The Gluten Open-Source Software Project: Modernizing Java-based Query Engines for the Lakehouse Era

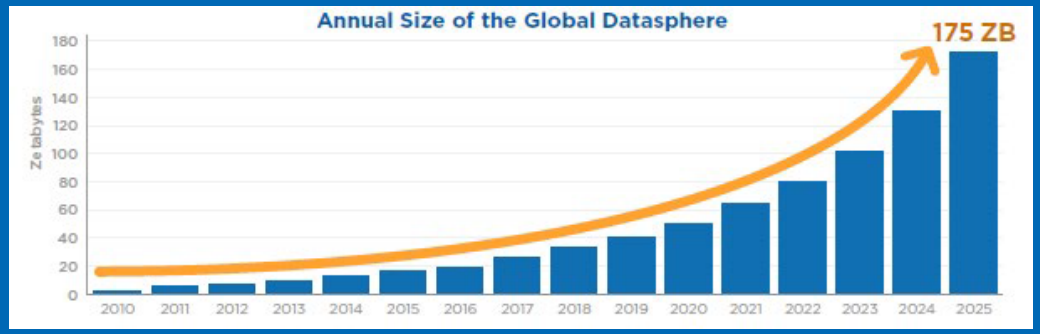
Presenter: Akash Shankaran

Authors: Akash Shankaran, George Gu, Weiting Chen, Binwei Yang, Chidamber Kulkarni, Mark Rambacher, Nesime Tatbul, David E. Cohen

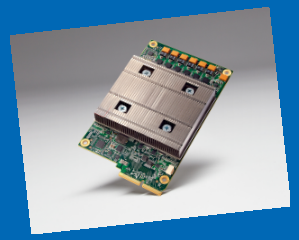
Intel Corporation

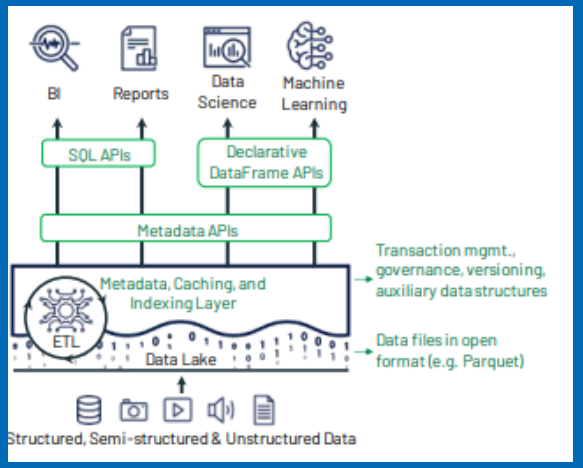
Background – Disruptive Trends

- Exponential growth in compute demand, largely driven by data supporting machine learning (ML), training models.
- Such data is increasingly processed over custom, heterogeneous hardware: CPUs, GPUs, TPUs, FPGAs, etc.
- Data lakehouses: Rise of open table formats, decouple query engine from storage, ability to perform mutable transactions on tables.



https://www.i-scoop.eu/big-data-action-value-context/data-age-2025-datasphere/





Motivation – SparkSQL as a Service

SparkSQL has been an **engine of innovation** in this market.

