splice MACHINE

Database Console

Last generated: March 01, 2018

Table of Contents

Database Console

Database Console

Splice Machine Database Console Guide

This topic introduces the *Splice Machine Database Console*, a browser-based tool that you can use to monitor database queries on your cluster in real time. The Console UI allows you to see the Spark queries that are currently running in Splice Machine on your cluster, and to then drill down into each job to see the current progress of the queries, and to identify any potential bottlenecks. If you see something amiss, you can also terminate a query.

> The *Splice Machine Database Console* leverages the Spark cluster manager *Web UI*, which is described here: http://spark.apache.org/docs/latest/monitoring.html.

This section is organized into the following topics:

- The next section, About the Splice Machine Database Console, tells you about the Database Console, including how to access it in your browser.
- The Features of the Splice Machine Database Console topic describes how to use major features of the console interface.
- The Managing Queries with the Console topic shows you how to review and monitor the progress of your Spark jobs.

About the Splice Machine Database Console

The *Splice Machine Spark Database Console* is a browser-based tool that you can use to watch your active Spark queries execute and to review the execution of completed queries. You can use the console to:

- View any completed jobs
- Monitor active jobs as they execute
- View a timeline chart of the events in a job and its stages
- View a Directed Acyclic Graph (DAG) visualization of a job’s stages and the tasks within each stage
- Monitor persisted and cached storage in realtime

How you access the Splice Machine Database Console depends on which Splice Machine product you’re using:

Product	DB Console Access
Database-as-Service	» To monitor the Splice Machine jobs running on your cluster, click the **DB Console** button at the top right of your Management screen or click the DB Console link in the cluster created email that you received from Splice Machine. » To monitor any non-Splice Machine Spark jobs that are running on your cluster, you need to use a different Spark console, which you can access by clicking the **External Spark Console** link that is displayed in the bottom left corner of your cluster's dashboard page.

Product	DB Console Access
On-Premise Database	`http://localhost:4040`

The Database Console URL will only be active after you've run at least one query on our Spark engine; prior to using the Spark engine, your browser will report an error such as *Connection Refused*.

Here are some of the terms you'll encounter while using the Database Console:

Term	Description
Accumulators	Accumulators are variables programmers can declare in Spark applications that can be efficiently supported in parallel operations, and are typically used to implement counters and sums.
Additional Metrics	You can indicate that you want to display additional metrics for a stage or job by clicking the `Show Additional Metrics` arrow and then selecting which metrics you want shown.
DAG Visualization	A visual depiction of the execution Directed Acyclic Graph (DAG) for a job or job stage, which shows the details and flow of data. You can click the `DAG Visualization` arrow to switch to this view.
Enable Zooming	For *event timeline* views, you can enable zooming to expand the view detail for a portion of the timeline. You can click the `Event Timeline` arrow to switch to this view.
Event Timeline	A view that graphically displays the sequence of all *jobs*, a specific job, or a *stage* within a job.
Executor	A process that runs *tasks* on a cluster node.
GC Time	The amount of time spent performing garbage collection in a stage.
Job	The basic unit of execution in the Spark engine, consisting of a set of stages. With some exceptions, each query submitted to the Spark engine is a single job. Each job is assigned a unique Job Id and is part of a unique Job Group.
Locality Level	To minimize data transfers, Spark tries to execute as close to the data as possible. The *Locality Level* value indicates whether a task was able to run on the local node.

Term	Description
Scheduling Mode	The scheduling mode used for a job. In FIFO scheduling, the first job gets priority on all available resources while its stages have tasks to launch. Then the second job gets priority, and so on. In FAIR scheduling, Spark assigns tasks between jobs in a round robin manner, meaning that all jobs get a roughly equal share of the available cluster resources. Which means that short jobs can gain fair access to resources immediately without having to wait for longer jobs to complete.
Scheduling Pool	The FAIR schedule groups jobs into pools, each of which can have a different priority weighting value, which allows you to submit jobs with higher or lower priorities.
ScrollInsensitive row	A row in a result set that is scrollable, and is not sensitive to changes committed by other transactions or by other statements in the same transaction.
Shuffling	Shuffling is the reallocation of data between multiple stages in a Spark job. *Shuffle Write* is amount of data that is serialized and written at the end of a stage for transmission to the next stage. *Shuffle Read* is the amount of serialized data that is read at the beginning of a stage.
Stage	The Splice Machine Spark scheduler splits the execution of a *job* into stages, based on the RDD transformations required to complete the job. Each stage contains a group of tasks that perform a computation in parallel.
Task	A computational command sent from the application driver to an *executor* as part of a *stage*.

See Also

- User Interface Features of the Splice Machine Database Console
- Managing Queries with the Console
- Using Spark Libraries with Splice Machine

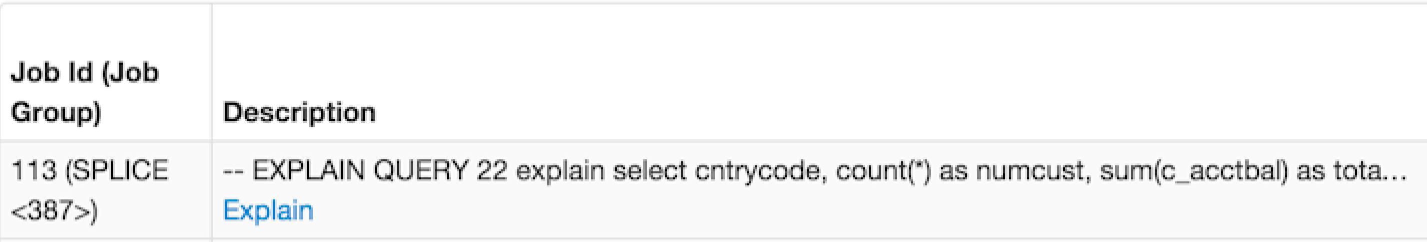
Features of the Splice Machine Database Console

This section summarizes the use of major features of the Database Console interface, including:

- Drilling Down
- Switching Views
- Hovering
- Refreshing the View
- Zooming the Timeline View

Drilling Down

In general, you can click anything that displays in blue (**like this**) to drill down into a more detailed view. For example, clicking **Explain** in the following description from the completed jobs table will drill down into the job details for *Job 113*:

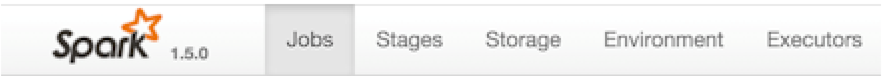


Job Id (Job Group)	Description
113 (SPLICE <387>)	-- EXPLAIN QUERY 22 explain select cntrycode, count(*) as numcust, sum(c_acctbal) as tota... Explain

You can continue to drill down from there to reveal increasing levels of detail.

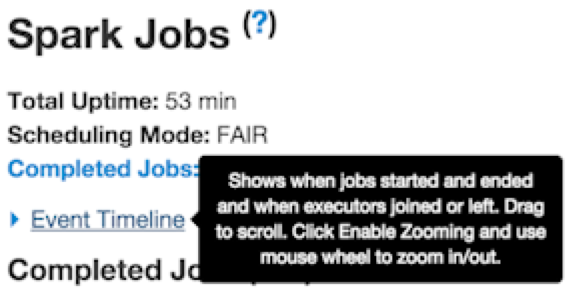
Switching Views

You can quickly switch to a different view by clicking a tab in the tab bar at the top of the console screen. The `Jobs` tab is selected in this screen shot:



Hovering

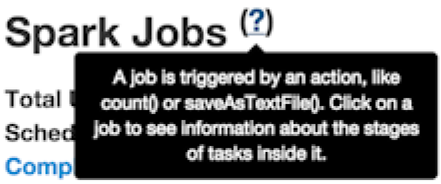
You can hover the cursor over interface element links, like the **Event Timeline** drop-down in the following image, to display a screen tip for the item:



Total Uptime: 53 min
Scheduling Mode: FAIR

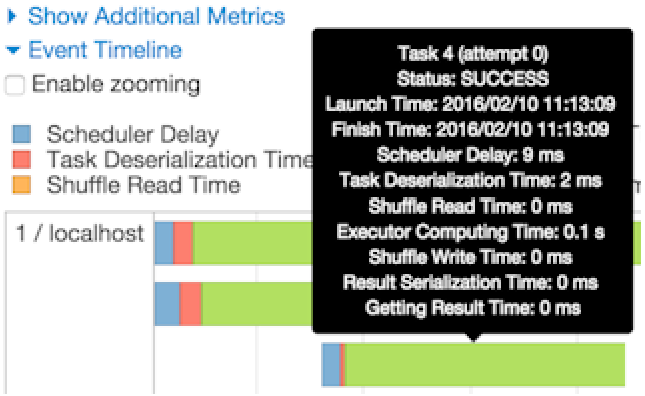
Shows when jobs started and ended and when executors joined or left. Drag to scroll. Click Enable Zooming and use mouse wheel to zoom in/out.

