
0	0	0	FAILED	PROCESS_LOCAL	driver / localhost	2017/02/01 09:52:07	0.3 s		org.apache.spark.api.python.PythonException: Traceback (most recent call last): +details
0	7	1	FAILED	PROCESS_LOCAL	driver / localhost	2017/02/01 09:52:07	57 ms		org.apache.spark.api.python.PythonException: Traceback (most recent call last): +details

```
org.apache.spark.api.python.PythonException: Traceback (most recent call last):
  File "/home/holden/repos/spark/python/lib/pyspark.zip/pyspark/worker.py", line 180, in main
    process()
  File "/home/holden/repos/spark/python/lib/pyspark.zip/pyspark/worker.py", line 175, in
process
    serializer.dump_stream(func(split_index, iterator), outfile)
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 2406, in pipeline_func
    return func(split, prev_func(split, iterator))
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 2406, in pipeline_func
    return func(split, prev_func(split, iterator))
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 2406, in pipeline_func
    return func(split, prev_func(split, iterator))
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 345, in func
    return f(iterator)
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 1040, in <lambda>
    return self.mapPartitions(lambda i: [sum(1 for _ in i)]).sum()
```

SPARK SUMMIT EAST 2017

A scroll down (not quite to the bottom)

PySparkShell - Deta

localhost:4040/stages/stage/?id=0&attempt=0

Apps | Bookmarks | Getting Started | Imported From F | Call and Respons | Conferences in S | stuf

parallelize at PythonRDD.scala:480

PythonRDD [1]
count at high_performance_pyspark/bad_pyspark.py:33

Show Additional Metrics

Event Timeline

Summary Metrics for 0 Completed Tasks

No tasks have reported metrics yet.

Aggregated Metrics by Executor

Executor ID ▲	Address	Task Time	Total Tasks	Failed Tasks	Killed Tasks	Succeeded Tasks	Blacklisted
driver	172.17.42.1:37813	2 s	8	8	0	0	0

Tasks (8)

Index ▲	ID	Attempt	Status	Locality Level	Executor ID / Host	Launch Time	Duration	GC Time	Errors
0	0	0	FAILED	PROCESS_LOCAL	driver / localhost	2017/02/01 09:52:07	0.3 s		org.apache.spark.api.python.PythonException: Traceback (most recent call last): +details
0	7	1	FAILED	PROCESS_LOCAL	driver / localhost	2017/02/01 09:52:07	57 ms		org.apache.spark.api.python.PythonException: Traceback (most recent call last): +details

```
  File "high_performance_pyspark/bad_pyspark.py", line 32, in <lambda>
    transform2 = transform1.map(lambda x: x / 0)
ZeroDivisionError: integer division or modulo by zero

        at org.apache.spark.api.python.PythonRunner$$anon$1.read(PythonRDD.scala:193)
        at org.apache.spark.api.python.PythonRunner$$anon$1.<init>(PythonRDD.scala:234)
        at org.apache.spark.api.python.PythonRunner.compute(PythonRDD.scala:152)
        at org.apache.spark.api.python.PythonRDD.compute(PythonRDD.scala:63)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:323)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:287)
        at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:87)
        at org.apache.spark.scheduler.Task.run(Task.scala:113)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:313)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
        at java.lang.Thread.run(Thread.java:745)
```


```
  File "high_performance_pyspark/bad_pyspark.py", line 32, in <lambda>
    transform2 = transform1.map(lambda x: x / 0)
ZeroDivisionError: integer division or modulo by zero
```

Or look at the bottom of console logs:

```
 File "/home/holden/repos/spark/python/lib/pyspark.zip/pyspark/worker.py", line
180, in main
    process()
  File "/home/holden/repos/spark/python/lib/pyspark.zip/pyspark/worker.py", line
175, in process
    serializer.dump_stream(func(split_index, iterator), outfile)
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 2406, in
pipeline_func
    return func(split, prev_func(split, iterator))
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 2406, in
pipeline_func
    return func(split, prev_func(split, iterator))
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 2406, in
pipeline_func
    return func(split, prev_func(split, iterator))
```

Or look at the bottom of console logs:

```
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 345, in func
    return f(iterator)
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 1040, in <lambda>
    return self.mapPartitions(lambda i: [sum(1 for _ in i)]).sum()
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 1040, in <genexpr>
    return self.mapPartitions(lambda i: [sum(1 for _ in i)]).sum()
  File "high_performance_pyspark/bad_pyspark.py", line 32, in <lambda>
    transform2 = transform1.map(lambda x: x / 0)
ZeroDivisionError: integer division or modulo by zero
```

