

A vertical photograph of the Golden Gate Bridge, showing its iconic orange-red towers and suspension cables against a hazy sky and water.

Spark Application Development Made Fast and Easy

Shivnath Babu

Lance Co Ting Keh



Lance Co Ting Keh

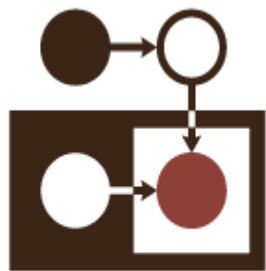
Machine Learning @ Box
Distributed ML Infrastructure
Go Blue Devils!



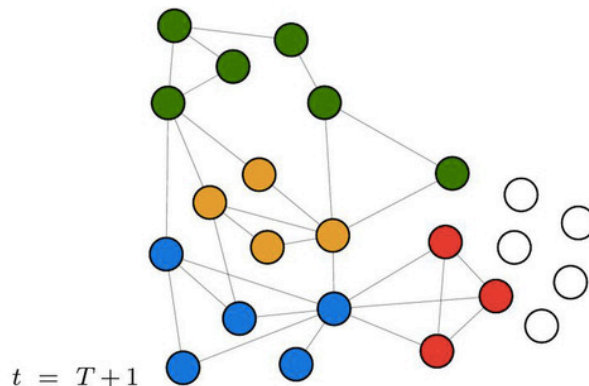
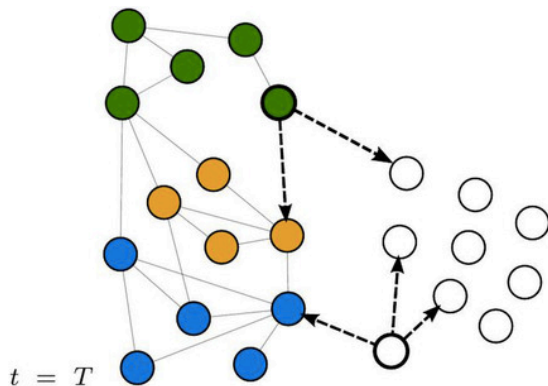
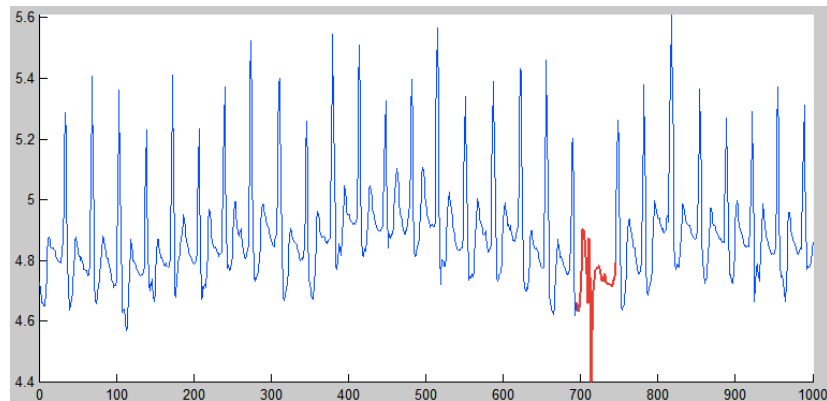
Shivnath Babu

Associate Professor @ Duke
Chief Scientist at Unravel Data Sys.
R&D in Management of Data Systems

Spark @Box



topic
modeling



What's so great about Spark?

We believe that Spark is the first system that allows a **general-purpose programming language** to be used at interactive speeds for in-memory data mining on clusters.

From: **Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing**

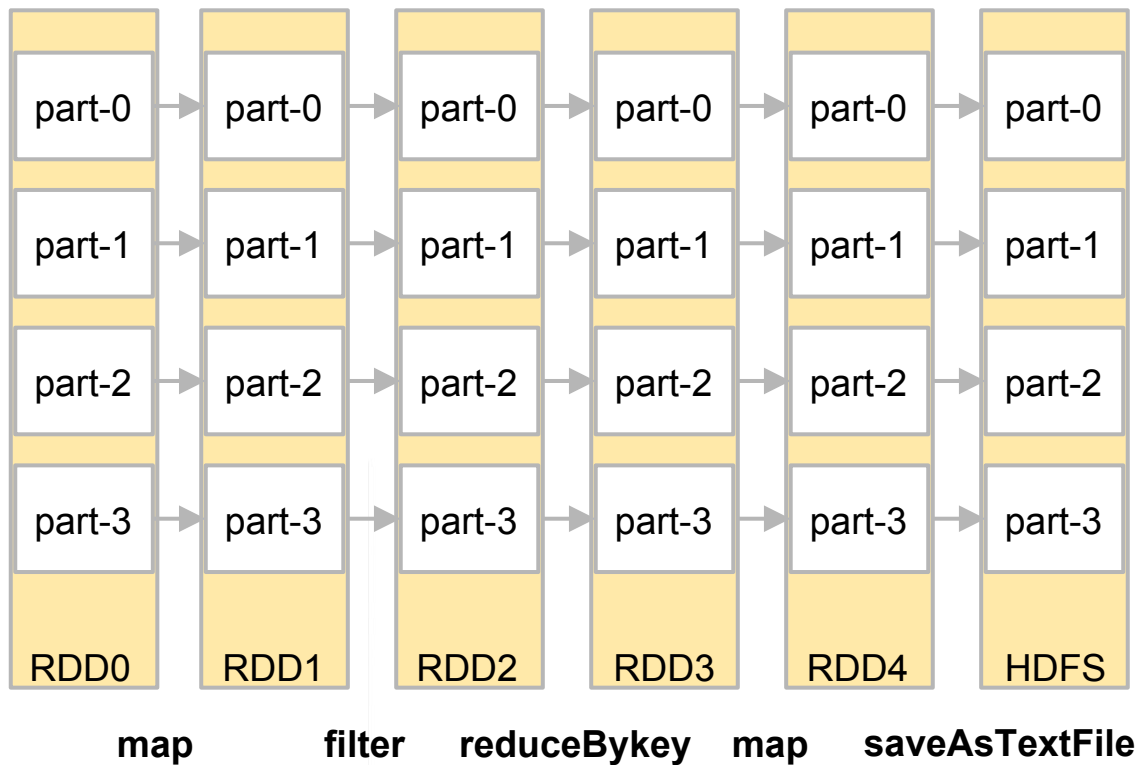
Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma,
Murphy McCauley, Michael J. Franklin, Scott Shenker, Ion Stoica
University of California, Berkeley





