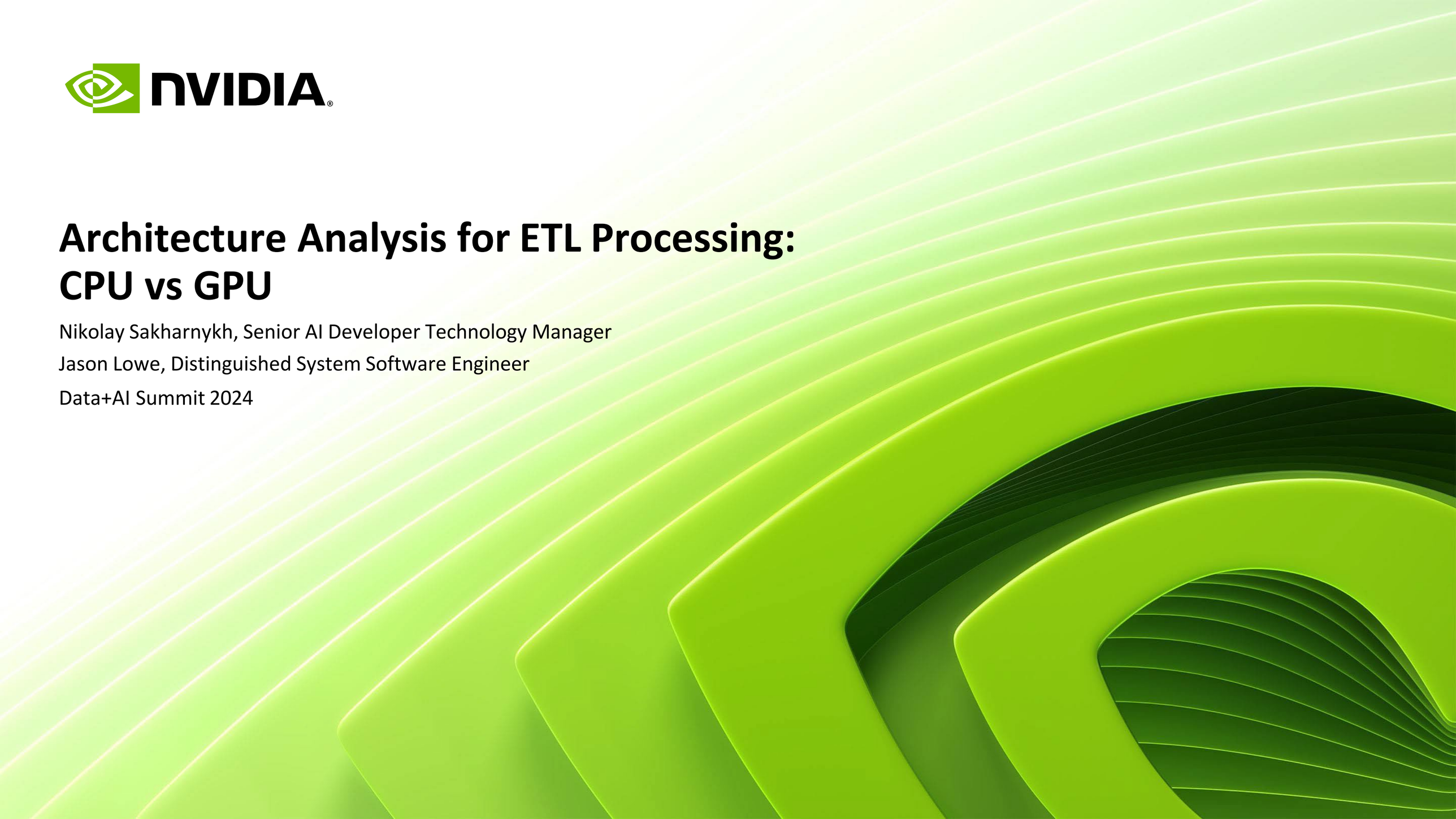


Architecture Analysis for ETL Processing: CPU vs GPU

Nikolay Sakharnykh, Senior AI Developer Technology Manager
Jason Lowe, Distinguished System Software Engineer

