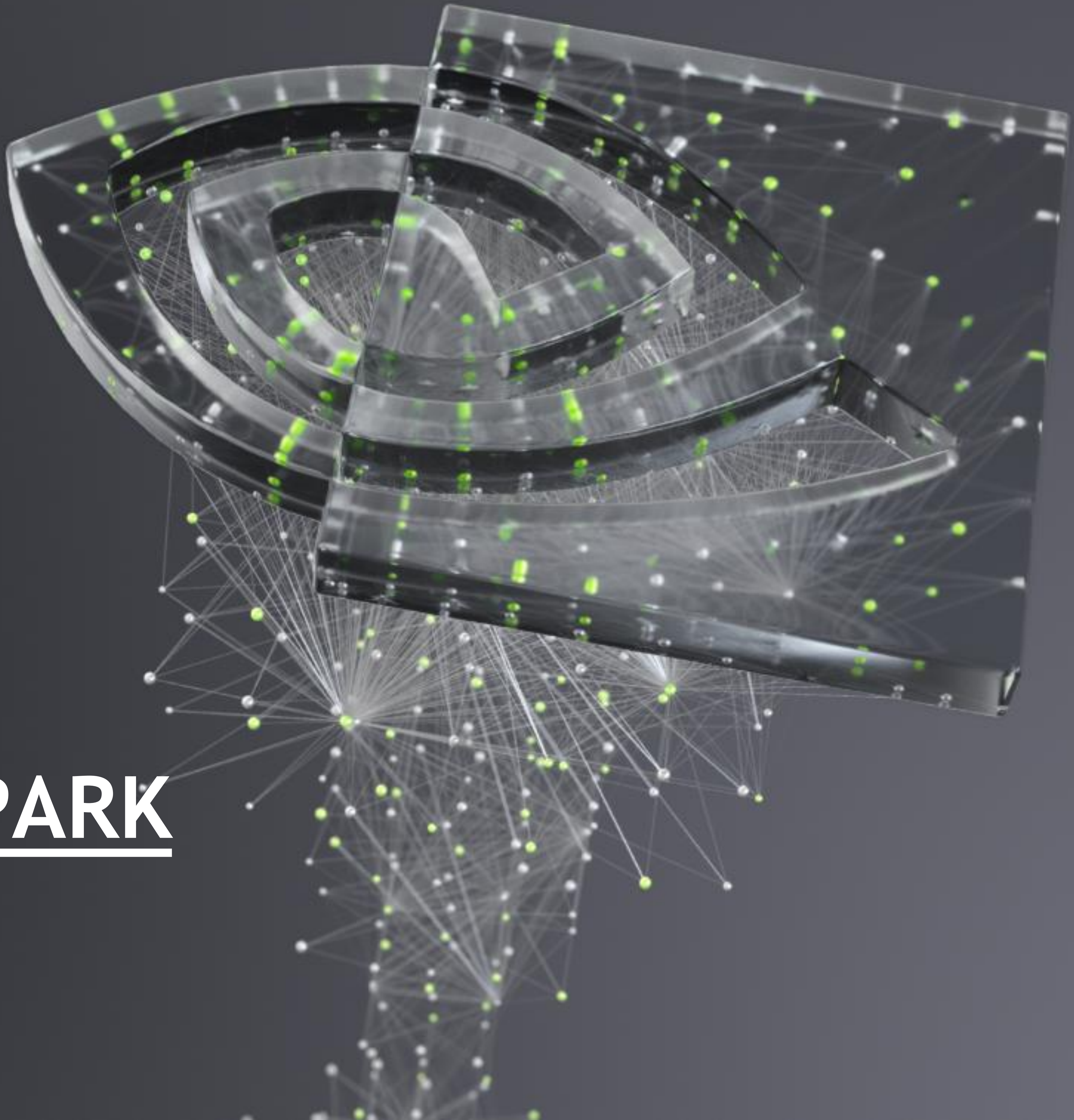


UCX FOR APACHE SPARK

Peter Rudenko (prudenko@nvidia.com)

November 2020

APACHE SPARK

Leading Framework for Distributed, Scale-Out Data Analytics

100s of 1000s of data scientists and over 16,000 enterprises use Spark

Spark is 100x faster at processing data than Hadoop

1000+ contributors across 250+ companies

Databricks platform alone spins up 1 million virtual machines per day

Examples of Increased Demand for AI-Driven Services and Analytics

2B Digital Buyers > All Want the Better Product at a Lower Price

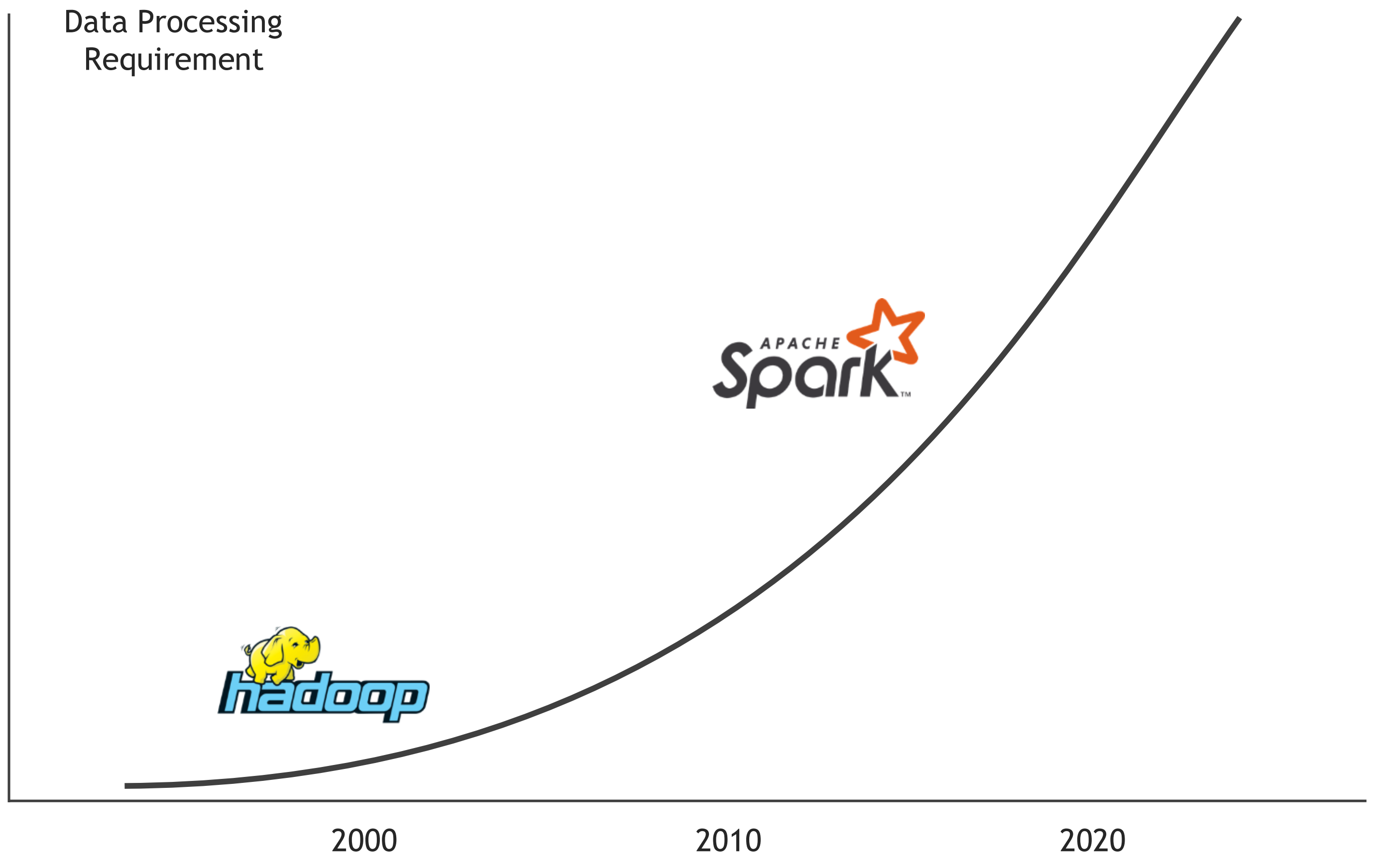
>1M Known Asteroids and Comets > Understand Where They're Going and When

