CS 744: SNOWFLAKE

Shivaram Venkataraman

Fall 2021

ADMINISTRIVIA

- Assignment 1 grades out! → View on Piazza / email
- Assignment 2 by mid-week
- Midterm on Thursday! Seating layout?

  - Randomize seating

booking
DB
confirm seats
online transaction

collection of logs or purchases over long time period, analyze this

SparkSQL/Scope: Given a query how do you run it efficiently?

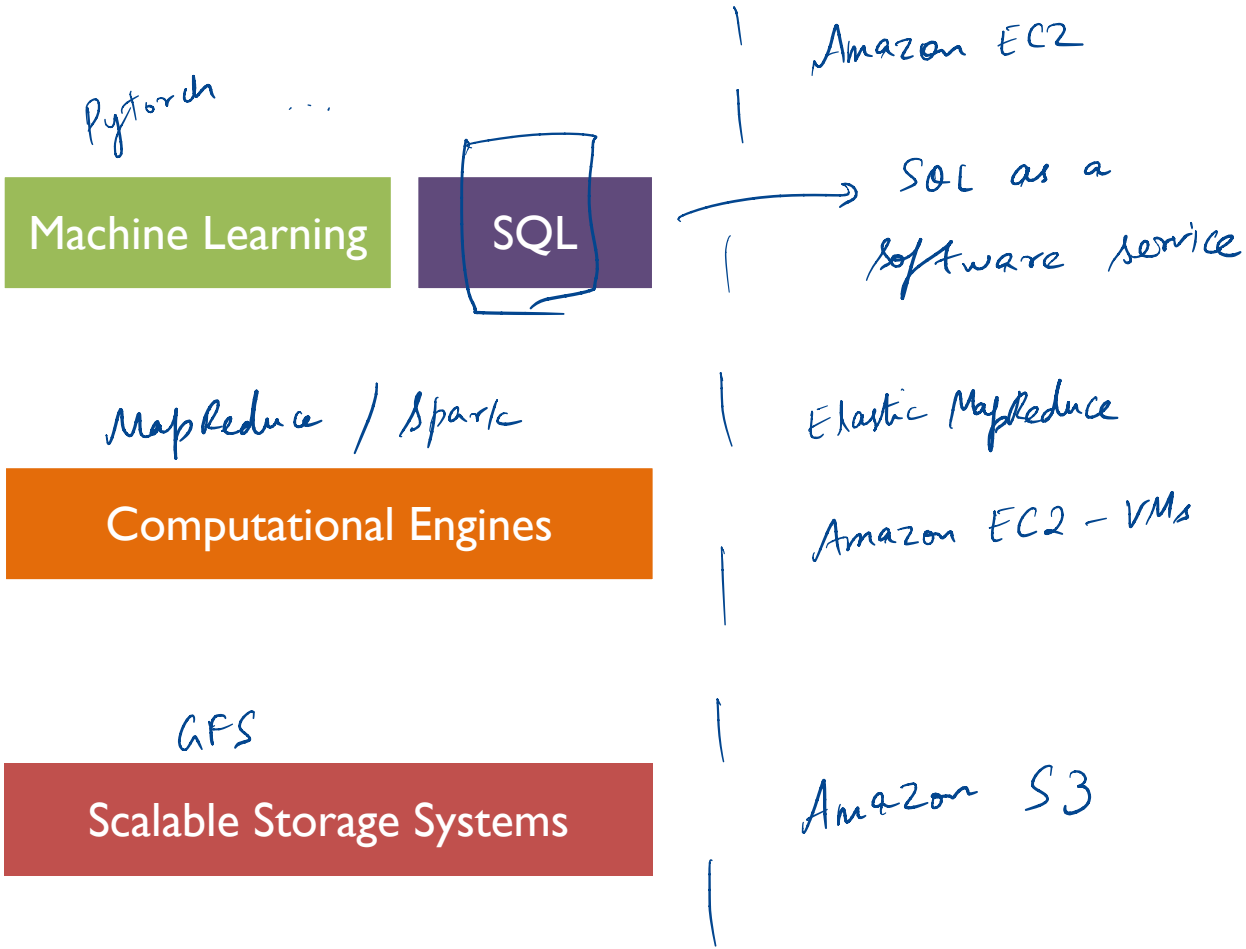
Snowflake: How do you build an elastic data warehouse?

