

**DATA+AI
SUMMIT**

BY  databricks

Determining When to Use GPU for Your ETL Pipelines at Scale

Chris Vo, AT&T

Hao Zhu, NVIDIA

Databricks
2023



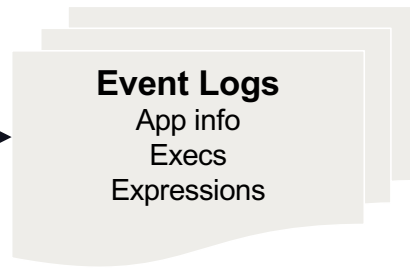
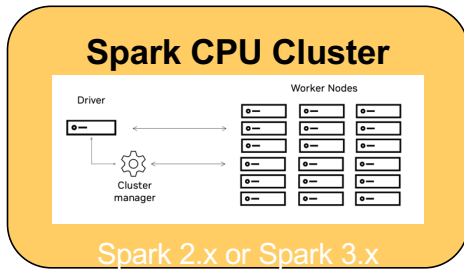
Agenda

Outline

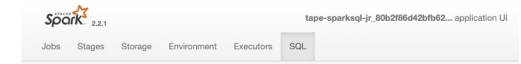
- NVIDIA Implementation of Spark RAPIDS (GPU) Qualification Tool
- AT&T Usage Pattern for Qualification Tool
- Actual Cost Saving after Conversion
- Key Takeaways
- What's Next For AT&T

Qualification Tool

Process Spark Event Logs

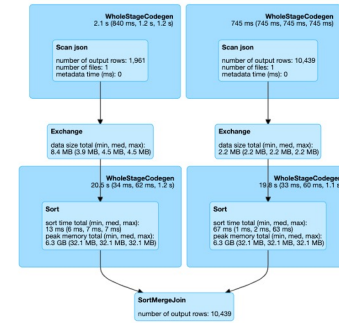


Spark 2.x or Spark 3.x



Details for Query 0

Submitted Time: 2019/09/11 02:30:37
Duration: 34 s
Succeeded Jobs: 2



```
== Parsed Logical Plan ==
Join FullOuter, (id#14 = person_id#958)
+-
  Relation[Birth_date#8,contact_details#9,death_date#10,family_name#11,gender#12,given_name#13,id#14,identifiers#15,
  ,image#16,images#17,links#18,name#19,other_names#20,sort_name#21] json
+-
  Relation[area_id#45,end_date#46,legislative_period_id#47,on_behalf_of_id#48,organization_id#49,person_id#958,role#
  51,start_date#52] json

== Analyzed Logical Plan ==
Birth_date: string, contact_details: array<struct<type:string,value:string>>, death_date: string, family_name:
string, gender: string, given_name: string, id: string, identifiers:
array<struct<identifier:string,scheme:string>>, image: string, images: array<struct<url:string>>, links:
array<struct<note:string,url:string>>, name: string, other_names:
array<struct<long:string,name:string,note:string>>, sort_name: string, area_id: string, end_date: string,
legislative_period_id: string, on_behalf_of_id: string, organization_id: string, person_id: string, role: string,
start_date: string
Join FullOuter, (id#14 = person_id#958)
+-
  Relation[Birth_date#8,contact_details#9,death_date#10,family_name#11,gender#12,given_name#13,id#14,identifiers#15,
  ,image#16,images#17,links#18,name#19,other_names#20,sort_name#21] json
+-
  Relation[area_id#45,end_date#46,legislative_period_id#47,on_behalf_of_id#48,organization_id#49,person_id#958,role#
  51,start_date#52] json

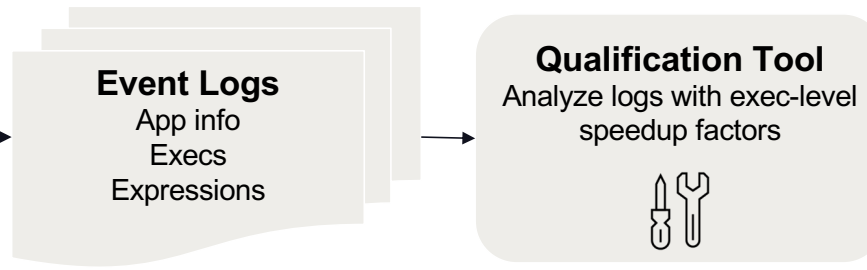
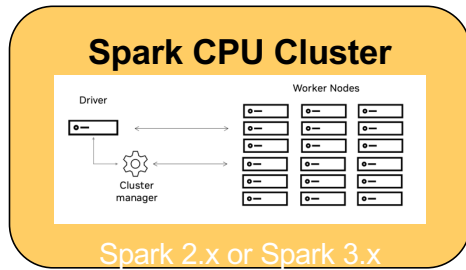
== Optimized Logical Plan ==
Join FullOuter, (id#14 = person_id#958)
+-
  Relation[Birth_date#8,contact_details#9,death_date#10,family_name#11,gender#12,given_name#13,id#14,identifiers#15,
  ,image#16,images#17,links#18,name#19,other_names#20,sort_name#21] json
+-
  Relation[area_id#45,end_date#46,legislative_period_id#47,on_behalf_of_id#48,organization_id#49,person_id#958,role#
  51,start_date#52] json

== Physical Plan ==
SortMergeJoin [id#14], [person_id#958], FullOuter
```



