GPU Accelerated High Performance Machine Learning Pipeline

Adobe Data Science and Production Team
Presenter: Lei Zhang

Agenda

- Adobe AI Services overview
- Machine Learning Algorithms on GPU
  - (Deep) Neural Networks
  - XGBoost
- GPU Based ETL, with Apache Spark 3.0 and Rapids
  - Introduction on RAPIDS
  - RAPIDS on SPARK
  - RAPIDS-accelerated Spark shuffles
  - Performance tuning
- Benchmark Results
- Conclusion

Adobe AI Services

AI-as-a-Service designed for marketers to power better customer experiences

Attribution

Understand incremental impact of every customer interaction

Customer Propensity

Deliver insights about each individual customer

Journeys

Optimize design & delivery of customer journeys

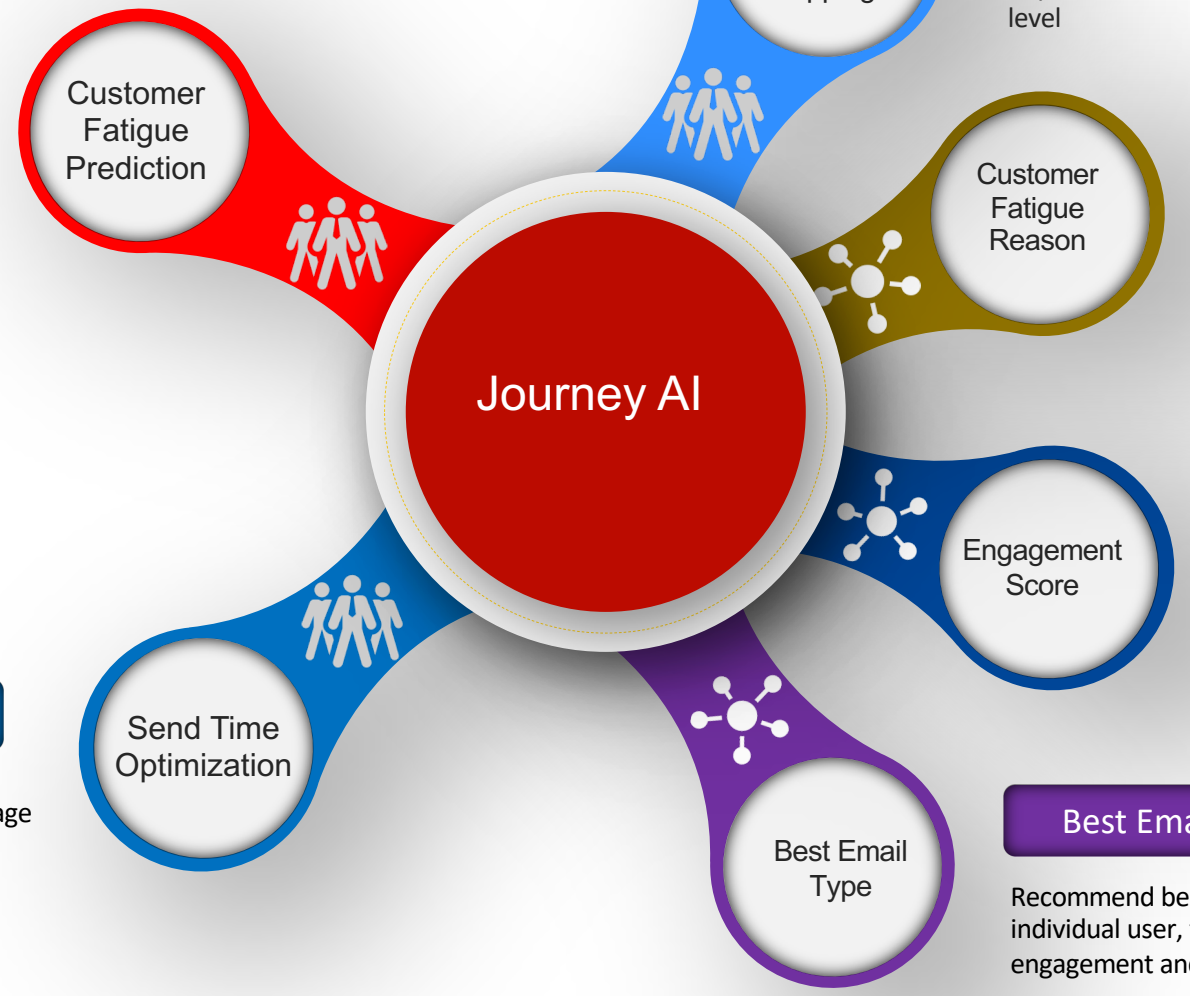
Content

Creation, selection and delivery of the most relevant content

Lead Scoring

Identify the most qualified leads for B2B

Journey AI



Delivery Frequency

Recommendation of max number of emails that should be sent to customer in the next x number of days to maintain low fatigue risk level

Customer Fatigue

Predict probability of user unsubscribe after sending email given email opened

Customer Segmentation

Reason why fatigue model thinks the user's un-subscription risk is high

Engagement Score

Predict probability of user engaging with email within the next 7 days after sending email

Best Send Time

Send emails at the time that a customer is most likely to engage

Best Email Type

Recommend best email type for individual user, to promote engagement and conversion