Able to increase or decrease resource use as workload changes

OLAP queries

A collection of data and you do analytics queries on it

CLOUD COMPUTING STACK



SNOWFLAKE: GOALS

Software-as-a-Service → Users don't need to install / manage this

Elastic → Use resources as required

Highly Available → Fault tolerance

Semi-Structured Data → Traditional warehouses only support structured data

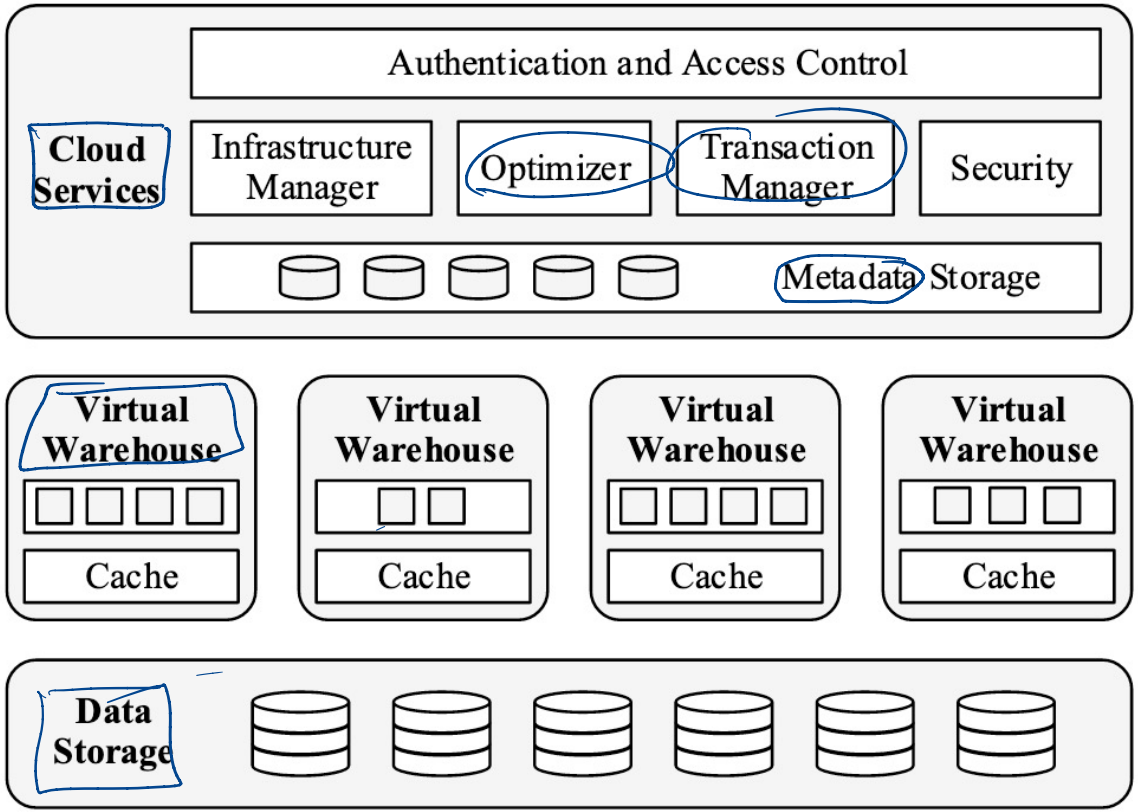
JSON
↳ queries on specific fields

SNOWFLAKE DESIGN

Admin & optimizer live in Cloud Services →

VWs are isolated from each other →

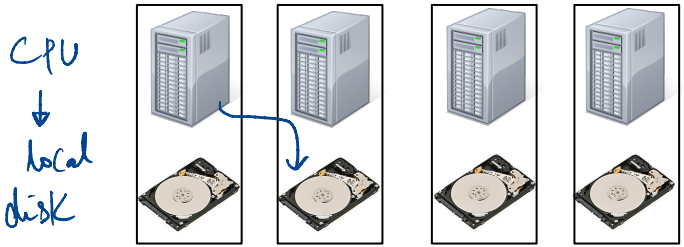
Data storage shared across VWs



STORAGE VS COMPUTE

ties together CPU, disk resources

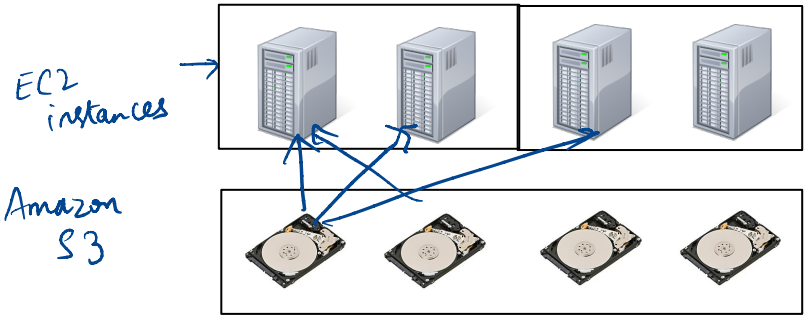
⇒ some queries need lot of disk, less CPU, can lead to under utilization



Shared Nothing

Separate out storage & compute
- Scale storage independently
- Handle irregular arrival of queries