500M Esports Viewers (Growing 20% YoY) > How to Increase Fan Engagement

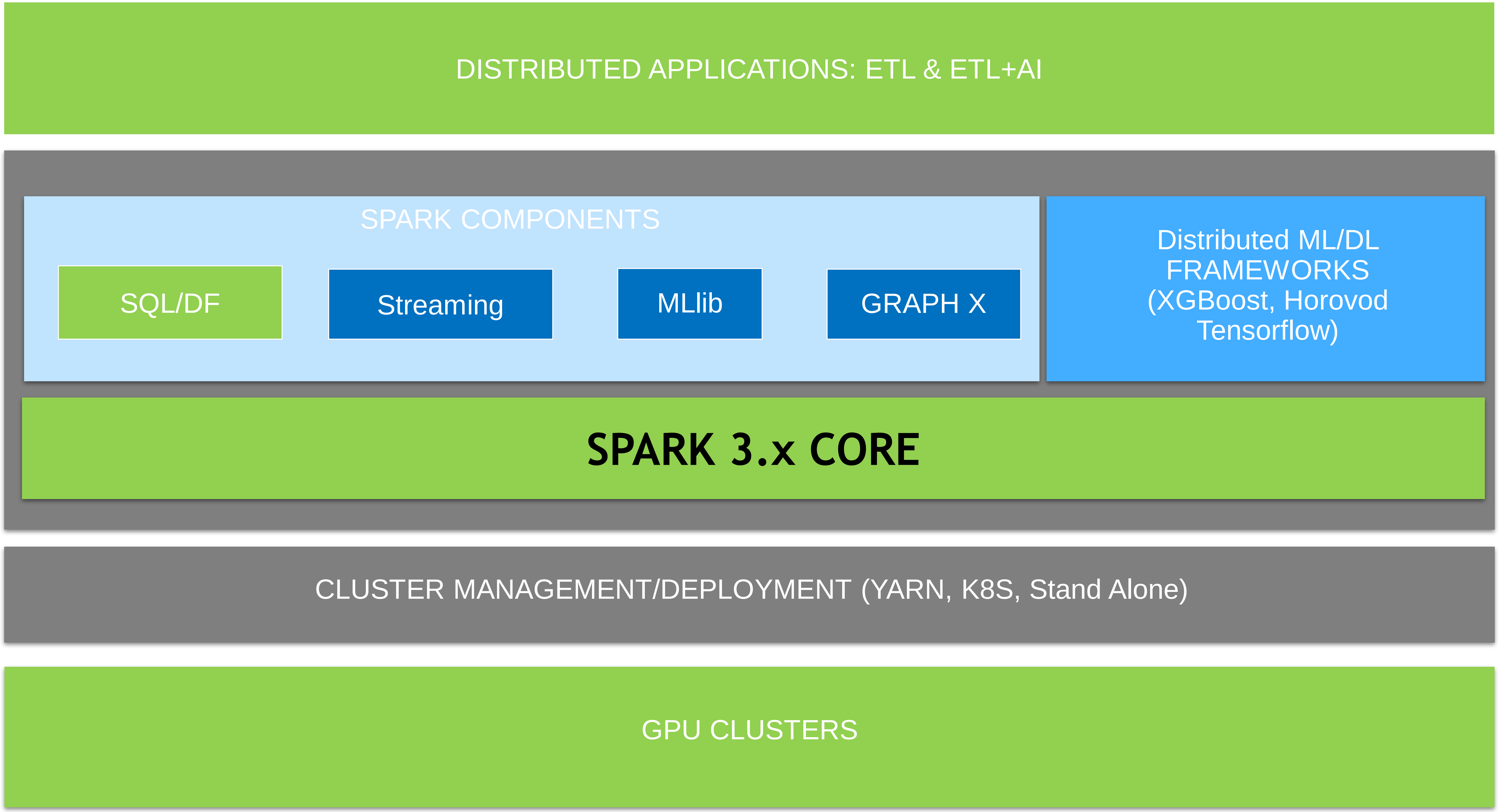
90% of US Homes Now Have Smart Meters > Determine More Efficient Uses of Electricity

