In-Memory Computing SUMMIT VIRTUAL EVENT 2020

Alluxio: Bridging the Gap Between SQL Engines and Storage for Physical Data Independence

Gene Pang, Software Engineer
Alluxio, Inc.
October 29th 2020

ALLUXIO

Agenda

- Overview & Motivation
- Alluxio Structured Data Services for Presto
- Demo

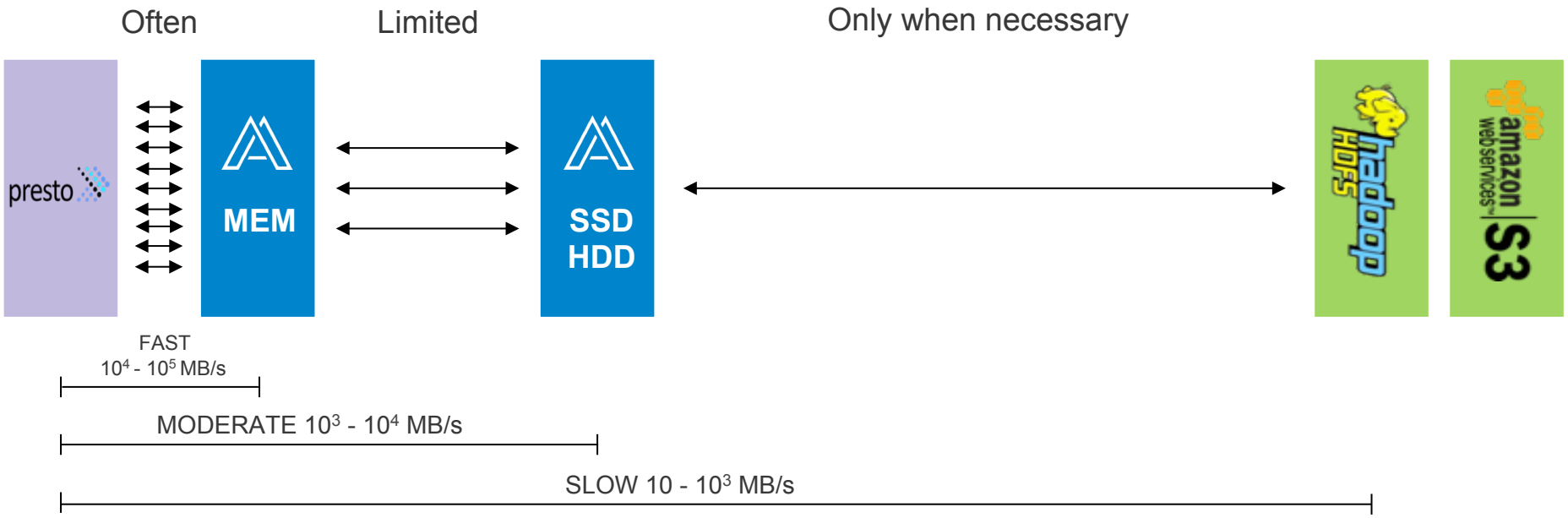
Alluxio – Open Source Data Orchestration

Java File API | HDFS Interface | S3 Interface | POSIX Interface | REST API

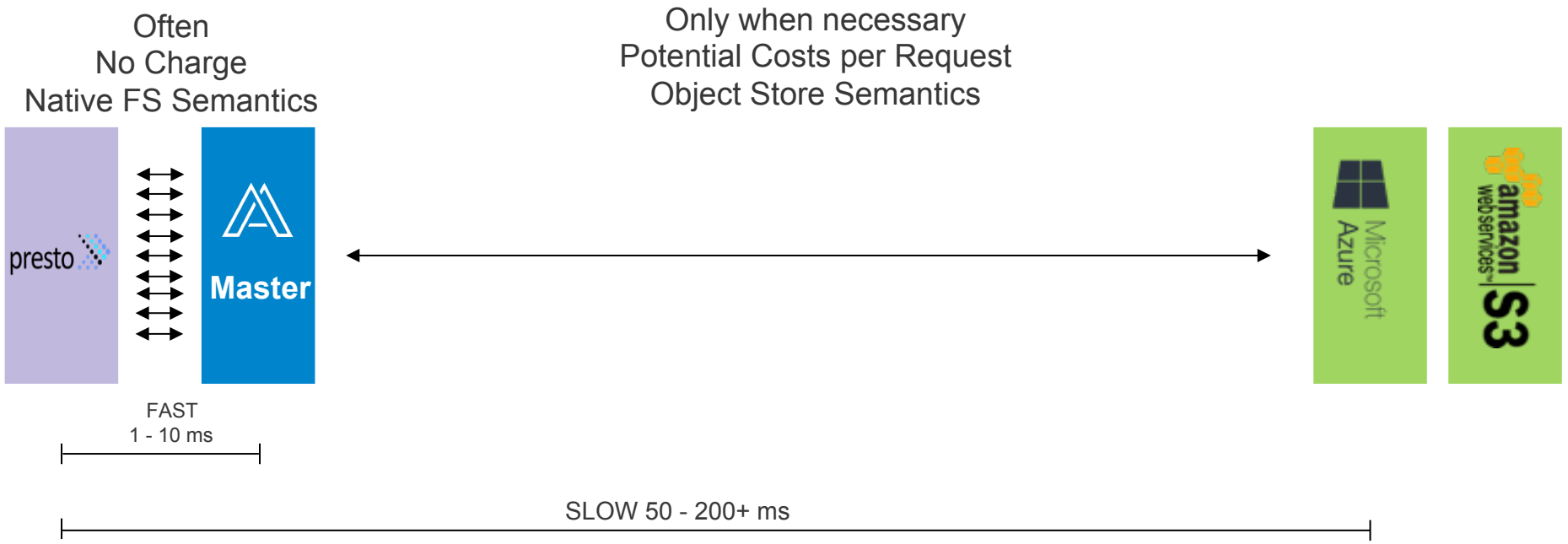
ALLUXIO Data Orchestration for the Cloud