Qualification Tool

Exec and Expression Analysis

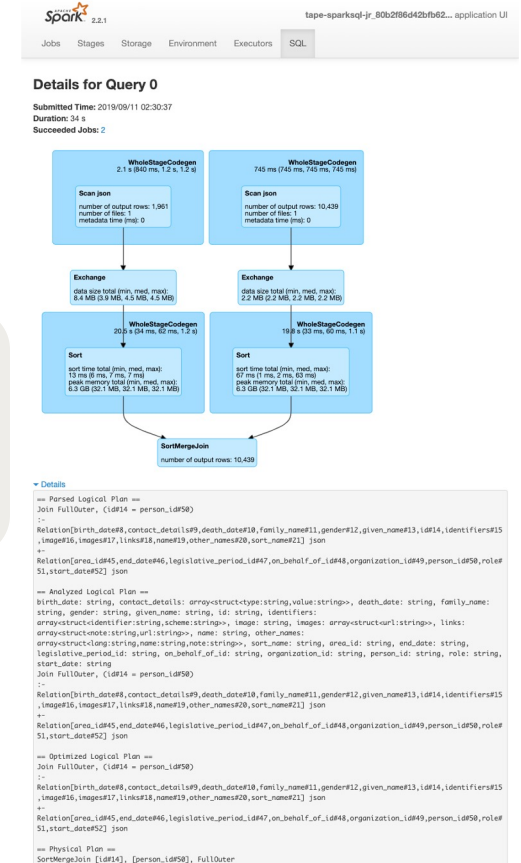


Spark 2.x or Spark 3.x

```
java ${QUALIFICATION_HEAP} \  
  -cp rapids-4-spark-tools_2.12-<version>.jar:$SPARK_HOME/jars/* \  
  com.nvidia.spark.rapids.tool.qualification.QualificationMain  
/usr/logs/app-name1
```

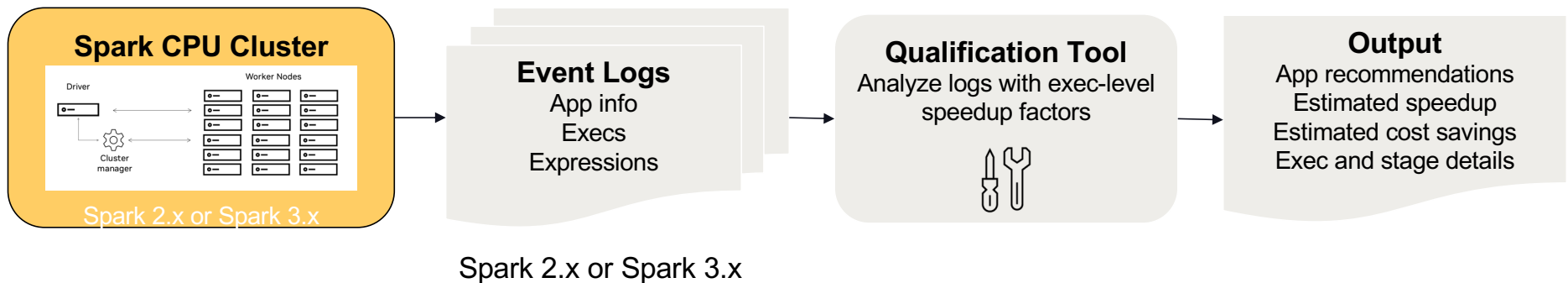
OR

```
spark_rapids_user_tools databricks_aws qualification \  
  --eventlogs $EVENTLOGS \  
  --cpu_cluster $CLUSTER_NAME \  
  --remote_folder $REMOTE_FOLDER
```