Similarly, you can hover over the ? to display the definition for a term, like the definition of a job:



A job is triggered by an action, like count() or saveAsTextFile(). Click on a job to see information about the stages of tasks inside it.

And you can hover over an event in timeline display to see summary information; for example:



Task 4 (attempt 0)
Status: SUCCESS
Launch Time: 2016/02/10 11:13:09
Finish Time: 2016/02/10 11:13:09
Scheduler Delay: 9 ms
Task Deserialization Time: 2 ms
Shuffle Read Time: 0 ms
Executor Computing Time: 0.1 s
Shuffle Write Time: 0 ms
Result Serialization Time: 0 ms
Getting Result Time: 0 ms

Refreshing the View

Currently, the console does not automatically or periodically refresh the view.

If you’re monitoring an active job, you’ll need to refresh your browser window to view the latest activity.

Zooming the Timeline View

When you’re viewing an event timeline, you can **Enable zooming**, which allows you to use mouse or touch gestures to zoom in on a portion or a timeline, zoom out, or scroll through the timeline.

See Also

- About the DB Console
- Managing Queries with the DB Console

Managing Queries with the DB Console

The Splice Machine Database Console allows you to view queries that are currently running and have completed running in your database. You typically start at the top level, viewing jobs, and then drill down into individual job details, job stages, and task details, as described in these sections:

- Viewing Summary Pages describes the console's top-level summary pages.
- Viewing Job Details describes the pages in which you can view details of active or completed jobs.
- Viewing Stage Details describes the pages in which you can view details of active and completed stages.
- Terminating a Stage shows you how to terminate a job stage that is not performing as you think it should.

Viewing Summary Pages

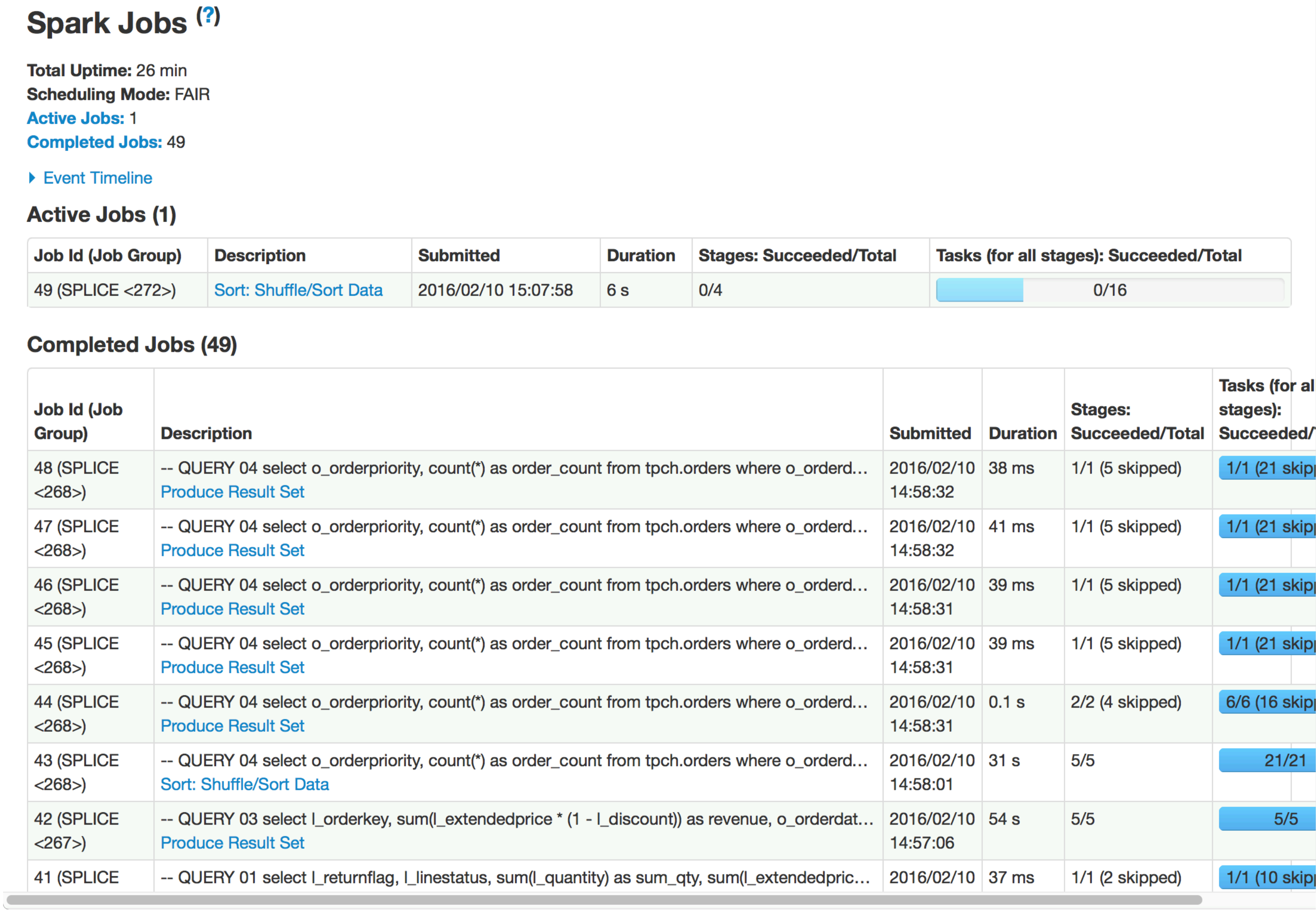
The console includes five summary pages, each of which can be accessed from the tab bar at the top of the console window:

- The Jobs Summary page shows information about all active and completed jobs.
- The Stages Summary Page shows all stages for all jobs, both active and completed.
- The Storage Summary Page shows any RDDs that you have persisted or cached to memory.
- The Environment Summary Page shows information about the Spark run-time environment.
- The Executors Summary Page shows the executors that are currently running.

The Jobs Summary page

The *Jobs Summary Page* is the top-level view in the Splice Machine Database Console, It shows you a summary of any currently active and all completed jobs.

You land on this page when you first view the *Database Console* in your browser, and you can view it at any time by clicking the `Jobs` tab in the tab bar at the top of the page.



Spark Jobs (?)