HDFS Driver | GCS Driver | S3 Driver | Azure Driver

Accelerating IO



Improving Metadata Access



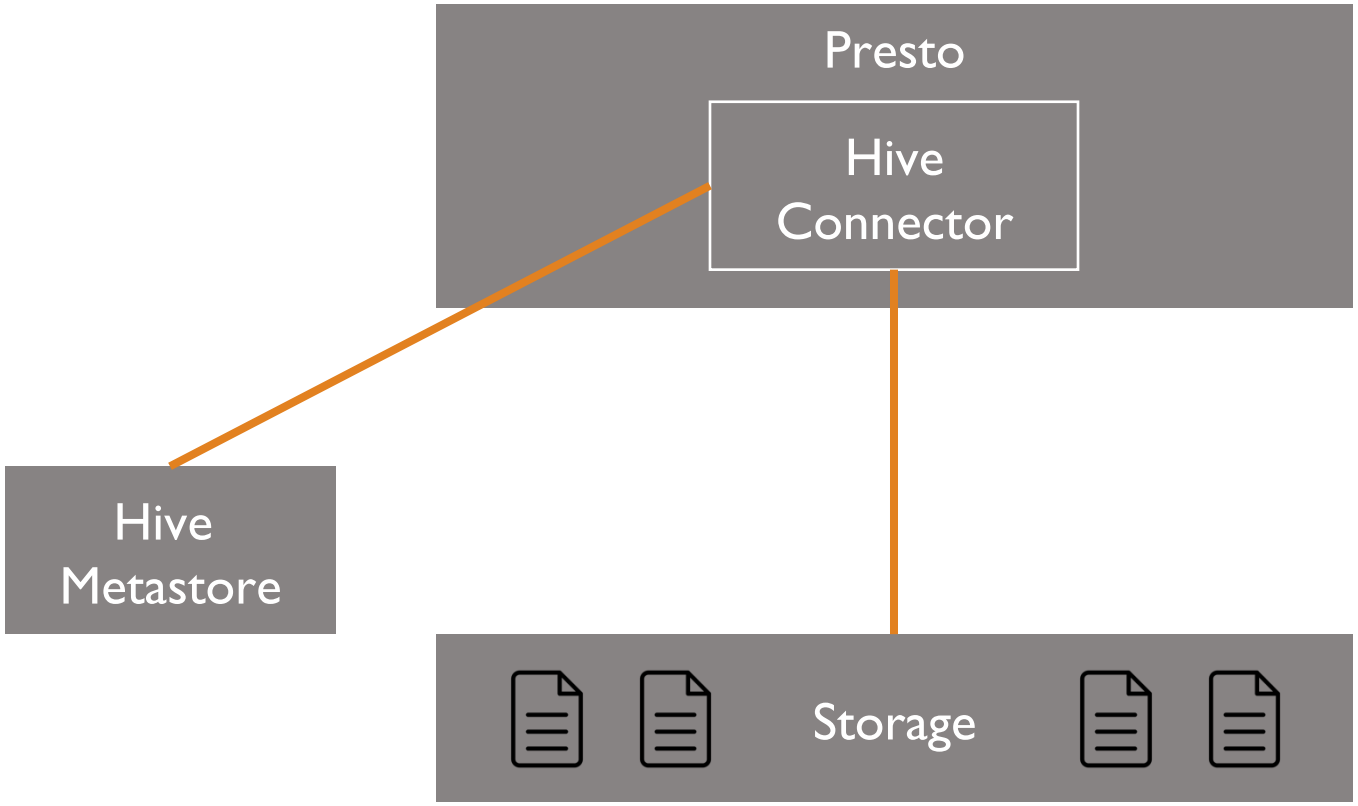
Beyond Files

- Significant portions of query performance are unrelated to file access
- Examples:
  - Hive Metastore operations - dependent on HMS performance
  - Unoptimized file format - CSVs can be much slower than Parquet
  - Inefficient Table Layout - Lack of data clustering, too many partitions or files
- To address these, we developed **Structured Data Services**