Python Pipelines

- Some pipelining happens inside of Python
  - For performance (less copies from Python to Scala)
- DAG visualization is generated inside of Scala
  - Misses Python pipelines :(

Regardless of language

- Can be difficult to determine which element failed
- Stack trace _sometimes_ helps (it did this time)
- take(1) + count() are your friends - but a lot of work :(

Side note: Lambdas aren't always your friend

- Lambda's can make finding the error more challenging
- I love lambda x, y: x / y as much as the next human but when y is zero :(
- A small bit of refactoring for your debugging never hurt anyone*
- If your inner functions are causing errors it's a good time to have tests for them!
- Difficult to put logs inside of them

*A blatant lie, but…. it hurts less often than it helps

Testing - you should do it!

- spark-testing-base is on pip now for your happy test adventures
- That’s a talk unto itself though (but it's on YouTube)

Adding your own logging:

- Java users use Log4J & friends
- Python users: use logging library (or even print!)
- Accumulators
  - Behave a bit weirdly, don’t put large amounts of data in them

Also not all errors are “hard” errors

- Parsing input? Going to reject some malformed records
- flatMap or filter + map can make this simpler
- Still want to track number of rejected records (see accumulators)

So using names & logging & accs could be:

```
data = sc.parallelize(range(10))
 rejectedCount = sc.accumulator(0)
 def loggedDivZero(x):
     import logging
     try:
         return [x / 0]
     except Exception as e:
         rejectedCount.add(1)
         logging.warning("Error found " + repr(e))
         return []
 transform1 = data.flatMap(loggedDivZero)
 transform2 = transform1.map(add1)
 transform2.count()
 print("Reject " + str(rejectedCount.value))
```

Spark accumulators

- Really “great” way for keeping track of failed records
- Double counting makes things really tricky
  - Jobs which worked “fine” don’t continue to work “fine” when minor changes happen
- Relative rules can save us* under certain conditions

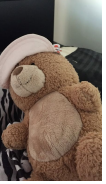
Could we just us -mtrace?

- Spark makes certain assumptions about how Python is launched on the workers this doesn’t (currently) work
- Namely it assumes PYSPARK_PYTHON points to a file
- Also assumes arg[0] has certain meanings :(

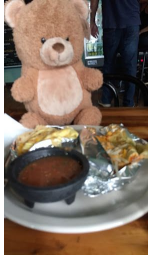
Ok what about if we run out of memory?

In the middle of some Java stack traces:

```
 File "/home/holden/repos/spark/python/lib/pyspark.zip/pyspark/worker.py", line 180, in main
    process()
  File "/home/holden/repos/spark/python/lib/pyspark.zip/pyspark/worker.py", line 175, in process
    serializer.dump_stream(func(split_index, iterator), outfile)
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 2406, in pipeline_func
    return func(split, prev_func(split, iterator))
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 2406, in pipeline_func
    return func(split, prev_func(split, iterator))
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 2406, in pipeline_func
    return func(split, prev_func(split, iterator))
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 345, in func
    return f(iterator)
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 1040, in <lambda>
    return self.mapPartitions(lambda i: [sum(1 for _ in i)]).sum()
  File "/home/holden/repos/spark/python/pyspark/rdd.py", line 1040, in <genexpr>
    return self.mapPartitions(lambda i: [sum(1 for _ in i)]).sum()
  File "high_performance_pyspark/bad_pyspark.py", line 132, in generate_too_much
    return range(10000000000000)
MemoryError
```



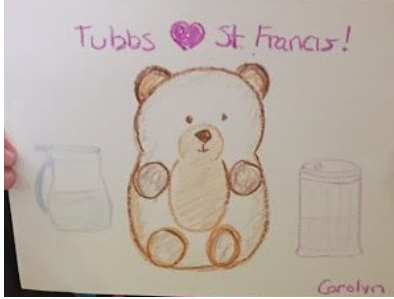
Tubbs doesn’t always look the same



- Out of memory can be pure JVM (worker)
  - OOM exception during join
  - GC timelimit exceeded
- OutOfMemory error, Executors being killed by kernel, etc.
- Running in YARN? “Application overhead exceeded”
- JVM out of memory on the driver side from Py4J

Reasons for JVM worker OOMs

(w/PySpark)



- Unbalanced shuffles
- Buffering of Rows with PySpark + UDFs
  - If you have a down stream select move it up stream
- Individual jumbo records (after pickling)

Reasons for Python worker OOMs

(w/PySpark)

- Insufficient memory reserved for Python worker
- Jumbo records
- Eager entire partition evaluation (e.g. sort + mapPartitions)
- Too large partitions (unbalanced or not enough partitions)
- Native code memory leak