Adobe

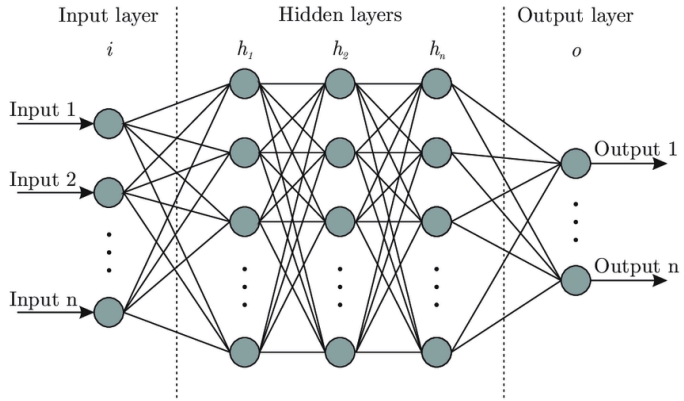
Machine Learning Algorithms on GPU

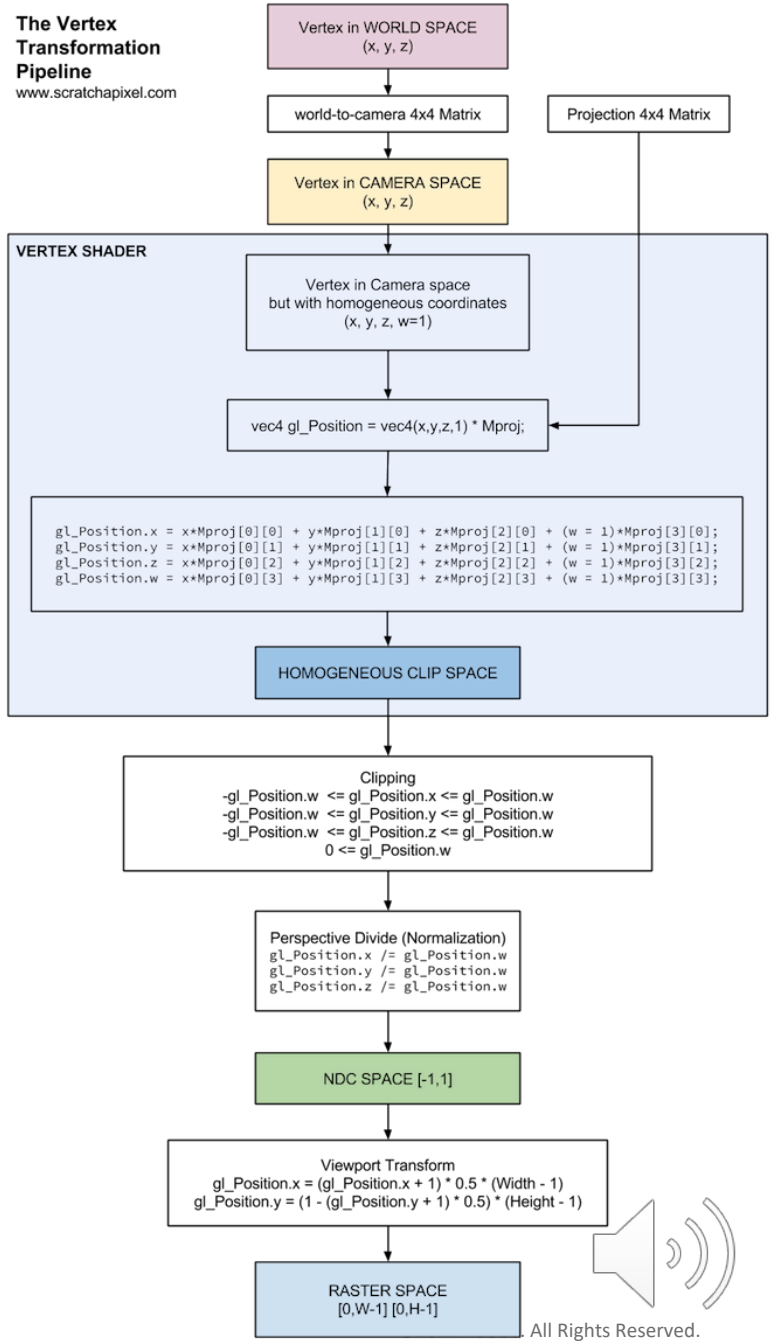
Neural Networks on GPU

Why can GPU do neural network training and scoring?

GPU is very good at big matrix multiplication

- GPU is designed to 3D transformations all the time
- 3D transformations are a series of matrix multiplication tasks
- Matrix multiplication tasks get fully hardware acceleration





XGBoost on GPU

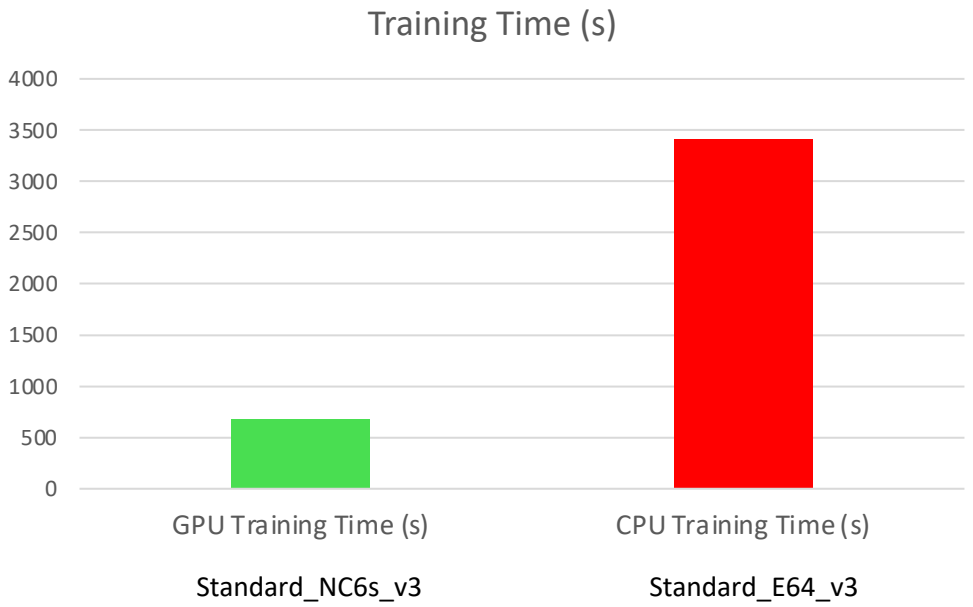
- Histogram method on GPU
  - Find quantiles over the input feature space and discretize the training examples into this space. This is fully implemented on GPU.
  - After histograms are built for each feature, the later tree build process is not required to go over every training sample.
  - It is an approximate, but efficient solution.
- Gradient calculation algorithms
  - Boost trees predictions are made on GPU, during each iteration.
- Memory Efficiency

- **Further Reading**: Rory Mitchell, Andrey Adinets, Thejaswi Rao: "XGBoost: Scalable GPU Accelerated Learning", 2018; http://arxiv.org/abs/1806.11248.