Total Uptime: 26 min
Scheduling Mode: FAIR
Active Jobs: 1
Completed Jobs: 49

▸ Event Timeline

Active Jobs (1)

Job Id (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
49 (SPLICE <272>)	Sort: Shuffle/Sort Data	2016/02/10 15:07:58	6 s	0/4	0/16

Completed Jobs (49)

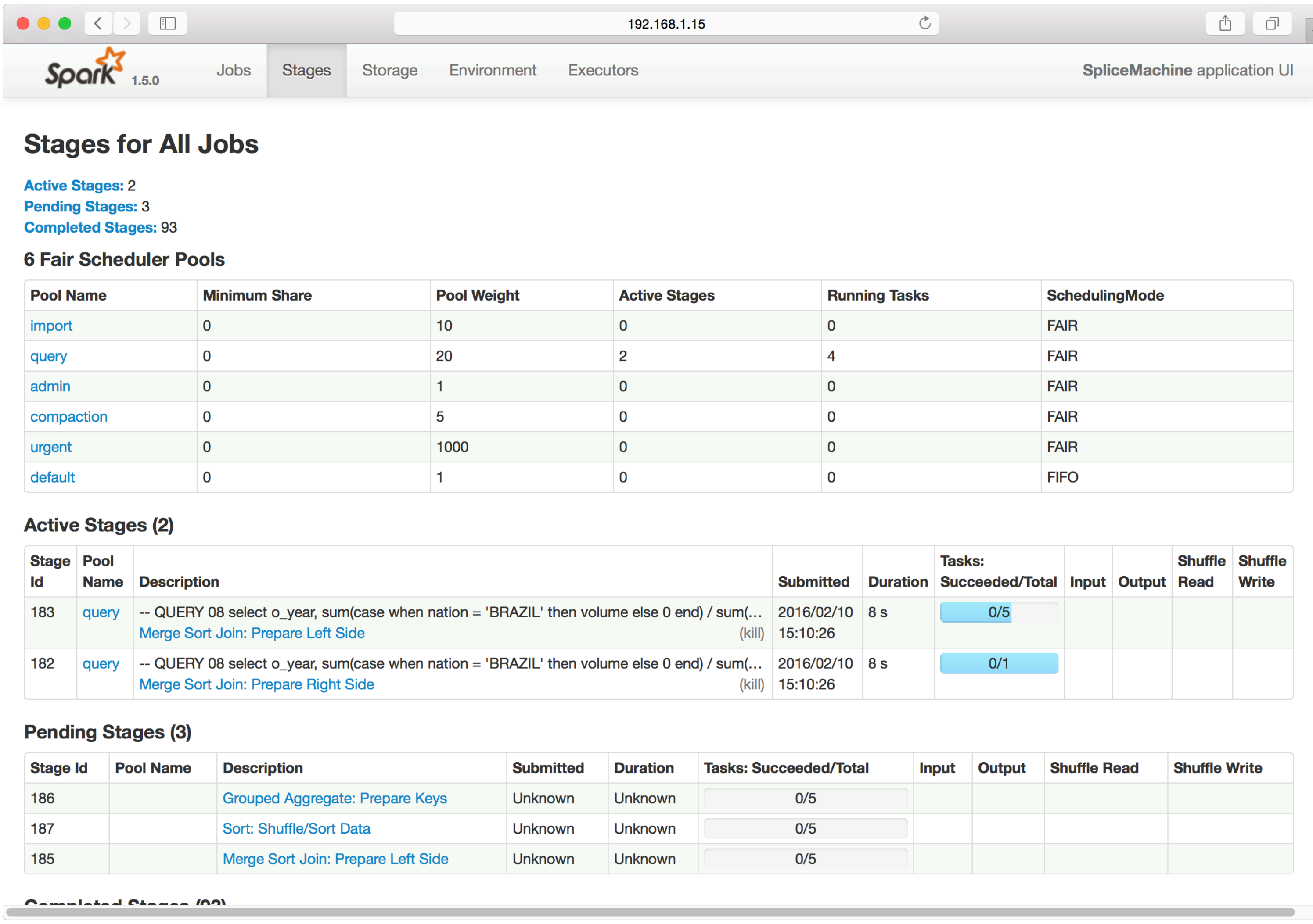
Job Id (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for al stages): Succeeded/
48 (SPLICE <268>)	-- QUERY 04 select o_orderpriority, count(*) as order_count from tpch.orders where o_orderd… Produce Result Set	2016/02/10 14:58:32	38 ms	1/1 (5 skipped)	1/1 (21 skip
47 (SPLICE <268>)	-- QUERY 04 select o_orderpriority, count(*) as order_count from tpch.orders where o_orderd… Produce Result Set	2016/02/10 14:58:32	41 ms	1/1 (5 skipped)	1/1 (21 skip
46 (SPLICE <268>)	-- QUERY 04 select o_orderpriority, count(*) as order_count from tpch.orders where o_orderd… Produce Result Set	2016/02/10 14:58:31	39 ms	1/1 (5 skipped)	1/1 (21 skip
45 (SPLICE <268>)	-- QUERY 04 select o_orderpriority, count(*) as order_count from tpch.orders where o_orderd… Produce Result Set	2016/02/10 14:58:31	39 ms	1/1 (5 skipped)	1/1 (21 skip
44 (SPLICE <268>)	-- QUERY 04 select o_orderpriority, count(*) as order_count from tpch.orders where o_orderd… Produce Result Set	2016/02/10 14:58:31	0.1 s	2/2 (4 skipped)	6/6 (16 skip
43 (SPLICE <268>)	-- QUERY 04 select o_orderpriority, count(*) as order_count from tpch.orders where o_orderd… Sort: Shuffle/Sort Data	2016/02/10 14:58:01	31 s	5/5	21/21
42 (SPLICE <267>)	-- QUERY 03 select l_orderkey, sum(l_extendedprice * (1 - l_discount)) as revenue, o_orderdat… Produce Result Set	2016/02/10 14:57:06	54 s	5/5	5/5
41 (SPLICE	-- QUERY 01 select l_returnflag, l_linestatus, sum(l_quantity) as sum_qty, sum(l_extendedpric…	2016/02/10	37 ms	1/1 (2 skipped)	1/1 (10 skip

> **NOTE:** A stage is shown as skipped when the data has been fetched from a cache and there was no need to reexecute the stage; this happens when shuffling data because the Spark engine automatically caches generated data.

You can click the a job description name (in **blue**) to view job details of any job in the `Active Jobs` or `Completed Jobs` sections.

The Stages Summary Page

The *StagesSummary Page* shows you the available scheduling pools, and a summary of the stages for all active and completed jobs. You can access this page by clicking the `Stages` tab in the tab bar at the top of the window.



Stages for All Jobs

Active Stages: 2
Pending Stages: 3
Completed Stages: 93

6 Fair Scheduler Pools

Pool Name	Minimum Share	Pool Weight	Active Stages	Running Tasks	SchedulingMode
import	0	10	0	0	FAIR
query	0	20	2	4	FAIR
admin	0	1	0	0	FAIR
compaction	0	5	0	0	FAIR
urgent	0	1000	0	0	FAIR
default	0	1	0	0	FIFO

Active Stages (2)

Stage Id	Pool Name	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
183	query	-- QUERY 08 select o_year, sum(case when nation = 'BRAZIL' then volume else 0 end) / sum(… Merge Sort Join: Prepare Left Side (kill)	2016/02/10 15:10:26	8 s	0/5				
182	query	-- QUERY 08 select o_year, sum(case when nation = 'BRAZIL' then volume else 0 end) / sum(… Merge Sort Join: Prepare Right Side (kill)	2016/02/10 15:10:26	8 s	0/1				

Pending Stages (3)

Stage Id	Pool Name	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
186		Grouped Aggregate: Prepare Keys	Unknown	Unknown	0/5				
187		Sort: Shuffle/Sort Data	Unknown	Unknown	0/5				
185		Merge Sort Join: Prepare Left Side	Unknown	Unknown	0/5				

You can click the descriptive name of a stage (in **blue**) to view the stage details.

