Lecture 35:
• VertexAI Pipelines (HW-6)
• PrestoDB

Announcements:
• R-5 out
• HW-6 out
• Quiz 6 Fri

HW-6

VertexAI Pipelines
• Another serverless pipeline approach in Google
• Two different options – Kubeflow or Tensorflow Extended (TFX)
• Can define a DAG of steps, where each step can be executed in parallel
• Workflow steps for data preprocessing, model training, model deployment

For HW-6, just do a basic pipeline (Kubeflow)
• A TFX example is an option (see assignment)

Tensorflow:
• a few labs to get started, get a taste
• choose two of a few additional labs
PrestoDB

High-Level Features and Usage (Review)

- Open-source cluster-based SQL engine for analytics developed at FB
- Decouples storage-layer from query-layer ... not a data store itself
- Can access various data systems/formats via connectors
- Circa 2018 “handled much of SQL analytic workload at FB” (*)
- 100s of petabytes and quadrillions (million billion) of rows per day (*)
- Developed for interactive analytics, various use cases ...
- Designed to be flexible and extensible (e.g., for different uses)

(*) R. Sethi et al. “Presto: SQL on Everything”, ICDE’19

---

PrestoDB High-Level Architecture

Each Presto cluster has one coordinator node and many worker nodes

R. Sethi et al. “Presto: SQL on Everything”

© S. Bowers  CPSC 324, Spring 2024 4
(1) The **coordinator node** (upon receiving a query):

- places in queues (based on priorities, etc.)
- parses and analyzes the query, generates a query plan
- performs optimization and creates distributed execution plan
- distributes plan to cluster workers

(2) The **coordinator node** (during execution):

- starts execution of **tasks**
- begins to assign **splits** (addressable data chunks in external systems)
- splits assigned to tasks responsible for reading corresponding data
PrestoDB High-Level Architecture

(3) The **worker nodes**: ... e.g., vs Hadoop-based execution

- workers process tasks from many queries concurrently (multitask)
- execution pipelined as possible (flows between tasks as becomes available)
- intermediate data and state stored in-memory whenever possible
- for some queries, Presto can return results before all data processed

---

PrestoDB System Design

The “Life-Cycle” of a Query:

1. **Parser**: converts query string to parse tree
2. **Logical Planner**: convert parse tree to **tree of plan nodes**

```sql
SELECT o.orderkey, SUM(tax)
FROM   orders o LEFT JOIN lineitem l
        ON o.orderkey = l.orderkey
WHERE  l.discount = 0
GROUP  BY o.orderkey
```

3. **Optimizer**: transforms plan into physical execution plan (strategy)
4. **Scheduler**: distributes plan to workers as executable tasks
5. **Execution**: the tasks are executed