CPU and GPU Cost/Performance Evaluation

Training data: 2.69GB Parquet (20.8GB CSV)

Node Type	Node Name	Price per hour	Description
GPU	Standard_NC6s_v3	$5.81	6 Cores, 1 NVIDIA Tesla V100 GPU, 112GB memory
CPU	Standard_E64_v3	$12.43	64 Cores, 432GB memory
CPU	Standard_L64_v2	$13.79	64 Cores, 512GB memory



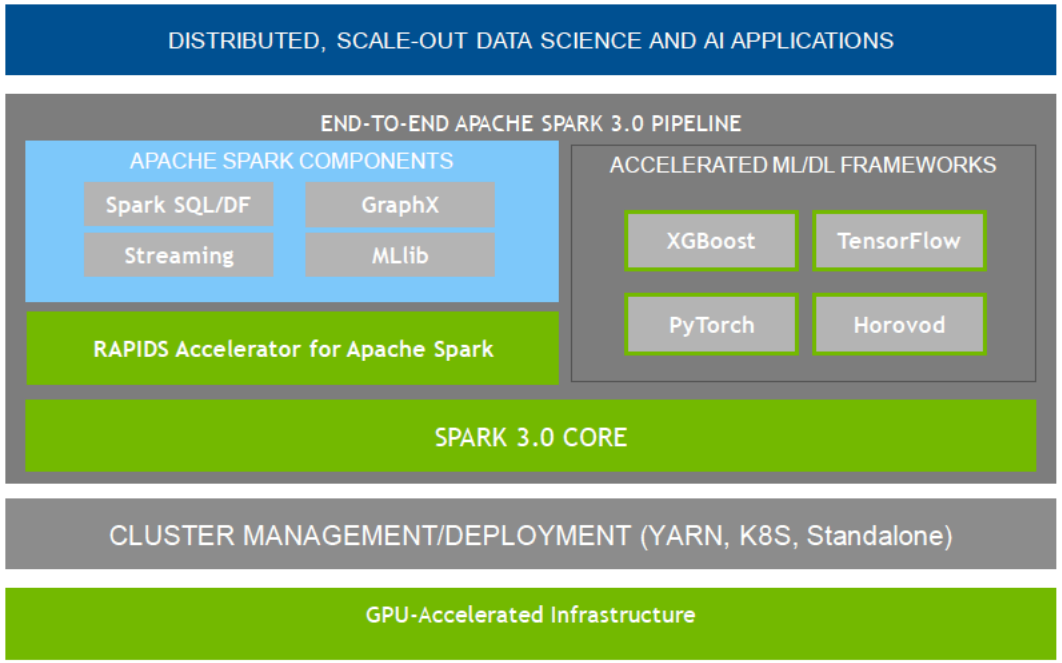
Data size (Parquet)	Data size (CSV)	GPU Training Time (s)	CPU Training Time (s)	GPU Run Cost	CPU Run Cost	Speed Up	Cost Saving	Relative Cost Saving
2.69GB	20.8GB	681	3410	$1.10	$11.77	5.01	$10.67	90.67%

GPU Based ETL with Apache Spark 3.0 and Rapids

RAPIDS on SPARK 3.0

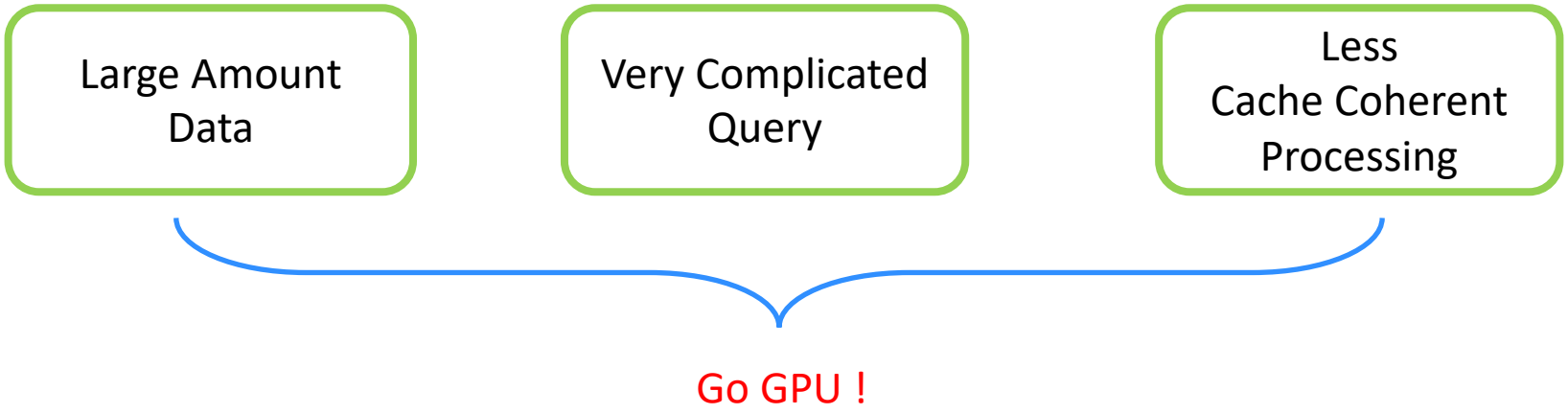
For Apache Spark 3.0, new RAPIDS APIs are used by Spark SQL and DataFrames for GPU-accelerated memory-efficient columnar data processing and query plans. When a Spark query executes, it goes through the following steps:

1. Creating a logical plan
2. Transforming the logical plan to a physical plan by the Catalyst query optimizer
3. Generating code
4. Executing the tasks on a cluster

