Presto and OpenLooKeng

SLIDES BY:

JINGLIN PENG AND JIANNAN WANG

https://sfu-db.github.io/bigdata-cmpt733/

Outline

- Presto
  - History and Motivation
  - Design of Presto
  - Presto vs Hive
- OpenLooKeng
  - Optimizations

Hadoop Ecosystem

Are they enough?

Use Cases

Dashboarding
- Low latency (<1s)

Ad-hoc Analysis
- Moderate latency (seconds to minutes)

Batch Processing
- High latency (tens of hours)

Slides adapted from Wenlei's talk 'Presto on Apache Spark: A Tale of Two Computation Engines'

How to support low latency analytics on big data?

Big data is not all about Hadoop

A Comparison of Approaches to Large-Scale Data Analysis

Andrew Pavlo
Brown University
pavlo@cs.brown.edu

Erik Paulson
University of Wisconsin
epaulson@cs.wisc.edu

Alexander Rasin
Brown University
alexr@cs.brown.edu

Daniel J. Abadi
Yale University
dna@cs.yale.edu

David J. DeWitt
Microsoft Inc.
dewitt@microsoft.com

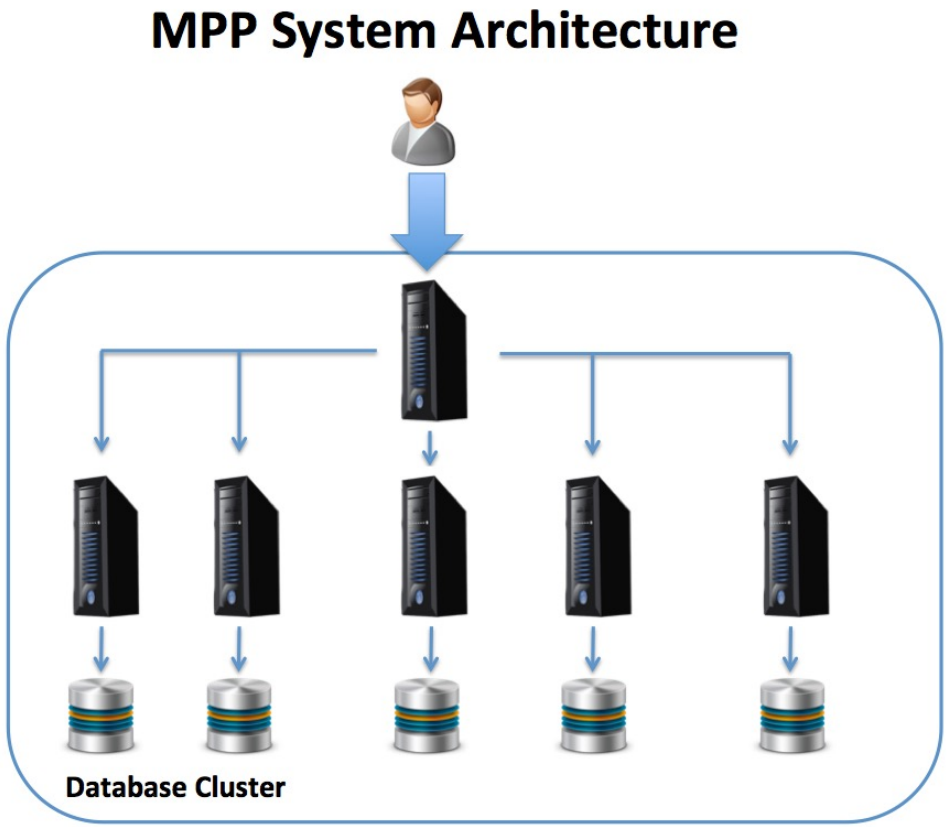
Samuel Madden
M.I.T. CSAIL
madden@csail.mit.edu

Michael Stonebraker
M.I.T. CSAIL
stonebraker@csail.mit.edu

ABSTRACT

There is currently considerable enthusiasm around the MapReduce (MR) paradigm for large-scale data analysis [17]. Although the basic control flow of this framework has existed in parallel SQL database management systems (DBMS) for over 20 years, some have called MR a dramatically new computing model [8, 17]. In this paper, we describe and compare both paradigms. Furthermore, we evaluate both kinds of systems in terms of performance and development complexity. To this end, we define a benchmark consisting of a collection of tasks that we have run on an open source version of MR as well as on two parallel DBMSs. For each task, we measure each system's performance for various degrees of parallelism on a cluster of 100 nodes. Our results reveal some interesting trade-offs. Although the process to load data into and tune the execution of parallel DBMSs took much longer than the MR system, the observed performance of these DBMSs was strikingly better. We speculate about the causes of the dramatic performance difference and consider implementation concepts that future systems should take from both kinds of architectures.