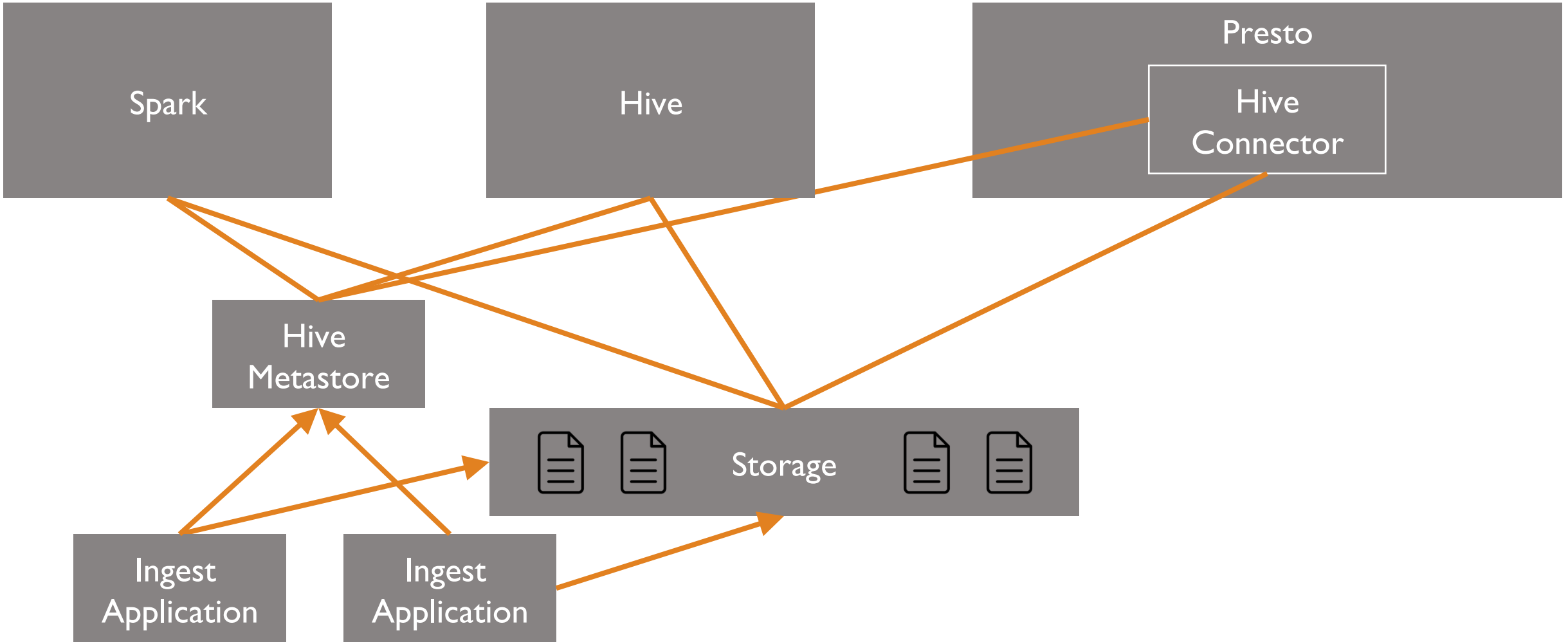
Alluxio Structured Data Services

Overview

Common Presto Environment



Common Presto Environment



Potential Inefficiencies

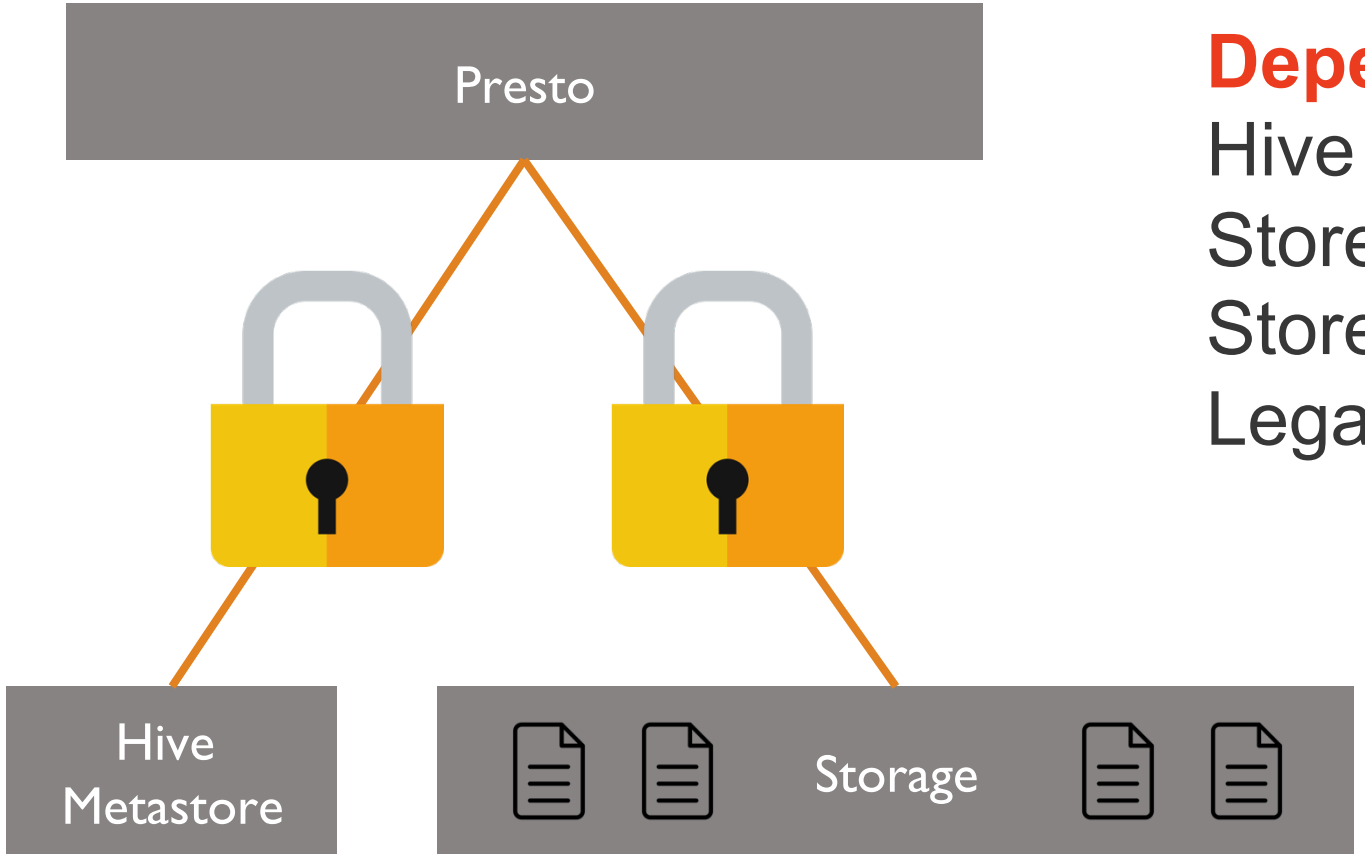
Overloaded/slow Hive Metastore

Un-optimized file formats (CSV vs parquet/ORC)

Inefficient table data organization (too many small files)

Inability to change or update how data is written or stored

Dependence on Hive Warehouse

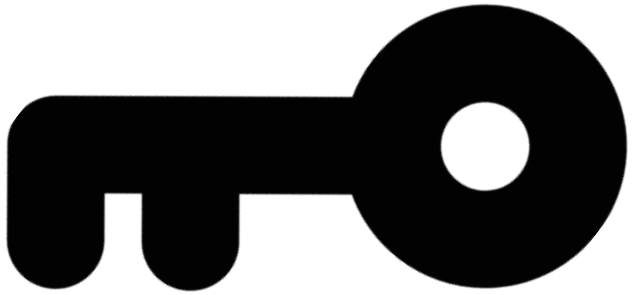


Dependencies

Hive Metastore
Stored File Formats
Stored Data Organization
Legacy Storage Decisions/Apps

How to unlock this dependence?