25B Connected Devices > Most Are Streaming Valuable Data that is Underutilized

50 Devices per House Concurrently Drawing Power > Need to Turn Off Things Not Being Used



SPARK 3.X IS AN UNIFIED AI PLATFORM



DISTRIBUTED APPLICATIONS: ETL & ETL+AI

SPARK COMPONENTS

SQL/DF

Streaming

MLlib

GRAPH X

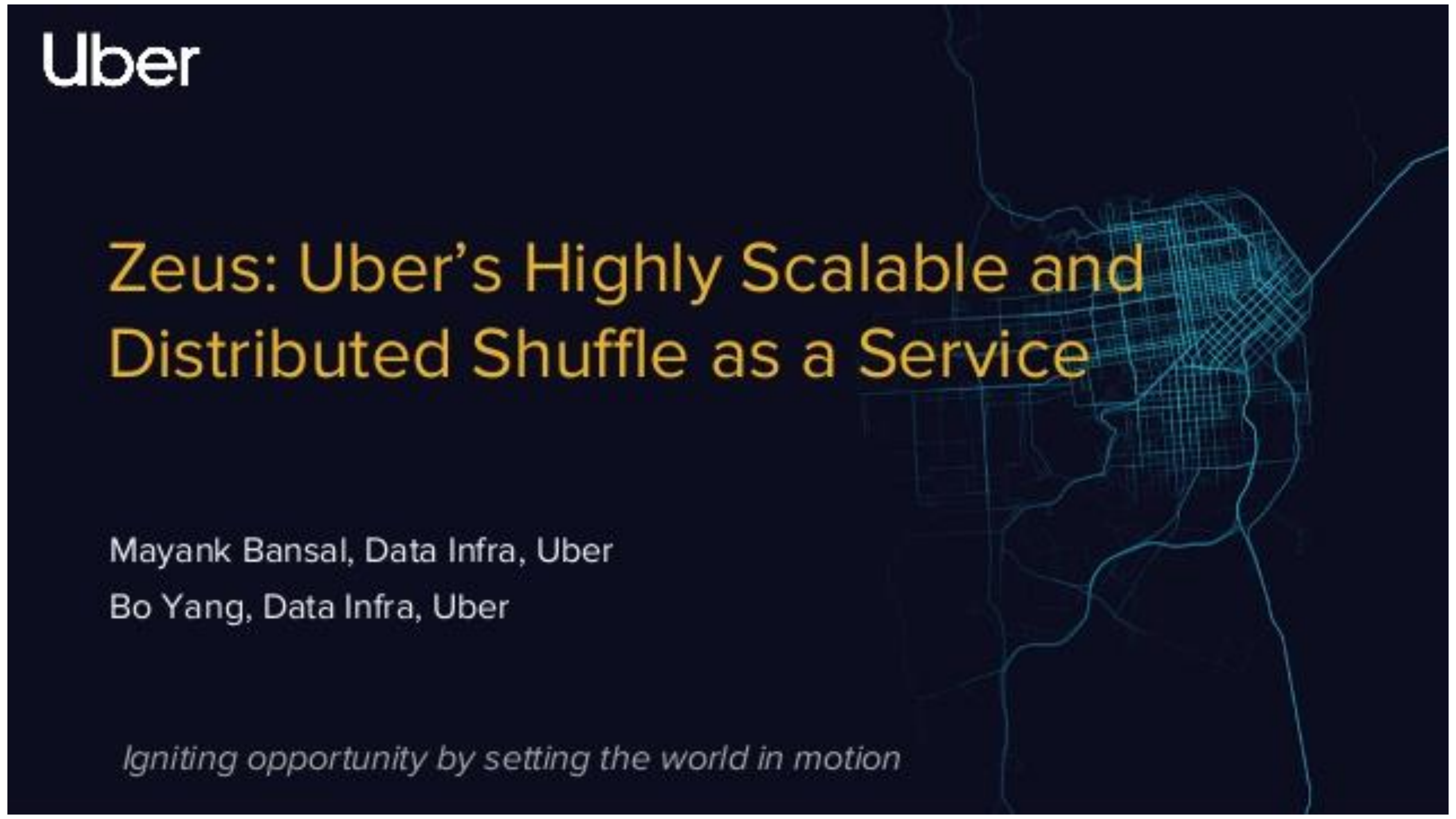
Distributed ML/DL FRAMEWORKS (XGBoost, Horovod Tensorflow)

SPARK 3.x CORE

CLUSTER MANAGEMENT/DEPLOYMENT (YARN, K8S, Stand Alone)