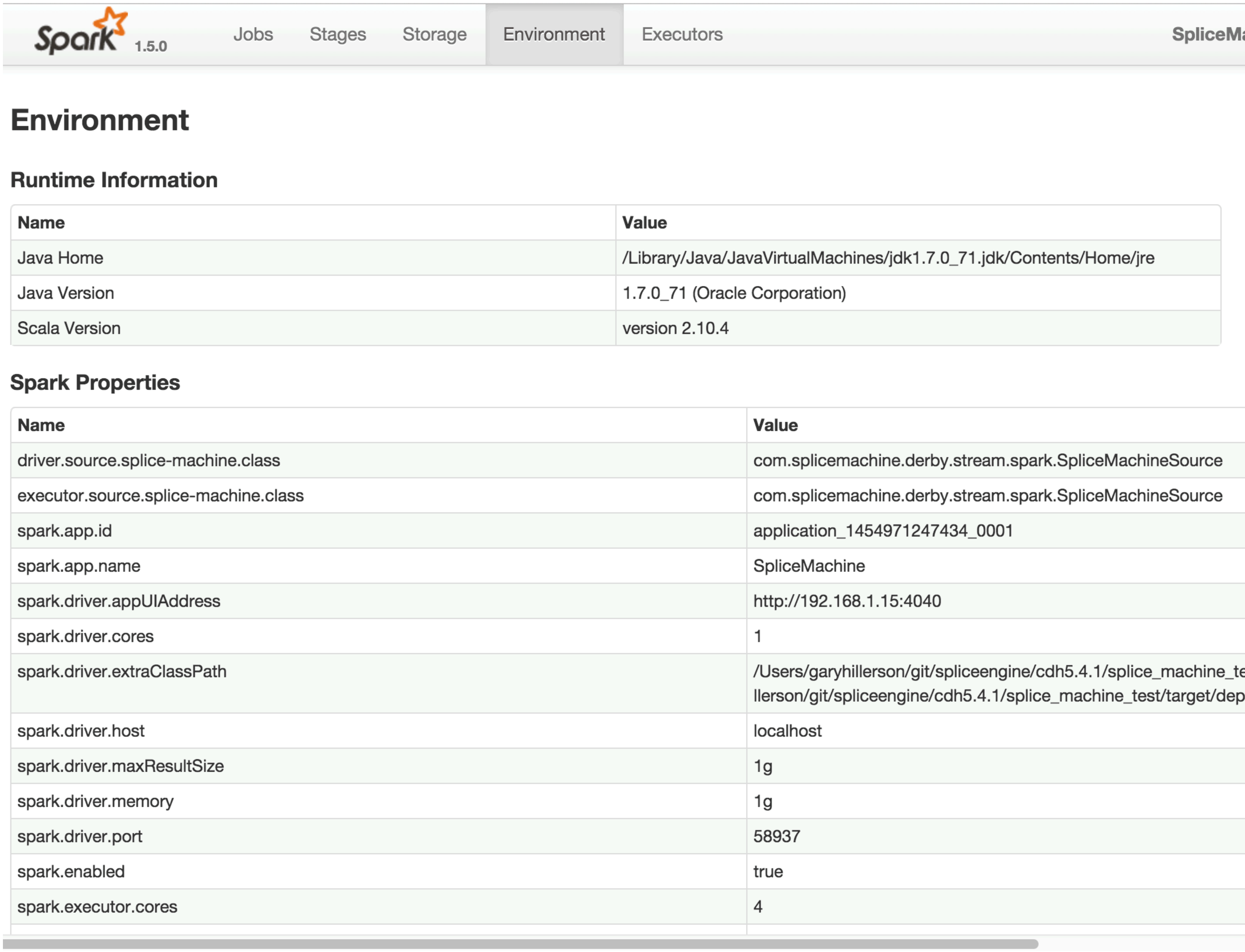
The `Fair Scheduler Pools` section at the top of the page shows the name and weighting value for each of the scheduler pools that have been defined for your database jobs.

The Storage Summary Page

The *Storage Summary Page* displays information about any RDDs that are currently persisted or cached. You can access this page by clicking the `Storage` tab in the tab bar at the top of the window:

The Environment Summary Page

The *Environment Summary Page* displays information about which software versions you're using, and shows the values of the Spark-related environment variables. You can access this page by clicking the `Environment` tab in the tab bar at the top of the window:



Environment

Runtime Information

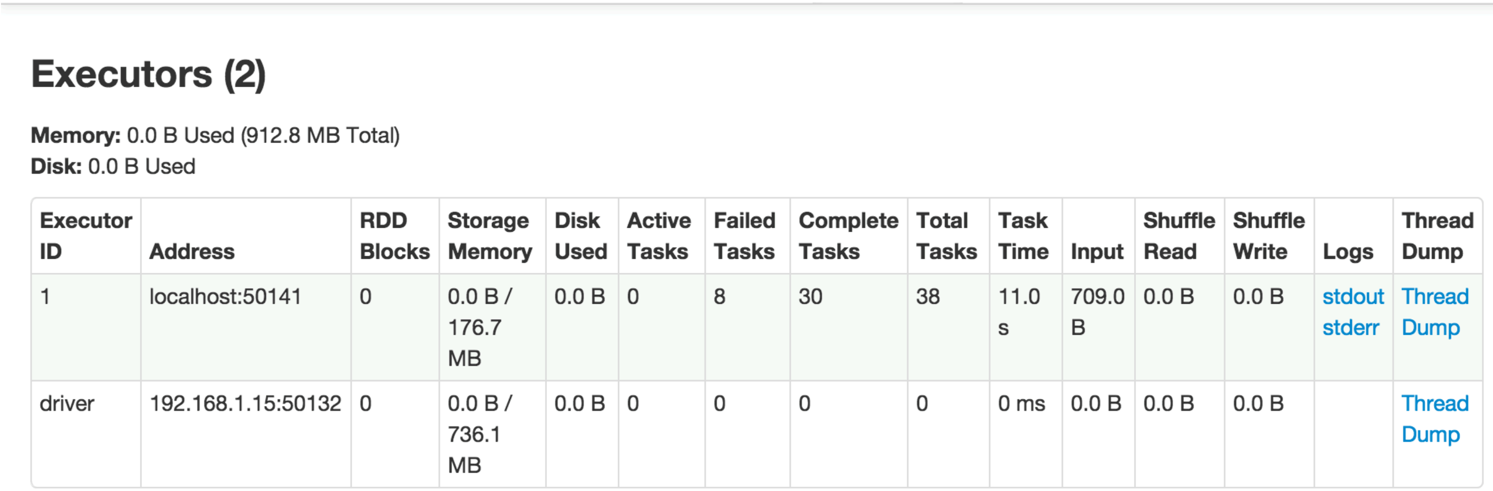
Name	Value
Java Home	/Library/Java/JavaVirtualMachines/jdk1.7.0_71.jdk/Contents/Home/jre
Java Version	1.7.0_71 (Oracle Corporation)
Scala Version	version 2.10.4

Spark Properties

Name	Value
driver.source.splice-machine.class	com.splicemachine.derby.stream.spark.SpliceMachineSource
executor.source.splice-machine.class	com.splicemachine.derby.stream.spark.SpliceMachineSource
spark.app.id	application_1454971247434_0001
spark.app.name	SpliceMachine
spark.driver.appUIAddress	http://192.168.1.15:4040
spark.driver.cores	1
spark.driver.extraClassPath	/Users/garyhillerson/git/spliceengine/cdh5.4.1/splice_machine_te llerson/git/spliceengine/cdh5.4.1/splice_machine_test/target/dep
spark.driver.host	localhost
spark.driver.maxResultSize	1g
spark.driver.memory	1g
spark.driver.port	58937
spark.enabled	true
spark.executor.cores	4

The Executors Summary Page

The *Executors Summary Page* shows you the Spark executors that are currently running. You can access this page by clicking the `Executors` tab in the tab bar at the top of the window:



Executors (2)

