

## Welcome



#### Who are we?

We provide our customers and partners a new way to design and build data systems.







#### **Standards Backed Data Systems**



### An engine to center a complex ecosystem



A GPU-Accelerated Query Engine for Large-Scale ETL



1

Accelerator-Native:

Distributed query engine built from the ground up to take advantage of full system hardware acceleration

A GPU-Accelerated Query Engine for Large-Scale ETL



#### Accelerator-Native:

Distributed query engine built from the ground up to take advantage of full system hardware acceleration



#### Petabyte Scale:

Focusing on problems too big and time sensitive for Spark

A GPU-Accelerated Query Engine for Large-Scale ETL







#### Petabyte Scale:

Focusing on problems too big and time sensitive for Spark



#### Composable:

Built on open source standards that enables interoperability from storage to application

A GPU-Accelerated Query Engine for Large-Scale ETL



#### **Accelerator-Native:**

Distributed query engine built from the ground up to take advantage of full system hardware acceleration



#### Petabyte Scale:

Focusing on problems too big and time sensitive for Spark



#### Composable:

Built on open source standards that enables interoperability from storage to application

4

#### Evolutionary:

A composable engine that grows over time so new languages and tools can leverage the power of accelerated data processing Next up:

# Why accelerator-native now? Why we built another distributed processing engine.

**Keith Kraus,** CTO & co-founder Voltron Data



## What is a data processing engine?

What does it look like at its core? How should I think about it?



#### Amdahl's Law

"

The overall performance improvement gained by optimizing a single part of a system is limited by the fraction of time that the improved part is actually used.

- Gene Amdahl

### Compute

CPUs have caused compute to become the bottleneck today



## Compute

#### How do GPUs alleviate the compute bottleneck from CPUs?



✓ Tens of thousands of cores

✓ Multiple TB/s of memory bandwidth ✓ Never run out of computational power relative to your memory bandwidth!

23.5 GB/s

#### New Bottleneck

Speeding up the compute just moves the bottleneck elsewhere, Networking and Storage



### • Networking and Storage

Need to be aware of the hardware when architecting the software



## **Accelerator-Native**

What does an ideal data system look like?

GPUs as offload
Processors

**PRO:** Both processors, the task agnostic CPU and the accelerated GPU, work together.

**CON:** Shipping data back and forth over PCIe is an order of magnitude slower than GPU memory bandwidth.

#### GPUs as primary Processors

**PRO:** Multiple GPUs coexist and operate on the same data.

**CON:** A distributed system transfers data many times between nodes - adding overhead. Networking quickly becomes a bottleneck.

## Accelerate the full-system

Accelerated compute with high memory bandwidth memory

Accelerated networking with RDMA

Accelerated storage with GPU Direct Storage (GDS)

## **Accelerator-Native**

Can't just leverage acceleration technology, need to embrace it and build around it

- Existing engines are built as monoliths making it difficult and brittle to integrate acceleration technology
- It's not feasible to change all of their behaviors to actually make accelerator technology integrations yield the possible speedups
- Run face first into Amdahl's Law



#### Theseus

What is different about Theseus?

Built from the ground up as a composable system to integrate new acceleration hardware, software, and techniques

#### Pluggable:

- Compute
- Network
- Storage

Built from the ground up to give the system level control needed to effectively leverage said acceleration hardware, software, and techniques

#### Control:

- Planning
- Sizing
- Scheduling
- Placement
- And more...

#### Theseus

Acceleration is only useful if easily and readily accessible.



✓ Data portability for workflows built on standards

✔ De-risk vendor lock-in

#### Theseus

System Level Architecture

A modular architecture built to evolve

- Multi-language (Python, SQL, R\*) •
- Multi-silicon (GPU, CPU) •
- Multi-node engine powered by Theseus •

**Flight Client** 



### • Theseus

GATEWAY (Flight Server / Client)

Relational Algebra + Data Source



## **Theseus TPC-H Performance at 10TB**



Note: Unofficial TPC-H run comparing Theseus: 1 Node = 8 x A100 80 GB, Spark: 1 Node = r5.8xlarge (AWS) 32 VCPU 32 GB | Date: March 2024

## **Theseus TPC-H Performance at 100TB**

X

X

X

X

Date: March 2024



www.voltrondata.com/benchmarks

If you have problems larger than 30TB, and existing engines like Spark or Trino aren't cutting it for you, let's keep the conversation going!

Email info@voltrondata.com

## VOLTRON DATA