Downside = No locality of access



Multi Cluster, Shared Data

STORAGE: HYBRID COLUMNAR

Entire table = 4 rows

Name	ID
Alice	32
Bob	22
Eve	24
Victor	27

Partition or File table

2 rows in each partition

Alice,32,Bob,22

Eve,24,Victor,27

Row-oriented

Alice, Bob, 32,22

Eve,Victor,24,27

Hybrid Columnar

range query start offset & length ID column

Compression saves space & also network traffic

VIRTUAL WAREHOUSES

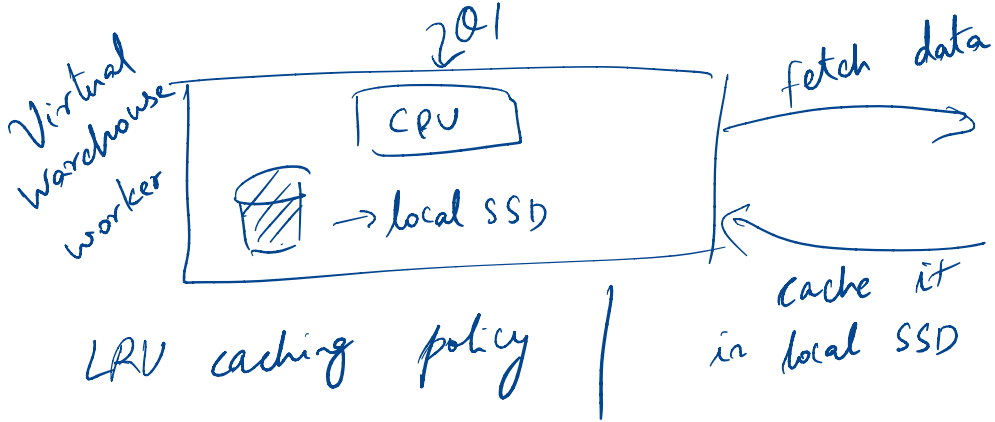
Elasticity, Isolation

↳ For each user spin up some EC2 instances = VW

→ Isolation as instances separate for each user

→ Can turn off these machines once you are done

Local caching, Stragglers



LRU caching policy

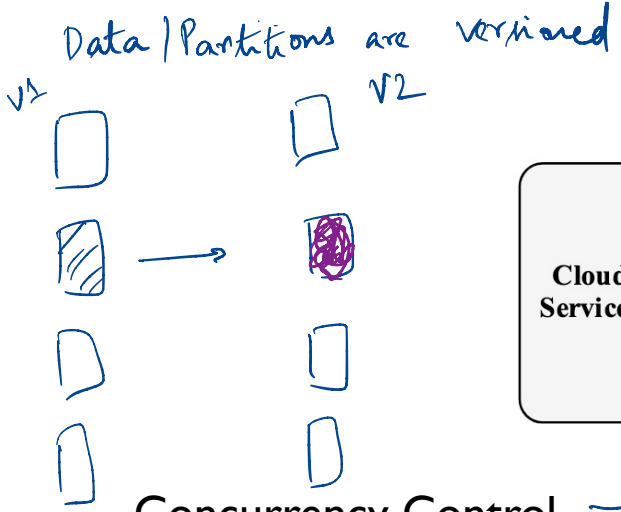
"File stealing" P2P communication

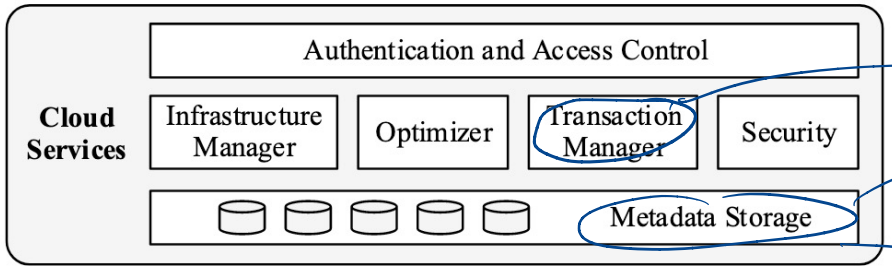
Stragglers happen due to data skew

↓

one machine has lot more work to do