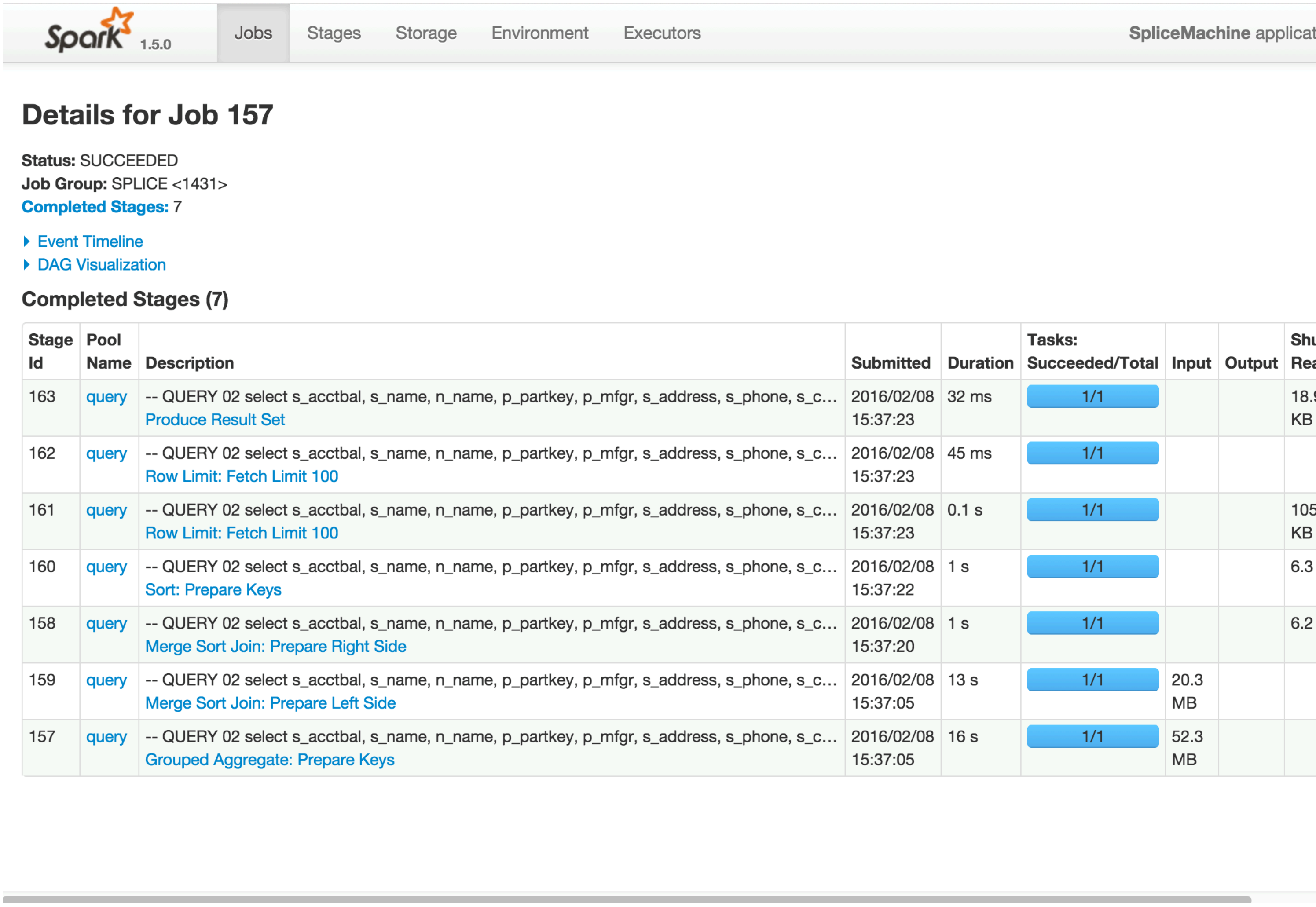
Memory: 0.0 B Used (912.8 MB Total)
Disk: 0.0 B Used

Executor ID	Address	RDD Blocks	Storage Memory	Disk Used	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time	Input	Shuffle Read	Shuffle Write	Logs	Thread Dump
1	localhost:50141	0	0.0 B / 176.7 MB	0.0 B	0	8	30	38	11.0 s	709.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
driver	192.168.1.15:50132	0	0.0 B / 736.1 MB	0.0 B	0	0	0	0	0 ms	0.0 B	0.0 B	0.0 B		Thread Dump

You can click **Thread Dump** to display a thread dump for an executor, or you can click a log name to see the contents of the log.

Viewing Job Details

If you click a job to see its details, you'll see a screen like the following displayed, which shows the stages of the job:



Details for Job 157

Status: SUCCEEDED
Job Group: SPLICE <1431>
Completed Stages: 7

- Event Timeline
- DAG Visualization

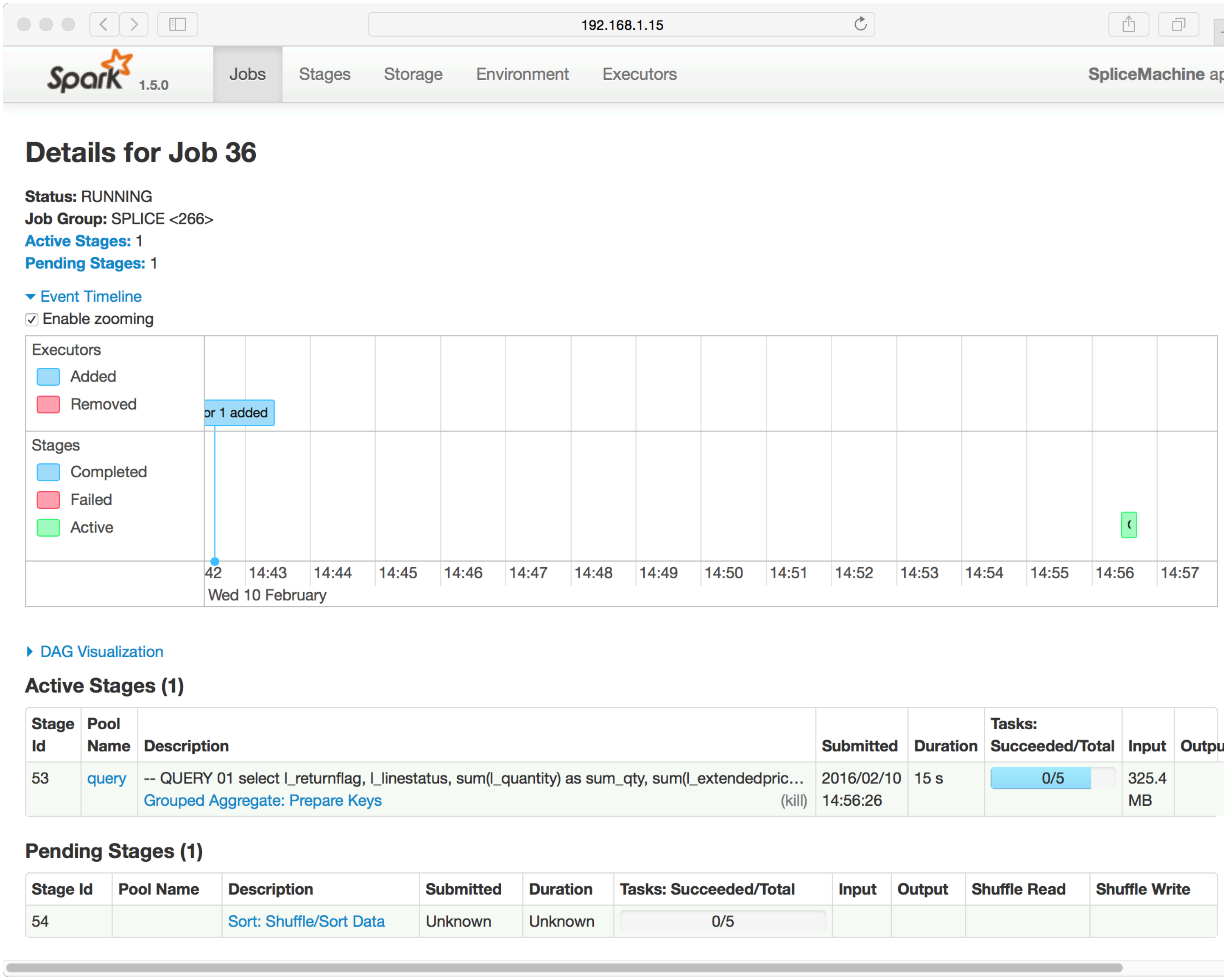
Completed Stages (7)