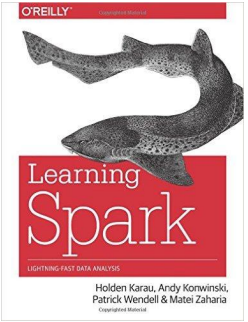
And loading invalid paths:

```
org.apache.hadoop.mapred.InvalidInputException: Input path does not exist: file:/doesnotexist
    at org.apache.hadoop.mapred.FileInputFormat.listStatus(FileInputFormat.java:251)
    at org.apache.hadoop.mapred.FileInputFormat.getSplits(FileInputFormat.java:270)
    at org.apache.spark.rdd.HadoopRDD.getPartitions(HadoopRDD.scala:202)
    at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:252)
    at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:250)
    at scala.Option.getOrElse(Option.scala:121)
    at org.apache.spark.rdd.RDD.partitions(RDD.scala:250)
    at org.apache.spark.rdd.MapPartitionsRDD.getPartitions(MapPartitionsRDD.scala:35)
    at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:252)
    at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:250)
    at scala.Option.getOrElse(Option.scala:121)
    at org.apache.spark.rdd.RDD.partitions(RDD.scala:250)
```

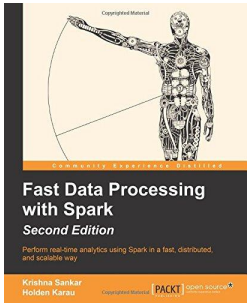
Oooh Boo found food! Let’s finish quickly :)

What about if that isn’t enough to debug?

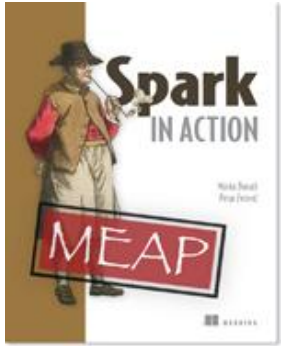
- Move take(1) up the dependency chain
- DAG in the WebUI -- less useful for Python :(
- toDebugString -- also less useful in Python :(
- Sample data and run locally



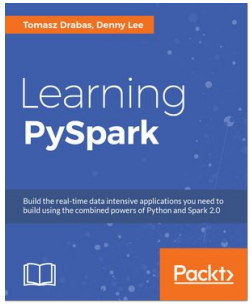
Learning Spark



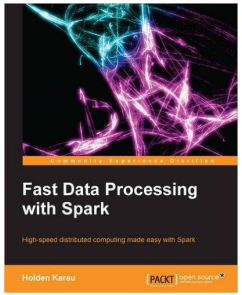
Fast Data Processing with Spark (2nd edition)



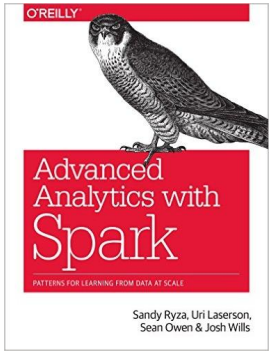
Coming soon: Spark in Action



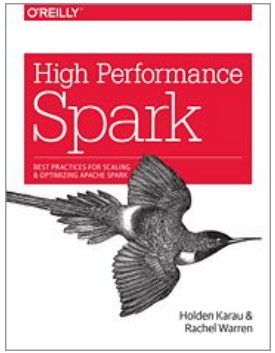
Coming Soon: Learning PySpark



Fast Data Processing with Spark (Out of Date)



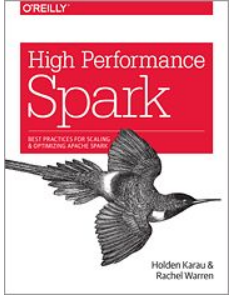
Advanced Analytics with Spark



Coming soon: High Performance Spark

SPARK SUMMIT EAST 2017

High Performance Spark (soon!)



First seven chapters are available in “Early Release”*:

- Buy from O’Reilly - http://bit.ly/highPerfSpark
- Python is in Chapter 7 & Debugging in Appendix

Get notified when updated & finished:

- http://www.highperformancespark.com
- https://twitter.com/highperfspark

* Early Release means extra mistakes, but also a chance to help us make a more awesome book.

K thnx bye!

Get in touch if you want:

@holdenkarau on twitter

Have some simple UDFs you wish ran faster?: http://bit.ly/pySparkUDF

If you care about Spark testing: http://bit.ly/holdenTestingSpark

Want to start contributing to PySpark? Talk to me IRL or

E-mail: holden.karau+contributing@gmail.com

SPARK SUMMIT EAST 2017