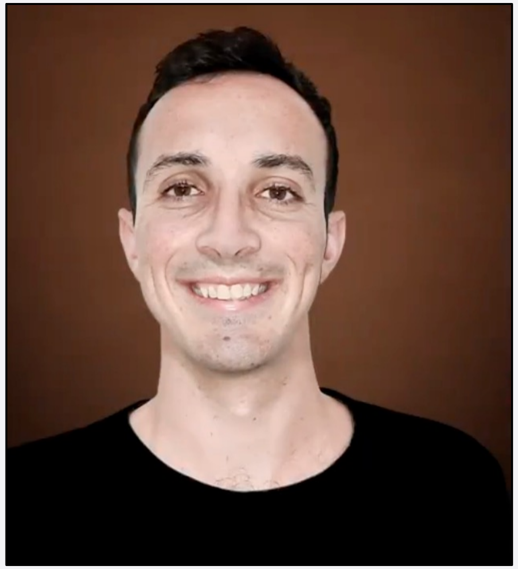
Big Data Processing with GPUs

Apache Spark 3 and GPUs to Reduce Cloud Cost by up to 70%

March 2024

Presenter:
Ilay Chen, PayPal

Experienced in designing, developing, and optimizing Big Data infrastructures and Machine Learning solutions



Overview

- Big Data and ML in PayPal
- Spark RAPIDS Introduction
- Running Spark with GPUs
- Cost Comparison
- Learnings and Actionable

Big Data and ML in PayPal

HUGE Scale:

- 430+ million active users
- 25+ billion transactions every year
- All kinds of valuable data

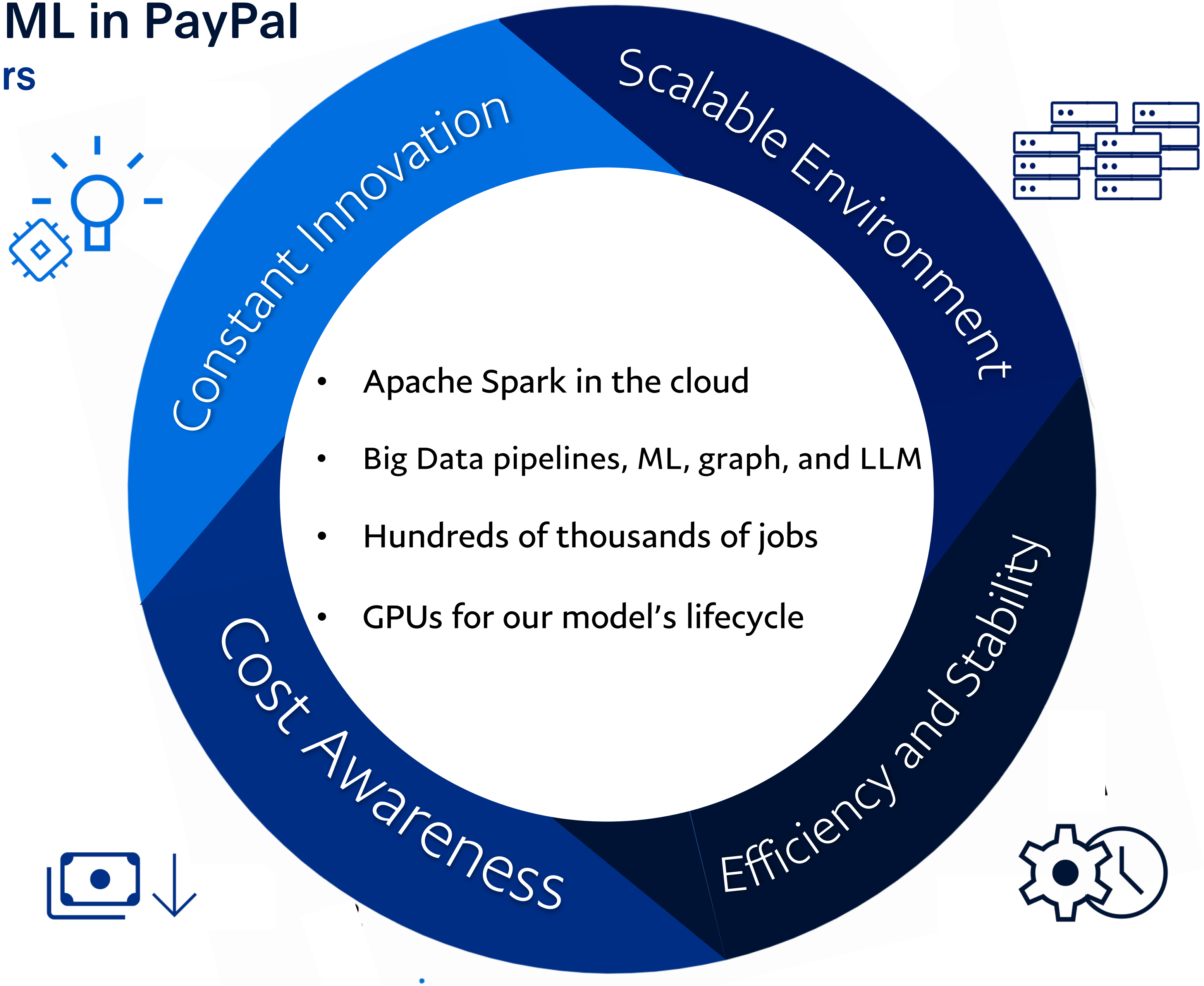
ML Across ALL PayPal’s Products and Domains:

- Fraud Detection
- Recommendation Systems
- Risk
- Credit
- Customer Support

And much more!

Big Data and ML in PayPal

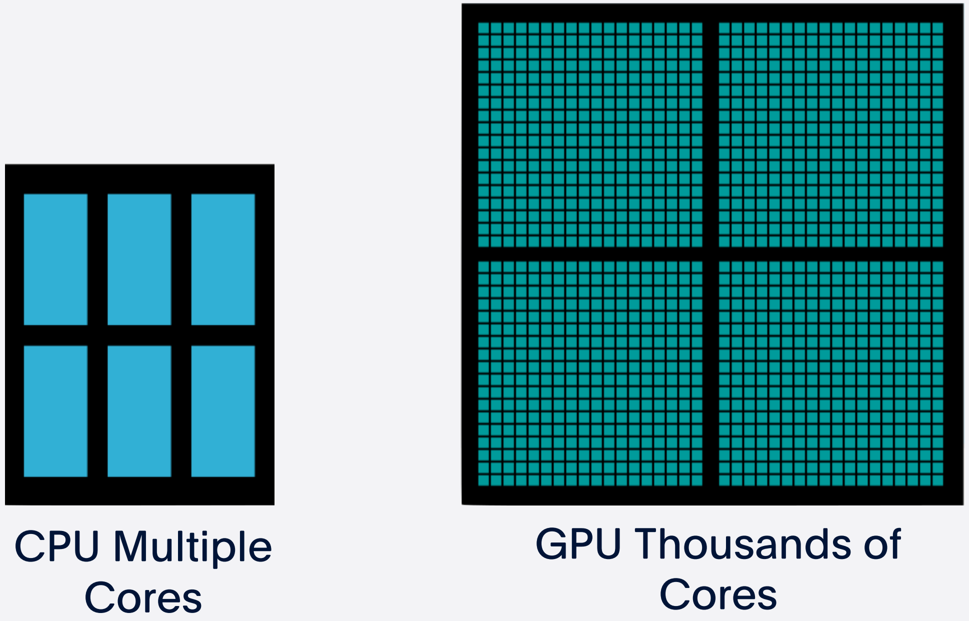
Tech Stack
Production Pillars



- Apache Spark in the cloud
- Big Data pipelines, ML, graph, and LLM
- Hundreds of thousands of jobs
- GPUs for our model's lifecycle

Spark RAPIDS Introduction
CPU vs GPU - Quick Alignment

- Computation wise
  - A few strong cores vs thousands of cores
  - Keyword - parallelism
- GPUs common use cases: AI, Crypto, Graphics applications and more
- Industry's standard for data processing workloads is to utilize CPUs



Can GPUs reduce big data processing costs?

In many cases

It depends

YES!

Spark RAPIDS Introduction

Project's Overview:

- NVIDIA's open-source project
- Apache Spark 3 accelerator that leverages GPUs
- PayPal focused on the cost reduction potential

Ease of Use:

- Can seamlessly run Python/Scala/Java/SQL code on GPU
- No code changes required
- Config Spark RAPIDS plugin

What Parts Do GPUs Accelerate Well?

- Large aggregations, joins, sorts
- Window operations (Data deduplication)
- Encoding Parquet
- Shuffle stages
- And more..