Stage Id	Pool Name	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shu Rea
163	query	-- QUERY 02 select s_acctbal, s_name, n_name, p_partkey, p_mfgr, s_address, s_phone, s_c… Produce Result Set	2016/02/08 15:37:23	32 ms	1/1			18.9 KB
162	query	-- QUERY 02 select s_acctbal, s_name, n_name, p_partkey, p_mfgr, s_address, s_phone, s_c… Row Limit: Fetch Limit 100	2016/02/08 15:37:23	45 ms	1/1			
161	query	-- QUERY 02 select s_acctbal, s_name, n_name, p_partkey, p_mfgr, s_address, s_phone, s_c… Row Limit: Fetch Limit 100	2016/02/08 15:37:23	0.1 s	1/1			105 KB
160	query	-- QUERY 02 select s_acctbal, s_name, n_name, p_partkey, p_mfgr, s_address, s_phone, s_c… Sort: Prepare Keys	2016/02/08 15:37:22	1 s	1/1			6.3
158	query	-- QUERY 02 select s_acctbal, s_name, n_name, p_partkey, p_mfgr, s_address, s_phone, s_c… Merge Sort Join: Prepare Right Side	2016/02/08 15:37:20	1 s	1/1			6.2
159	query	-- QUERY 02 select s_acctbal, s_name, n_name, p_partkey, p_mfgr, s_address, s_phone, s_c… Merge Sort Join: Prepare Left Side	2016/02/08 15:37:05	13 s	1/1	20.3 MB		
157	query	-- QUERY 02 select s_acctbal, s_name, n_name, p_partkey, p_mfgr, s_address, s_phone, s_c… Grouped Aggregate: Prepare Keys	2016/02/08 15:37:05	16 s	1/1	52.3 MB		

You can expand the job detail display by selecting the **Event Timeline** and/or **DAG Visualization** buttons.

Job Details Event Time Line View

The job details time-line view looks like the following screen shot:



Details for Job 36

Status: RUNNING
Job Group: SPLICE <266>
Active Stages: 1
Pending Stages: 1

▾ Event Timeline
☑ Enable zooming

▸ DAG Visualization

Active Stages (1)

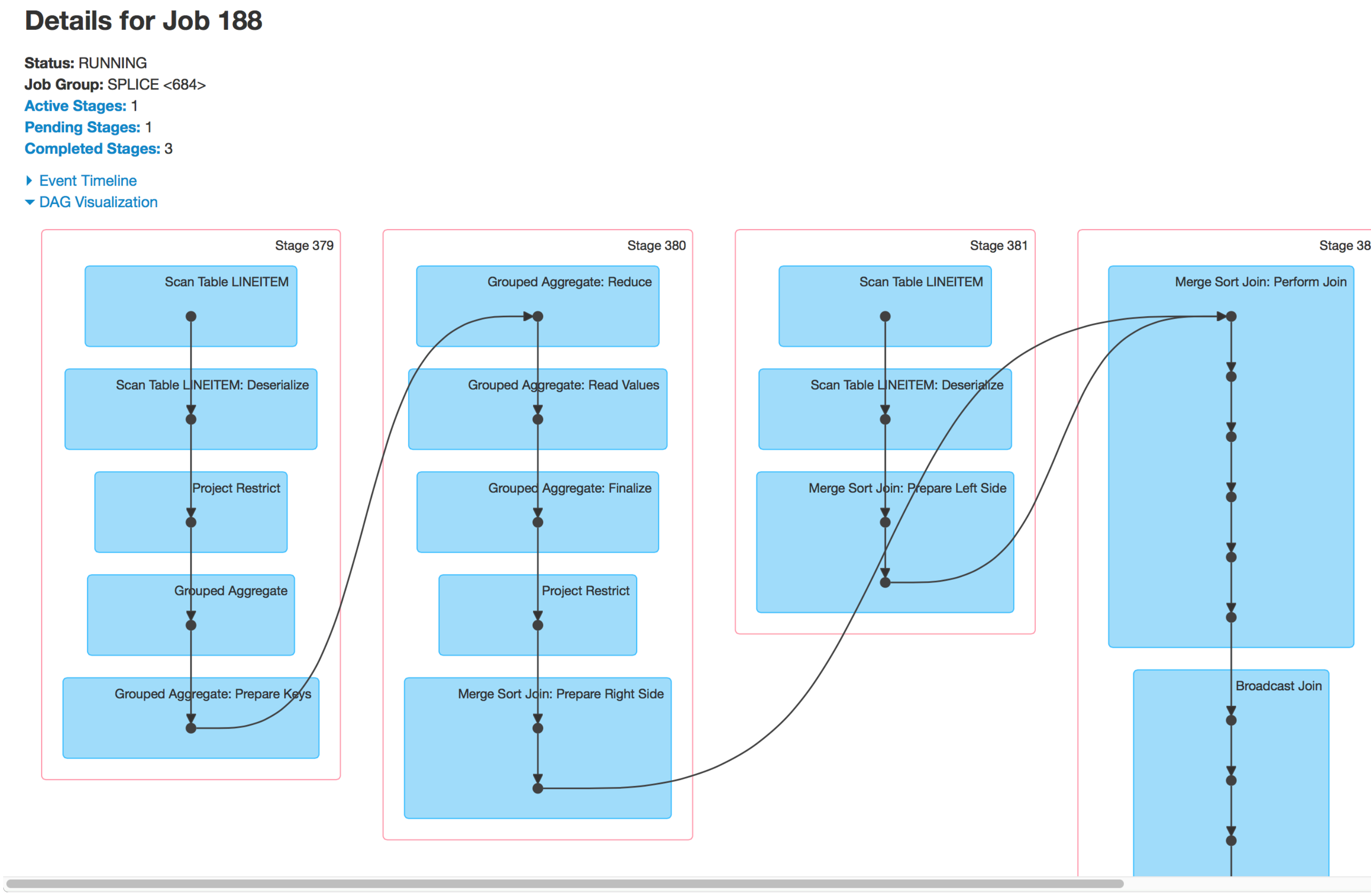
Stage Id	Pool Name	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Outpu
53	query	-- QUERY 01 select l_returnflag, l_linestatus, sum(l_quantity) as sum_qty, sum(l_extendedpric… Grouped Aggregate: Prepare Keys (kill)	2016/02/10 14:56:26	15 s	0/5	325.4 MB	

Pending Stages (1)

Stage Id	Pool Name	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
54		Sort: Shuffle/Sort Data	Unknown	Unknown	0/5				

Job Details Graphical Visualization View

The DAG Visualization view for a job looks like this:



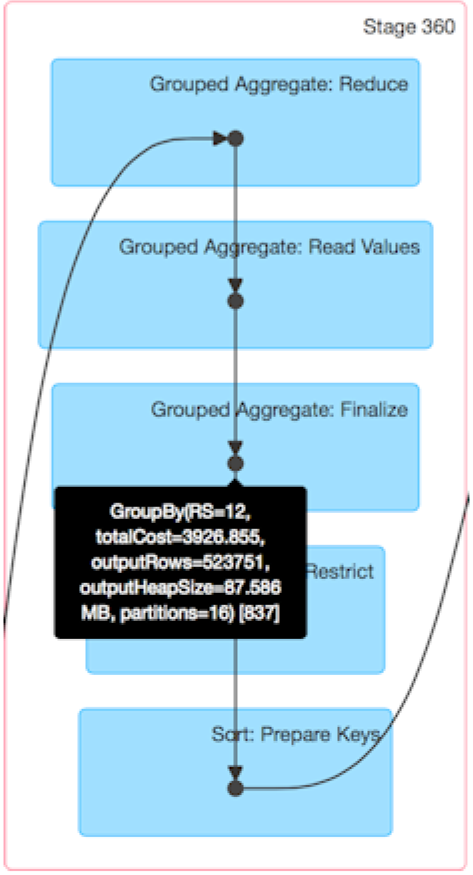
Details for Job 188

Status: RUNNING
Job Group: SPLICE <684>
Active Stages: 1
Pending Stages: 1
Completed Stages: 3

▸ Event Timeline
▾ DAG Visualization
Stage 379
Scan Table LINEITEM
Scan Table LINEITEM: Deserialize
Project Restrict
Grouped Aggregate
Grouped Aggregate: Prepare Keys
Stage 380
Grouped Aggregate: Reduce
Grouped Aggregate: Read Values
Grouped Aggregate: Finalize
Project Restrict
Merge Sort Join: Prepare Right Side
Stage 381
Scan Table LINEITEM
Scan Table LINEITEM: Deserialize
Merge Sort Join: Prepare Left Side
Stage 38
Merge Sort Join: Perform Join
Broadcast Join