CLOUD SERVICES





Track version numbers of blocks / files

↳ about blocks in S3

Concurrency Control → MVCC

A query will see all the files at a specific version

Query → V1 is associated with it

all the reads come from a consistant version

"Snapshot Isolation"

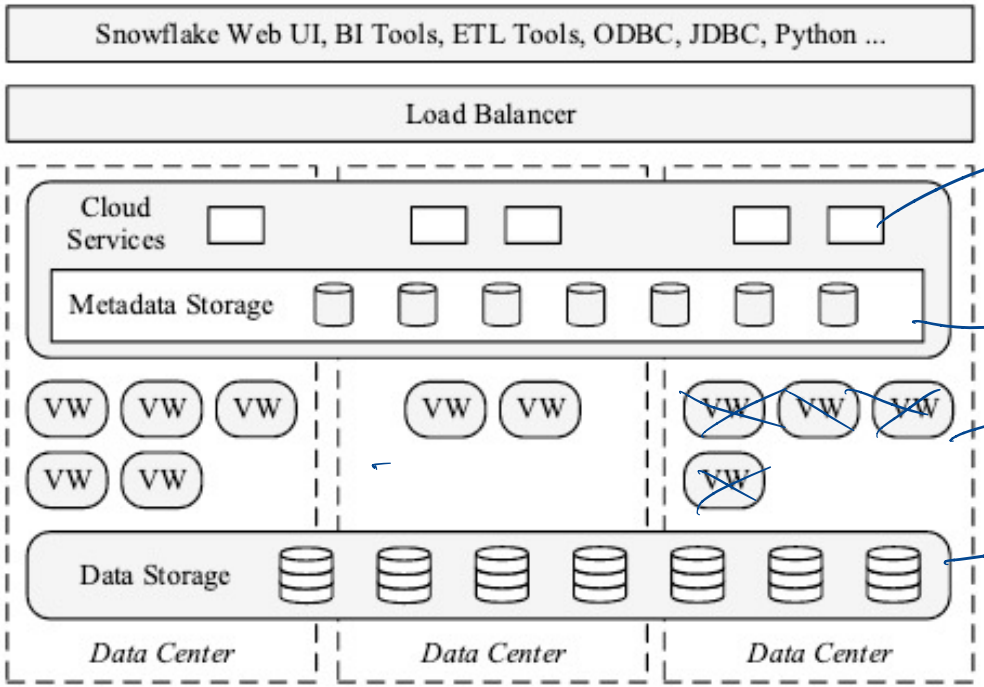
Pruning

→ the range of various Columns in each block

"min: 22, max: 34"

→ limit or exclude files which are not going to be used in the query

FAULT TOLERANCE



these are again stateless

replicated across AZs

Just spin up new ones. "Retry queries"

replicate across data centers "Availability Zone"

SEMI-STRUCTURED DATA → Key reason / challenge for enterprises

```
{
    first_name: “john”,
    last_name: “doe”,
    order_id: “1234”,
}
{
    first_name: “bucky”,
    last_name: “badger”,
    order_id: “52342”,
    order_date: “3/3/2020”,
}
```