GPU CLUSTERS

SHUFFLE IS THE KEY

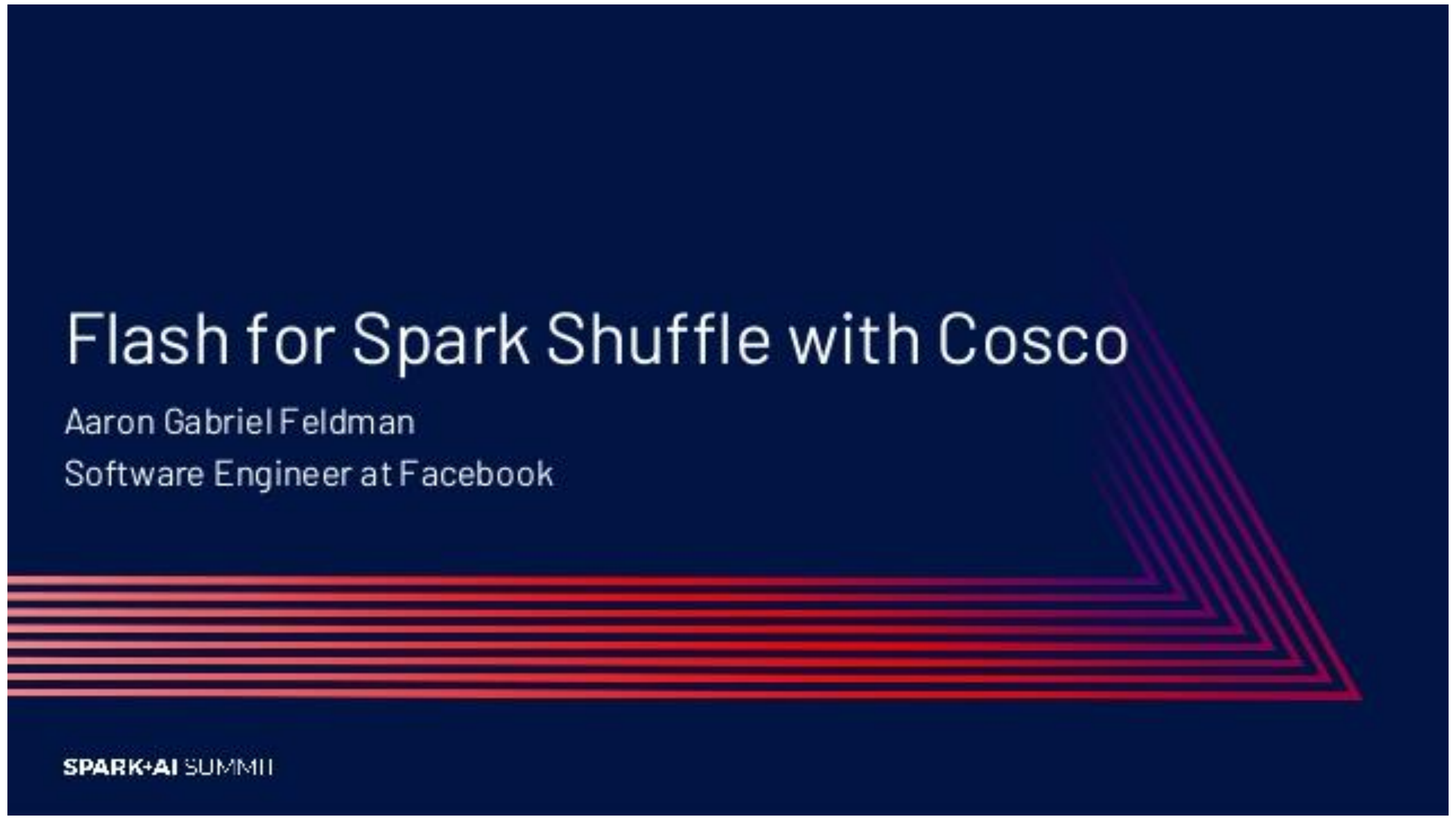


Uber

Zeus: Uber's Highly Scalable and Distributed Shuffle as a Service

Mayank Bansal, Data Infra, Uber
Bo Yang, Data Infra, Uber

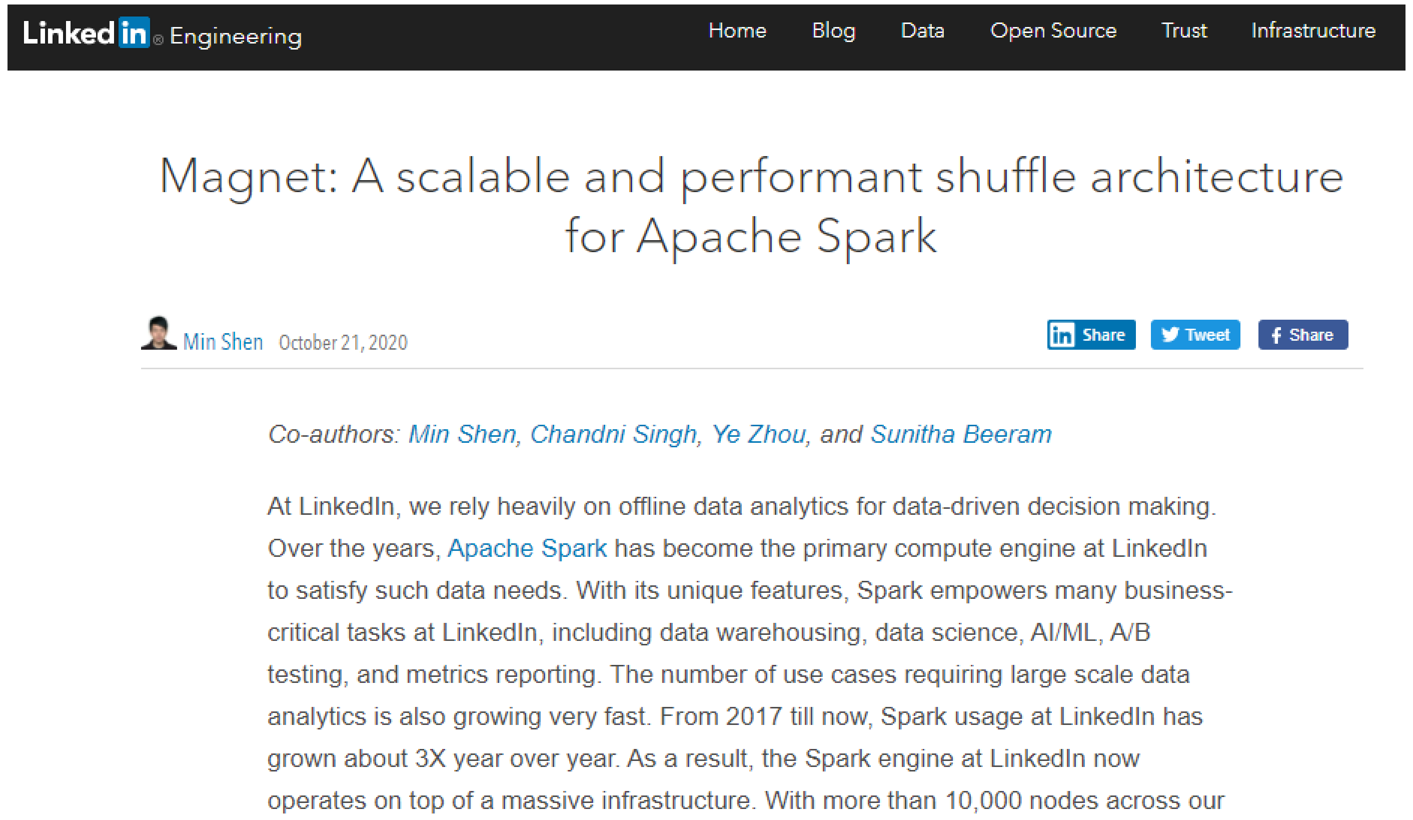
Igniting opportunity by setting the world in motion



Flash for Spark Shuffle with Cosco

Aaron Gabriel Feldman
Software Engineer at Facebook

SPARK+AI SUMMIT



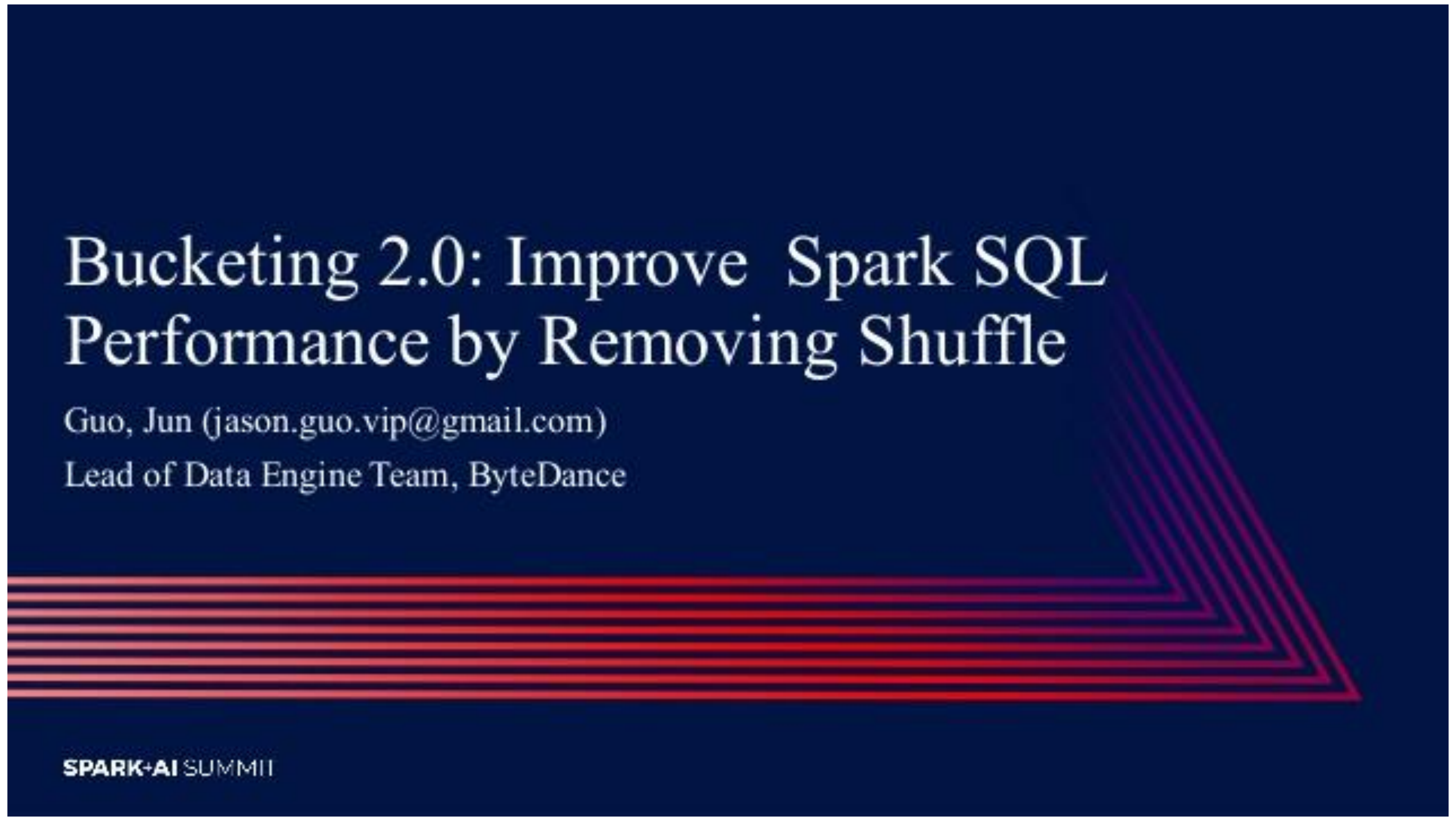
LinkedIn Engineering — Home | Blog | Data | Open Source | Trust | Infrastructure