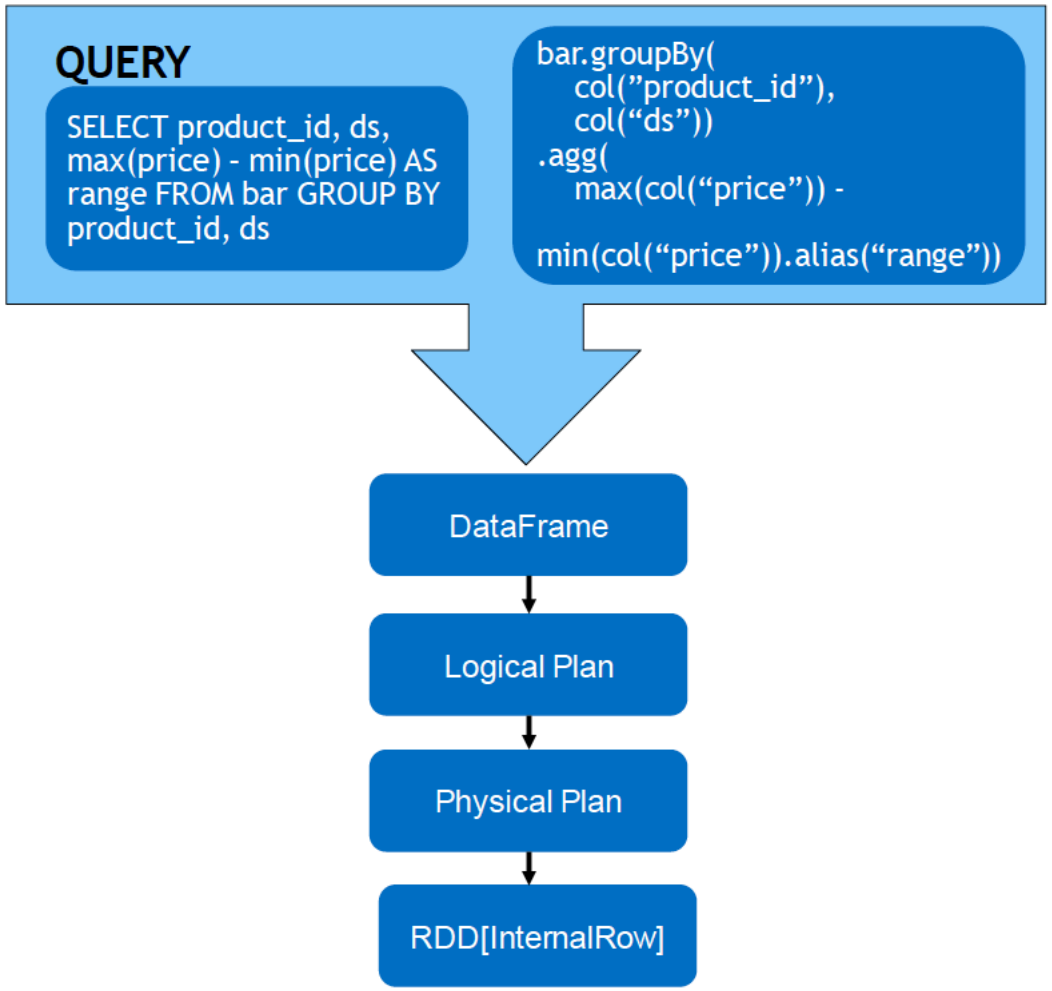


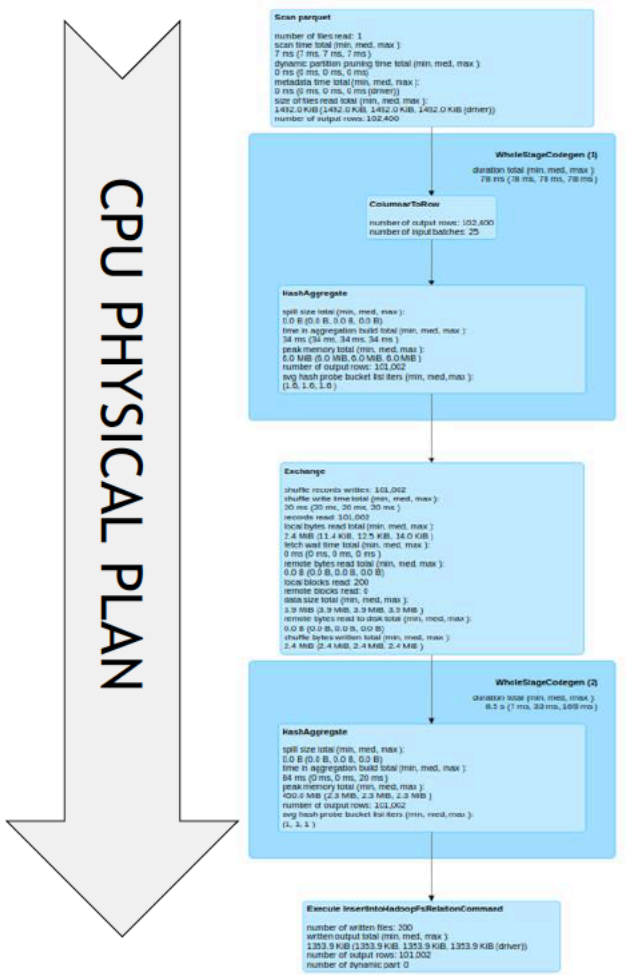
Why use GPU for Query Tasks?

- GPU excels at the following query tasks:
  - High cardinality joins
  - High cardinality aggregates
  - High cardinality sort
  - Window operations (especially on large windows)
  - Complicated processing
  - Transcoding (Writing Parquet and ORC is hard, reading CSV is hard)