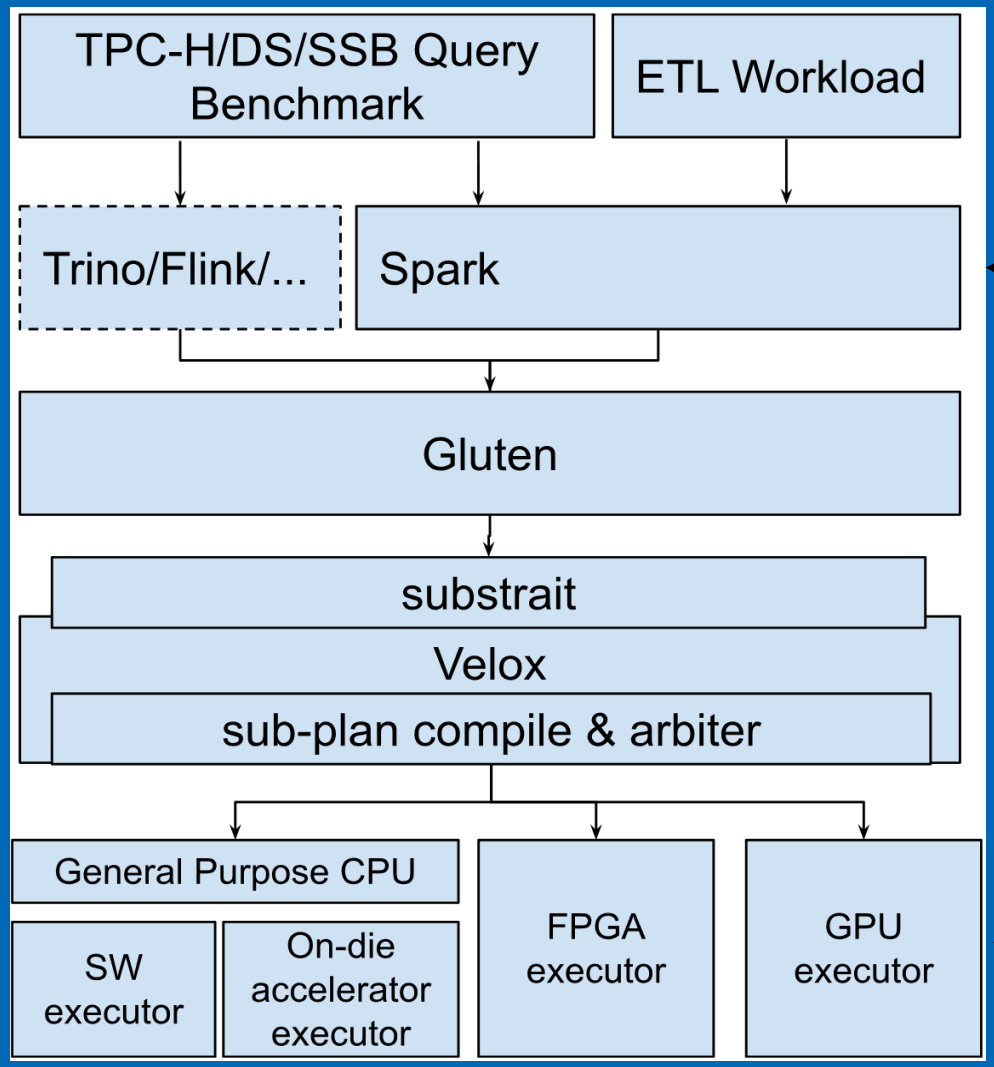
The shift towards lakehouses has created opportunities for SparkSQL-as-a-Service providers.

For external customers:	For internal constituents:
❖ Hyperscalers: Amazon EMR, Azure Synapse, Google Dataproc, Alibaba EMR, etc. ❖ Databricks (available on all public clouds)	❖ Pinterest, Meituan, Netflix, Stripe, etc. ❖ Operates at a scale that rivals the hyperscalers.

Motivation for an **open-source initiative** (Gluten):

- ❖ The Photon solution is not available to the Spark community.
- ❖ Due to scale/TCO for above customers, the Delta Lake approach is not feasible.