Some key things to know about the DAG view are:

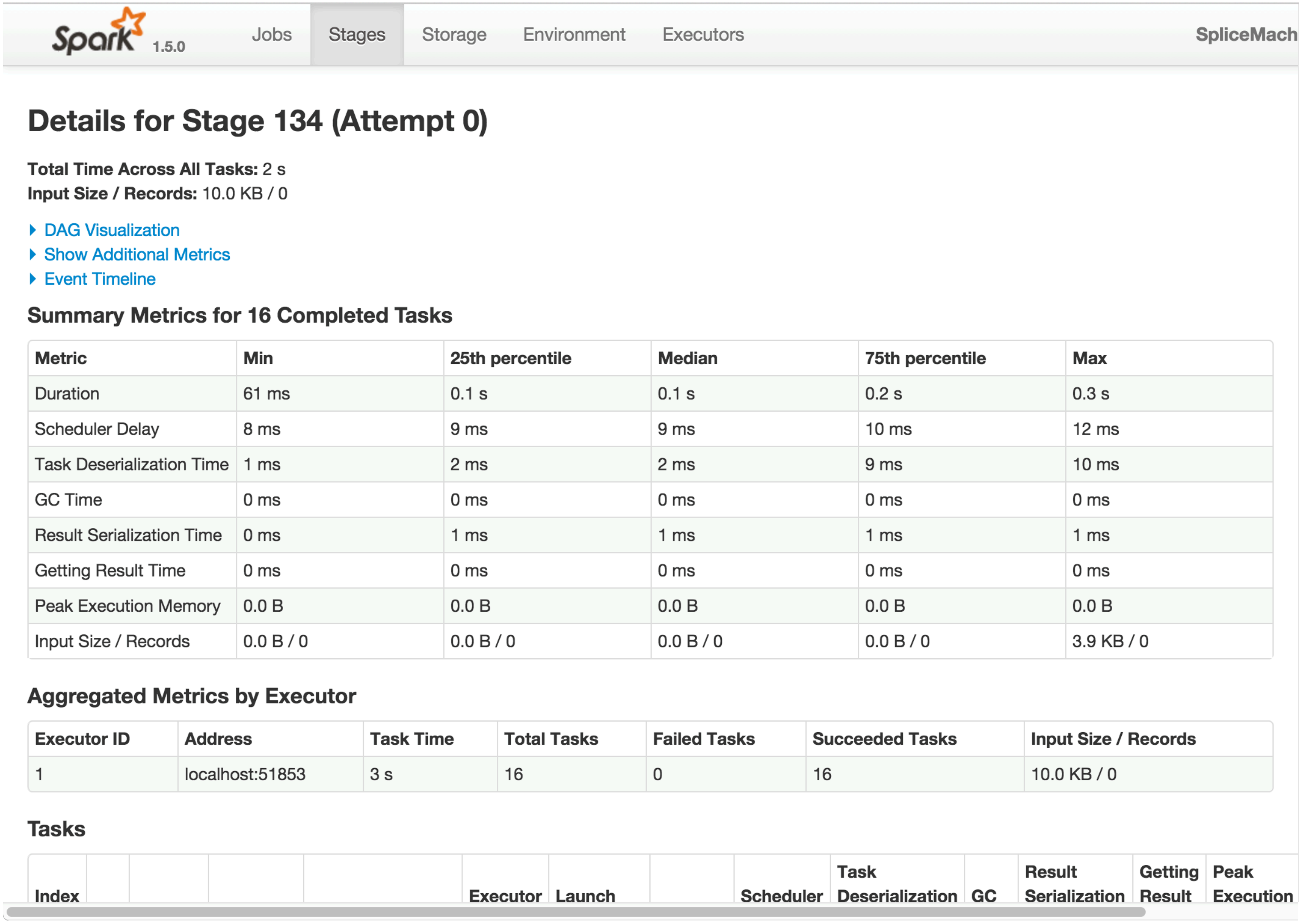
- You can click in the box representing a stage to view the detailed tasks within that stage. For an example, see Graphical View of the Tasks in a Stage, in the next section.
- You can hover over any of the black dots inside a task box to display information about the task. For example:


Stage 360
Grouped Aggregate: Reduce
Grouped Aggregate: Read Values
Grouped Aggregate: Finalize
Restrict
Sort: Prepare Keys

GroupBy(RS=12, totalCost=3926.855, outputRows=523751, outputHeapSize=87.586 MB, partitions=16) [837]

Viewing Stage Details

Viewing stage details is very much the same as viewing job details. If you click the name of a stage in another page, the detailed view of that stage displays:



Details for Stage 134 (Attempt 0)

Total Time Across All Tasks: 2 s
Input Size / Records: 10.0 KB / 0

Summary Metrics for 16 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	61 ms	0.1 s	0.1 s	0.2 s	0.3 s
Scheduler Delay	8 ms	9 ms	9 ms	10 ms	12 ms
Task Deserialization Time	1 ms	2 ms	2 ms	9 ms	10 ms
GC Time	0 ms	0 ms	0 ms	0 ms	0 ms
Result Serialization Time	0 ms	1 ms	1 ms	1 ms	1 ms
Getting Result Time	0 ms	0 ms	0 ms	0 ms	0 ms
Peak Execution Memory	0.0 B	0.0 B	0.0 B	0.0 B	0.0 B
Input Size / Records	0.0 B / 0	0.0 B / 0	0.0 B / 0	0.0 B / 0	3.9 KB / 0

Aggregated Metrics by Executor

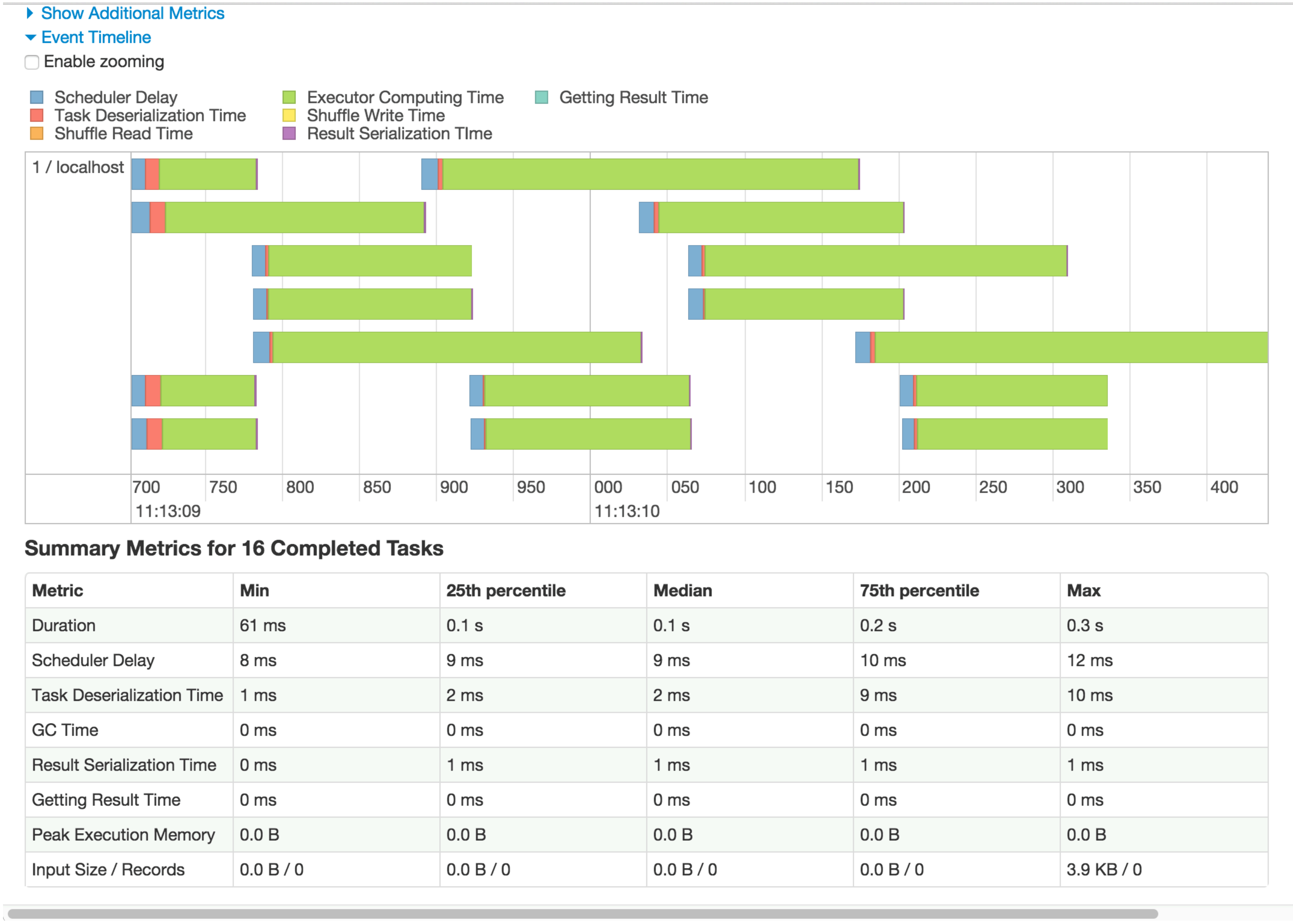
Executor ID	Address	Task Time	Total Tasks	Failed Tasks	Succeeded Tasks	Input Size / Records
1	localhost:51853	3 s	16	0	16	10.0 KB / 0