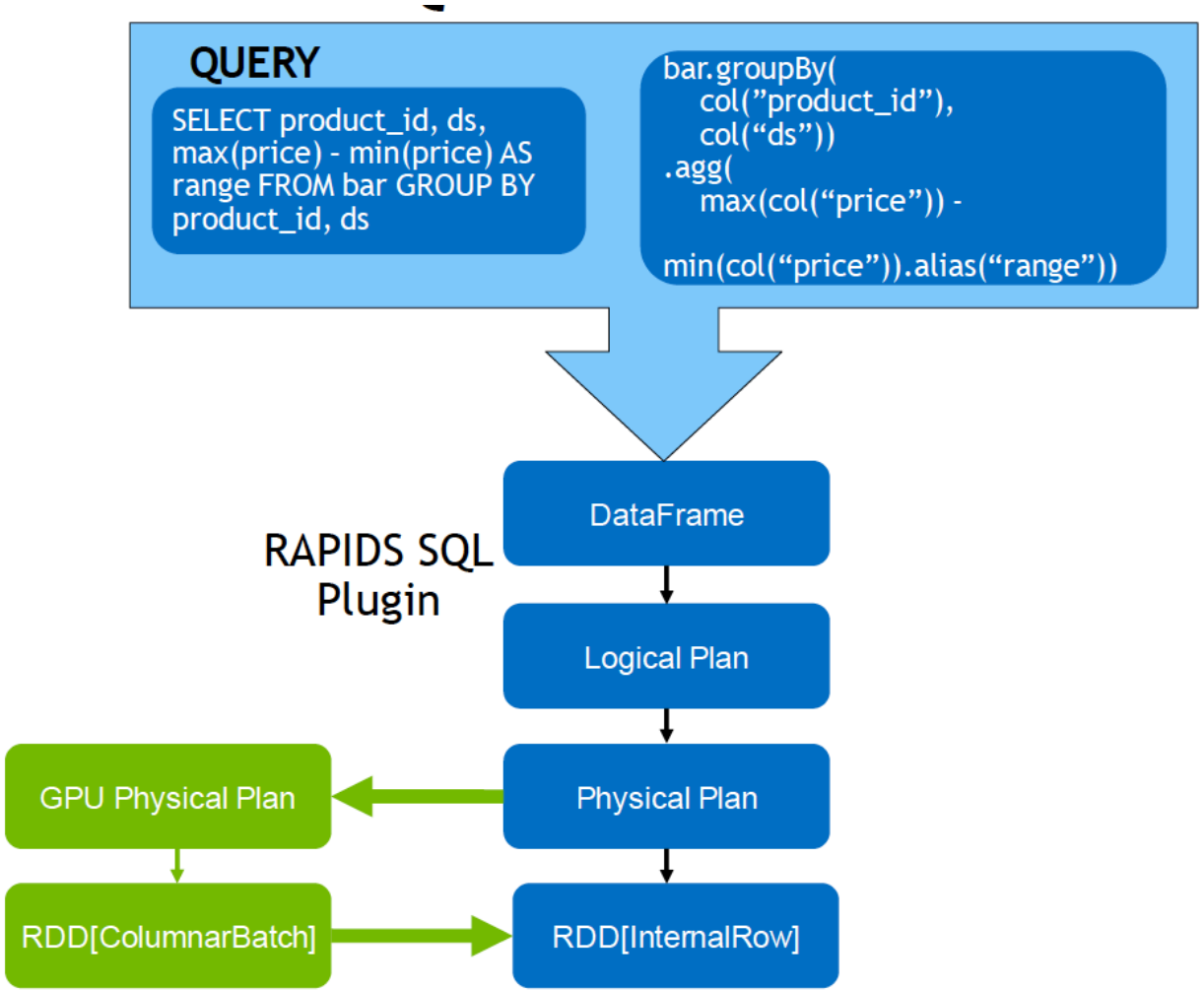


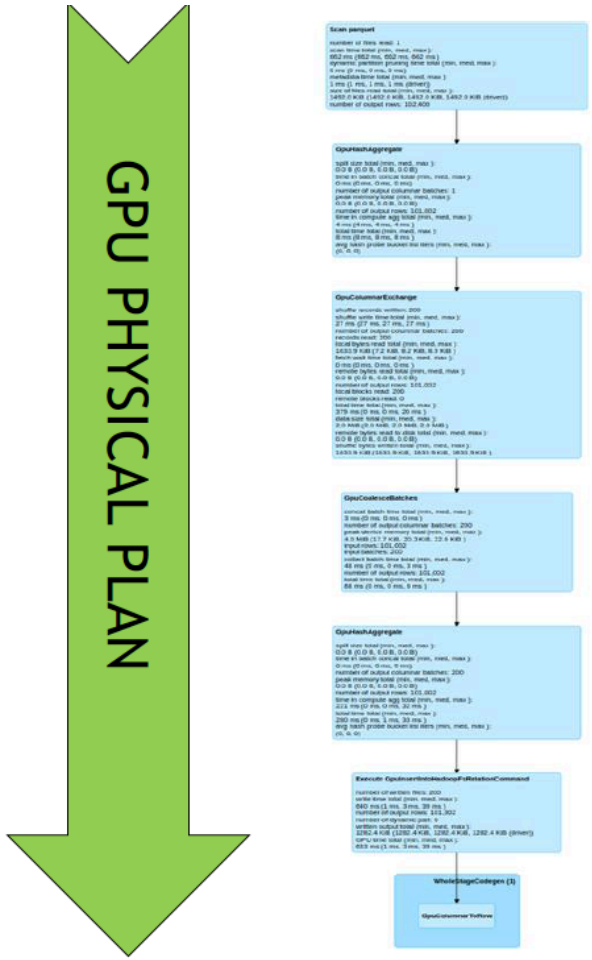
Spark SQL & Dataframe Compilation Flow





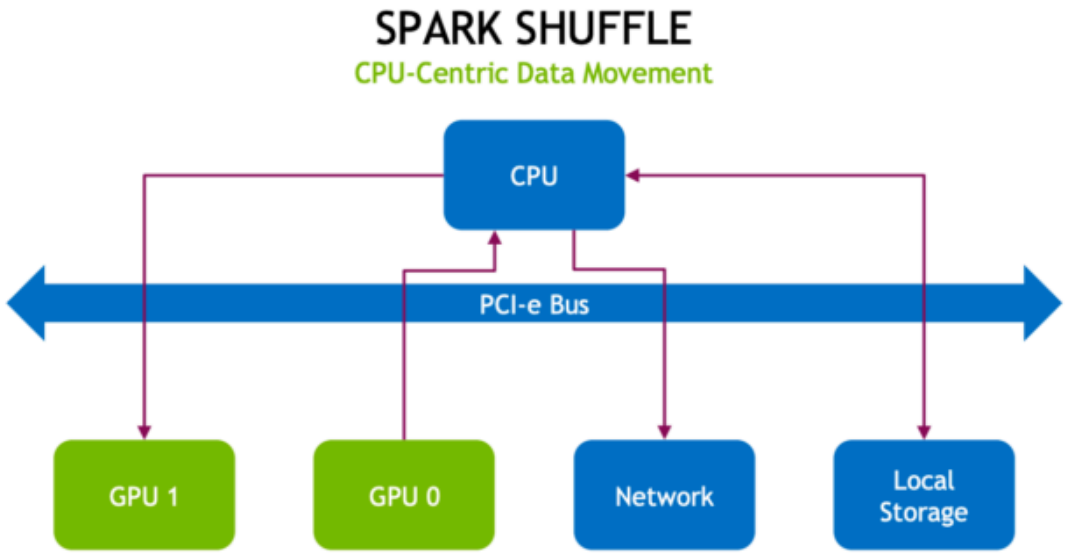
Spark SQL & Dataframe Compilation Flow with RAPIDS Plugin