Magnet: A scalable and performant shuffle architecture for Apache Spark

Min Shen October 21, 2020 — Share | Tweet | Share

Co-authors: Min Shen, Chandni Singh, Ye Zhou, and Sunitha Beeram

At LinkedIn, we rely heavily on offline data analytics for data-driven decision making. Over the years, Apache Spark has become the primary compute engine at LinkedIn to satisfy such data needs. With its unique features, Spark empowers many business-critical tasks at LinkedIn, including data warehousing, data science, AI/ML, A/B testing, and metrics reporting. The number of use cases requiring large scale data analytics is also growing very fast. From 2017 till now, Spark usage at LinkedIn has grown about 3X year over year. As a result, the Spark engine at LinkedIn now operates on top of a massive infrastructure. With more than 10,000 nodes across our

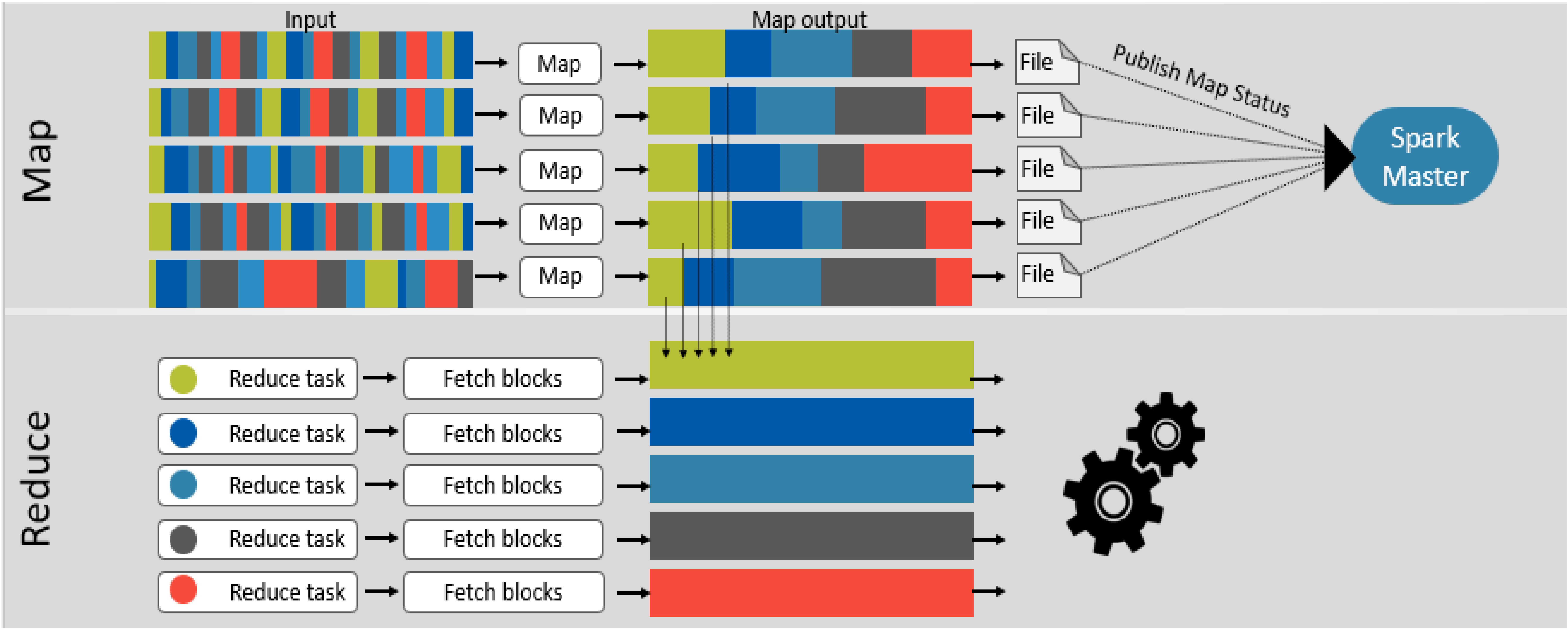


Bucketing 2.0: Improve Spark SQL Performance by Removing Shuffle

Guo, Jun (jason.guo.vip@gmail.com)
Lead of Data Engine Team, ByteDance

SPARK+AI SUMMIT

SHUFFLE BASICS



MELLANOX + NVIDIA SHUFFLE ACCELERATION

- **2017** SparkRDMA shuffle plugin open sourced https://github.com/Mellanox/SparkRDMA
  - Based on disni library (thin wrapper over verbs)
  - Promote RDMA technology in Spark community (AI Spark summit talks Accelerating Shuffle: A Tailor-Made RDMA Solution for Apache Spark, Accelerated Spark on Azure: Seamless and Scalable Hardware Offloads in the Cloud)
  - Initial customers POC, collected requirements and feedback.