Spark RAPIDS Introduction

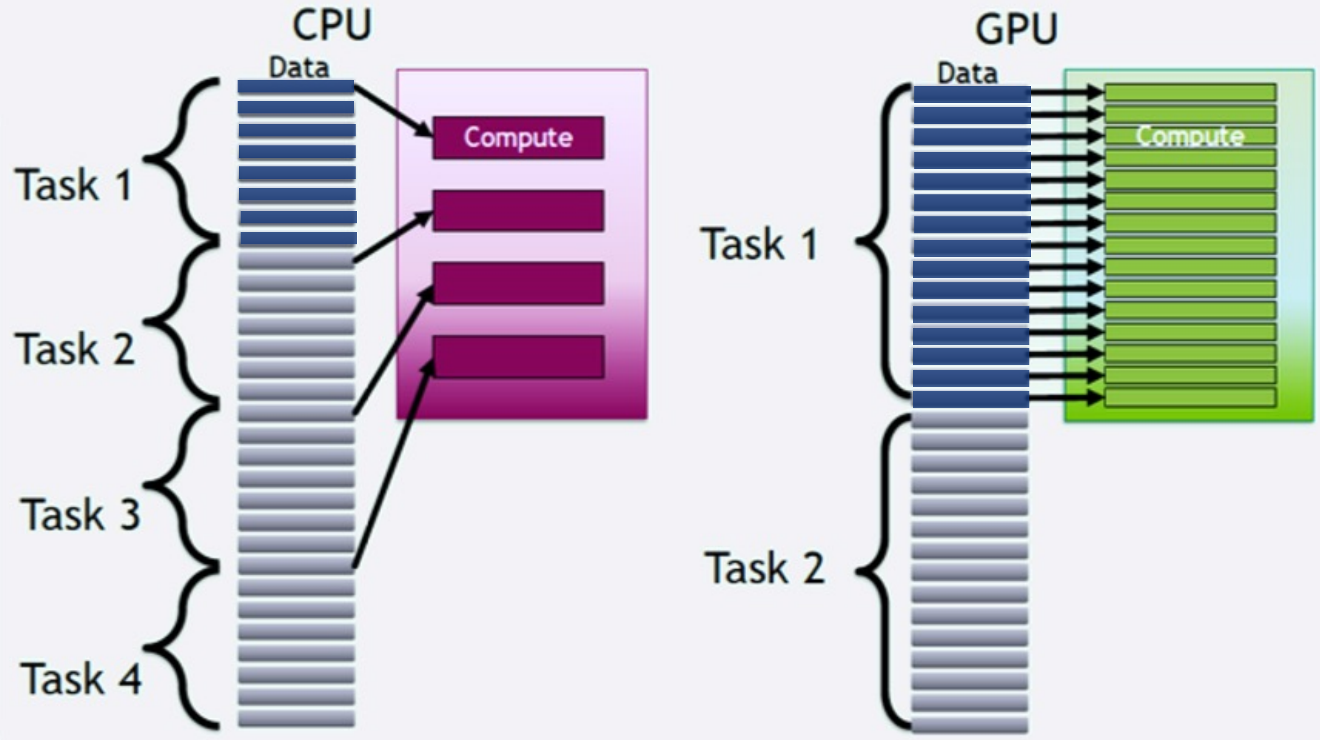
Design Changes - Under the Hood

Apache Spark - Batch Processing:

- Data in each stage is divided into tasks

How the Data is Being Processed?

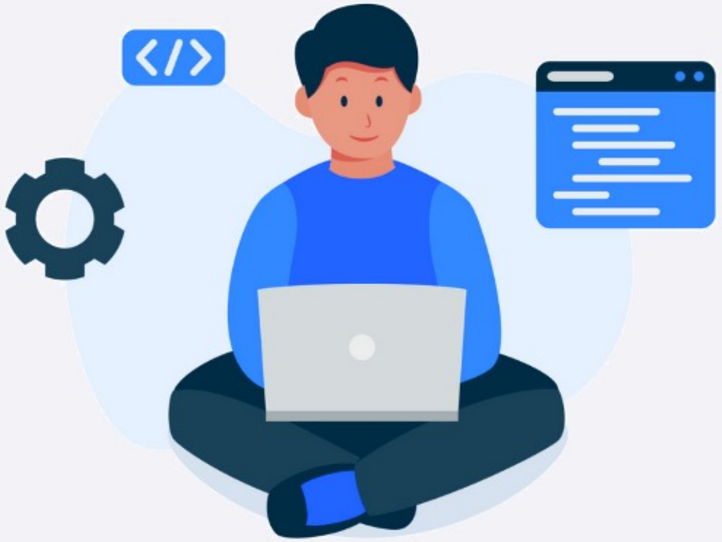
- Task level parallelism vs Data level parallelism
- Computation bound vs I/O bound
- The motivation of working with large partitions



Running Spark with GPUs
Getting Started

1. Experimenting in Research Environment
- Tried some common heavy functions (window – dedup, JOIN, GROUP BY, sort, read/write and more)