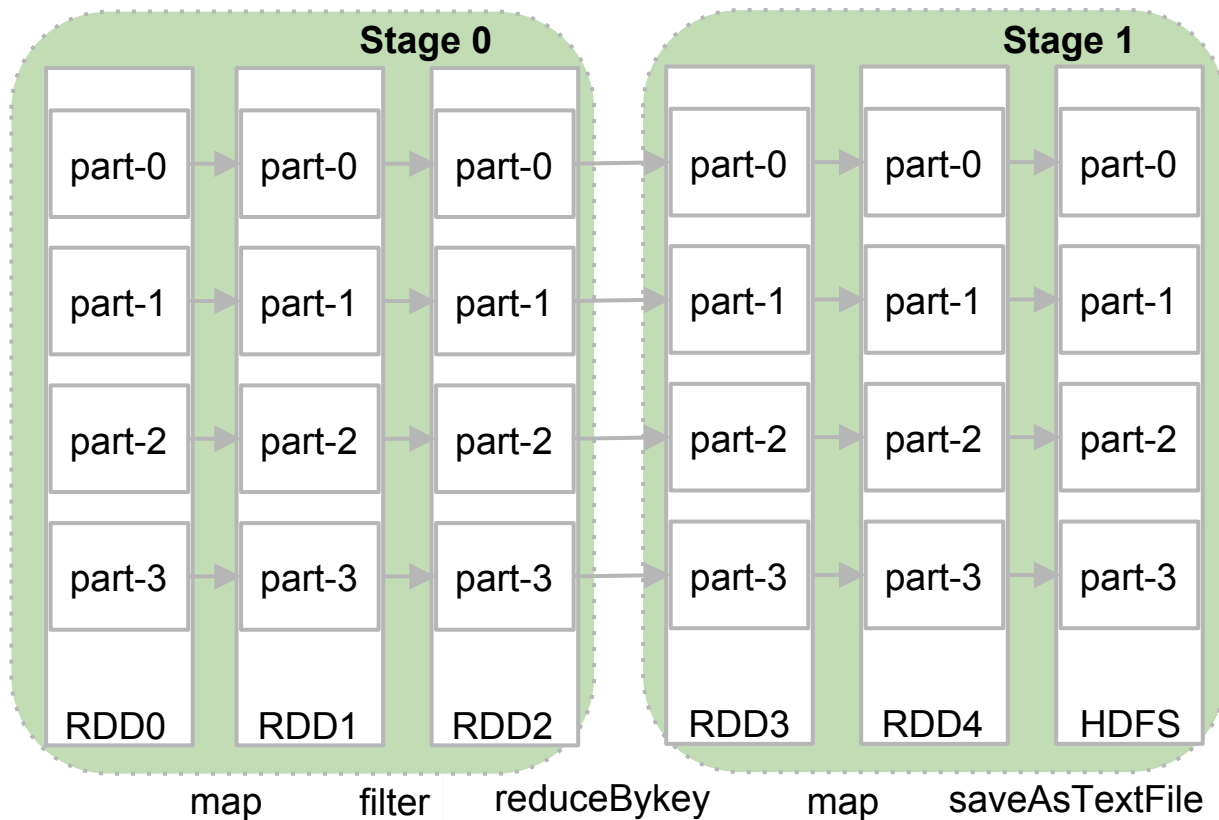
Spark Execution

```
sc.textFile(hdfsPath)
  .map(parseInput)
  .filter(subThreshold)
  .reduceByKey(tallyCount)
  .map(formatOutput)
  .saveAsTextFile(outPath)
```