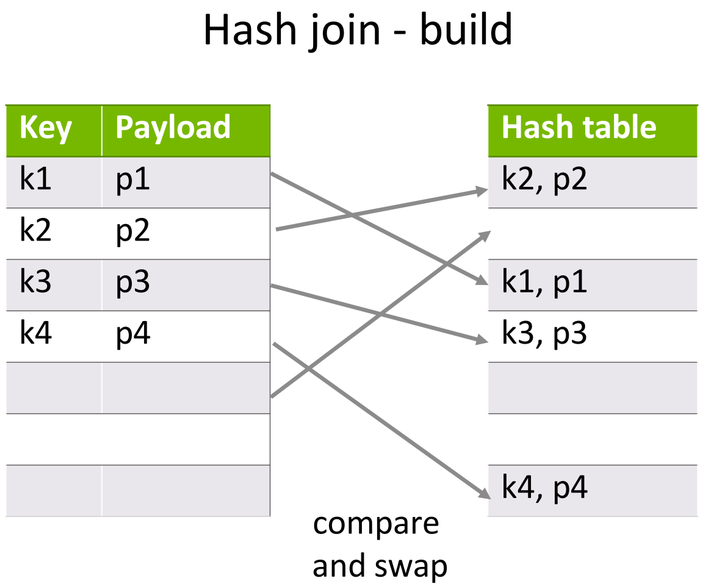
Data+AI Summit 2024

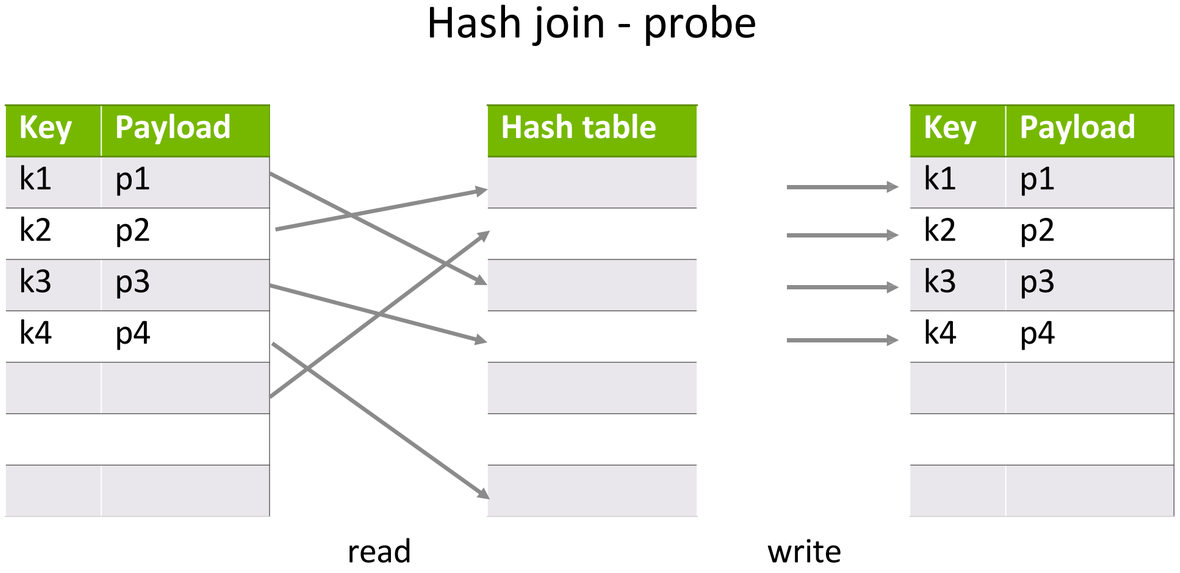
Can GPUs accelerate ETL?

Agenda

- Performance limiters of Database Operations
- CPU and GPU architectures overview
- GPU performance on join and full SQL Queries
- RAPIDS Accelerator for Apache Spark
- RAPIDS Accelerator Benchmarks

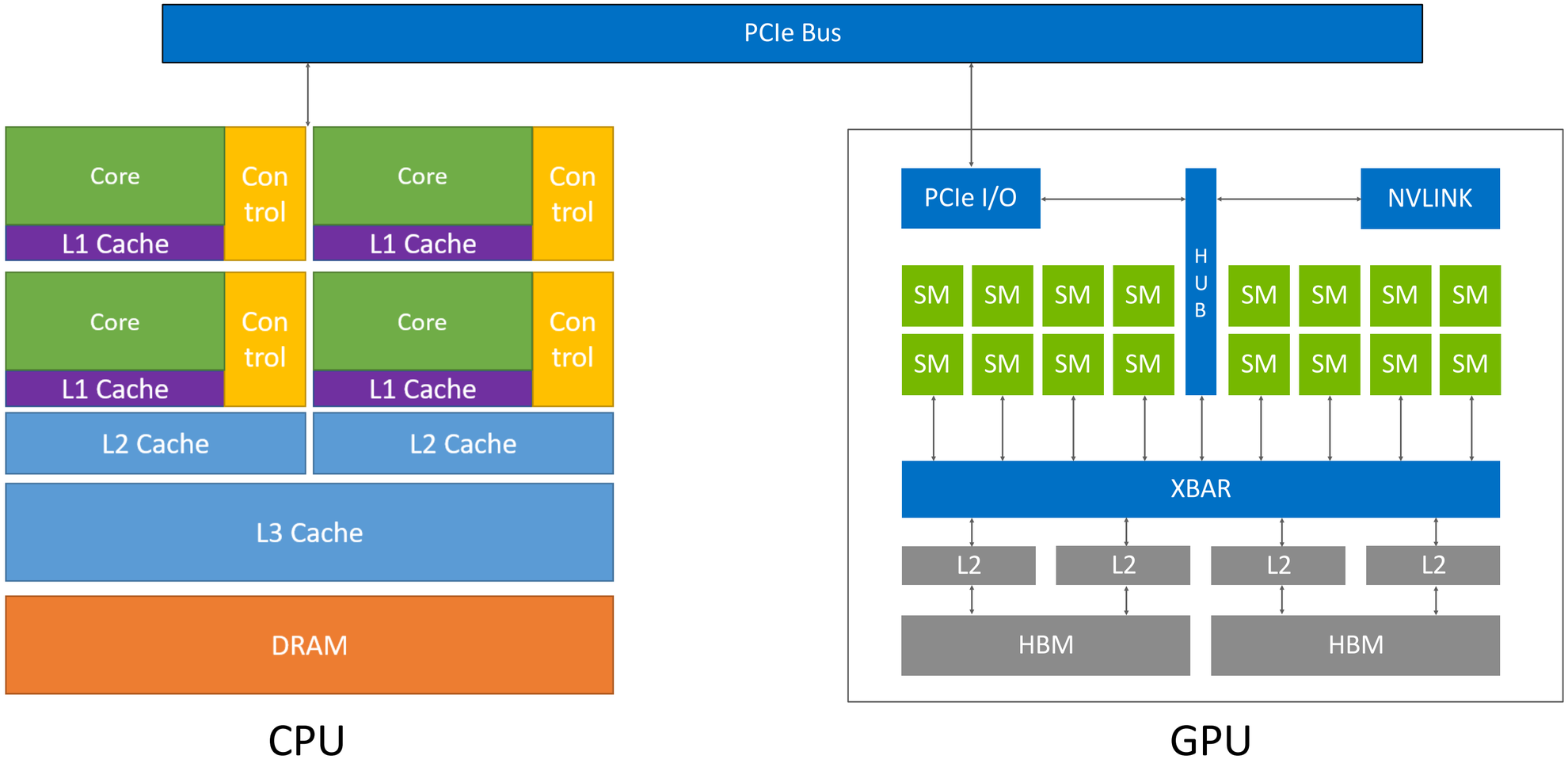
Database Operations and Their Performance Limiters