- **2019** SparkUCX shuffle plugin https://github.com/openucx/sparkucx
  - Java wrapper for UCX library implementation
  - Fixes architectural bottlenecks in SparkRDMA

- **2020** Nvidia Rapids for Spark https://github.com/NVIDIA/spark-rapids
  - Based on UCX java library for communication
  - GPU + RDMA acceleration

- **2021** SparkUCX – unified shuffle architecture
  - Public transport API, that can be utilized in other Spark and big data solutions
  - Works for both GPU and host memory RDMA

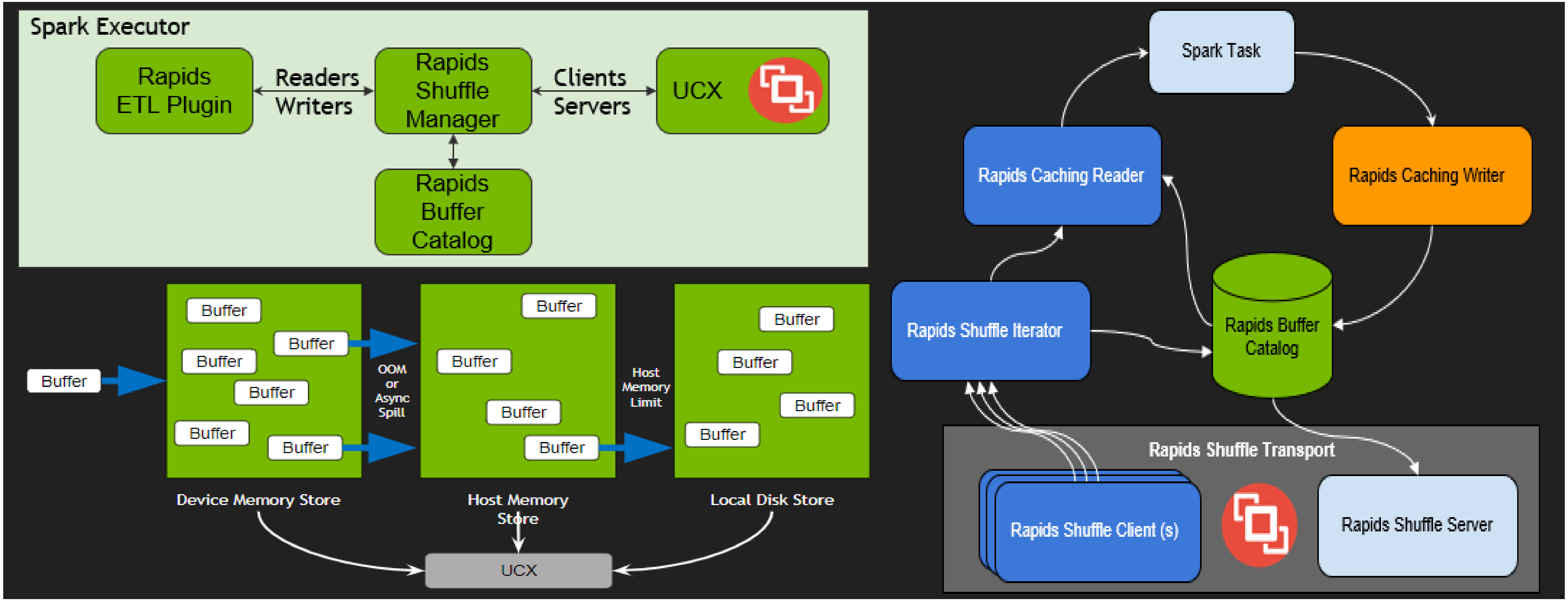
SPARKUCX ARCHITECTURE

- Initialization:
    - Spark driver allocates global metadata buffer per shuffle stage, to hold addresses and memory keys of data and index files on mappers.

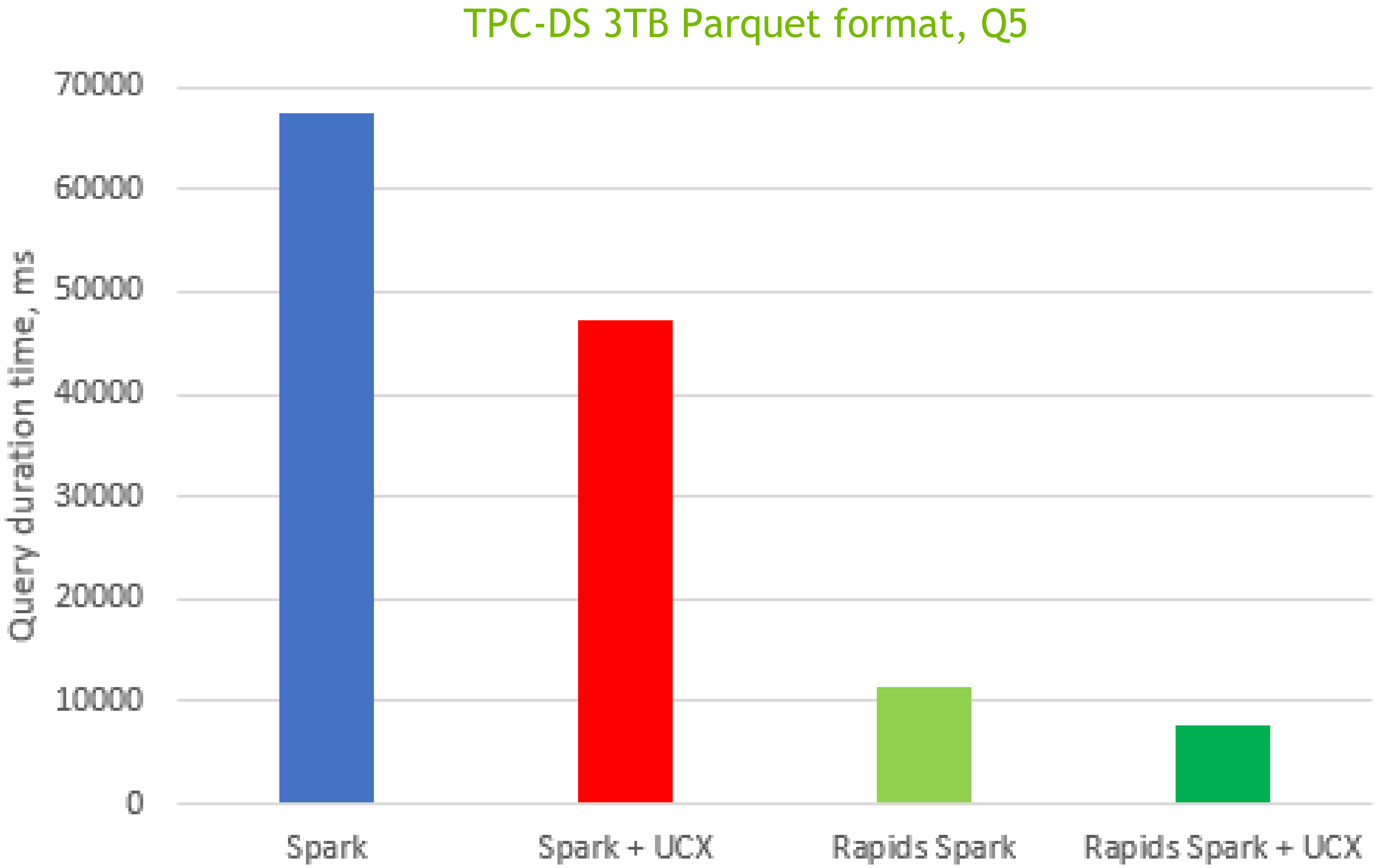
- Mapper phase:
    - mmap() and register index and data files
    - Publish {address, rkey} to driver metadata buffer (ucp_put).