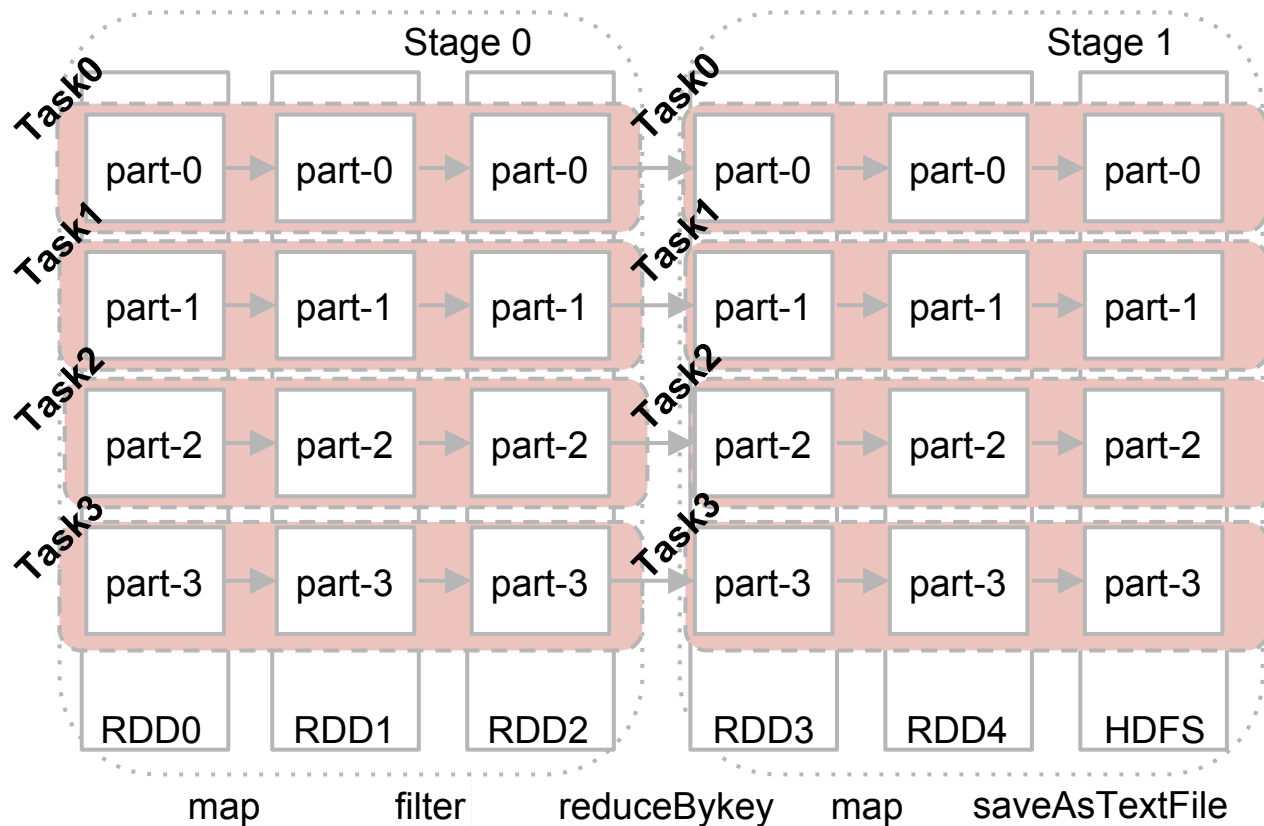
Spark Execution

```
sc.textFile(hdfsPath)
  .map(parseInput)
  .filter(subThreshold)
  .reduceByKey(tallyCount)
  .map(formatOutput)
  .saveAsTextFile(outPath)
```



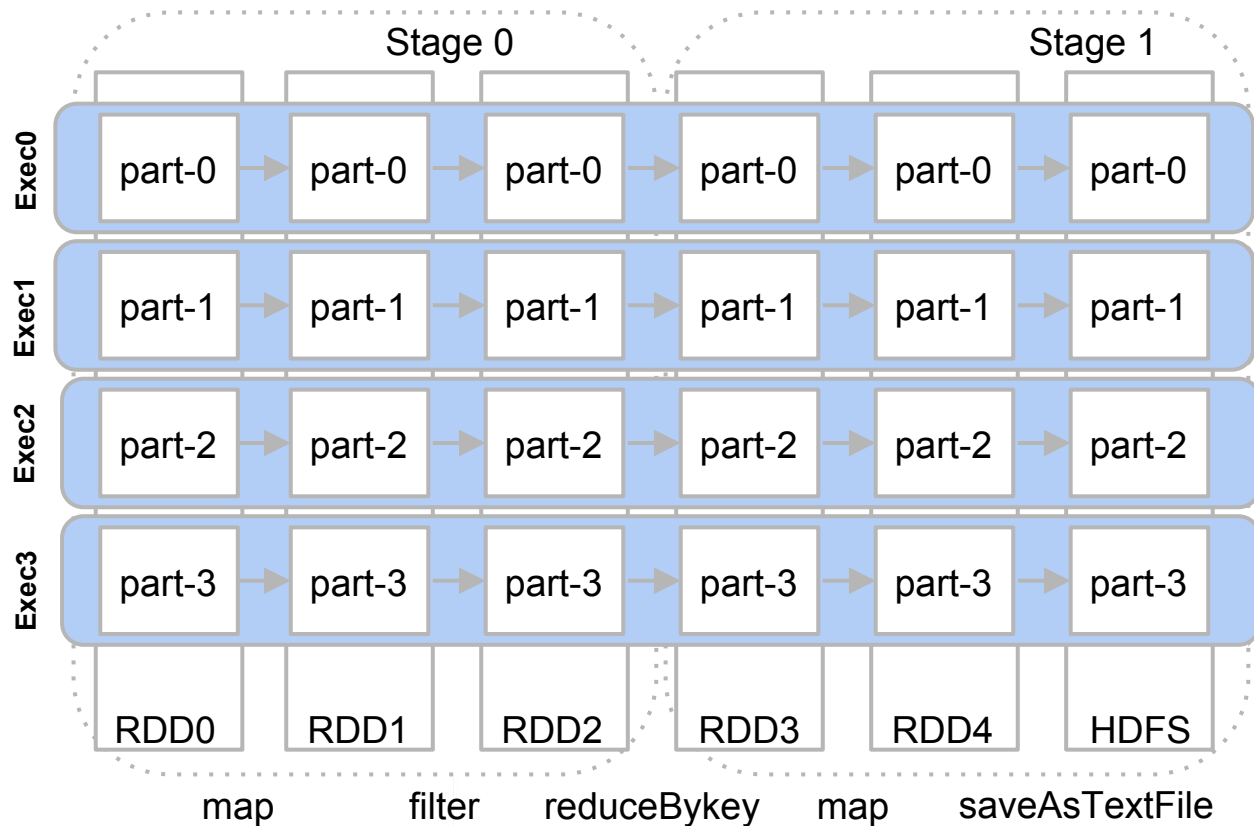
Spark Execution

```
sc.textFile(hdfsPath)
  .map(parseInput)
  .filter(subThreshold)
  .reduceByKey(tallyCount)
  .map(formatOutput)
  .saveAsTextFile(outPath)
```