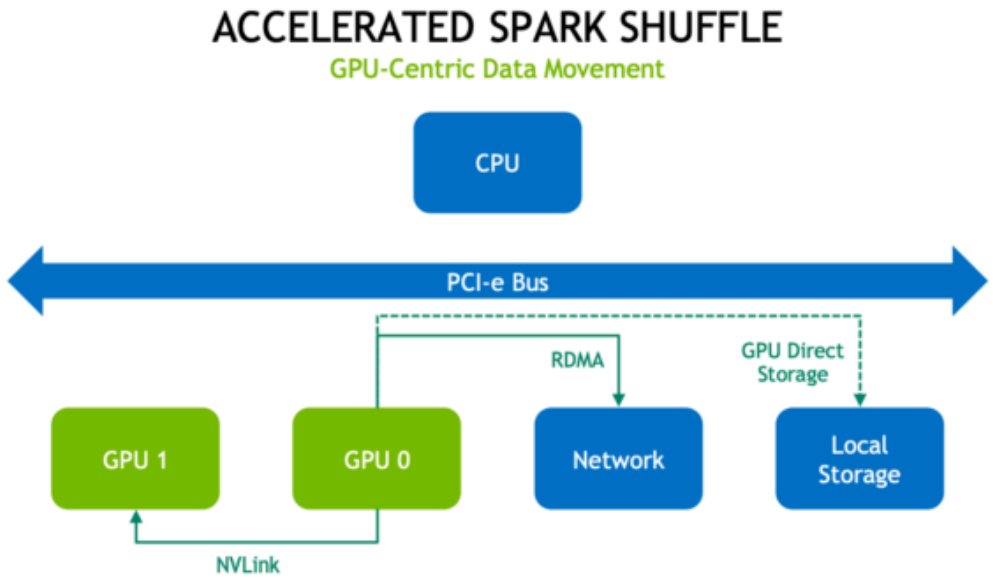




Although underlining mechanism is different,
the query is the same as before

RAPIDS-accelerated Spark shuffles





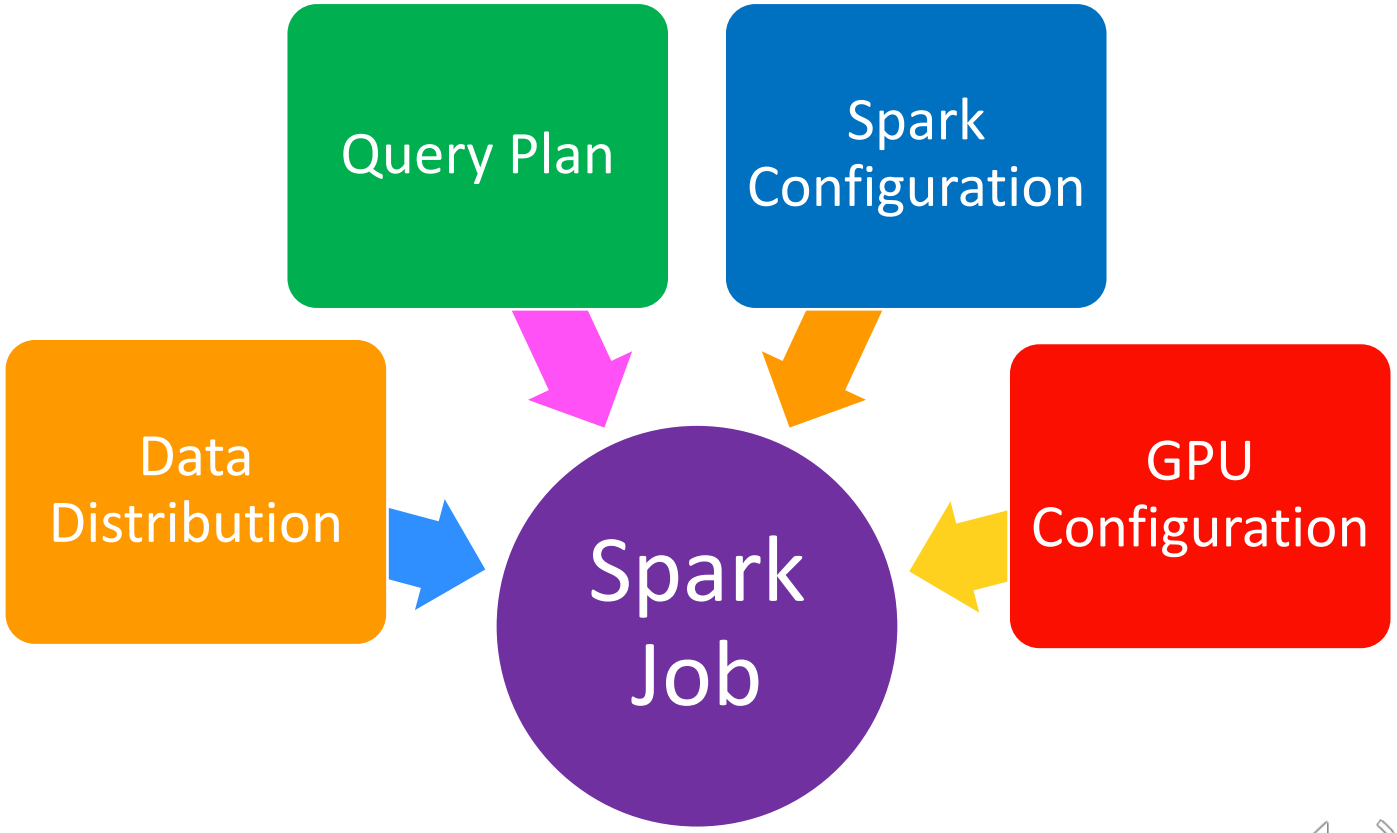
- Data moves to CPU for shuffle
- Data is copied from GPU memory to CPU memory
- PCI-e Bus may become bottleneck

- Data directly moves from GPU to other destination
- No CPU involved
- PCI-e Bus traffic is greatly alleviated