Qualification Tool

Predicting the benefits of Spark + GPUs



```
java ${QUALIFICATION_HEAP} \
    -cp rapids-4-spark-tools_2.12-<version>.jar:$SPARK_HOME/jars/* \
    com.nvidia.spark.rapids.tool.qualification.QualificationMain
/usr/logs/app-name1
```

OR

```
spark_rapids_user_tools databricks_aws qualification \
  --eventlogs $EVENTLOGS \
  --cpu_cluster $CLUSTER_NAME \
  --remote_folder $REMOTE_FOLDER
```



Qualification Tool

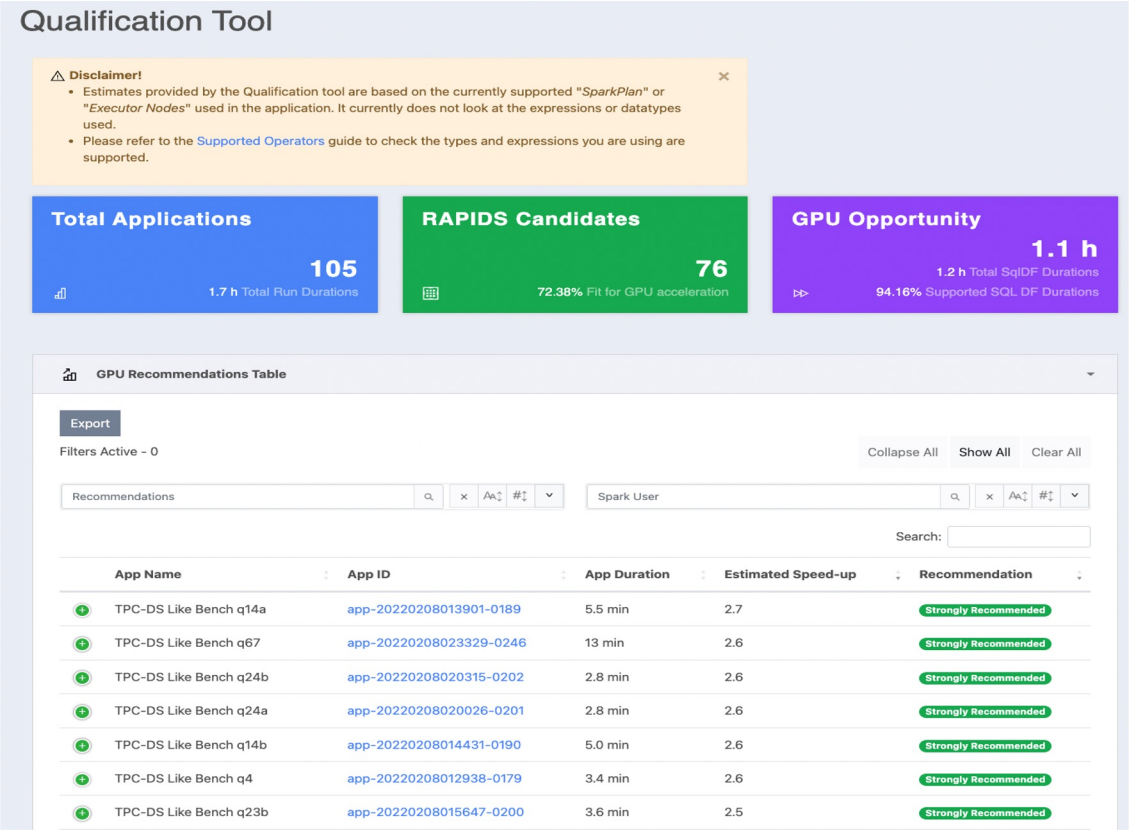
Example Output

App Name	Recommendation	Estimated GPU Speedup	Estimated GPU Duration(s)	App Duration(s)	Estimated GPU Savings(%)
Customer App #1	Strongly Recommended	3.66	651.24	2384.32	64.04
Sales App #1	Strongly Recommended	3.14	89.61	281.62	58.11
Sales App #2	Strongly Recommended	3.12	300.39	939.21	57.89
Customer App #2	Strongly Recommended	2.55	698.16	1783.65	48.47



Qualification Tool

Example UI Output



Qualification Tool

Download Links and Documentation

spark-rapids-user-tools 23.4.0

✓ Latest version

`pip install spark-rapids-user-tools`

Released: Apr 26, 2023

A simple wrapper process around cloud service providers to run tools for the RAPIDS Accelerator for Apache Spark.

Navigation

Project description

Release history

Download files