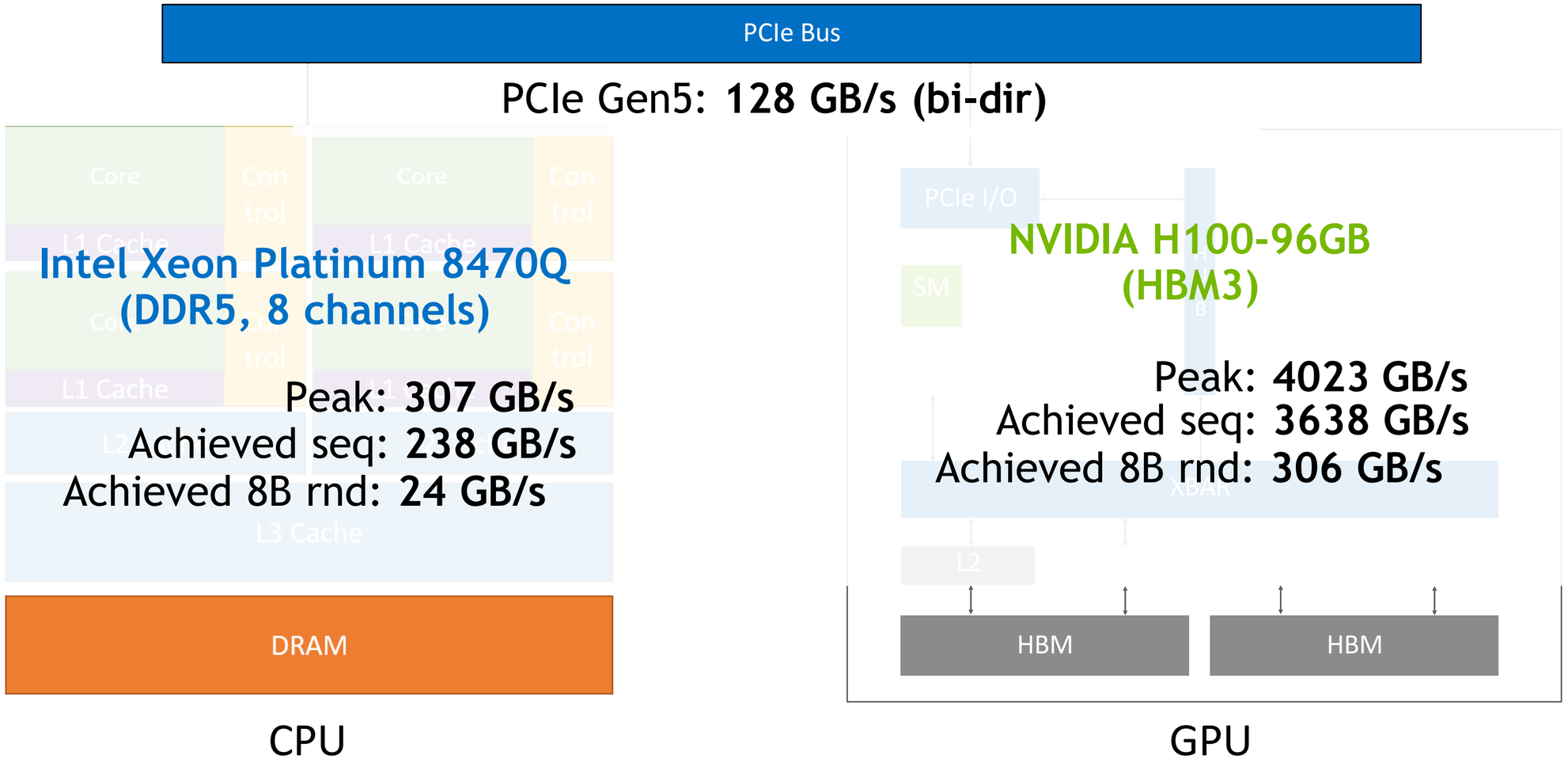


- Sequential access to input/output tables (read/write)
- Random fine-grained access to hash tables (CAS/read)
- Integer computations for hashing