Massively Parallel Processing (MPP)



Coordinator

Workers

Limitations of MPP

Possible Solution

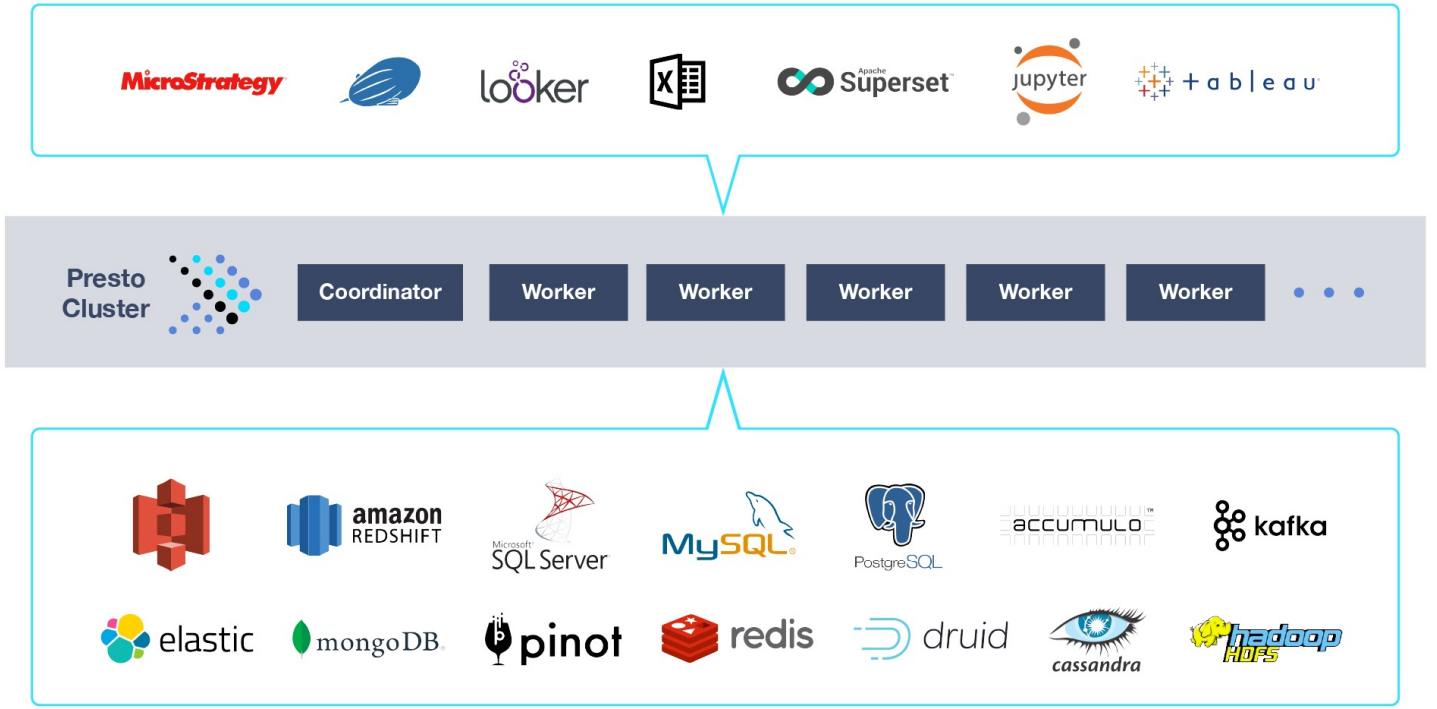
But…

- ETL is expensive
- Data keeps updating
- MPP costs $$$

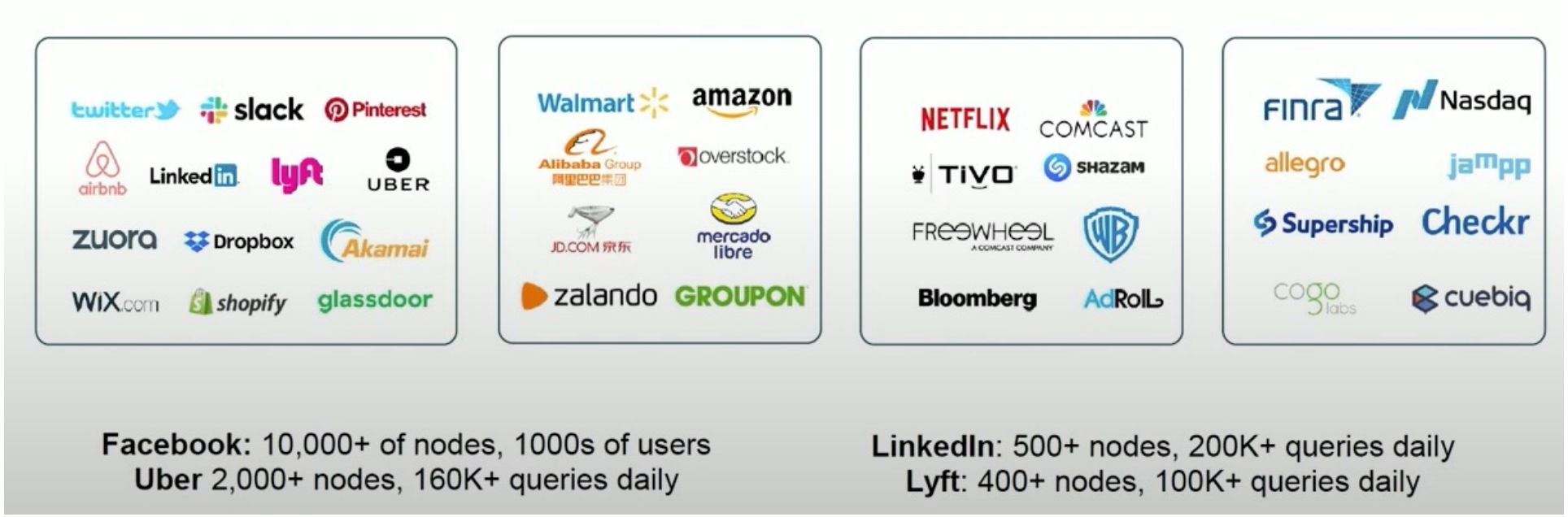
Presto

https://trino.io/blog/2020/12/27/announcing-trino.html

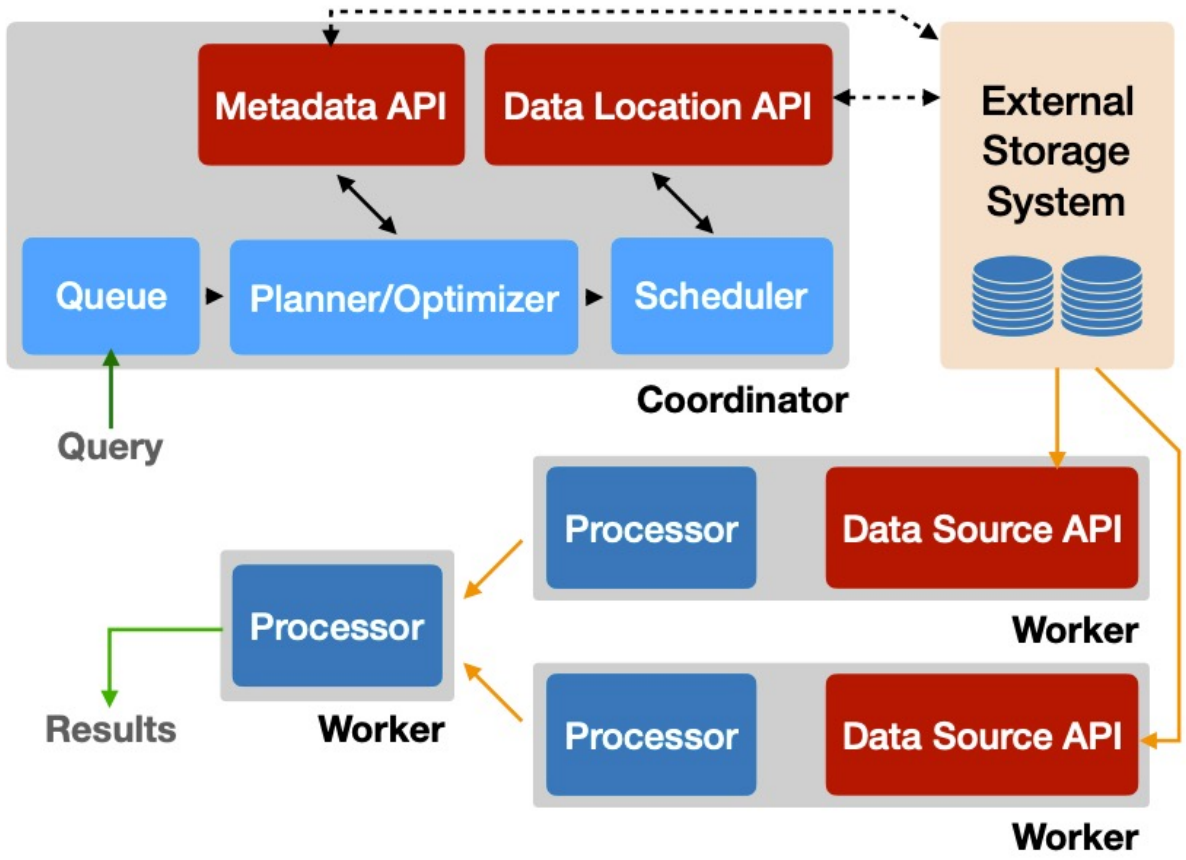
Presto: SQL on Everything



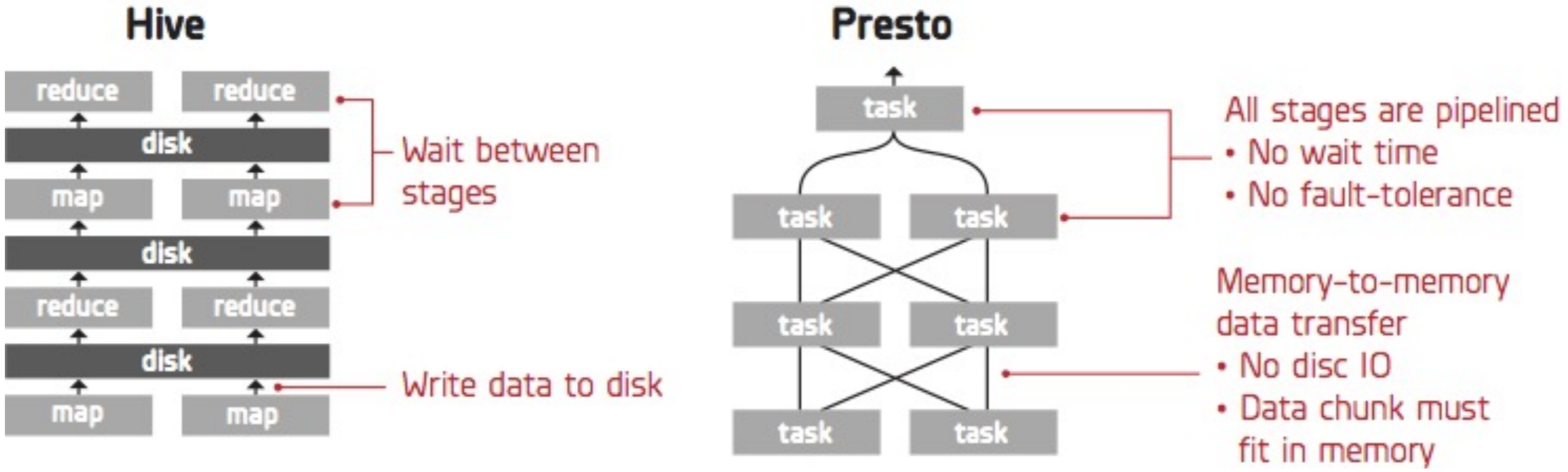
Presto is popular



Architecture



Presto vs Hive



http://spark.apache.org/docs/latest/rdd-programming-guide.html#shuffle-operations

Outline

Presto

- History and Motivation
- Design of Presto
- Presto vs Hive

OpenLooKeng

- Optimizations

OpenLooKeng

- Launched by Huawei in 2018
- Several optimizations on Presto
- Long term goal: a unified data virtualization engine

Optimizations

- Heuristic Index
- Adaptive Dynamic Filtering
- Cross Region Connector

Cross Region

- Support clusters deployed in different region

- Cross region access is close to 'local' performance
  - Parallel data access of remote data sources
  - Reduce network transmission by data compression
  - Cross region dynamic filtering

Summary

Presto: Low latency and unified SQL engine

Presto vs Hive: MPP vs. MapReduce

OpenLooKeng: Optimizations on Presto