Statistics

View statistics for this project via [Libraries.io](#), or by using [our public dataset on Google BigQuery](#)

Meta

License: Apache Software License

Project description

spark-rapids-user-tools

User tools to help with the adoption, installation, execution, and tuning of RAPIDS Accelerator for Apache Spark.

The wrapper improves end-user experience within the following dimensions:

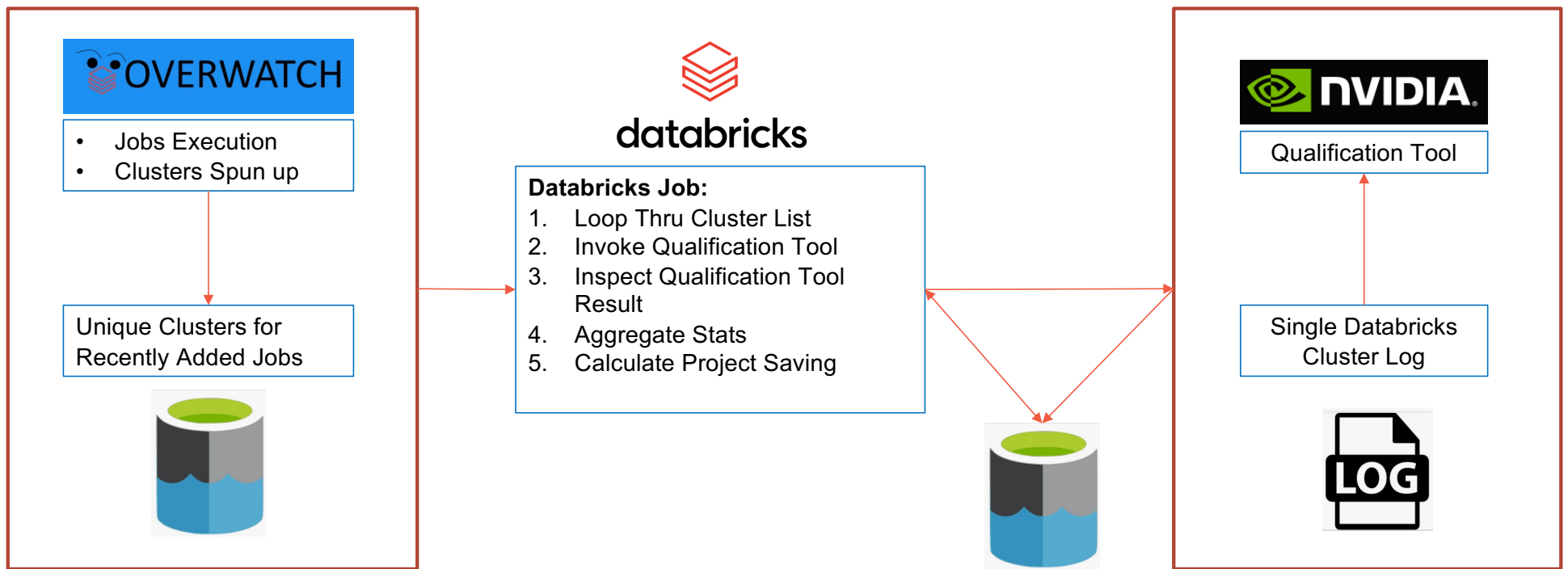
1. Qualification: Educate the CPU customer on the cost savings and acceleration potential of RAPIDS Accelerator for Apache Spark. The output shows a list of apps recommended for RAPIDS Accelerator for Apache Spark with estimated savings and speed-up.
2. Bootstrap: Provide optimized RAPIDS Accelerator for Apache Spark configs based on GPU cluster shape. The output shows updated Spark config settings on driver node.
3. Tuning: Tune RAPIDS Accelerator for Apache Spark configs based on initial job run leveraging Spark event logs. The output shows recommended per-app RAPIDS Accelerator for Apache Spark config settings.
4. Diagnostics: Run diagnostic functions to validate the Dataproc with RAPIDS Accelerator for Apache Spark environment to make sure the cluster is healthy and ready for Spark jobs.