CPU and GPU Architectures

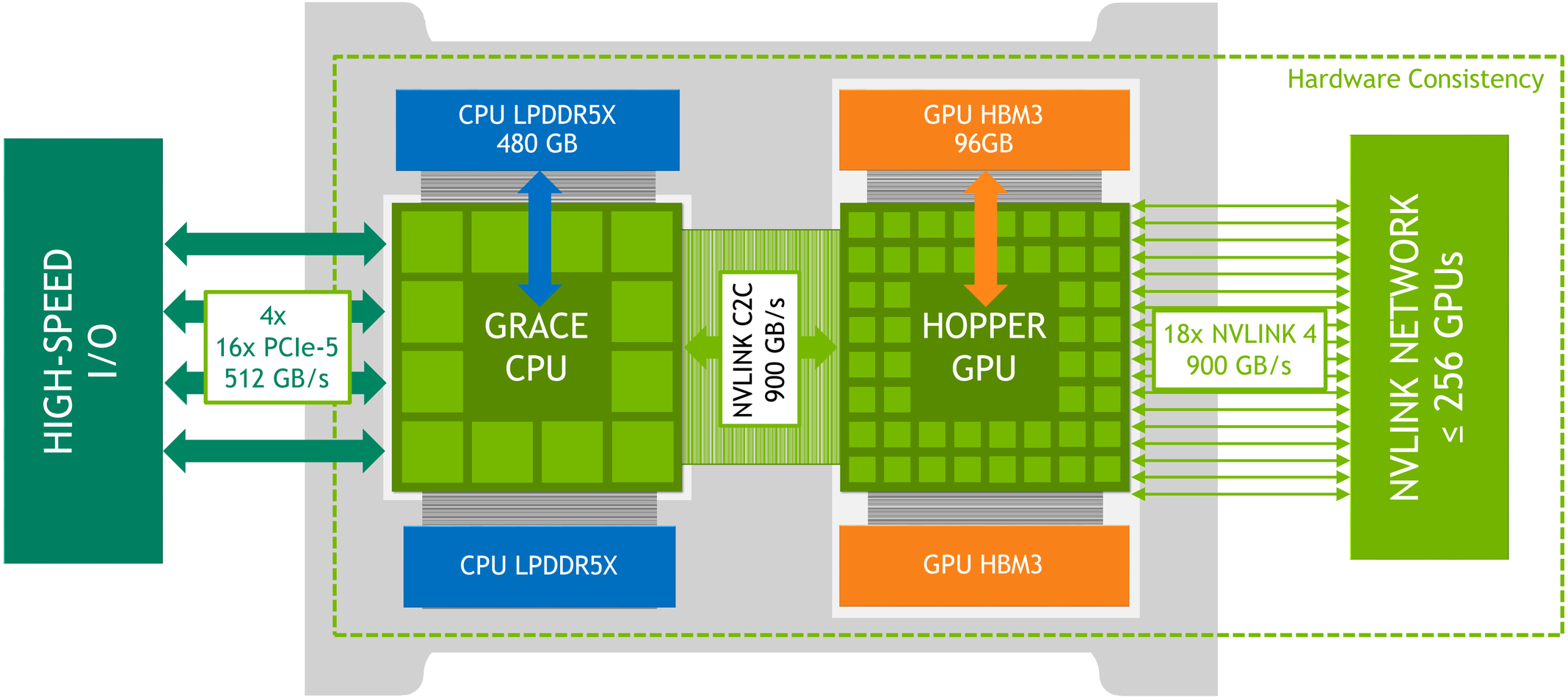


CPU and GPU Architectures – Memory Speeds



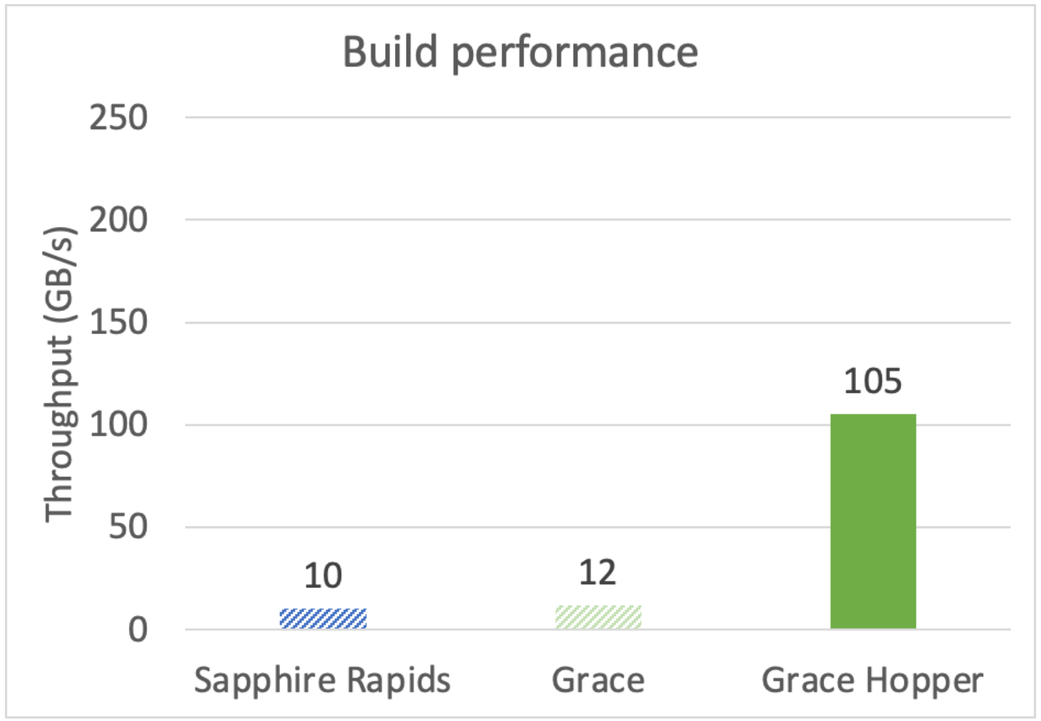
Bandwidth is calculated as the size of access multiplied by the number of accesses divided by time

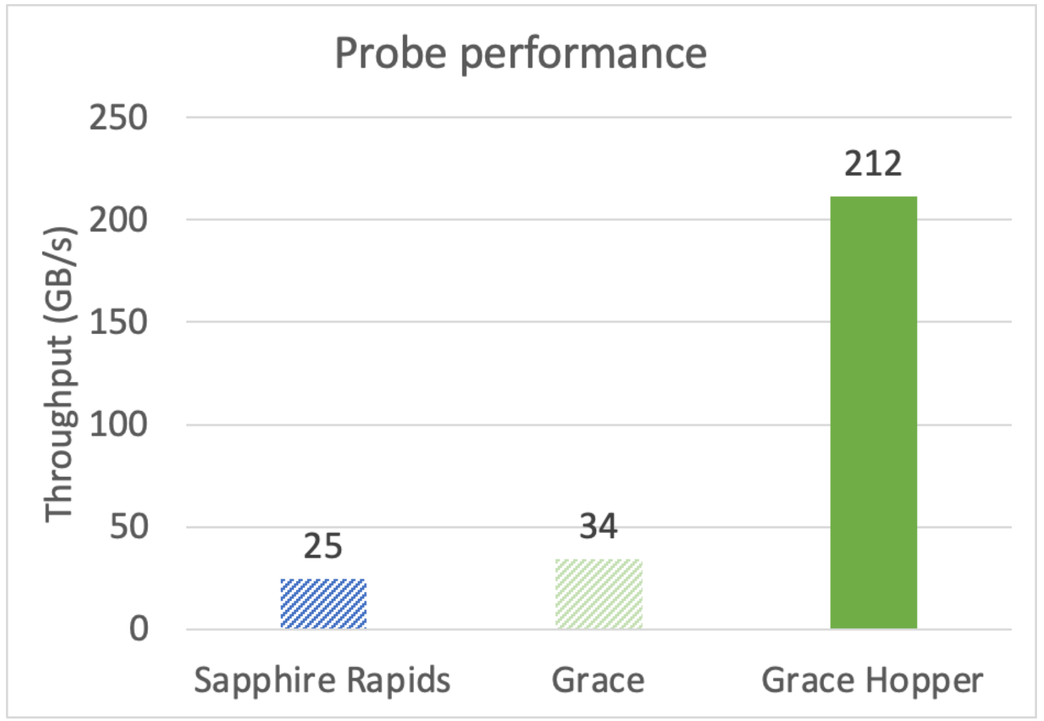
Grace Hopper Superchip



https://resources.nvidia.com/en-us-grace-cpu/nvidia-grace-hopper

Join Micro-benchmark





Grace Hopper achieved perf **is 8-10x faster** than *projected* best x86 performance
Performance limiter – random 64-bit CAS/read