Tasks

									Task	GC	Result	Getting	Peak
Index					Executor	Launch		Scheduler	Deserialization		Serialization	Result	Execution

The Event Time Line View of a Stage

The Event Timeline view of a stage looks like this:

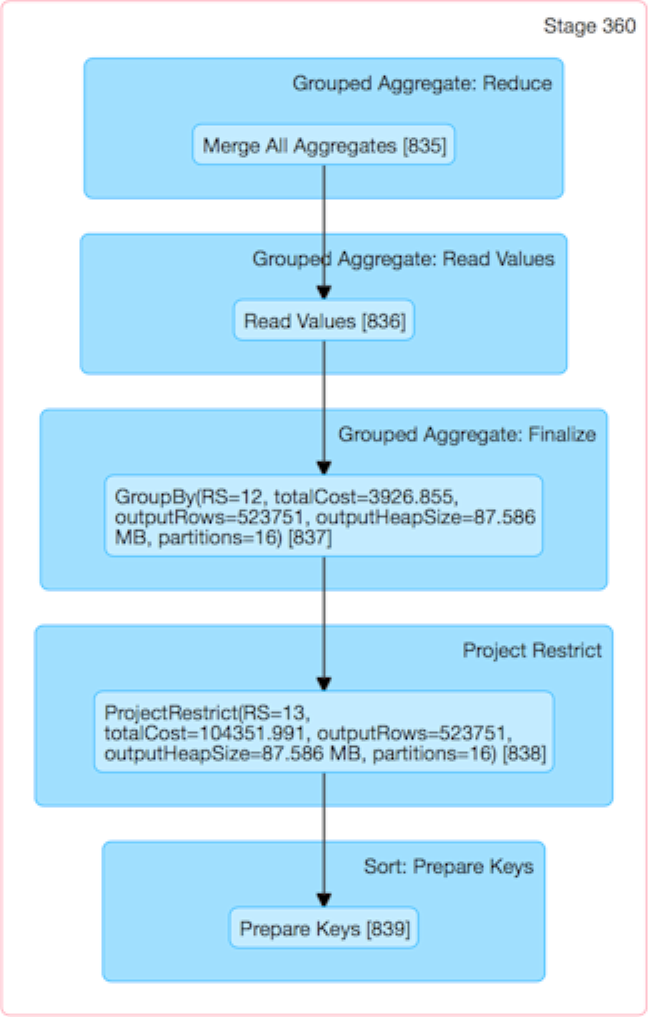


Summary Metrics for 16 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	61 ms	0.1 s	0.1 s	0.2 s	0.3 s
Scheduler Delay	8 ms	9 ms	9 ms	10 ms	12 ms
Task Deserialization Time	1 ms	2 ms	2 ms	9 ms	10 ms
GC Time	0 ms	0 ms	0 ms	0 ms	0 ms
Result Serialization Time	0 ms	1 ms	1 ms	1 ms	1 ms
Getting Result Time	0 ms	0 ms	0 ms	0 ms	0 ms
Peak Execution Memory	0.0 B	0.0 B	0.0 B	0.0 B	0.0 B
Input Size / Records	0.0 B / 0	0.0 B / 0	0.0 B / 0	0.0 B / 0	3.9 KB / 0

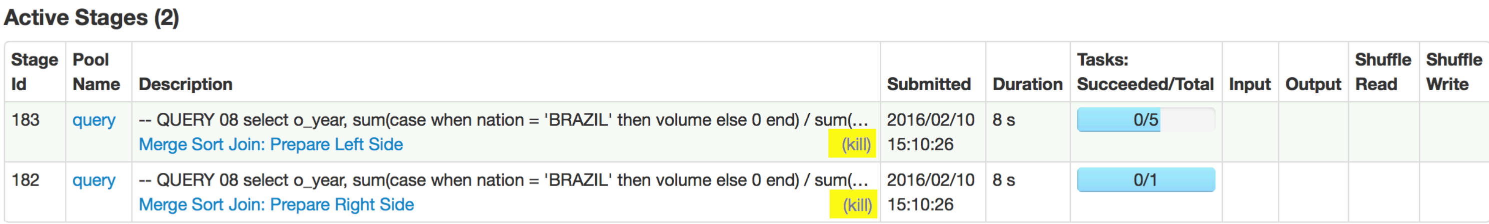
Graphical View of the Tasks in a Stage

The DAG Visualization view of a stage looks like this:


Stage 360
Grouped Aggregate: Reduce
Merge All Aggregates [835]
Grouped Aggregate: Read Values
Read Values [836]
Grouped Aggregate: Finalize
GroupBy(RS=12, totalCost=3926.855, outputRows=523751, outputHeapSize=87.586 MB, partitions=16) [837]
Project Restrict
ProjectRestrict(RS=13, totalCost=104351.991, outputRows=523751, outputHeapSize=87.586 MB, partitions=16) [838]
Sort: Prepare Keys
Prepare Keys [839]

Terminating a Stage

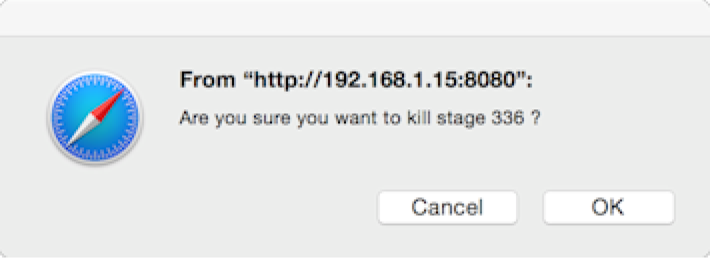
If you conclude that an active job stage is not performing the way you think it should, you can terminate a stage by clicking the `Kill` button shown in the description of every active stage. The following image highlights the kill buttons that you'll find in the console display:



Active Stages (2)

Stage Id	Pool Name	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
183	query	-- QUERY 08 select o_year, sum(case when nation = 'BRAZIL' then volume else 0 end) / sum(… Merge Sort Join: Prepare Left Side (kill)	2016/02/10 15:10:26	8 s	0/5				
182	query	-- QUERY 08 select o_year, sum(case when nation = 'BRAZIL' then volume else 0 end) / sum(… Merge Sort Join: Prepare Right Side (kill)	2016/02/10 15:10:26	8 s	0/1				

You'll be prompted to verify that you want the stage terminated:



From “http://192.168.1.15:8080”:

Are you sure you want to kill stage 336 ?

You can access the `Kill` button by drilling down into a job’s stages, or by selecting the `Stages` tab in the tab bar, which displays all stages for all jobs.

See Also

- About the Splice Machine Database Console
- User Interface Features of the Splice Machine Database Console
- Using Spark Libraries with Splice Machine