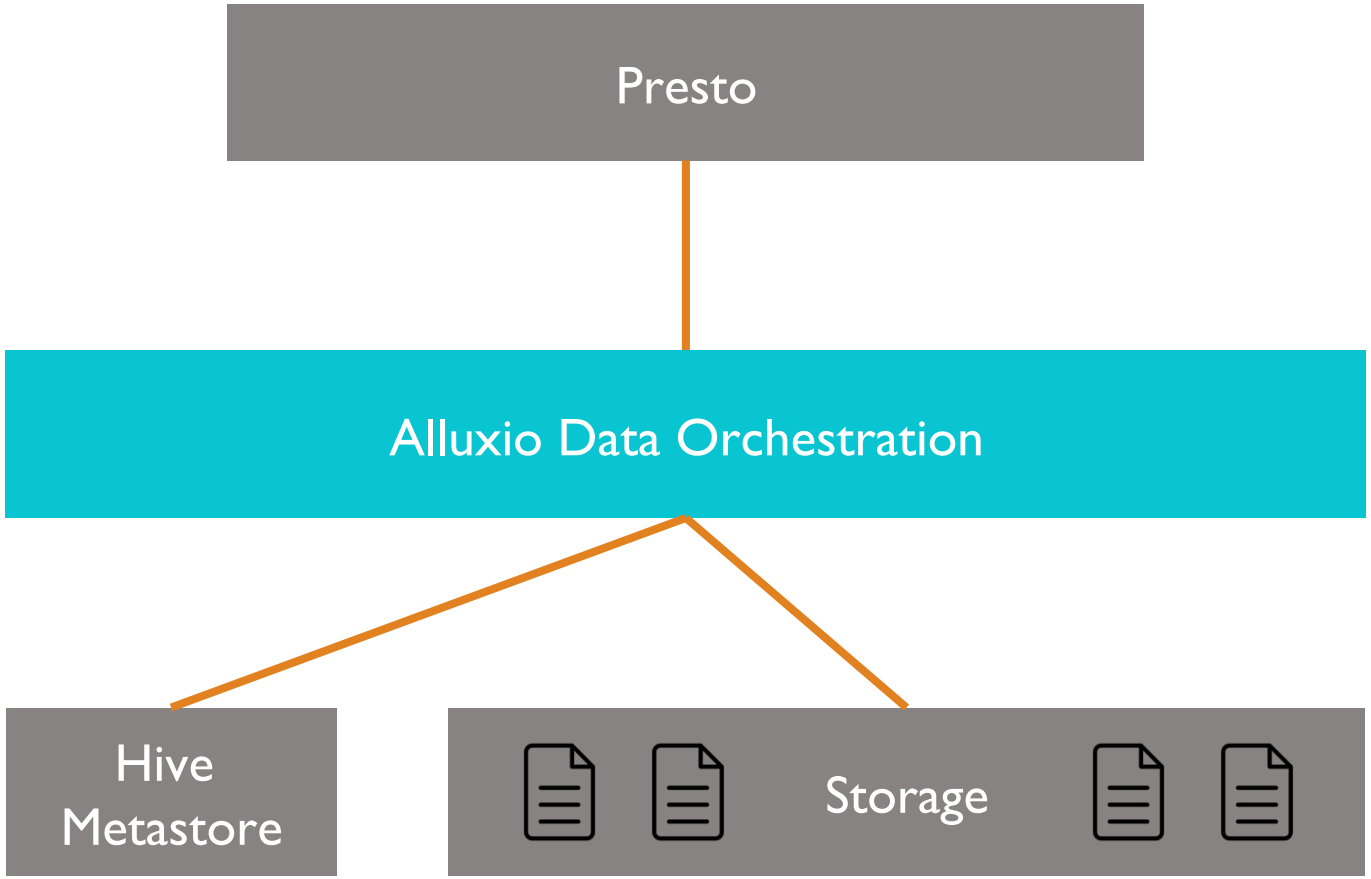
Unlock Potential with Data Orchestration