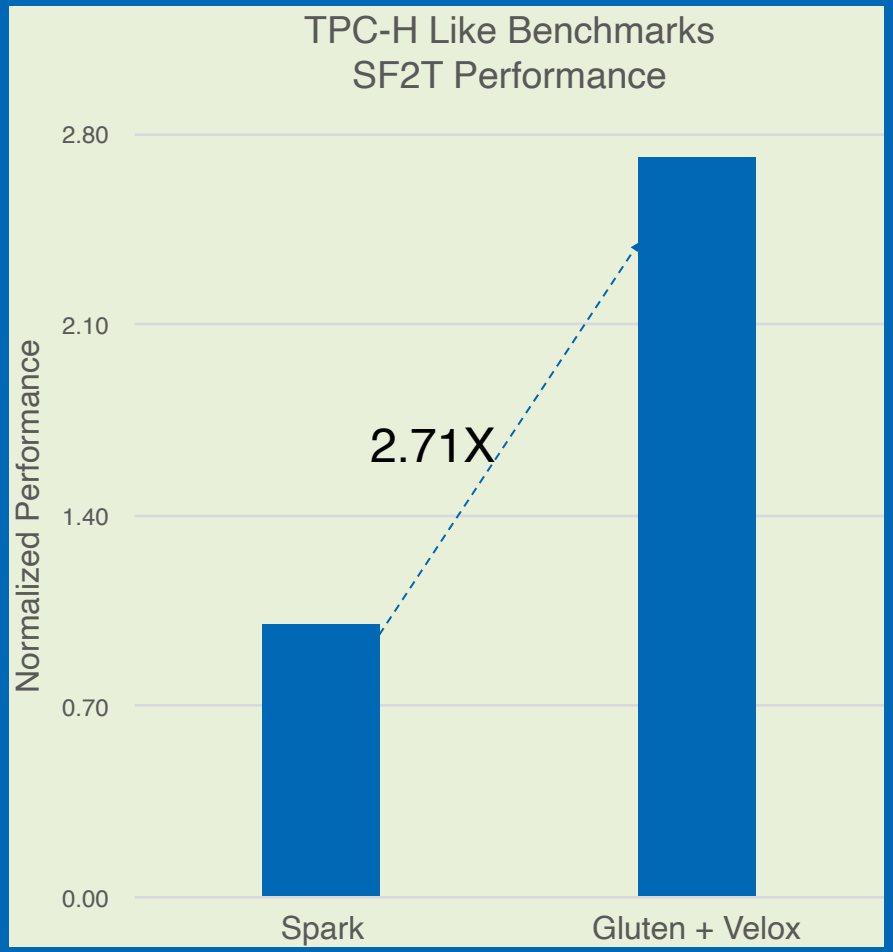
The Gluten Open-Source Software Project

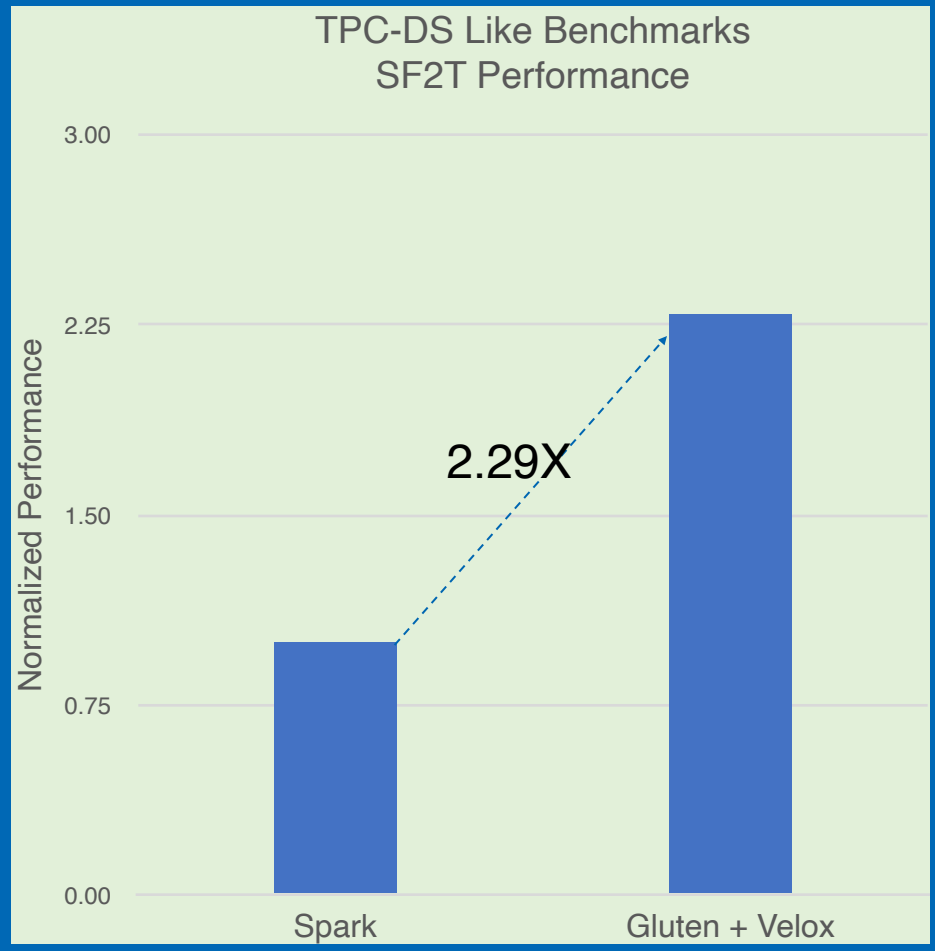


We envision Gluten to:

- ❖ support a variety of workloads & execution frameworks
- ❖ generalize across query engines (e.g., Velox, Clickhouse, Arrow)
- ❖ run on heterogeneous hardware (e.g., CPUs, GPUs, FPGAs)

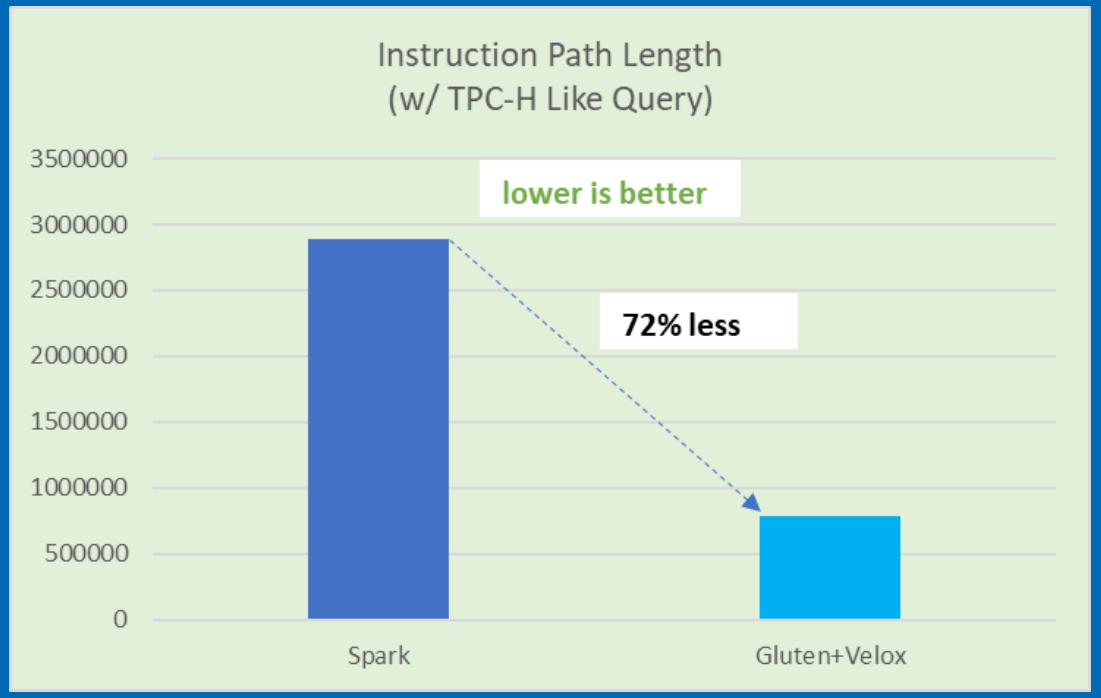
Performance Characterization
Spark vs. Spark+Gluten+Velox