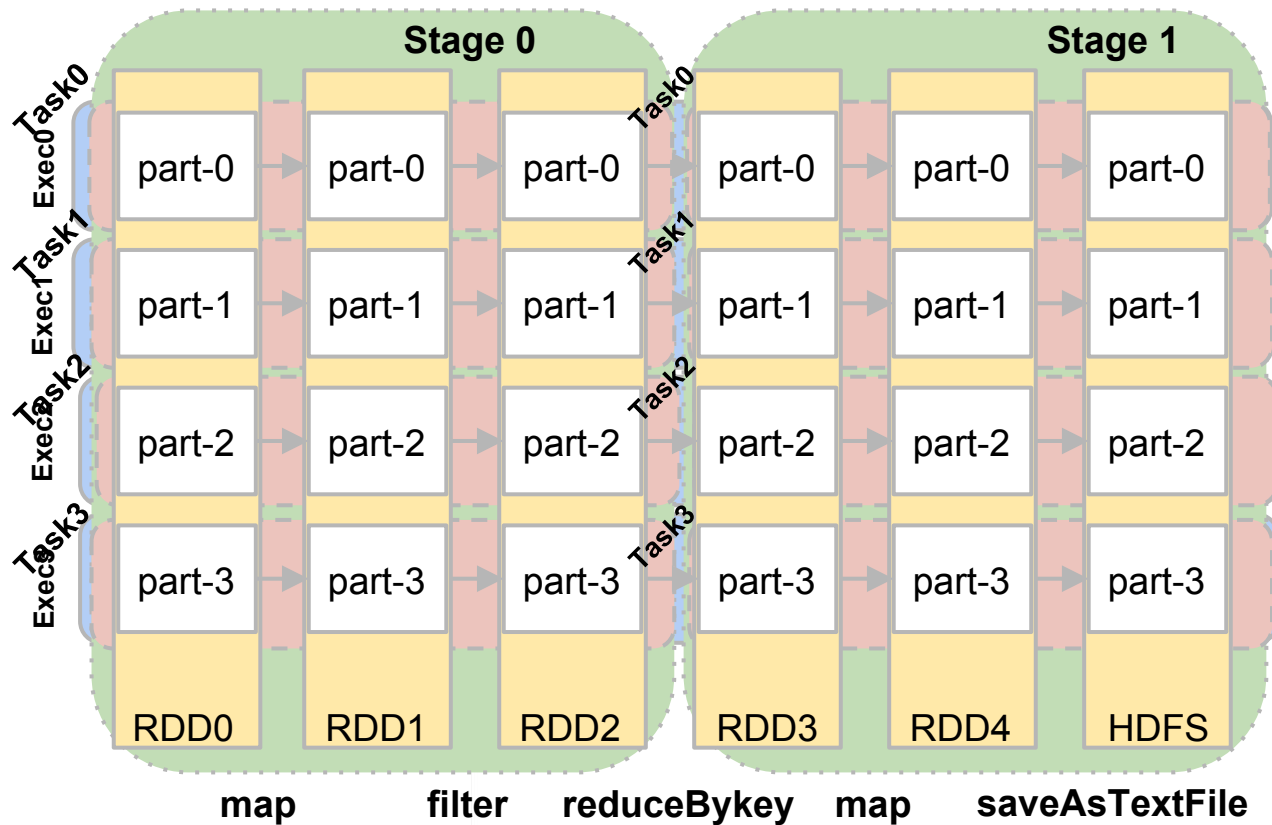
Spark Execution

```
sc.textFile(hdfsPath)
  .map(parseInput)
  .filter(subThreshold)
  .reduceByKey(tallyCount)
  .map(formatOutput)
  .saveAsTextFile(outPath)
```



Spark Execution

```
sc.textFile(hdfsPath)
  .map(parseInput)
  .filter(subThreshold)
  .reduceByKey(tallyCount)
  .map(formatOutput)
  .saveAsTextFile(outPath)
```



Spark Execution

VERY COMPLEX!

```
sc.textFile(hdfsPath)
  .map(parseInput)
  .filter(subThreshold)
  .reduceByKey(tallyCount)
  .map(formatOutput)
  .saveAsTextFile(outPath)
```

The diagram illustrates the Spark execution process across two stages, Stage 0 and Stage 1. Stage 0 consists of four tasks (Task0, Task1, Task2, Task3) each executing a 'map' operation on RDD0, RDD1, RDD2, and RDD3 respectively. Stage 1 consists of four tasks (Task0, Task1, Task2, Task3) each executing a 'filter' operation on the output of Stage 0. The final output is a 'reduceByKey' operation on RDD4, which is then saved to HDFS. The diagram is overlaid with a large, diagonal watermark reading 'VERY COMPLEX!'.