NVIDIA Decision Support-H Benchmark

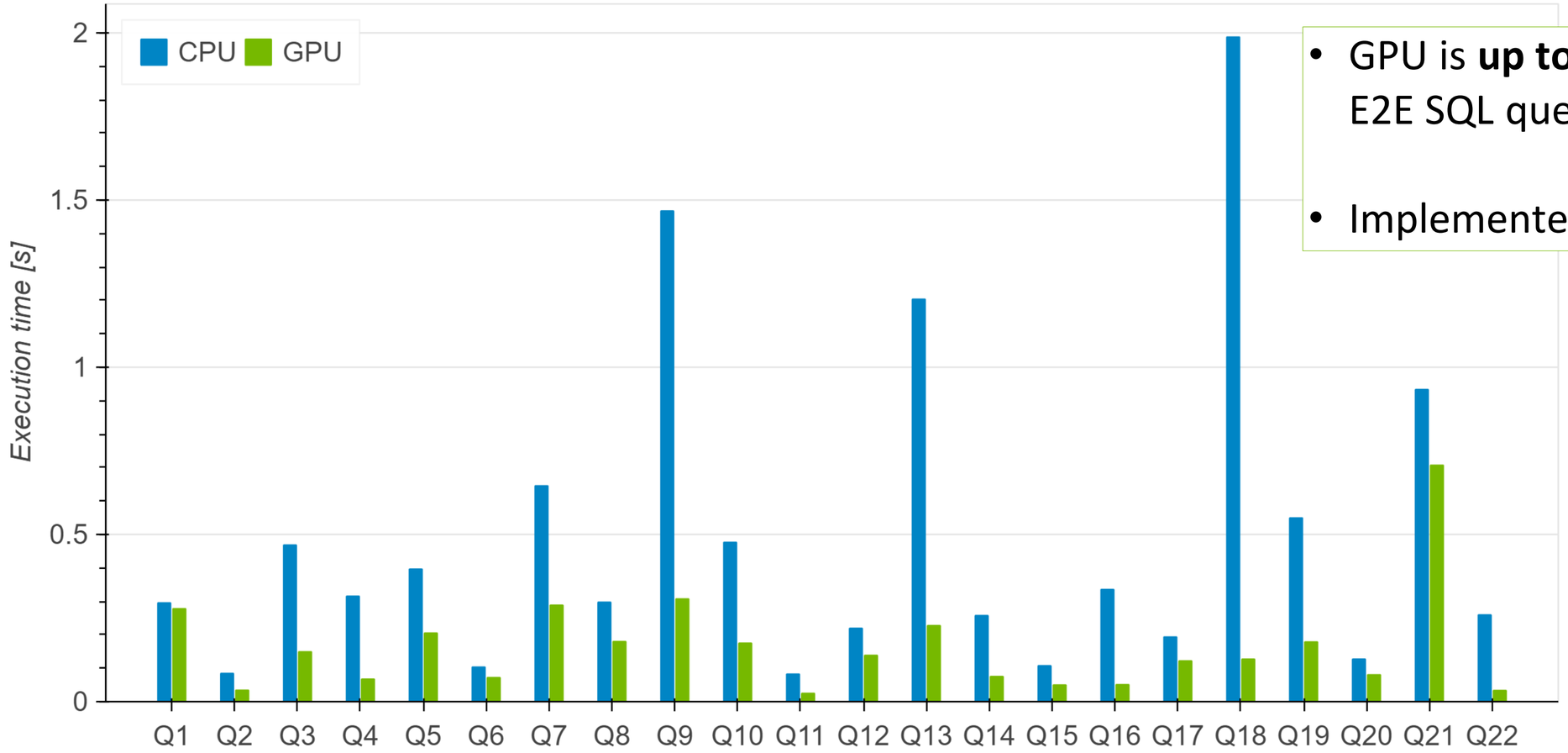
NVIDIA Decision Support-H (NDS-H) is our adaptation of the TPC-H benchmark often used by database customers and providers

NDS-H consists of the same 22 SQL queries as the industry standard benchmark

The NDS-H benchmark is derived from the TPC-H benchmark and as such is not comparable to published TPC-H results, as the NDS-H results do not comply with the TPC-H Specification

SQL Queries on Grace Hopper

NDS-H **SF100**, input tables in **CPU memory** (lower is better)



- GPU is **up to 15.7x faster** than CPU for E2E SQL query perf
- Implemented with **RAPIDS cuDF**