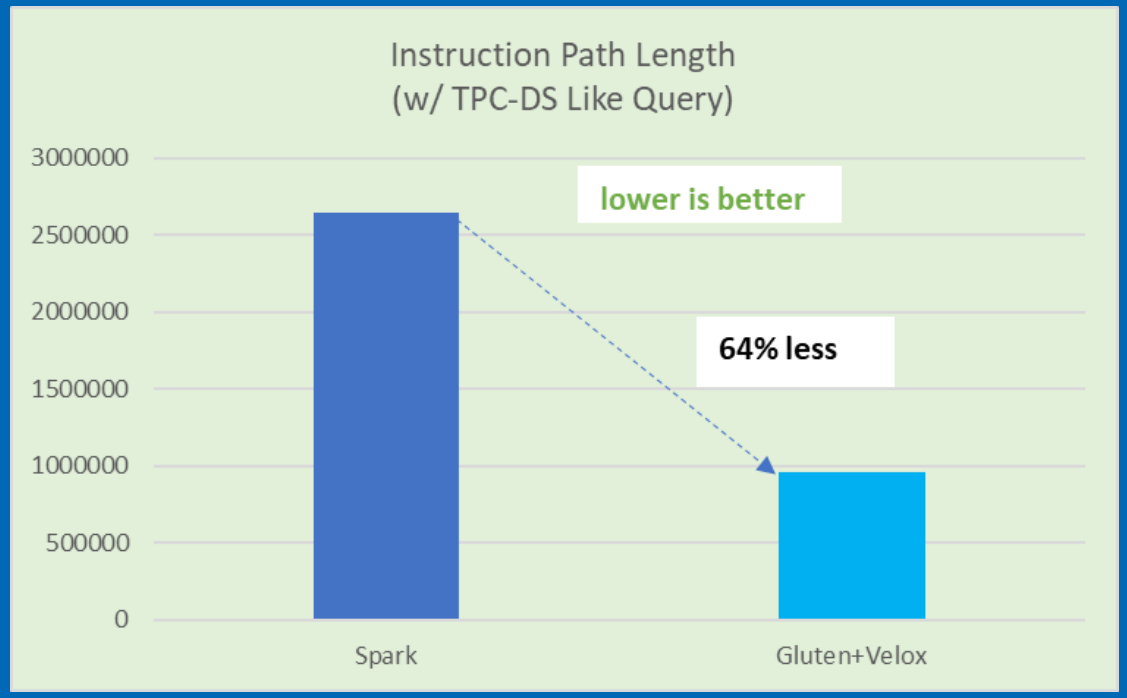




Performance Characterization
Microarchitectural Analysis

Instruction Path Length with and without Gluten+Velox



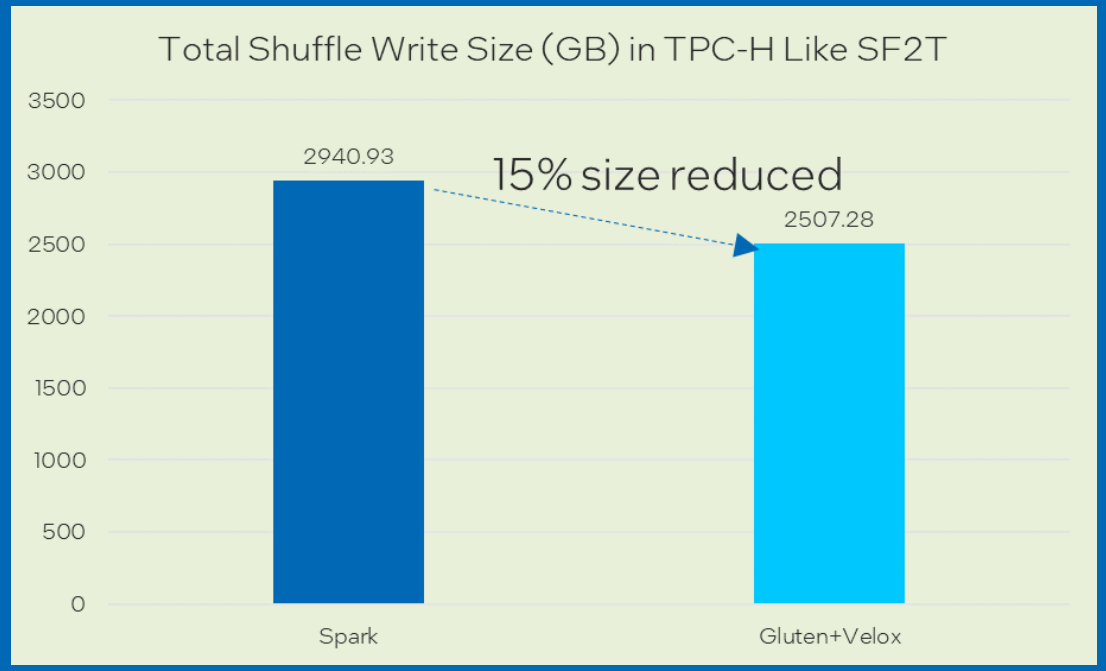


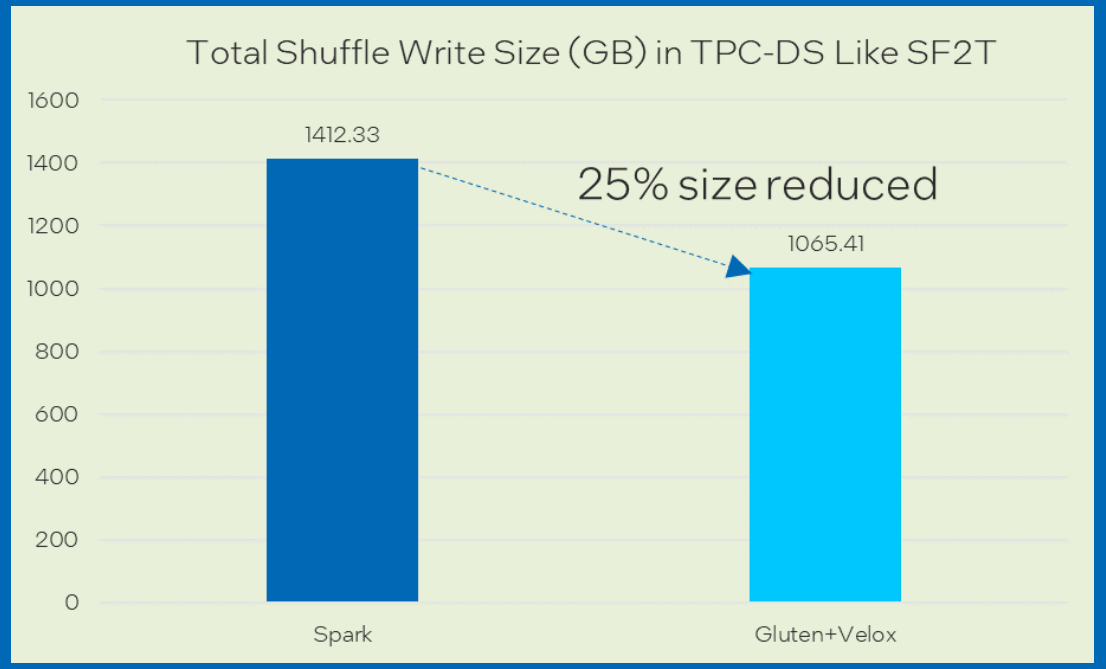
Reduction in Path Length leads to huge performance savings.

- fewer instructions required to produce the same output

Performance Characterization
Shuffle Write Reduction

Shuffle Write Sizes with and without Gluten+Velox (compression: lz4)





- Columnar-based shuffle can take advantage of compression codecs (e.g., lz4, zlib, zstd).
- Users can choose codec based on columnar data type to get higher compression ratio for better performance.

Gluten Community Participation and Adoption

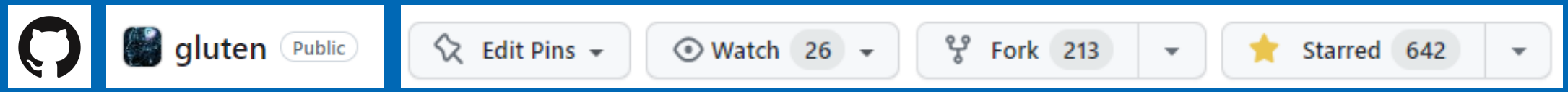
Active Community Participation:

- 10+ companies across the globe have made contributions to the Gluten Project.

Increasing Adoption:

- Strong interest from various hyperscalers and large Spark customers in trying Gluten for their offerings.

https://github.com/oap-project/gluten



Future Directions

- Develop an Application Binary Interface (ABI)
- Extend Gluten as a JNI bridge to other query engines
- Target heterogeneous hardware
- Offload Spark shuffle to hardware accelerators (e.g., Intel-QAT) for additional performance improvements

With Velox and Substrait, extend composability to the underlying heterogeneous hardware, supporting the vision of Composable Data Management Systems.