Spark
summit 2015

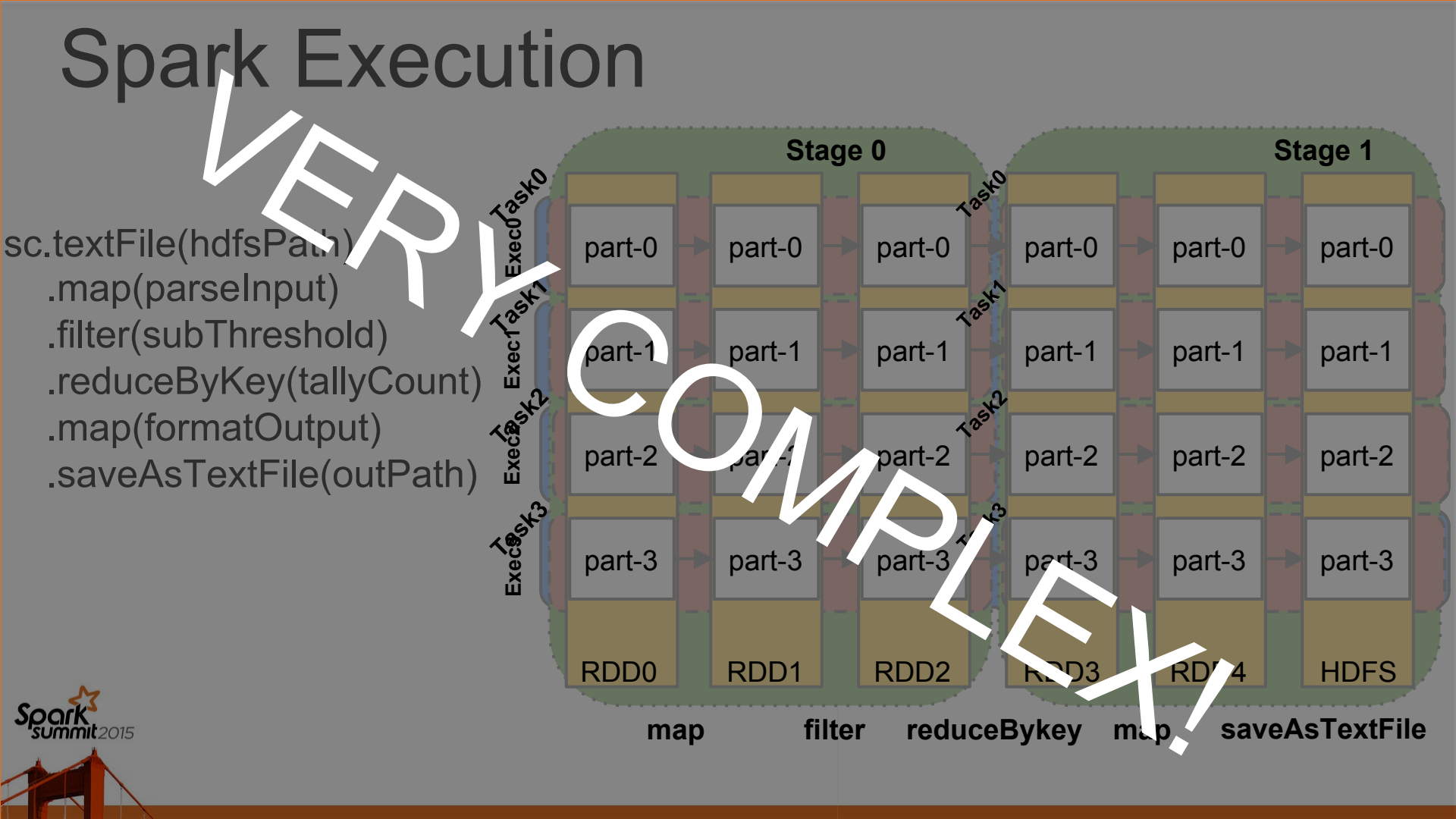
Spark Execution

VERY COMPLEX!

```
sc.textFile(hdfsPath)
  .map(parseInput)
  .filter(subThreshold)
  .reduceByKey(tallyCount)
  .map(formatOutput)
  .saveAsTextFile(outPath)
```

The diagram illustrates the Spark execution process across two stages, Stage 0 and Stage 1. Stage 0 consists of four tasks (Task0, Task1, Task2, Task3) each executing a 'map' operation on RDD0, RDD1, RDD2, and RDD3 respectively. Stage 1 consists of four tasks (Task0, Task1, Task2, Task3) each executing a 'filter' operation on the output of Stage 0. The final output is saved as a text file using 'saveAsTextFile'. The diagram also shows the 'reduceByKey' operation as a separate stage. The text 'VERY COMPLEX!' is overlaid on the diagram.

Spark
summit 2015



Anything that can go wrong,
will go wrong (at some point)



What can go wrong?

Failures

- My query failed after 6 hours!
- What does this exception mean?

```
15/06/16 00:33:34 INFO TaskSchedulerImpl: Cancelling stage 0
15/06/16 00:33:34 INFO TaskSchedulerImpl: Stage 0 was cancelled
15/06/16 00:33:34 INFO DAGScheduler: Stage 0 (map at NaiveLabelProp.scala:59) failed in 7.070 s
15/06/16 00:33:34 INFO DAGScheduler: Job 0 failed: saveAsTextFile at NaiveLabelProp.scala:69, took 7.233958 s
Exception in thread "main" org.apache.spark.SparkException: Job aborted due to stage failure: Task 2 in stage 0.0 failed 4 times, most
recent failure: Lost task 2.3 in stage 0.0 (TID 17, 10.199.10.16): org.apache.spark.SparkException: RDD transformations and actions can
only be invoked by the driver, not inside of other transformations; for example, rdd1.map(x => rdd2.values.count() * x) is invalid bec
ause the values transformation and count action cannot be performed inside of the rdd1.map transformation. For more information, see SP
ARK-5063.
    at org.apache.spark.rdd.RDD.sc(RDD.scala:87)
    at org.apache.spark.rdd.RDD.count(RDD.scala:1006)
    at com.demo.main.NaiveLabelProp$$anonfun$4.apply(NaiveLabelProp.scala:59)
    at com.demo.main.NaiveLabelProp$$anonfun$4.apply(NaiveLabelProp.scala:59)
    at scala.collection.Iterator$$anon$11.next(Iterator.scala:328)
    at com.unraveldata.spark.UnravelSpecificIterator.next(UnravelSpecificIterator.scala:46)
    at org.apache.spark.util.collection.ExternalSorter.insertAll(ExternalSorter.scala:204)
    at org.apache.spark.shuffle.sort.SortShuffleWriter.write(SortShuffleWriter.scala:56)
    at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:68)
    at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:41)
    at org.apache.spark.scheduler.Task.run(Task.scala:64)
    at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:203)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
    at java.lang.Thread.run(Thread.java:745)

Driver stacktrace:
    at org.apache.spark.scheduler.DAGScheduler.org$apache$spark$scheduler$DAGScheduler$$failJobAndIndependentStages(DAGScheduler.sc
ala:1204)
    at org.apache.spark.scheduler.DAGScheduler$$anonfun$abortStage$1.apply(DAGScheduler.scala:1193)
    at org.apache.spark.scheduler.DAGScheduler$$anonfun$abortStage$1.apply(DAGScheduler.scala:1192)
    at scala.collection.mutable.ResizableArray$class.foreach(ResizableArray.scala:59)
    at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala:47)
    at org.apache.spark.scheduler.DAGScheduler.abortStage(DAGScheduler.scala:1192)
    at org.apache.spark.scheduler.DAGScheduler$$anonfun$handleTaskSetFailed$1.apply(DAGScheduler.scala:693)
    at org.apache.spark.scheduler.DAGScheduler$$anonfun$handleTaskSetFailed$1.apply(DAGScheduler.scala:693)
    at scala.Option.foreach(Option.scala:236)
    at org.apache.spark.scheduler.DAGScheduler.handleTaskSetFailed(DAGScheduler.scala:693)
    at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:1393)
```