<https://pypi.org/project/spark-rapids-user-tools/>





GPU-Qualification for ETL Jobs Scanning Pipeline



Estimated Saving Projection for Conversion Candidates

Job Name	Contact	Recommendation	Est Hours Saving	Est Unit Saving (\$)	Frequency	Est Yearly Saving (\$)
Job ABC	xx1234	Recommended	0.5	9.52	Hourly	83,356
Job EFG	xx5678	Recommended	0.4	6.92	Hourly	60,587
...						
Job XXX	xx1111	Strongly Recommended	1.0	28.72	Daily	10,481
...						
Job YYY	xx2222	Strongly Recommended	0.5	15.18	Monthly	182

Databricks Jobs Conversion Checklist

1. Rank Job Candidates List By Highest Saving Projection
2. Convert Job Candidates Where Projected Saving > \$10k/yr
3. Record Current CPU Cluster Size and Execution Time
4. Convert Job to Spark RAPIDS GPU
5. Record New GPU Cluster Size and Execution Time
6. Calculate Actual Saving Based on #3 and #5
7. Rollback to CPU If:
 1. Encountered Error
 2. Job Cost Saving is Low (Azure GPU Quota is Limited)

CPU Cluster Configuration

Policy ?

job-connect-2-ext-ms-and-set-proxy | v

☒ Multi node ☐ Single node

Access mode ?

Custom | v

Performance

Databricks runtime version ?

Runtime: 10.4 LTS (Scala 2.12, Spark 3.2.1) | v

☐ Use Photon Acceleration ?

Worker type ?

Standard_DS5_v2 | 56 GB Memory, 16 Cores | v

Workers

8

Driver type

Same as worker | 56 GB Memory, 16 Cores | v

☐ Enable autoscaling ?

Tags ?

Spark RAPIDS (GPU) Cluster Configuration

Cluster Configuration

Shared_job_cluster [UI preview](#) [Provide feedback](#)

Cluster name

Shared_job_cluster

Policy [?](#)

job-connect-2-ext-ms-and-set-proxy

☒ Multi node ☐ Single node

Access mode [?](#)

Custom

Performance

Databricks runtime version [?](#)

Runtime: 10.4 LTS **ML** (GPU, Scala 2.12, Spark 3.2.1)

[NVIDIA EULA](#) [?](#)

☐ Use Photon Acceleration [?](#)

Worker type [?](#)

Standard_NC8as_T4_v3

56 GB Memory

1 GPU

Workers

4

Driver type

Same as worker

56 GB Memory, 1 GPU

☐ Enable autoscaling [?](#)

Advanced Options - Spark Config

Spark [Logging](#) [Init Scripts](#) [Docker](#)

Spark config [?](#)

```
spark.excludeOnFailure.enabled true
spark.rapids.sql.reader.batchSizeBytes 768M
spark.rapids.sql.batchSizeBytes 512M
spark.rapids.sql.enabled true
spark.sql.files.maxPartitionBytes 512M
spark.rapids.sql.concurrentGpuTasks 2
```

Advanced Options - Init Scripts

Spark [Tags](#) [Logging](#) [Init Scripts](#) [Permissions](#)

Init scripts [?](#)