2. Production Job Characteristics
- Consumes lots of data
  - Heavy shuffle, large JOINs, GROUP BYs, and more
- NVIDIA's Spark Qualification tool



3. Upgrading to Spark 3

Running Spark with GPUs
Tuning Spark 3

Description	Duration	Tasks: Succeeded/Total	Input	Shuffle Write
Saving stats output.entity_map.cnt sql at ZonkeyContext.scala:778	1.0 h	185497/185497	9.5 TB	3.6 TB

Description	Duration	Tasks: Succeeded/Total	Input	Shuffle Write
Saving stats output.entity_map.cnt sql at ZonkeyContext.scala:798	40 min	10291/10291	9.5 TiB	3.3 TiB

AQE:

- New optimization technique in Spark 3
- Default is 128 MB, we changed it to 1 GB

Changing the Input Partitions Size:

- Default it 128 MB, we changed it to 2 GB

Intermediate Results:

- We made our job to be computation bound rather than I/O bound
- 30% less machines and 25% overall runtime reduction
- Resulting in ~45% cost reduction

Running Spark with GPUs

1. Infrastructure Update

- Support GPU parameters of our cloud vendor
- Enable Spark RAPIDS plugin

2. Running our Candidate Job

- Encountered a few runtime errors
    - For example:

RMM failure at: arena.hpp:382: Maximum pool size exceeded

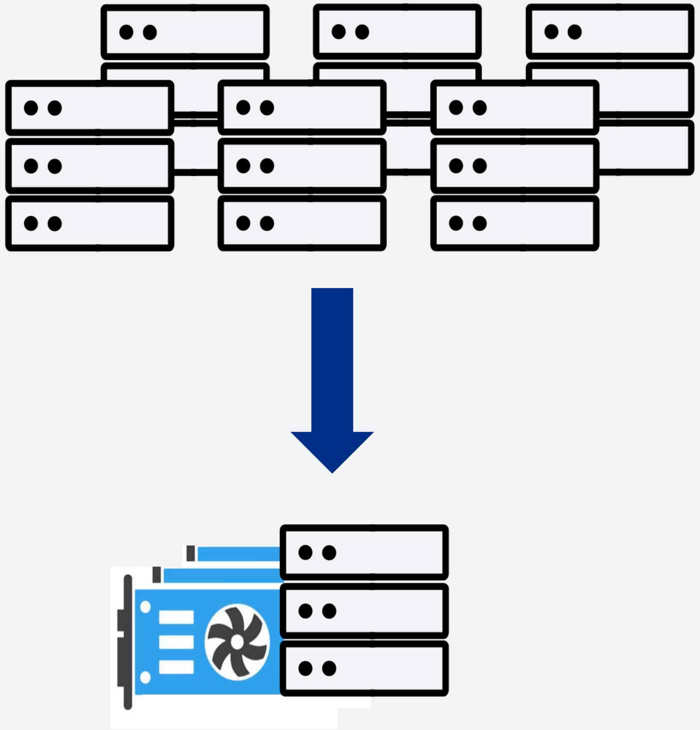
- The Solution:
  Finding the sweet spot of “spark.rapids.sql.concurrentGpuTasks”

Running Spark with GPUs
Cost Optimization

The Key: Reducing Machines Count: 140 ➔ 30

- Reduce machines count until reaching fair utilization
- Eventually, we encountered:

java.io.IOException: No space left on device



- **The Solution:**
- Increasing the disk counts per node: 4 ➔ 8
  - Also Improves I\O performance
  - Relatively cheap in cloud vendors
  - Configuring NVME protocol

Cost Comparison

	Spark 2 - Baseline	Spark 3 with AQE	Spark 3 with GPUs
# of Machines	140	100	30
Machines Type	16 cores, 100GB	16 cores, 100GB	32 cores, 120GB
GPU			2x Tesla T4
Local SSDs	4x 375GB (SCSI)	4x 375GB (SCSI)	8x 375GB (NVME)
Runtime	2 hours	1.6 hours	1.3 hours
Cost Percentage (vs Baseline)	100%	~55% (45% cost savings)	~30% (70% cost savings!)

*The price of the machine's hardware is factored into the cost calculation

Learnings and Actionable
Spark RAPIDS Optimization

- Choosing the Right GPU
  - NVIDIA's Tesla T4 (L4 should give better results)
- Considering Memory Overhead
  - The executor's memory was 16 GB,
    we set the memory overhead to 16 GB too
- Auto-scaling GPU cluster

Scan the QR code to read more

Thank You!

Questions?

Scan the QR code to read more