Preliminary unoptimized results from a prototype GPU-accelerated query engine

RAPIDS Accelerator for Apache Spark

NVIDIA RAPIDS Accelerator

Key technologies for GPU acceleration

DATA ANALYTICS APPLICATIONS AND AI/ML PIPELINES

APACHE SPARK PLATFORM

ACCELERATED BATCH DATA PROCESSING

Spark SQL

DataFrames

ACCELERATED SPARK MACHINE LEARNING

MLlib

RAPIDS Accelerator for Apache Spark

RAPIDS Accelerator for Apache Spark ML

RAPIDS

GPU-ACCELERATED INFRASTRUCTURE

No Query Changes

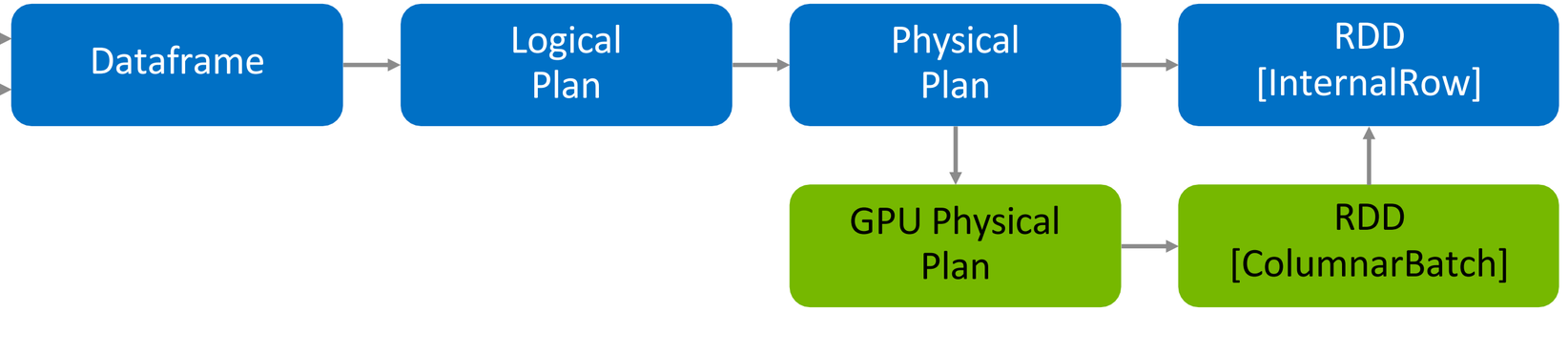
- Add jar to classpath and set spark.plugins config
- Same SQL and DataFrame code
- Compatible with PySpark, SparkR, Java, Scala and other DataFrame-based APIs
- Seamless fallback to CPU for unsupported operations

```
spark.sql( """

  SELECT
    o_order_priority
    count(*) as order_count
  FROM
    orders
  WHERE
    o_orderdate >= DATE '1993-07-01'
    AND o_orderdate < DATE '1993-07-01' +
interval '3' month
    AND EXISTS (
      SELECT
        *
      FROM lineitem
      WHERE
        l_orderkey = o_orderkey
          AND l_commitdate < l_receiptdate
    )
  GROUP BY
    o_orderpriority ORDER BY o_orderpriority

""" ).show()
```

Spark SQL & DataFrame Query Execution

```
bar.groupBy(
    col(”product_id”),
    col(“ds”))
.agg(
    max(col(“price”)) -
    min(col(“price”)).alias(“range”))
```

```
SELECT product_id, ds,
    max(price) – min(price)
AS range
FROM bar
GROUP BY product_id, ds
```



NVIDIA Decision Support Benchmark

NVIDIA Decision Support (NDS) is our adaptation of the TPC-DS benchmark often used by Spark customers and providers

NDS consists of the same 100+ SQL queries as the industry standard benchmark but has modified parts for execution scripts.

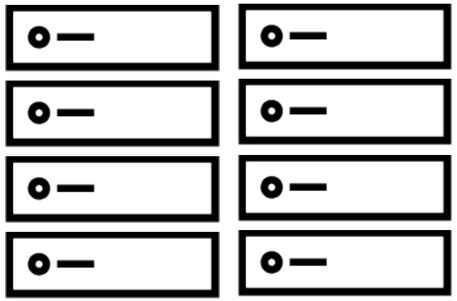
The NDS benchmark is derived from the TPC-DS benchmark and as such is not comparable to published TPC-DS results, as the NDS results do not comply with the TPC-DS Specification

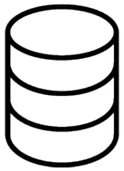
https://github.com/nvidia/spark-rapids-benchmarks

AWS EC2 cluster

Parquet data, scale factor 3k, stored on S3

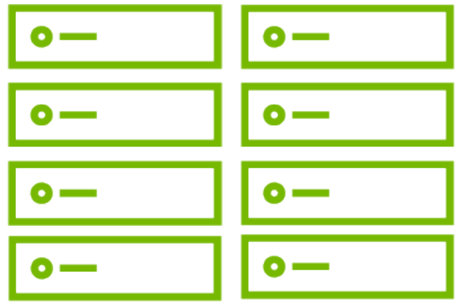
8 x r6id.8xlarge





S3

8 x g6.8xlarge



	CPU Cores	CPU Mem (GB)	Network BW (Gbps)	Storage	GPU	On Demand $ Cost / Hr
r6id.8xlarge	32	256	12.5	1900GB local SSD		$2.419
g6.8xlarge	32	128	25	2x450GB local SSD	L4	$2.014

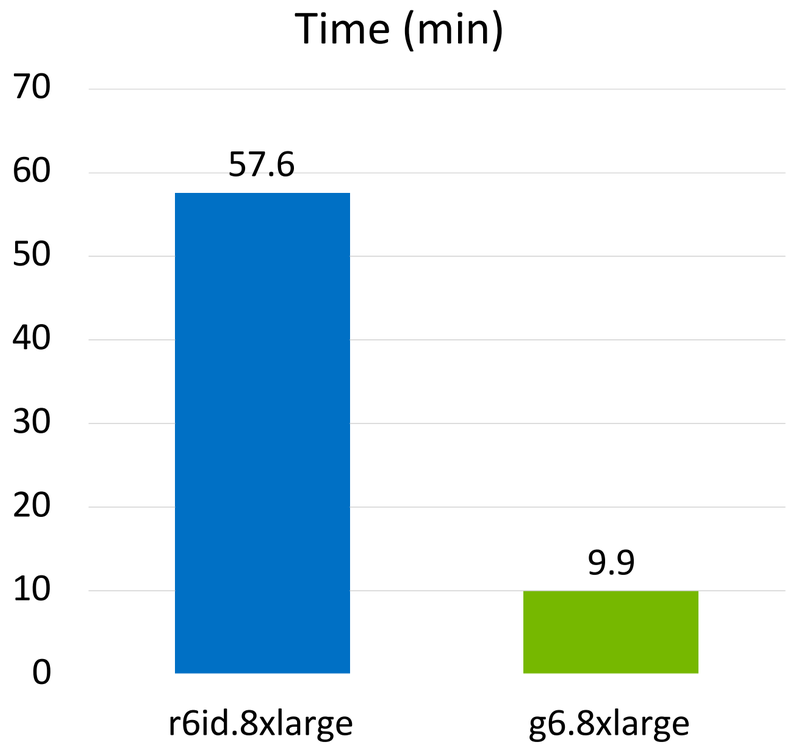
Source: https://aws.amazon.com/ec2/pricing/on-demand/