Decouples Compute from Hive Data Warehouse

Enables Compute-Optimized Data Access

Alluxio Data Orchestration



Decouples Compute from Hive Data Warehouse

Enables Compute-Optimized Data Access

Benefits of Alluxio Data Orchestration

Storage Systems

In-Memory Caching

Unified Interface/Namespace

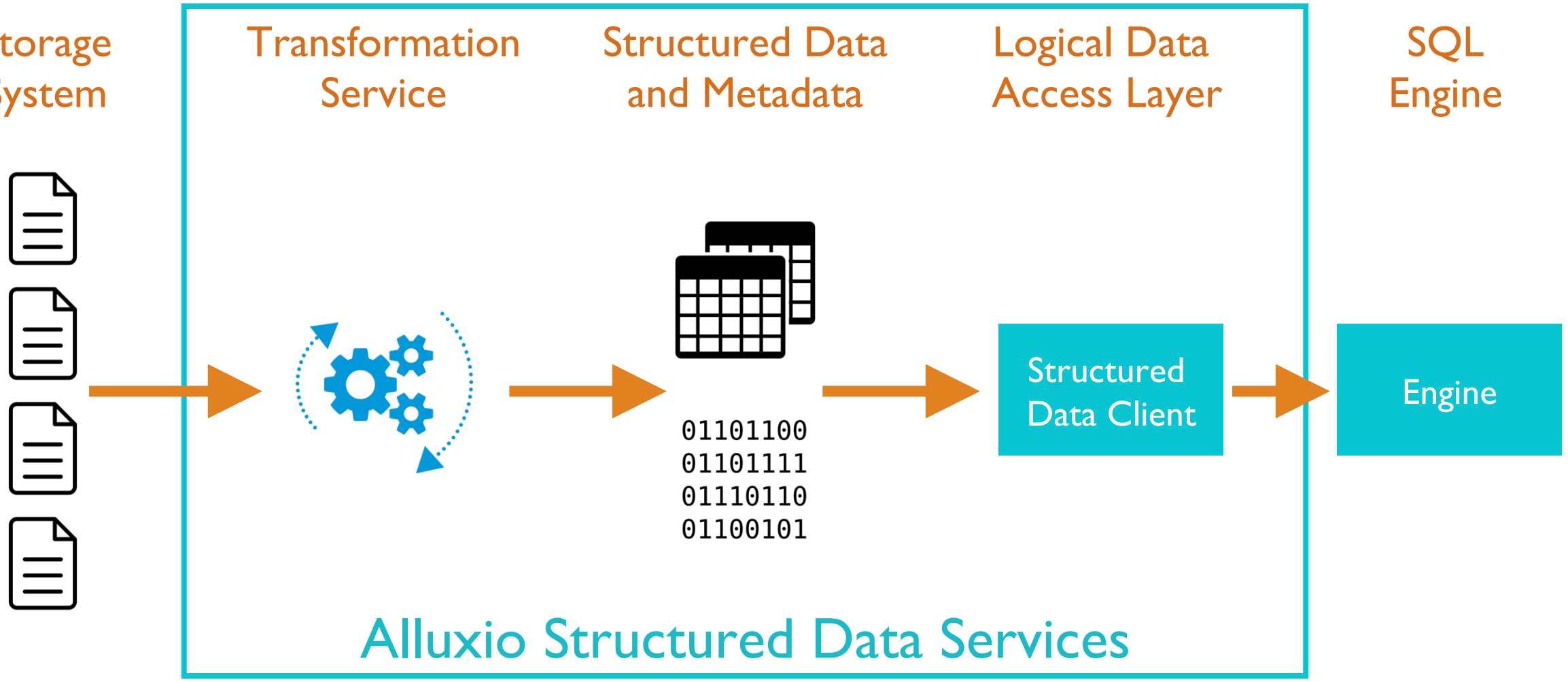
Schema-Aware Optimizations

Compute-Optimized Formats

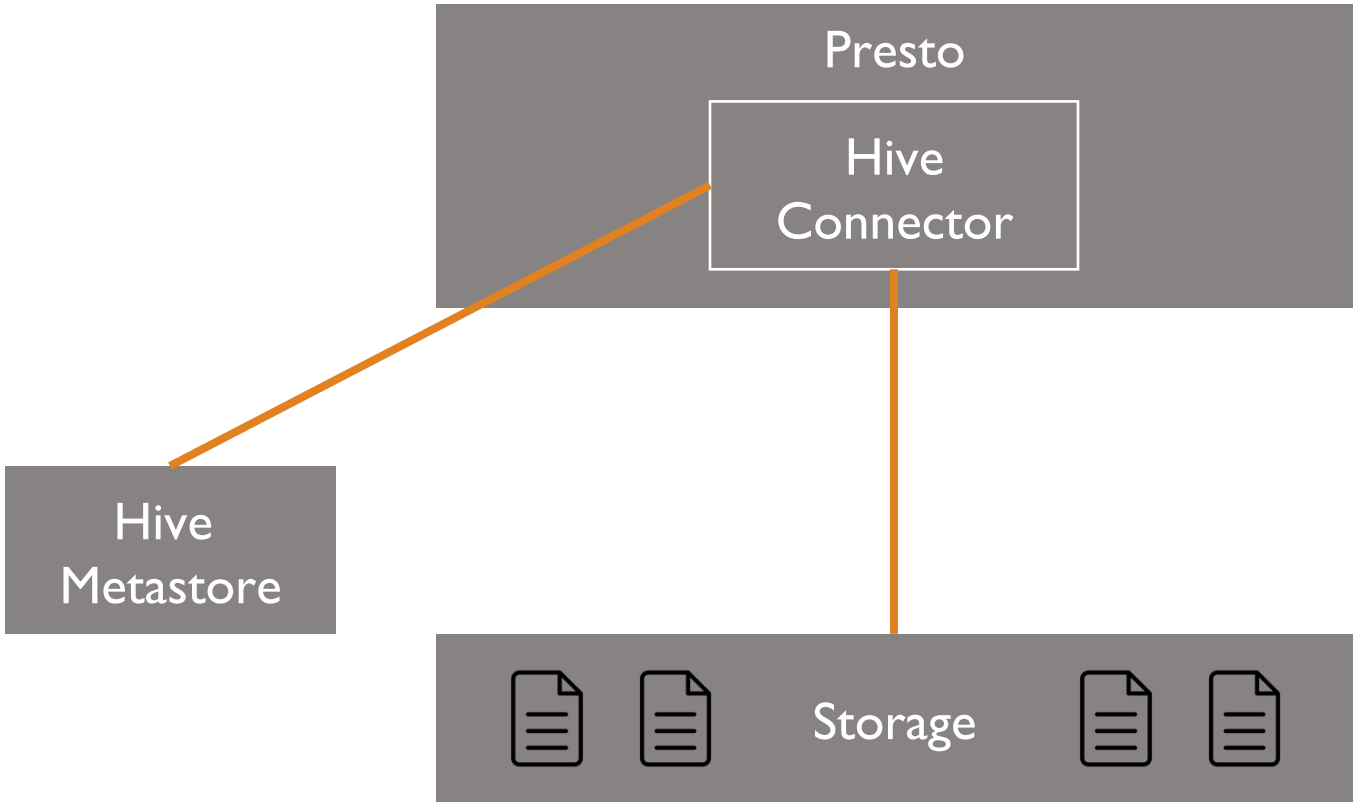
Physical Data Independence

SQL Frameworks (Presto)

