Project #2: Monday May 1st

Project #3: Friday May 5th @ 5:30pm

Final Exam: Sunday May 7th @ 11:59pm
→ Submit on Gradescope
→ Please also fill out the form on whether you use ChatGPT.

Amazon Redshift Guest Lecture (In-Class)
→ Wednesday April 26th @ 2:00pm ET
OBSERVATION

Since the late 2000s, Facebook/Meta has made contributions to DBMS development.
→ They still run the largest MySQL deployment in the world.

But unlike some of the other major tech companies, they have open-sourced nearly all their projects.
DATA SYSTEMS AT META

- Scribe (2008)
- Hive (2010)
- Scuba (2013)
- PrestoDB (2013) ➡️ Ahana, Trino/Starburst
- WebScaleSQL (2014)
- Gorilla/Beringei (2015)
- LogDevice (2017)
- Velox (2021)

Not Open-Source ➡️
DATA SYSTEMS AT META

- Scribe (2008)
- Cassandra (2008)
- Hive (2010)
- RocksDB (2012)
- Scuba (2013)
- PrestoDB (2013)
- WebScaleSQL (2014)
- Gorilla/Beringei (2015)
- LogDevice (2017)
- Velox (2021)

Not Open-Source →

- DataStax
- Ahana

IBM acquires SaaS-based PrestoDB provider Ahana

IBM’s will join the Presto Foundation and aid in the development of PrestoDB.

By Anirban Ghoshal
Senior Writer, InfoWorld | APR 14, 2023 5:19 AM PDT

IBM has acquired Ahana, a software-as-a-service (SaaS)-based provider of PrestoDB, for an undisclosed sum.

PrestoDB, or Presto, is an open source, distributed SQL query engine created at Facebook (now Meta) that is tailored for ad hoc analytics against data of all sizes.

IBM said that its acquisition of Ahana is in line with its strategy to invest in open source projects and foundations. The company acquired Red Hat in 2018, cementing its open source strategy.

"IBM is now a prominent contributor to open source communities — working across the cloud native ecosystem, artificial intelligence, machine learning, blockchain, and quantum computing. One example is our role as a founding member of the Cloud Native Computing Foundation (CNCF), which fostered the growth of Kubernetes. We see our involvement with Presto Foundation as a similar relationship," IBM’s vice president of hybrid data management Vikram Murali and CEO of Ahana Steven Mih wrote in a joint statement.
HISTORICAL CONTEXT

Meta collects a lot of data. Over the last decade, their engineering teams have built many (redundant) internal tools to analyze this data. This fracturing has led to wasted efforts and inconsistent features/performance.

Like Databricks, Meta recognized that it was not feasible to replace all the existing systems with a new one...
META VELOX

Extensible C++ library to support high-performance single-node query execution.
→ No SQL parser!
→ No meta-data catalog!
→ No cost-based optimizer!

Velox takes in a physical plan (DAG of operators) as its input for execution. It then produces the output to the specified location.
**VELOX OVERVIEW**

Push-based Vectorized Query Processing
Precompiled Primitives + Codegen Expressions (C++)
Arrow Compatible (extended)
Adaptive Query Optimization
Sort-Merge + Hash Joins
VELOX COMPONENTS

Type System
Vector Internal Representation
Expression Engine
Function API
Operator Engine
Storage Connectors / Adapters
Resource Manager

Source: Pedro Pedreira
Velox does not "own" data and it does not have a proprietary data format.

Instead, it exposes APIs to define connectors to retrieve data from systems and adapters to decode/encode storage formats.

→ Systems: S3, HDFS
→ Formats: Parquet, ORC/DWRF, Alpha
VECTOR INTERNAL REPRESENTATION

Velox uses in-memory vectors to move data to and from operators at runtime.
→ Extends Apache Arrow columnar layout to support more encoding/compression schemes.

Optimizations:
→ Lazy Vector Materialization
→ German-style String Storage
→ Out-of-order Writes/Population

"AndyP"

"AndyP smells bad!"

"AndyP smells bad!"
Velox converts expression trees into a flattened intermediate representation that they then execute during query processing. → Think of it like an array of function pointers to precompiled (untemplated) primitives.

**Experimental branch** transpiles IR into C++ code and then compiles to machine code via exec.
QUERY ADAPTIVITY

Predicate Reordering
Column Prefetching
Elide ASCII Encoding Checks
→ Bonus: Reuse buffers for output!

Source: Pedro Pedreira
CMU-DB
15-721 (Spring 2023)
PRESTISSIMO

Replace PrestoDB's Java-based runtime engine with Velox-based engine.
→ Think of it like Databricks replacing Spark SQL runtime with Photon via JNI.

Uses Velox API to re-implement SQL functions and operators beyond base Velox to provide compatibility with PrestoDB.
Replace PrestoDB's Java-based runtime engine with a Velox-based engine.

→ Think of it like Databricks replacing Spark SQL runtime with Photon via JNI.

Uses Velox API to re-implement SQL functions and operators beyond base Velox to provide compatibility with PrestoDB.

In essence, the direction of Trino while still including bare metal performance improvements in the JVM, will instead focus on not wasting time with suboptimal query plans before or during runtime. There are so many optimizations that are constantly being added to every release that ultimately makes for a work-smarter not-harder query engine.
SUBSTRAIT (2021)

Open-source specification to represent relational algebra query plans. → Think of it like Arrow but for query plans.

The idea is that systems can share physical query plans with each other without having to convert them into a native API/DSL. → Federated DBMSs are hard.
Extensible vectorized execution library for Apache Arrow data.
→ Written in Rust for the kids!

Provides more front-end functionality features to build a complete DBMS than Velox
→ SQL and DataFrame APIs.
→ Query Optimizer

Examples: InfluxDB, CeresDB, CnosDB, Seafowl
POLARS (2020)

Extensible vectorized execution library for Apache Arrow data.
→ Written in Rust for the kids!

Provides more front-end functionality features to build a complete DBMS than Velox
→ SQL and DataFrame APIs.
→ Query Optimizer

Examples: ???
Instead of building an OLAP DBMS from scratch or forking an existing DBMS, starting with something like Velox that was designed to be embedded + extended seems like the better move.

That means the differentiating factors between DBMSs will be UI/UX factors and query optimization.
NEXT CLASS

Amazon Redshift with Ippokratis Pandis (PhD'07)