What can go wrong?

- **Failures**

- My query failed after 6 hours!
- What does this exception mean?

- **Wrong results**

- Result of my job looks wrong

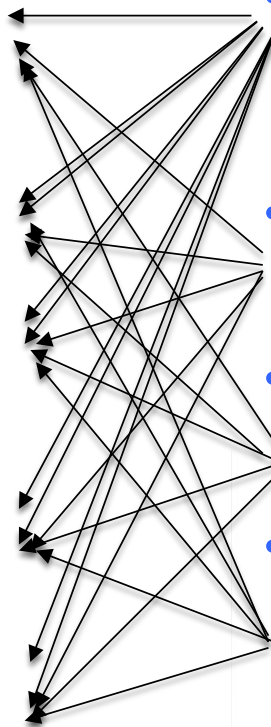
- **Bad performance**

- My app is very slow
- Pipeline is not meeting the 4hr SLA

- **Poor scalability**

- Oh, but it worked on the dev cluster!

- **Bad App(le)s**



And Why?

- **Application Problems**

- Poor choice of transformations
- Ineffective caching
- Bloated data structures

- **Data/Storage Problems**

- Skewed data, load imbalance
- Small files, poor data partitioning

- **Spark Problems**

- Shuffle
- Lazy evaluation causes confusion

- **Resource Problems**

- Resource contention
- Performance degradation

How do application developers
detect & fix these problems today?



Details for Stage 4

Total task time across all tasks: 1.1 min
Shuffle write: 152.3 MB

Summary Metrics for 144 Completed Tasks

| Metric | Min | 25th percentile | Median | 75th percentile | Max |
|----------------------------------|-----------|-----------------|-----------|-----------------|-----------|
| Result serialization time | 0 ms | 0 ms | 0 ms | 0 ms | 1 ms |
| Duration | 0.3 s | 0.4 s | 0.4 s | 0.4 s | 0.6 s |
| Time spent fetching task results | 0 ms | 0 ms | 0 ms | 0 ms | 0 ms |
| Scheduler delay | 7 ms | 9 ms | 11 ms | 15 ms | 0.3 s |
| Shuffle Write | 1081.0 KB | 1082.3 KB | 1082.7 KB | 1083.2 KB | 1084.7 KB |

Aggregated Metrics by Executor

| Executor ID | Address | Task Time | Total Tasks | Failed Tasks | Succeeded Tasks | Shuffle Read | Shuffle Write | Shuffle Spill (Memory) | Shuffle Spill (Disk) |
|-------------|-------------------------------|-----------|-------------|--------------|-----------------|--------------|---------------|------------------------|----------------------|
| 0 | dwh-lab4004.dev.box.net:33356 | 23 s | 48 | 0 | 48 | 0.0 B | 50.8 MB | 0.0 B | 0.0 B |
| 1 | dwh-lab4002.dev.box.net:58672 | 24 s | 51 | 0 | 51 | 0.0 B | 53.9 MB | 0.0 B | 0.0 B |
| 2 | dwh-lab4003.dev.box.net:44774 | 21 s | 45 | 0 | 45 | 0.0 B | 47.6 MB | 0.0 B | 0.0 B |

Tasks

| Task Index | Task ID | Status | Locality Level | Executor | Launch Time | Duration | GC Time | Result Ser Time | Write Time | Shuffle Write | Errors |
|------------|---------|---------|----------------|-------------------------|---------------------|----------|---------|-----------------|------------|---------------|--------|
| 0 | 432 | SUCCESS | PROCESS_LOCAL | dwh-lab4003.dev.box.net | 2014/07/17 10:15:33 | 0.5 s | | | 1 ms | 1082.4 KB | |
| 1 | 433 | SUCCESS | PROCESS_LOCAL | dwh-lab4002.dev.box.net | 2014/07/17 10:15:33 | 0.5 s | | | 1 ms | 1083.8 KB | |
| 2 | 434 | SUCCESS | PROCESS_LOCAL | dwh-lab4004.dev.box.net | 2014/07/17 10:15:33 | 0.6 s | | | 1 ms | 1082.6 KB | |
| 3 | 436 | SUCCESS | PROCESS_LOCAL | dwh-lab4002.dev.box.net | 2014/07/17 10:15:33 | 0.5 s | | | 1 ms | 1082.4 KB | |
| 4 | 435 | SUCCESS | PROCESS_LOCAL | dwh-lab4004.dev.box.net | 2014/07/17 10:15:33 | 0.5 s | | | 1 ms | 1082.7 KB | |
| 5 | 437 | SUCCESS | PROCESS_LOCAL | dwh-lab4004.dev.box.net | 2014/07/17 10:15:33 | 0.6 s | | 1 ms | 1 ms | 1082.2 KB | |
| 6 | 438 | SUCCESS | PROCESS_LOCAL | dwh-lab4002.dev.box.net | 2014/07/17 10:15:34 | 0.4 s | | | 1 ms | 1083.4 KB | |
| 7 | 439 | SUCCESS | PROCESS_LOCAL | dwh-lab4003.dev.box.net | 2014/07/17 10:15:34 | 0.4 s | | | 1 ms | 1082.5 KB | |
| 8 | 440 | SUCCESS | PROCESS_LOCAL | dwh-lab4002.dev.box.net | 2014/07/17 10:15:34 | 0.4 s | | | 1 ms | 1082.3 KB | |
| 9 | 441 | SUCCESS | PROCESS_LOCAL | dwh-lab4004.dev.box.net | 2014/07/17 10:15:34 | 0.4 s | | | 1 ms | 1081.4 KB | |
| 10 | 442 | SUCCESS | PROCESS_LOCAL | dwh-lab4004.dev.box.net | 2014/07/17 10:15:34 | 0.5 s | | 1 ms | 1 ms | 1083.5 KB | |