- Reduce phase:
    - Fetch metadata from driver (ucp_get)
    - For each block:
    - Fetch offset in data file, from index file (ucp_get).
    - Fetch block contents from data file (ucp_get).

RAPIDS SPARK UCX SHUFFLE



ACCELERATED SPARK SHUFFLE RESULTS

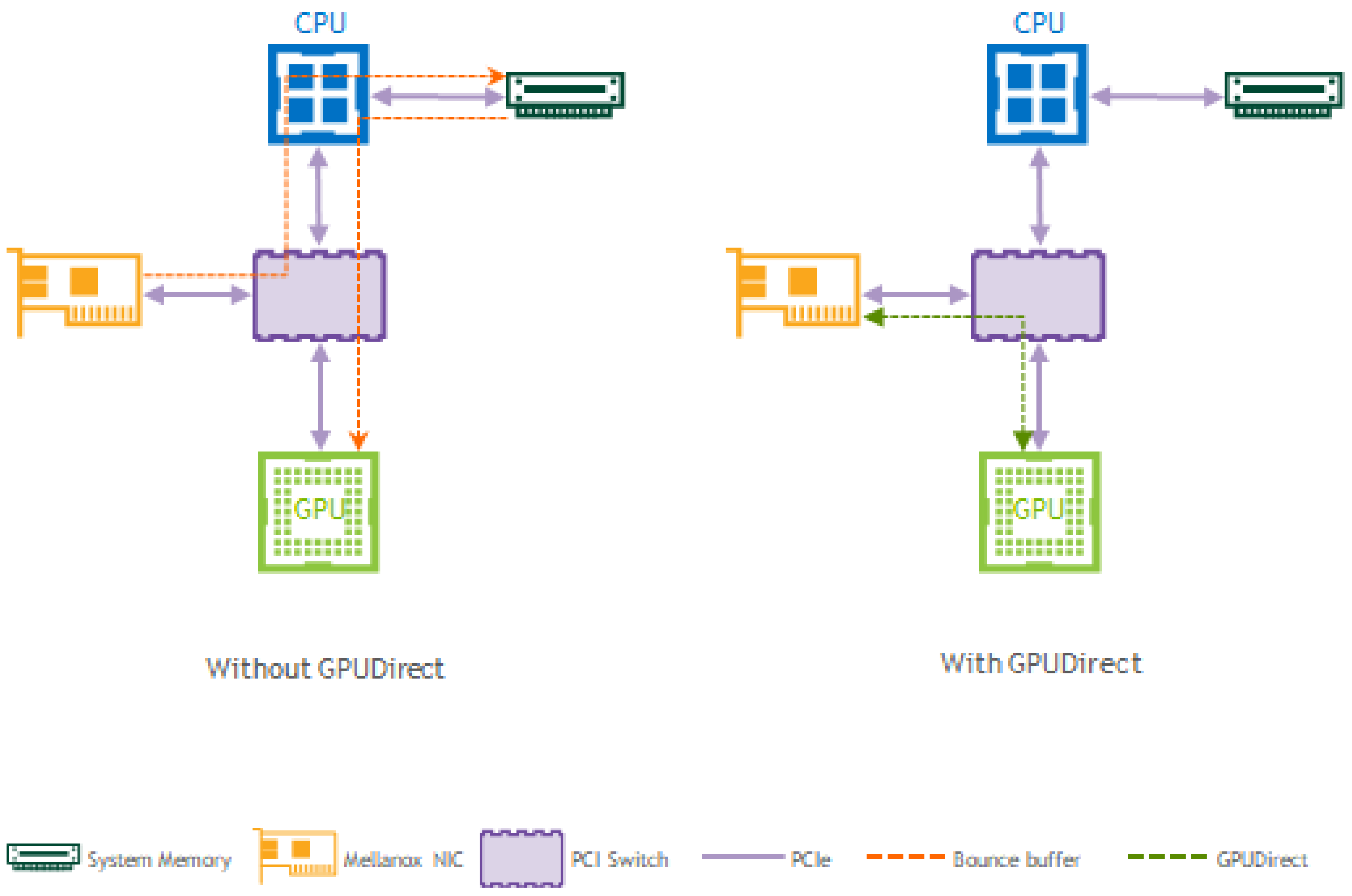


STEP BY STEP SETUP

Reference Deployment Guide

RDG: Accelerating Apache Spark 3.0 with RAPIDS Accelerator over RoCE network.

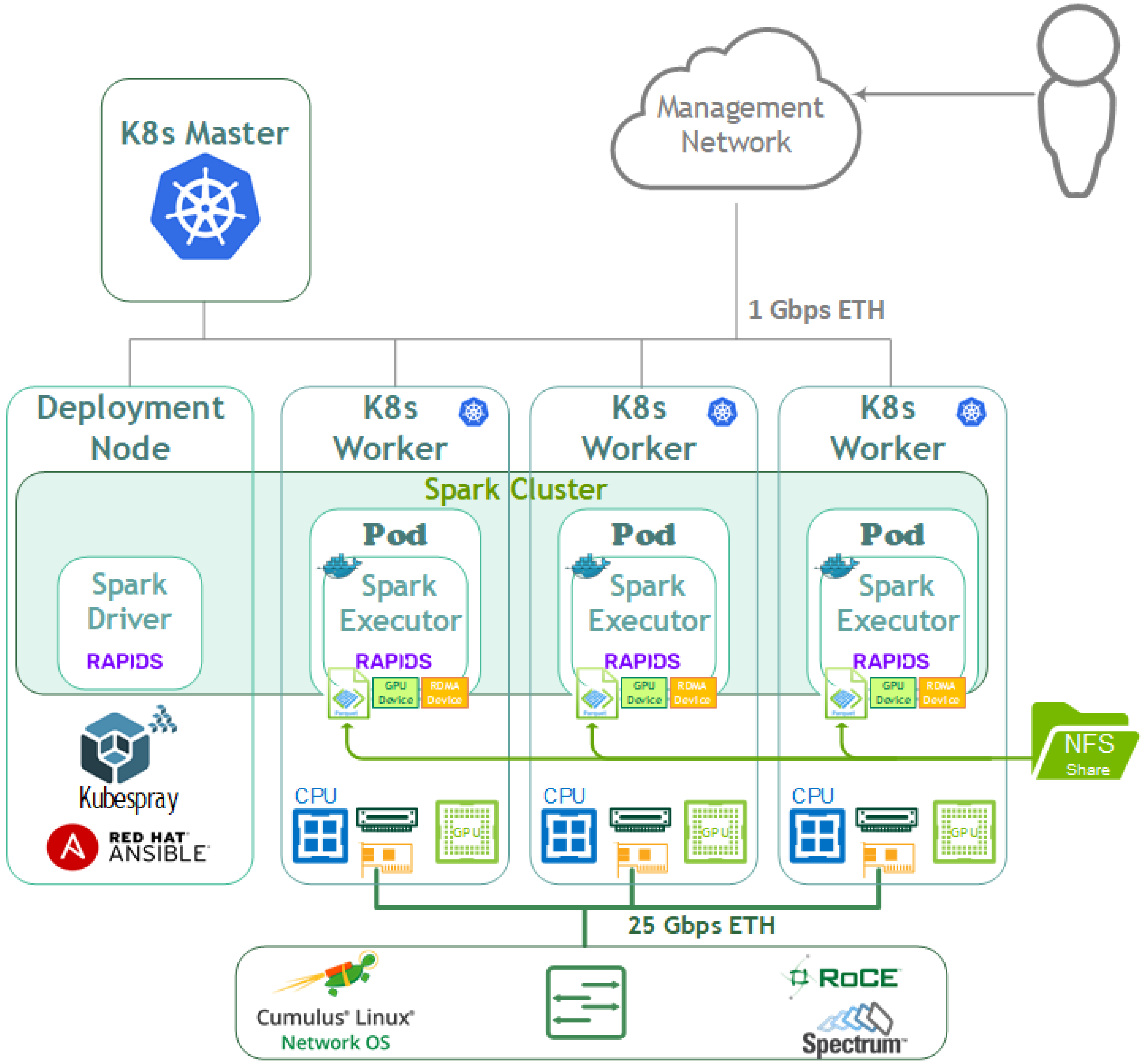
- **GPUDirect RDMA**
  GPUDirect (GDR) RDMA provides a direct P2P (Peer-to-Peer) data path between the GPU Memory directly to and from NVIDIA Mellanox HCA devices, which reduces GPU-to-GPU communication latency and completely offloads the CPU, removing it from all GPU-to-GPU communications across the network.