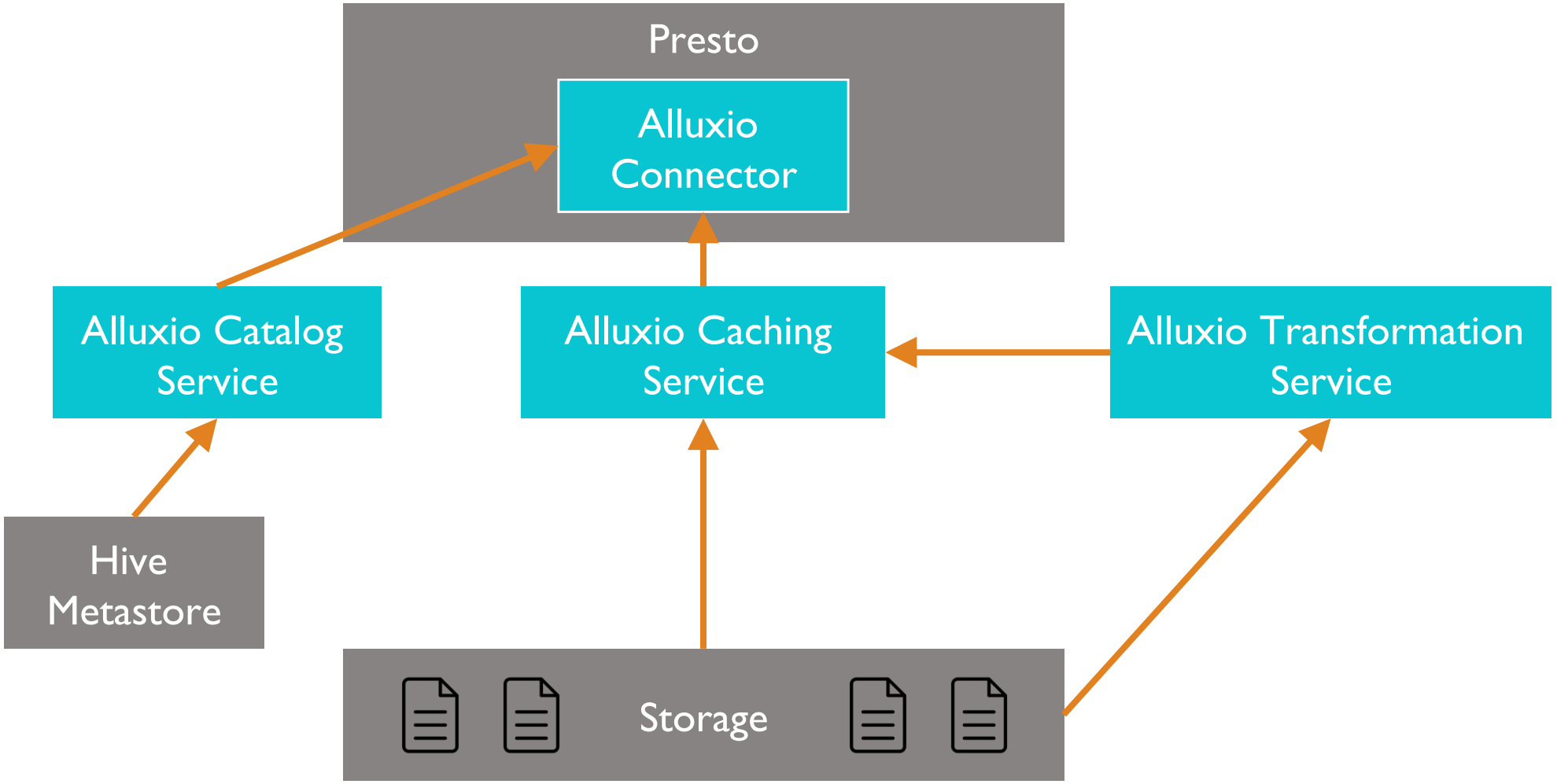
Alluxio Structured Data Services



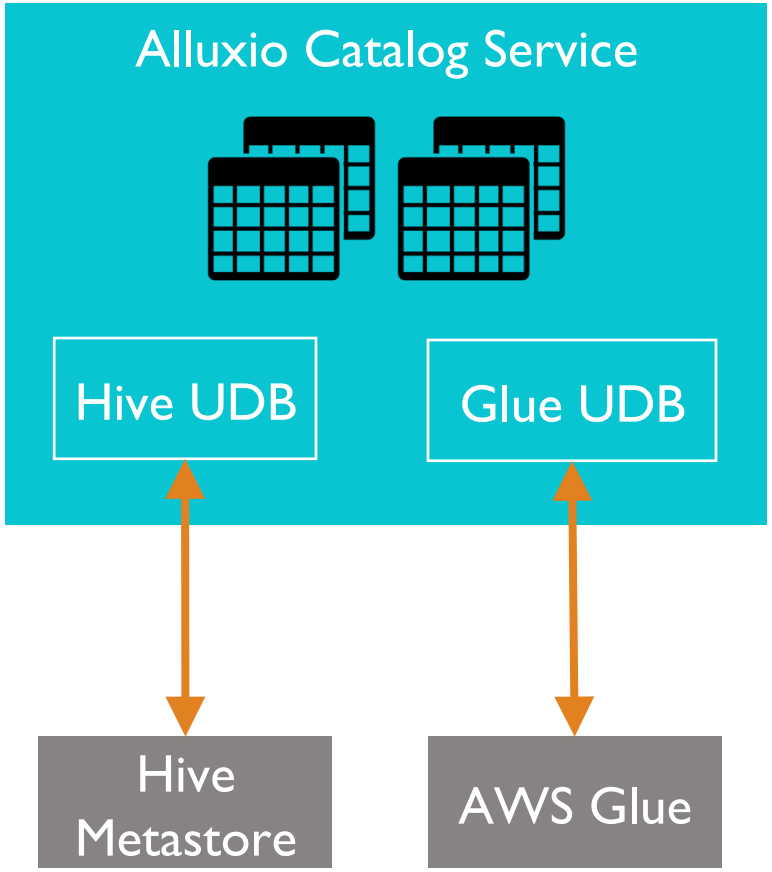
Target Environment



Alluxio Structured Data Services



Alluxio Catalog Service



Functionality

Manages metadata for structured data

Abstracts other database catalogs as Under Database (UDB)

Benefits

Schema-aware optimizations

Simple deployment

Alluxio Presto Connector

Tighter integration with Presto

New plugin based on the Presto Hive connector

Source code merged into PrestoSQL and PrestoDB repositories

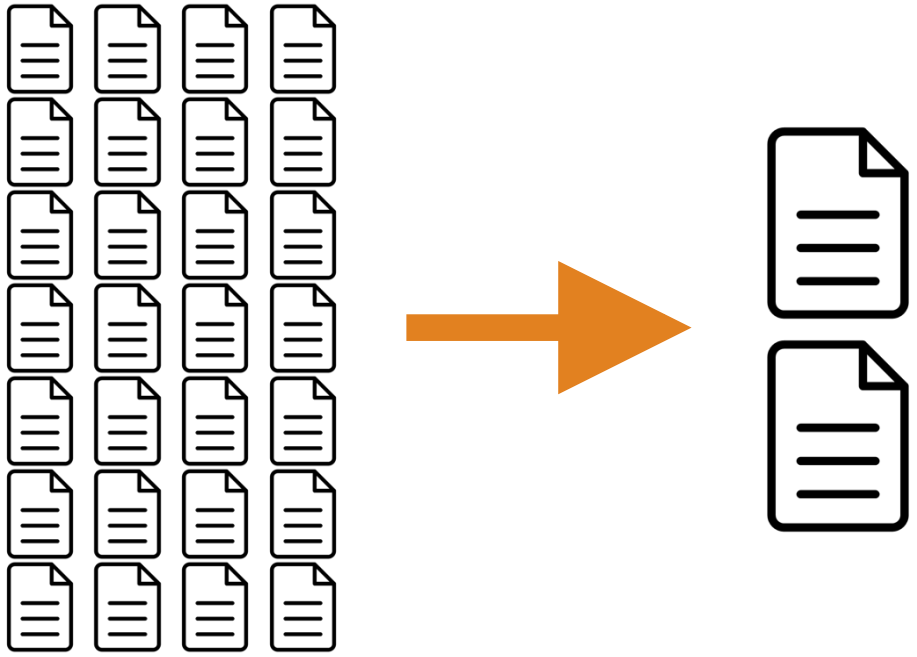
PrestoSQL 332+

PrestoDB 0.232+

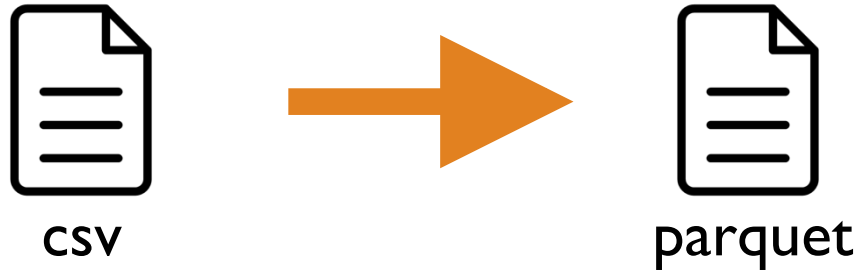
Transformation Service

Transform data to be compute-optimized independent from storage-optimized format

Coalesce



Format Conversion



Demo

Demo Setup

2 isolated AWS 5-node clusters

- Presto + Hive Metastore + S3 Data
- Presto + Alluxio + Hive Metastore + S3 Data

TPCDS sample dataset on S3

- Some tables have ~10,000 CSV files (inefficient organization)

Demo Summary

Attached existing Hive database into Alluxio Catalog

Alluxio Catalog served table metadata for Presto

Transformed `store_sales` table by coalescing and converting CSV to Parquet

Presto Without Alluxio	Alluxio Transformations	Alluxio Transformations w/ Caching
23s	8s	4s

Future Work

User community feedback/collaboration is important!

Future projects

- New UDB implementations
- More conversion formats (json)
- DDL/DML workloads (CREATE TABLE, INSERT, etc.)
- New Client APIs for structured data (Arrow)

Available Now

Try it out in Alluxio 2.4!

Documentation

Provide feedback

Feature requests and issues in Github Alluxio/alluxio

Discuss on Alluxio Slack channel

Thank You!