Spark Stages

Total Duration: 6.0 min
Scheduling Mode: FIFO
Active Stages: 1
Completed Stages: 4
Failed Stages: 0

Active Stages (1)

| Stage Id | Description | Submitted | Duration | Tasks: Succeeded/Total | Shuffle Read | Shuffle Write |
|----------|---|---------------------|----------|------------------------|--------------|---------------|
| 3 | (kill) collectAeMap at KMeansBench.scala:58 | 2014/07/17 10:15:45 | 5.5 min | 0/144 | | |

Completed Stages (4)

| Stage Id | Description | Submitted | Duration | Tasks: Succeeded/Total | Shuffle Read | Shuffle Write |
|----------|------------------------------------|---------------------|----------|------------------------|--------------|---------------|
| 4 | map at KMeansBench.scala:51 | 2014/07/17 10:15:33 | 12 s | 144/144 | | 152.3 MB |
| 2 | takeSample at KMeansBench.scala:48 | 2014/07/17 10:15:33 | 0.5 s | 144/144 | | |
| 1 | takeSample at KMeansBench.scala:48 | 2014/07/17 10:15:32 | 0.4 s | 144/144 | | |
| 0 | count at KMeansBench.scala:45 | 2014/07/17 10:15:25 | 7 s | 144/144 | | |

Failed Stages (0)

| Stage Id | Description | Submitted | Duration | Tasks: Succeeded/Total | Shuffle Read | Shuffle Write | Failure Reason |
|----------|-------------|-----------|----------|------------------------|--------------|---------------|----------------|
|----------|-------------|-----------|----------|------------------------|--------------|---------------|----------------|

cloudera manager Home Clusters Hosts Diagnostics Audits Charts Administration

Status All Health Issues All Configuration Issues All Recent Commands

March 17 2014, 1:13:30 PM PDT

Status

Cluster 1 (CDH 5.0.0, Packages)

- Hosts
- FLUME-1
- HBASE-1
- HDFS-1
- HIVE-1
- HUE-1
- MAPALA-1
- KS_INDEXER-1
- MAPREDUCE-1
- OOZIE-1
- SOULR-1
- SPARK-1
- SQOOP-1
- YARN-1
- ZOOKEEPER-1

Cloudera Management Service

- mgmt

Charts

Cluster 1 (CDH 5.0.0)

Cluster CPU

Cluster Disk IO

Cluster Network IO

HDFS IO

Running MapReduce Jobs

Completed Impala Queries

Per Pool Running Applications

ec2-107-20-75-123.compute-1.amazonaws.com/ganglia/m=load_one&h=hour&s=descending&c=datacluster&h=&sh=1&hc=4&z=small

Overview of datacluster

CPU: 7
Total: 7
Hosts up: 7
Hosts down: 0

Avg Load (15, 5, 1m): 58%, 66%, 56%

Localtime: 2011-09-03 00:09

datacluster Cluster Load last hour

datacluster Cluster CPU last hour

datacluster Cluster Memory last hour

datacluster Cluster Network last hour

datacluster load_one last hour sorted descending

ip-10-203-14-99.ec2.intern

ip-10-203-147-147.ec2.intern

ip-10-243-42-147.ec2.intern

ip-10-204-85-110.ec2.intern

ip-10-243-150-241.ec2.intern

ip-10-112-58-84.ec2.intern

localhost

(Nodes colored by 1-minute load) Legend

(Nodes colored by 1-minute load) | Legend

Look at Logs?



Logs in distributed systems are spread out, incomplete,
& usually very difficult to understand

There has to be a better way for
application developers to
detect & fix problems



Visualize:

Show me all relevant data in one place



Optimize:

Analyze the data for me and give me
diagnoses and fixes



Strategize:

Help me prevent the problems from happening and meet my goals



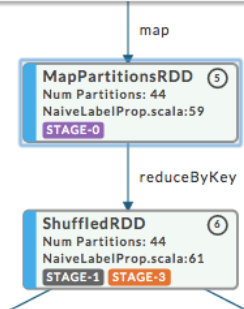


Demo

Visualize:

Show me all relevant data in one place

| MAPPARTITIONSRDD |
|--|
| DataSamples |
| ((FileId(51),NDA),(List(FileId(50)),1)) |
| ((FileId(51),Tech),(List(FileId(2)),1)) |
| ((FileId(1),Legal),(List(FileId(2)),1)) |
| Description |
| Returns a new RDD by applying a function to all elements of this RDD |
| CallSite |
| NaiveLabelProp.scala:59 |



| | DURATION | DATA I/O | RESOURCES | # OF STAGES |
|--|----------|----------|-----------|-------------|
| | 17m 44s | 1.16GB | 4 | 4 |