Without GPUDirect

With GPUDirect

RDG: Apache Spark 3.0 on Kubernetes accelerated with RAPIDS over RoCE network.



NEXT STEPS

Unified transport API

1. **RegisterBlock** (blockId, address, length) – associates memory block with a blockId

2. **MutateBlock** (blockId, newAddress, newLength, callback) – changes block location on spill

3. **FetchBlockByBlockId** (blockId, destinationBuffer, callback) - fetches remote block. Transport selects best protocol (one sided, AM) to transfer the data

4. **Unregister**(blockId) – tells transport block is not needed

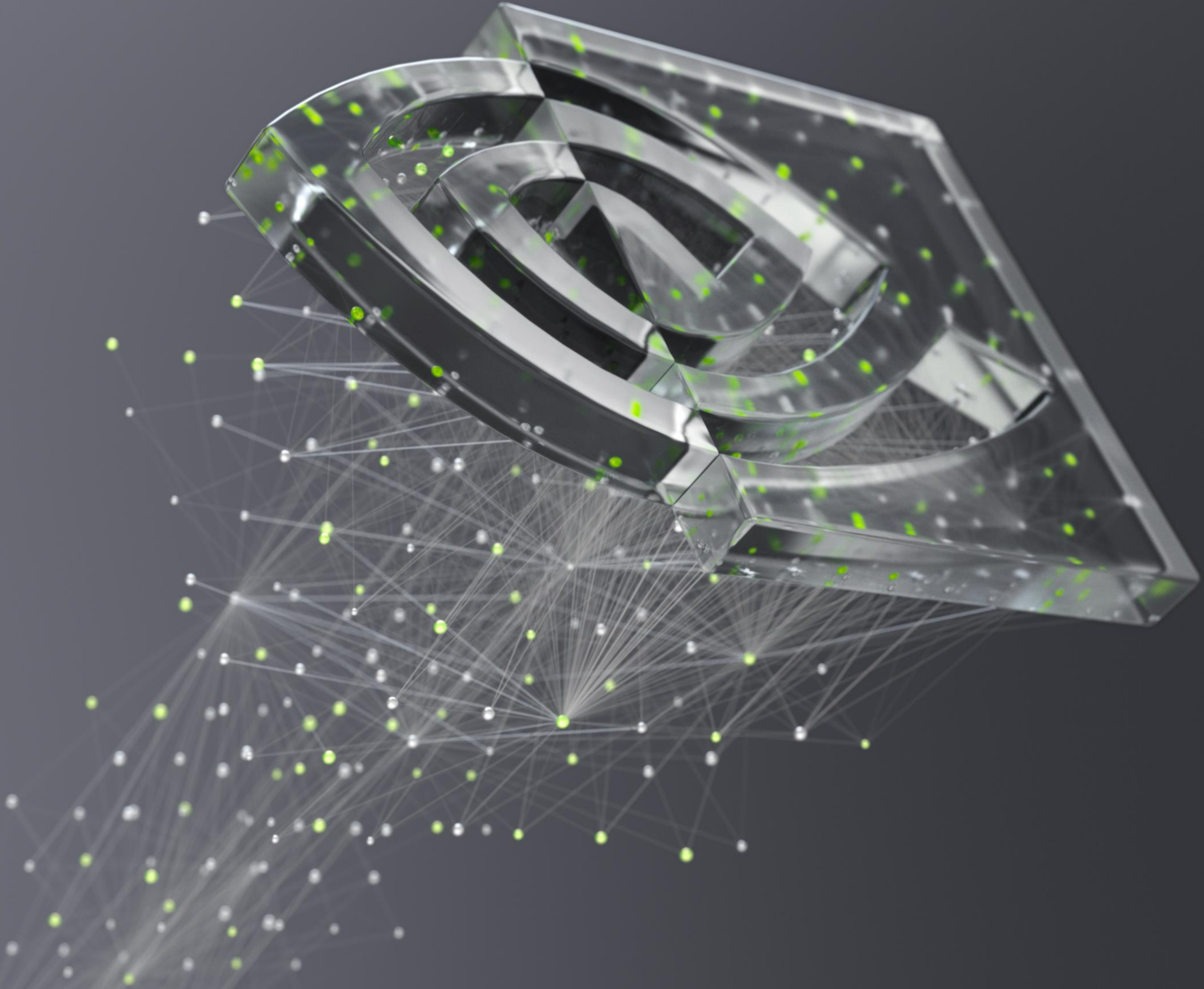
NEXT STEPS

Transport optimization

1. One sided GPU RDMA
2. GPU topology awareness
3. GPU bounce buffers
4. Error handling
5. Commodity architecture optimization (cloud, non GPUDIRECT).

SPARK+UCX BENEFITS

- Accelerating Spark
  - Lower Block transfer times (latency and total transfer time)
  - Lower Memory consumption and management
  - Lower CPU utilization
  - GPU Direct
- Easy to deploy and configure
  - Packed into a single JAR file
  - Plugin is enabled through a simple configuration handle
  - Allows finer tuning with a set of configuration handles
- Configuration and deployment are on a per job basis
  - Can be deployed incrementally



Thanks,
QA