AWS EC2 Configurations

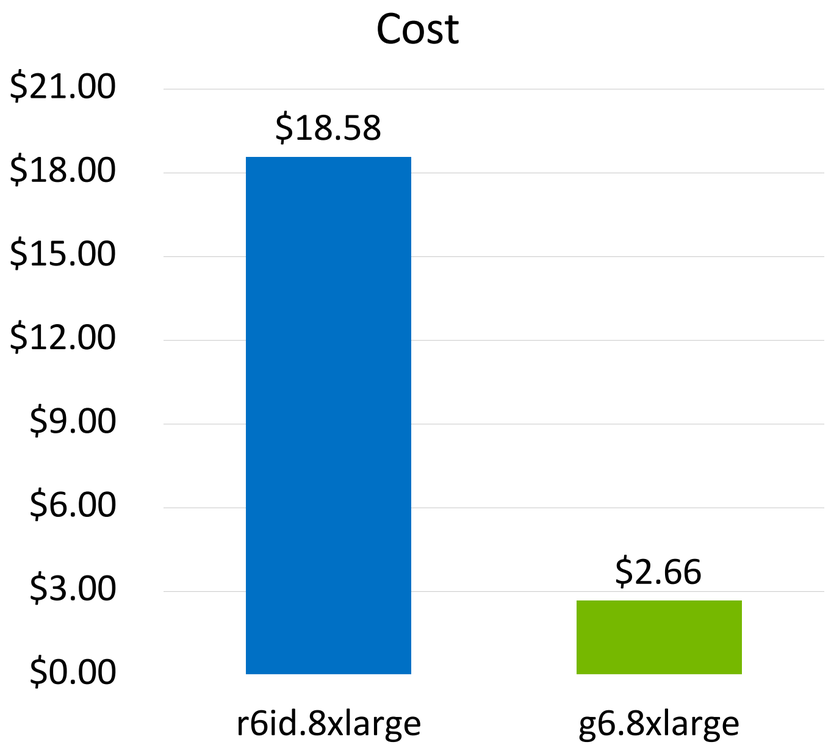
	CPU	GPU	Config type
spark.executor.cores	16	16	Resource
spark.executor.instances	16	8	
spark.executor.memory	64G	64G	
spark.rapids.filecache.enabled		true	
spark.executor.resource.gpu.amount		1	
spark.task.resource.gpu.amount		0.0625	
spark.scheduler.minRegisteredResourcesRatio	1.0	1.0	Scheduling
spark.locality.wait	0	0	
spark.sql.files.maxPartitionBytes	128M	2GB	
spark.shuffle.manager		com.nvidia.spark.rapids.spark341.RapidsShuffleManager	Shuffle
spark.rapids.shuffle.multiThreaded.{reader|writer}.threads		32	
spark.rapids.sql.multiThreadedRead.numThreads		100	
spark.plugins		com.nvidia.spark.SQLPlugin	GPU
spark.rapids.memory.host.spillStorageSize		16G	
spark.rapids.memory.pinnedPool.size		8G	
spark.rapids.sql.concurrentGpuTasks		3	

NVIDIA Decision Support Benchmark 3TB, AWS EC2

Apache Spark 3.4.1, RAPIDS Accelerator 24.04



5.8x faster



85% cost savings

Gluten+Velox vs A100 on 8 Node Cluster

Parquet data stored on HDFS

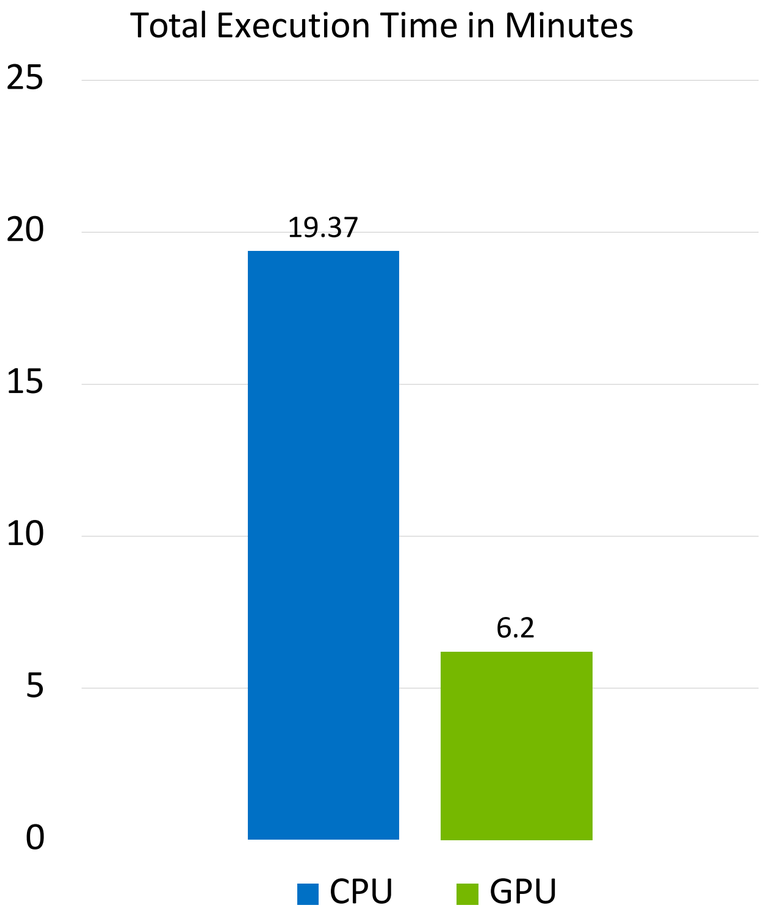
	CPU	CPU Mem (GB)	Network BW (Gbps)	Storage	GPU
NVIDIA EGX Certified Server	2 x AMD EPYC 7452 64 cores 128 threads	512	100	4 x 7.68 TB Gen4 U.2 NVMe	1 x A100 80GB