SPARK PROGRAM

COPY PROGRAM

```
val fileIdAndNeighborLabels = filesWithinT
hresh.map(ffSim => ((ffSim._1, ffSim._2._2), (
List(ffSim._2._1), 1)))

val adjacencyList = fileIdAndNeighborLabel
s.reduceByKey{
  case (adjFileAndCountA, adjFileAndCountB
) =>
    (adjFileAndCountA._1 ++ adjFileAndCountB
._1, adjFileAndCountA._2 + adjFileAndCountB._2)
}

println('adjacencyList toDebugString')
println(adjacencyList.toDebugString)

adjacencyList.saveAsTextFile(s'adjacency_1
ist/${System.currentTimeMillis()}')
// adjacencyList.cache()

adjacencyList.map{ a => (a._1._1, (a._1._2
, a._2._2))}
```


Visualize:

Show me all relevant data in one place

STAGE DETAIL

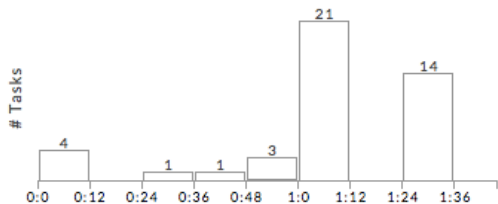
app-20150609030059-0109:stage-0

TIMELINE

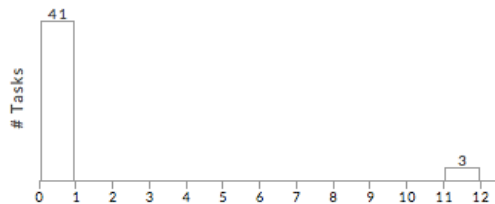
CONFIGURATION

RESOURCE UTILIZATION

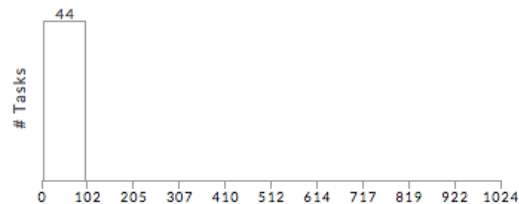
Distribution Charts Drag and highlight graphs to select specific tasks



Task Time (min)



Task Input (MB)



Task Output (KB)

TIMELINE

SELECTED TASKS

Success Killed Failed

Tasks


Killed/Failed

5.72 minutes




Optimize:

Analyze the data for me and give me diagnoses and fixes




EVENTS PANEL
1 EVENTS FOUND




SPARK

SPARK STAGE FAILED

- Error in MapPartitionsRDD Id 5 (NaiveLabelProp.scala:59).
- Transformations and actions cannot be invoked inside other transformations. For example, `rdd1.map(x => rdd2.values.count() * x)` is invalid because the values transformation and count action cannot be performed inside of the `rdd1.map` transformation. For more information, see [SPARK-5063](#)



EVENTS PANEL
1 EVENTS FOUND



SPARK

RESOURCE WASTAGE

- This application has one RDD that will benefit from caching.
- RDD Id 6 (adjacencyList) is recomputed multiple times. Add a `cache()` statement to cache it and improve performance

Strategize:

Help me prevent the problems from happening
and meet my goals

APPLICATION

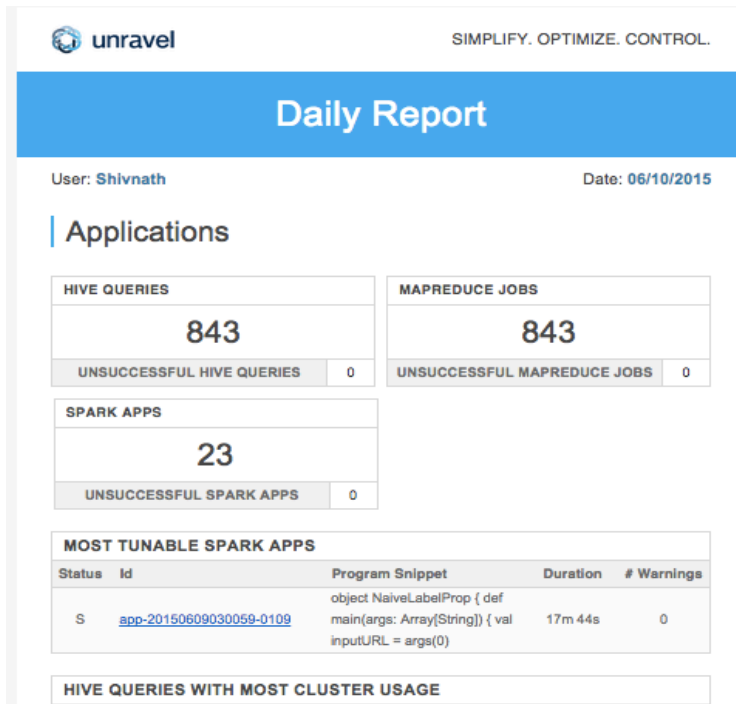
Search... SHOW All **Inefficient** Templates Ad-hoc Workflow Spark [Add Workflow](#)

SHOWING ☒ Applications with Resource Wastage
Applications by Degree of Parallelism
Applications with Load Imbalance
Applications with excessive wait
Applications with inefficient joins
Applications with small files
Longest running applications (multiple nodes)
Longest running applications
Applications using most number of map tasks
Applications using most number of reduce tasks
Applications Using Most Map Time
Applications Using Most Reduce Time
Applications Doing Most DFS Read I/O
Applications Doing Most DFS Write I/O

| ID | APPLICATION | USER | QUEUE | JOBS | EVENTS | RESOURCES |
|----------------------|-------------|--------|-------------|------|--------|-----------|
| hue_2014-0cc-45e4 | 7m 53s | sandra | MODELING | 4 | 2 | 244.63 |
| karma_2014-8-fddc-4a | 2m 29s | rajeev | ADVERTISING | 2 | 1 | 45.02 |

Strategize:

Help me prevent the problems from happening
and meet my goals





Sign up for early access at:
bit.ly/unravelspark

We are hiring: jobs@unraveldata.com