intel

BACKUP

Configurations

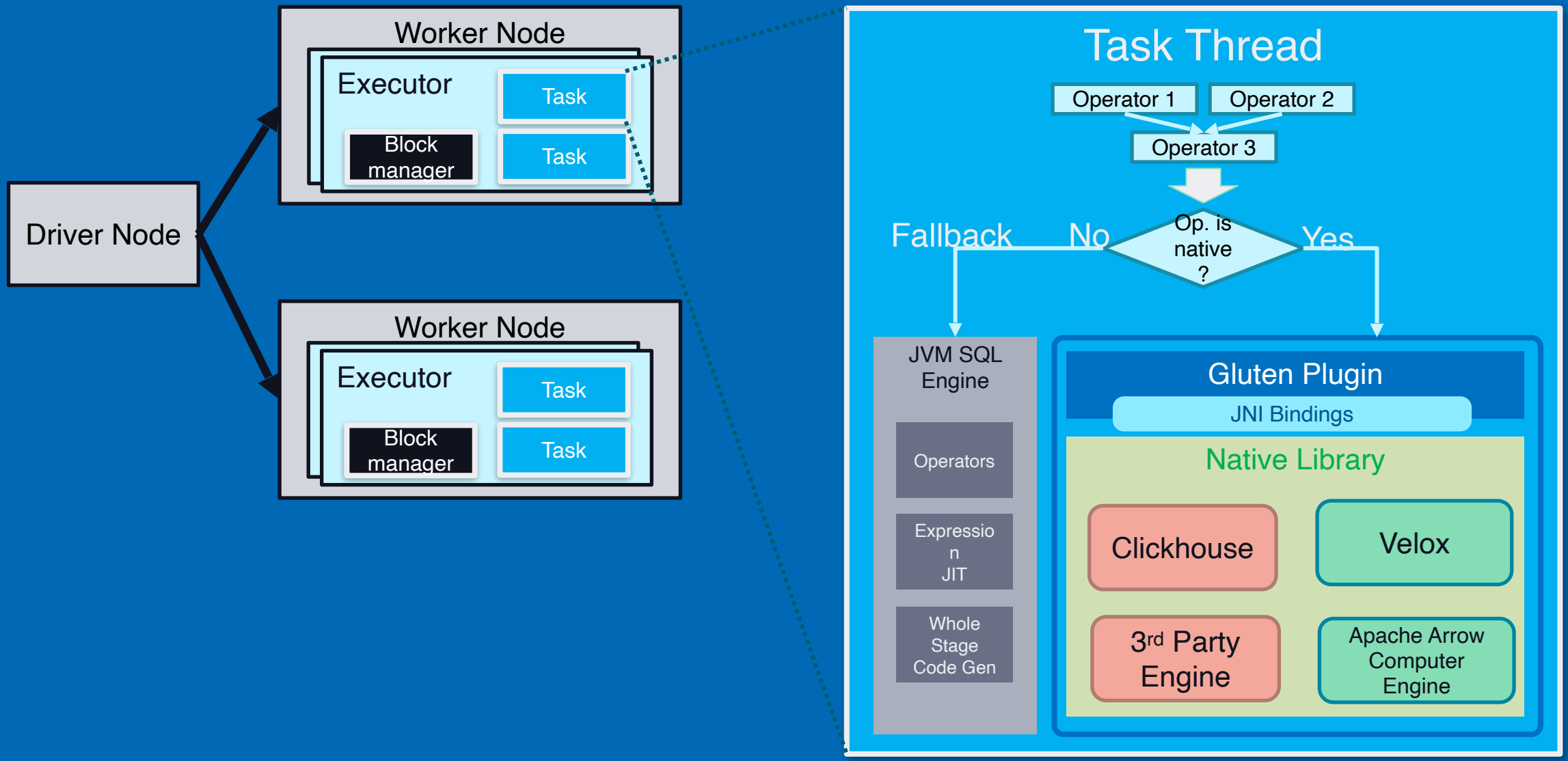
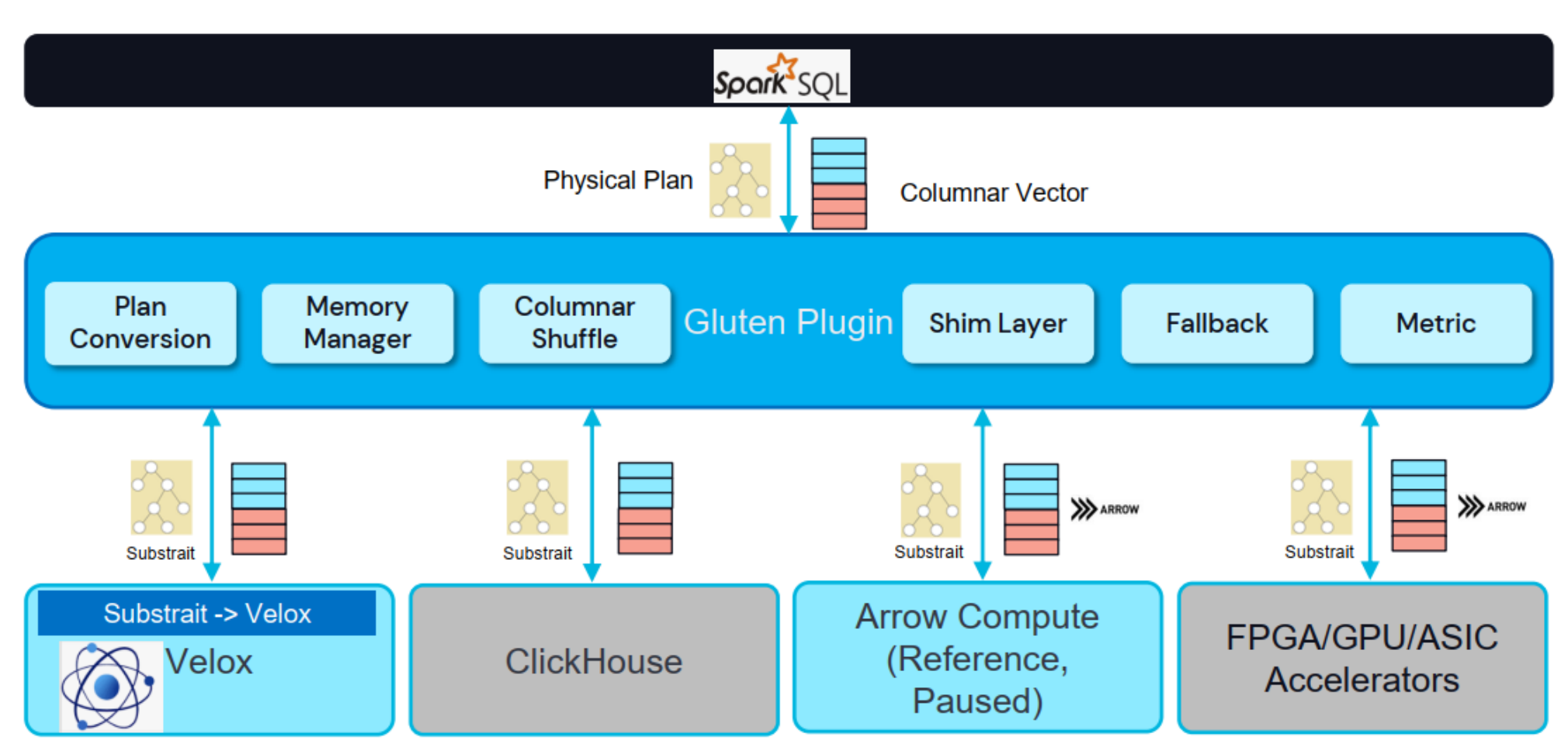
Name	Hardware Platform
CPU Model	Intel® Xeon® Platinum 8480+
Micro-architecture	Sapphire Rapids
CPUs	224
Memory	1024GB
NIC	1x Ethernet Controller I225-LM 1x Ethernet interface
Disks	2x 1.5T INTEL SSDPE2KE016T8 1x 447.1G INTEL SSDSC2BB48 1x 447.1G INTEL SSDSC2KB48 7x 3.5T INTEL SSDPF2KX038TZ

Table 1: Hardware Configuration.

Name	Software Platform
Operating System	Ubuntu 22.04.1 LTS
Linux Kernel	5.16.0-051600rc5-generic
JDK version	1.8
GCC version(Gluten only)	11
Spark version	3.3.1
Hadoop version	3.2

Table 2: Software Configuration.

Gluten working model



Gluten - architecture