	Type	File path	
	DBFS	dbfs:/databricks/init_scripts/init.sh	x

```
#!/bin/bash
sudo wget -O /databricks/jars/rapids-4-spark_2.12-23.04.0.jar
https://repo1.maven.org/maven2/com/nvidia/rapids-4-spark_2.12/23.04.0/rapids-4-spark_2.12-23.04.0.jar
```

Comparing Query Plans

```
SELECT id1, sum(counter)
FROM BaseView
WHERE SpecialText rlike '{TextSearch}'
GROUP BY id1
```

CPU Plan

```

== Physical Plan ==
AdaptiveSparkPlan (22)
+- == Final Plan ==
   * HashAggregate (13)
   +- ShuffleQueryStage (12)
      +- Exchange (11)
         +- * HashAggregate (10)
            +- * HashAggregate (9)
               +- AQEShuffleRead (8)
                  +- ShuffleQueryStage (7)
                     +- Exchange (6)
                        +- * HashAggregate (5)
                           +- * Project (4)
                              +- * Filter (3)
                                 +- * ColumnarToRow (2)
                                    +- Scan parquet (1)

```

GPU Plan

```

== Physical Plan ==
  GpuColumnarToRow (13)
    +- GpuHashAggregate (12)
      +- GpuShuffleCoalesce (11)
        +- GpuColumnarExchange (10)
          +- GpuHashAggregate (9)
            +- GpuHashAggregate (8)
              +- GpuShuffleCoalesce (7)
                +- GpuColumnarExchange (6)
                  +- GpuHashAggregate (5)
                    +- GpuProject (4)
                      +- GpuCoalesceBatches (3)
                        +- GpuFilter (2)
                          +- GpuScan parquet (1)

```

Compute Cost Calculation

VM Cost = Cost Per VM x Number of Nodes

DBU Cost = DBU x DBU Price x Number of Nodes

Compute Cost = VM Cost + DBU Cost

Databricks Cluster Creation

Node type ⓘ

Standard_D8s_v3 32 GB Memory, 8 Cores ▼ ⓘ

DBU / hour: 1.5 ⓘ

Standard_D8s_v3

Databricks Website

Workload	DBU prices—standard tier
All-Purpose Compute**	\$0.40/DBU-hour
Jobs Compute**	\$0.15/DBU-hour

Microsoft Azure Website

Instance	vCPU(s)	RAM	Temporary storage	Pay as you go
D2s v3	2	8 GiB	16 GiB	\$0.096/hour
D4s v3	4	16 GiB	32 GiB	\$0.192/hour
D8s v3	8	32 GiB	64 GiB	\$0.384/hour

CPU-GPU Conversion Actual Saving

Job Name	Contact	Frequency	CPU Cost (\$)	GPU Cost (\$)	Saving (\$)	Saving (%)	Multiplier	Yearly Saving (\$)
Job ABC	xx1234	Hourly	7.64	2.96	4.68	61%	8760	41,039
Job EFG	xx5678	Hourly	5.59	1.34	4.25	76%	8760	37,195
...								
Job XXX	xx1111	Daily	30.57	8.65	21.92	72%	365	8,000
Job ZZZ	zz3333	Daily	31.75	19.90	11.85	37%	365	4,326
...								

Key Takeaways

- Qualification Tool
 - Useful for identifying candidate for Spark RAPIDS
 - Can get estimation for cost saving
- A recommendation status doesn't mean actual cost saving
- Focus on the jobs with high execution frequency (ie: hourly and daily)

What's Next for AT&T

- **Automate Cluster Configuration, Adjustment, Rollback**
 - Optimize CPU
 - Optimize GPU
 - Optimize Photon
- **Automate Tracking**
 - Compute Costs
 - Savings

NVIDIA RAPIDS Accelerator for Apache Spark

Enterprise Support available with license purchase

5x

Faster Execution Time

- Take advantage of faster analytics
- Accelerate AI pipelines

4x

Lower Costs

- Save on cloud usage costs
- Reduce power consumption and carbon footprint



Full Enterprise Support

NVIDIA AI Enterprise 3.1 offers

- Mission critical support
- Bug fixes
- Professional services



Amazon EMR



Google Cloud
Dataproc



aws

Microsoft
Azure

 databricks

nvidia.com/spark