Performance Tuning

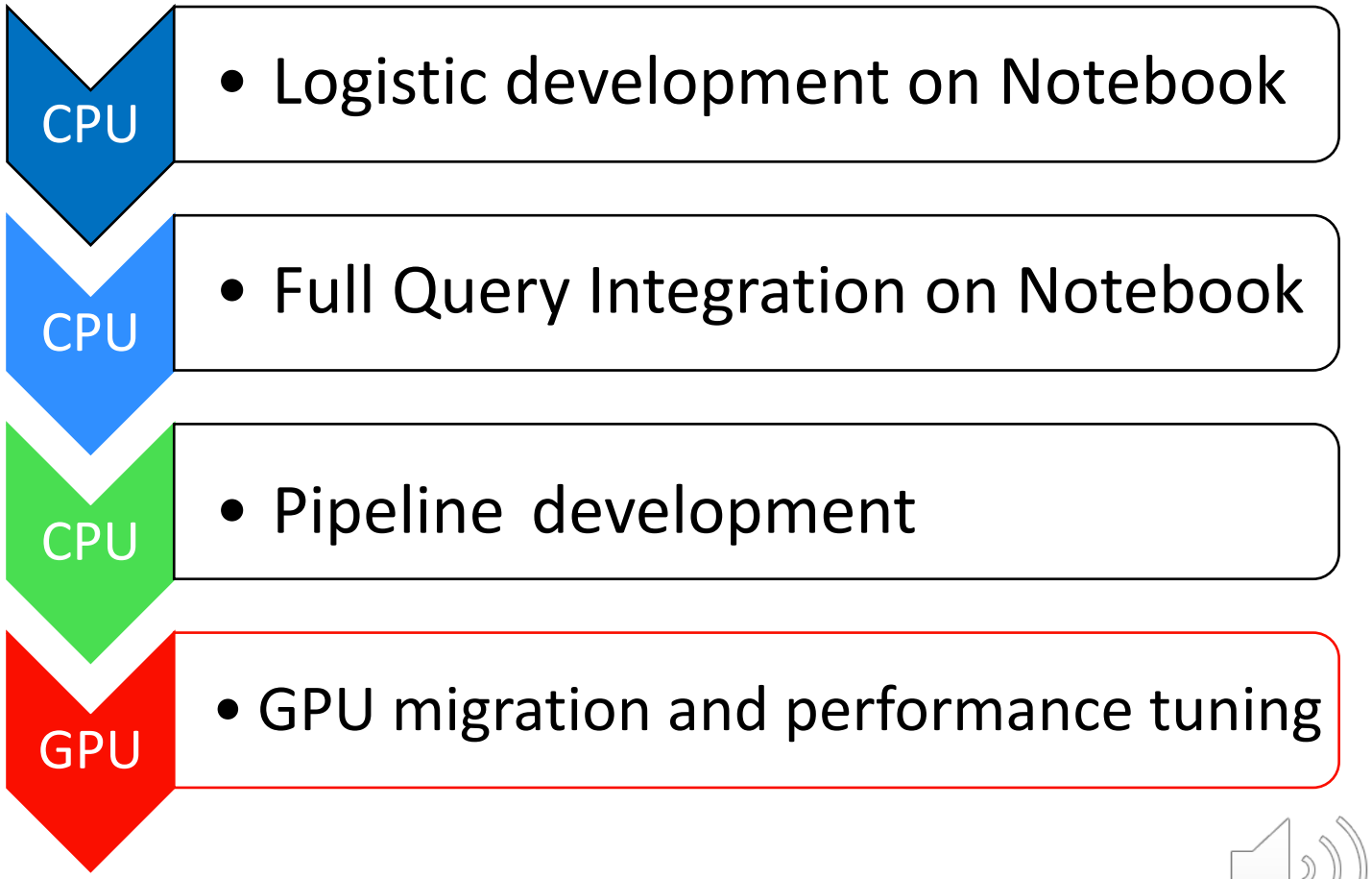
Your Spark job performance depends on:

1. Data Distribution (Skew)
2. Query Plan
3. Spark Configuration
4. GPU Configuration



Development Procedure

1. Start with CPU cluster for logic development
2. Migrate to GPU cluster for performance tuning



RAPIDS Current Progress

- RAPIDS supports many SQL syntaxes, but not all. If RAPIDS does not support a syntax:
  - Send data back to CPU to process
  - Get result back to GPU for the next tasks
- Performance may be affected depending on the specific task.
- About Spark UDF (under evaluation)
  - RAPIDS support basic Spark UDF
  - For complicated UDF, it is possible to use Pandas UDF to operate on CUDF

Benchmark Results

Evaluation Environment

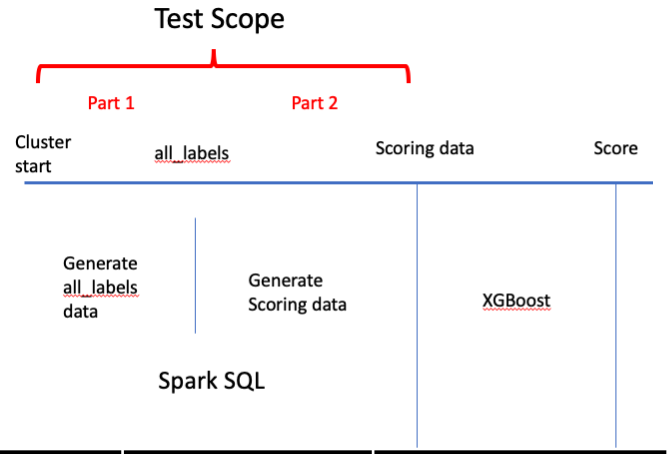
- Two types of workers in Azure were used for the tests:

	Node Name	CPU Cores	Memory	GPU	1 Year Reserved Cost
CPU Worker	Standard_L4	4	32 GB	-	$ 0.853
GPU Worker	Standard NC6s_v3	6	112 GB	1 (TESLA V100)	$ 4.7

- Evaluation Goal
  - Evaluate Cluster Cost and Run time*

* Note: driver cost not included

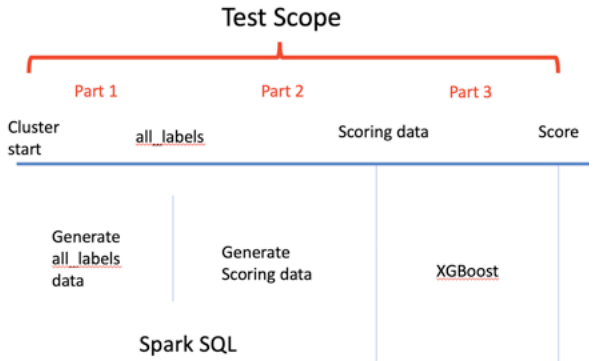
Benchmark Test 1



Pure Spark Query Test

Data Size	Customer	Cluster Type	Cluster Size	Run Time (s)	Cost ($)	Speed Up Ratio	Time Saving %	Cost Saving %
Small size	Customer 1	GPU worker	1	193	0.251972222	1.367875648	26.89%	19.44%
		CPU workers	5	264	0.312766667			
	NVIDIA	GPU worker	1	332	0.433444444	1.668674699	40.07%	33.96%
		CPU workers	5	554	0.656336111			
Medium Size	Customer 3	GPU workers	2	370	0.966111111	2.281081081	56.16%	51.69%
		CPU workers	10	844	1.999811111			
	Customer 4	GPU workers	2	419	1.094055556	2.618138425	61.80%	57.91%
		CPU workers	10	1097	2.599280556			
	Customer 5	GPU workers	2	1225	3.198611111	1.486530612	32.73%	25.87%
		CPU workers	10	1821	4.314758333			
Large Size	Customer 6	GPU workers	5	1309	8.544861111	3.07868602	67.52%	64.21%
		CPU workers	25	4030	23.87215278			