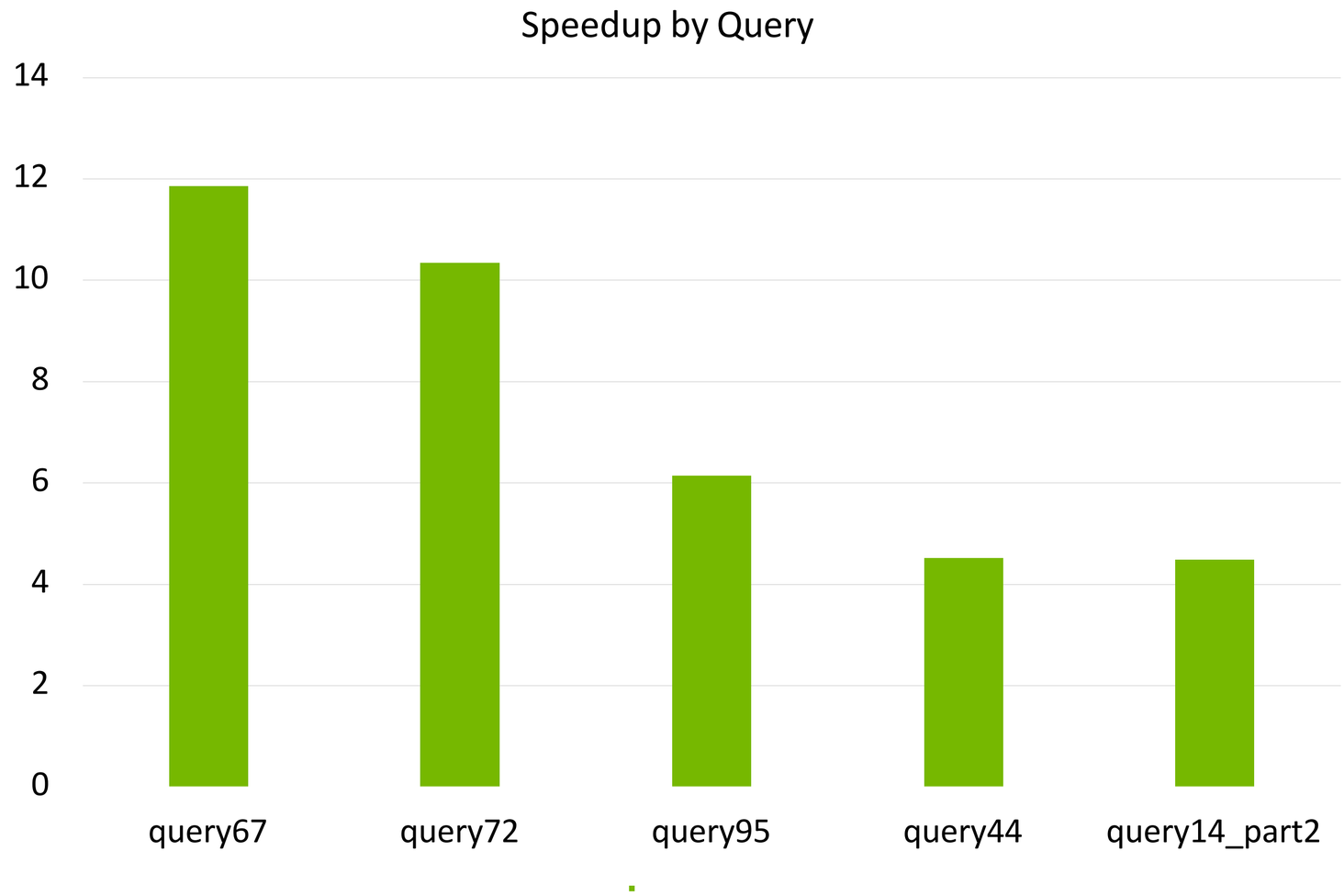
Gluten+Velox vs A100 Configurations

	CPU	GPU	Config type
spark.executor.cores	16	16	Resource
spark.executor.instances	64	8	
spark.executor.memory	16G	16G	
spark.memory.offHeap.enabled	true		
spark.memory.offHeap.size	20G		
spark.executor.resource.gpu.amount		1	
spark.task.resource.gpu.amount		0.0625	
spark.sql.files.maxPartitionBytes	2GB	2GB	Shuffle
spark.sql.shuffle.partitions	1024	200	
spark.shuffle.manager	org.apache.spark.shuffle.sort.ColumnarShuffleManager	com.nvidia.spark.rapids.spark342.RapidsShuffleManager	
spark.rapids.memory.host.spillStorageSize		32G	GPU
spark.rapids.memory.pinnedPool.size		8G	
spark.rapids.sql.concurrentGpuTasks		4	
spark.plugins	org..apache.gluten.GlutenPlugin	com.nvidia.spark.SQLPlugin	Plugin

NVIDIA Decision Support Benchmark 3TB Gluten+Velox

Apache Spark 3.4.2, RAPIDS Accelerator 24.06





Grace Hopper (GH200) 10 Node Cluster

Parquet data stored on HDFS

	CPU Cores	CPU Mem (GB)	Network BW (Gbps)	Storage	GPU	Retail Price / node
Quanta GH200	72	512	100	4 x 3.8TB local SSD	H100	$45763

Grace Hopper Configuration

	GPU	Config type
spark.executor.cores	16	Resource
spark.executor.memory	16G	
spark.rapids.filecache.enabled	true	
spark.executor.resource.gpu.amount	1	
spark.task.resource.gpu.amount	0.0625	
spark.locality.wait	0s	Scheduling
spark.sql.files.maxPartitionBytes	2GB	
spark.shuffle.manager	com.nvidia.spark.rapids.spark341.RapidsShuffleManager	Shuffle
spark.rapids.shuffle.multiThreaded.{reader|writer}.threads	32	
spark.plugins	com.nvidia.spark.SQLPlugin	GPU
spark.rapids.memory.host.spillStorageSize	32G	
spark.rapids.memory.pinnedPool.size	8G	
spark.rapids.sql.concurrentGpuTasks	4	

Grace Hopper (GH200) 10 Node Cluster

Apache Spark 3.4.1, RAPIDS Accelerator 24.06