Building IBM watsonx.data from Composable Parts

Aditi Pandit
IBM
San Jose, California USA
Aditi.Pandit@ibm.com

ABSTRACT

In this talk, we discuss how we built IBM watsonx.data, an open, hybrid and governed data lakehouse, by composing modular open-source and proprietary technology.

VLDB Workshop Reference Format:
Aditi Pandit. Building IBM watsonx.data from Composable Parts. VLDB 2025 Workshop: Third International Workshop on Composable Data Management Systems.

VLDB Workshop Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://www.ibm.com/products/watsonx-data.

1 INTRODUCTION

IBM watsonx.data is an open, hybrid and governed data lakehouse optimized for enterprise data and AI workloads.

The platform architecture is based on ideas of composability in modern data management systems and the usage of modular components, both open-source and proprietary [4]. As a result the product has a consistent user experience, leverages open standards to eliminate vendor lock-in, and streamlines new development to avoid wasting engineering cycles. In this talk, we will outline the key design decisions in watsonx.data to achieve these goals.

This product was built in just over 3 years by a team with an open-source first mindset. Several components in this tech stack are also used in Big Tech companies, most notably at Meta and Uber.

2 WATSONX.DATA ARCHITECTURE

The composability and modularity in watsonx.data happens at several layers of the development stack.

2.1 Storage

Users can share a single copy of data between engines through open table formats like Apache Iceberg. The openness facilitates collaboration and eliminates vendor lock-in of prior IBM products.

2.2 Execution Engine

To deliver on the breadth of all analytics and AI usage, users can choose between multiple fit for purpose open-source engines:

- Presto C++ for BI workloads
- Spark/Gluten for machine learning
- Milvus vector database for GenAI

Both Presto C++ and Spark Gluten replace their original Java/Scala execution with a new native runtime.

This runtime is based on the Velox library [3]. Velox is a pioneer in composable query execution as it provides a library of re-usable data processing primitives. A single Velox team built both engines.

2.3 Optimizer

Good query plans are critical for performance. Watsonx.data uses IBM proprietary DB2 optimizer in a Presto query plugin, and executes the generated plans with the C++ engine. By gluing these, it gets the benefits of statistics and runtime optimizations. This strategy lead it to rival its competitors on TPC-DS 100 TB workloads-[2].

2.4 Metastore

The key to compose modular query engines is to have a common understanding of the table schema and metadata for queries. Watsonx.data uses the Hive Metastore API for this. Users can ingest data with the Gluten engine, and use Presto C++ for reading.

2.5 Data federation

Presto supports federated query across multiple data sources. To avoid re-implementing the old Presto Java connectors, Presto C++ uses Arrow Flight-[1] to abstract Flight protocol servers to retrieve data from data services. watsonx.data uses an Arrow flight service exposed via a Table function for Presto C++ federation.

2.6 Platform services

The engines can be seamlessly deployed across cloud or on-premise environments using OpenShift containers which scale up or down automatically as the workload changes.

In this talk, we will elaborate on each of these aspects and give insight into the benefits and challenges encountered.

REFERENCES

[1] https://arrow.apache.org 2019. *Introducing Apache Arrow Flight: A Framework for Fast Data Transport.* Retrieved 13 Oct 2019 from https://arrow.apache.org/blog/2019/10/13/introducing-arrow-flight/
[2] ibm.com 2024. *Delivering superior price-performance and enhanced data management for AI with IBM watsonx.data.* Retrieved May 27, 2024 from https://www.ibm.com/new/announcements/delivering-superior-price-performance-and-enhanced-data-management-for-ai-with-ibm-watsonx-data
[3] Pedro Pedreira, Orri Erling, Masha Basmanova, Kevin Wilfong, Laith Sakka, Krishna Pai, Wei He, and Biswapesh Chattopadhyay. 2022. Velox: meta's unified execution engine. *Proc. VLDB Endow.* 15, 12 (Aug. 2022), 3372–3384. https://doi.org/10.14778/3554821.3554829
[4] Pedro Pedreira, Orri Erling, Konstantinos Karanasos, Scott Schneider, Wes McKinney, Satya R Valluri, Mohamed Zait, and Jacques Nadeau. 2023. The Composable Data Management System Manifesto. *Proc. VLDB Endow.* 16, 10 (June 2023), 2679–2685. https://doi.org/10.14778/3603581.3603604

Proceedings of the VLDB Endowment. ISSN 2150-8097.