Benchmark Test 2

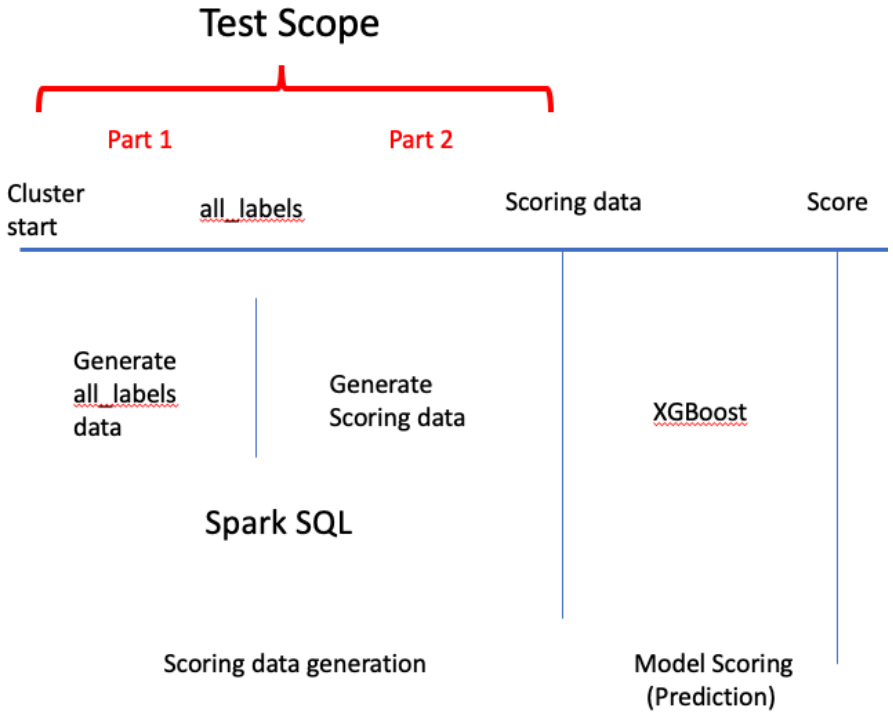


Spark Query Test + XGBoost Model Scoring, All on GPU, in one pass

Data Size	Customer	Cluster Type	Cluster Size	Run Time (s)	Cost ($)	Speed Up Ratio	Time Saving %	Cost Saving %
Small size	Customer 1	GPU worker	1	264	0.34466667	1.32575758	24.57%	16.88%
		CPU workers	5	350	0.41465278			
	NVIDIA	GPU worker	1	539	0.70369444	1.57884972	36.66%	30.20%
		CPU workers	5	851	1.00819861			
Medium Size	Customer 3	GPU workers	2	323	0.84338889	2.11764706	52.78%	47.96%
		CPU workers	10	684	1.6207			
	Customer 4	GPU workers	2	433	1.13061111	2.37413395	57.88%	53.58%
		CPU workers	10	1028	2.43578889			
	Customer 5	GPU workers	2	1170	3.055	1.44444444	30.77%	23.71%
		CPU workers	10	1690	4.00436111			
Large Size	Customer 6	GPU workers	5	1621	10.5815278	2.6668723	62.50%	58.68%
		CPU workers	25	4323	25.6077708			

Benchmark Test 3

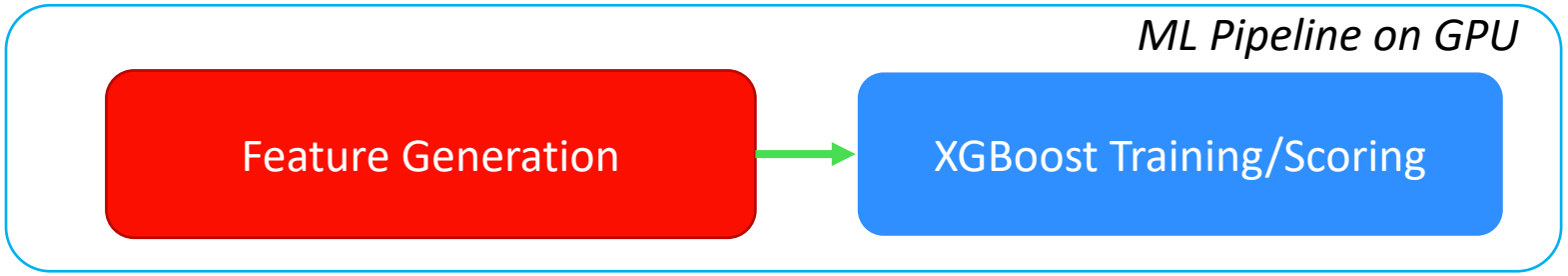
- Our Largest Customer - Adobe
  - 5 data sources
  - 2.88 TB data in total
  - More complicated join and aggregation
  - Real production pipeline



Customer	Cluster Type	Cluster Size	Run Time (s)	Cost ($)	Speed Up Ratio	Time Saving %	Cost Saving %
Adobe	GPU workers	20	2692	70.29	2.20653789	54.68%	50.06%
	CPU workers	100	5940	140.75			

Adobe is the Pioneer

- First to implement GPU based machine learning pipeline
  - Successfully proved the concept that running feature generation and model scoring on GPU in one pass
  - No intermedium results saved



- First to processed 2.88 TB data on GPU in real-world application

Conclusion

1. Some machine learning algorithms running on GPU see significantly faster speeds on training and prediction tasks
2. RAPIDS uses GPU to accelerate data science related tasks
3. GPU-based machine learning pipeline Spark + XGBoost is the future for handling large amount of data with complicated Spark and ML training tasks
4. Benchmark tests show significant performance improvement and cost reduction

Q & A

This project is a collaboration with the NVIDIA [illegible]rk Team

Adobe