→ doesn't strictly follow a schema

↳ not present in all records

Extraction operation

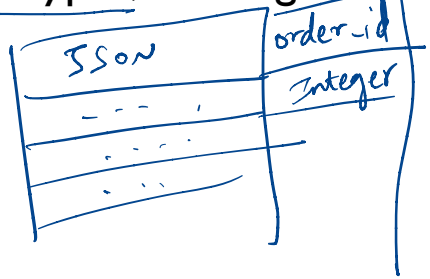
↳ access individual fields with JSON as part of query

Flattening

↳ flatten nested JSON objects

Infer types, Pruning

Inside each Partition



min, max pruning on this column

TIME TRAVEL?

```
SELECT * FROM my_table AT(TIMESTAMP =>
  'Mon, 01 May 2015 16:20:00 -0700'::timestamp);
SELECT * FROM my_table AT(OFFSET => -60*5); -- 5 min ago
SELECT * FROM my_table BEFORE(STATEMENT =>
  '8e5d0ca9-005e-44e6-b858-a8f5b37c5726');
```

Multiple versions of table (MVCC) → Run queries which say processes data till yesterday

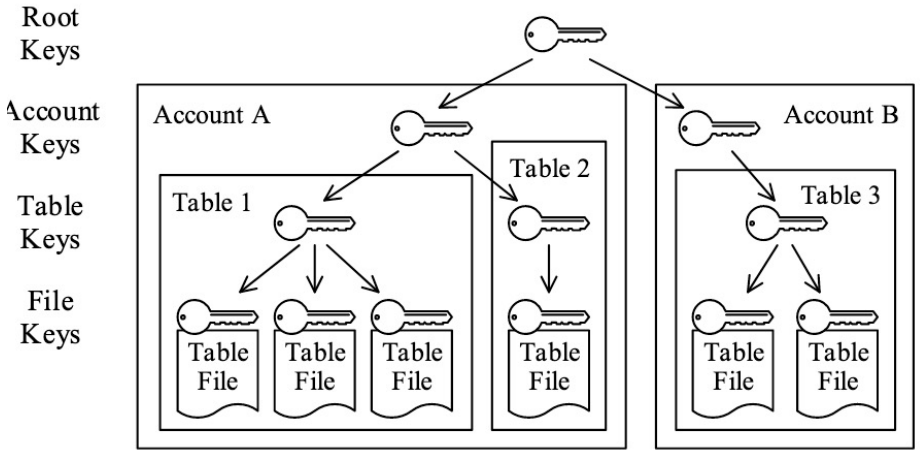
Undo accidental deletes

Cheap to clone / snapshot a table

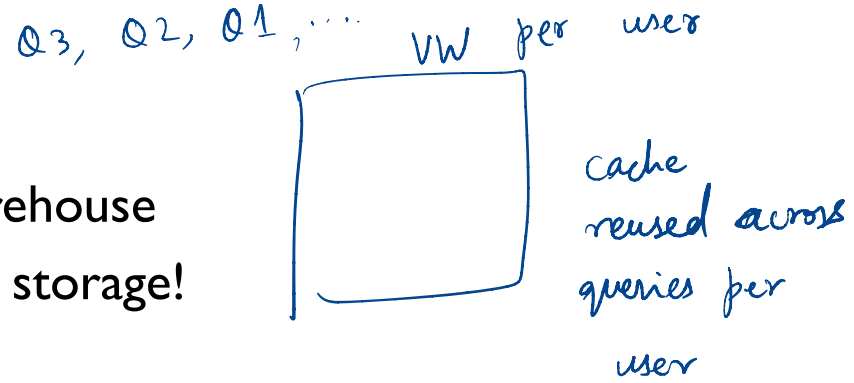
SECURITY

Hierarchical key management

Key rotation, re-keying



SUMMARY, TAKEAWAYS



Snowflake

- Cloud computing → Elastic data warehouse
- Key idea: Separation of compute and storage!

- Hybrid columnar storage format
- Elastic compute with virtual warehouses
- Pruning, semi-structured optimizations, fault tolerant

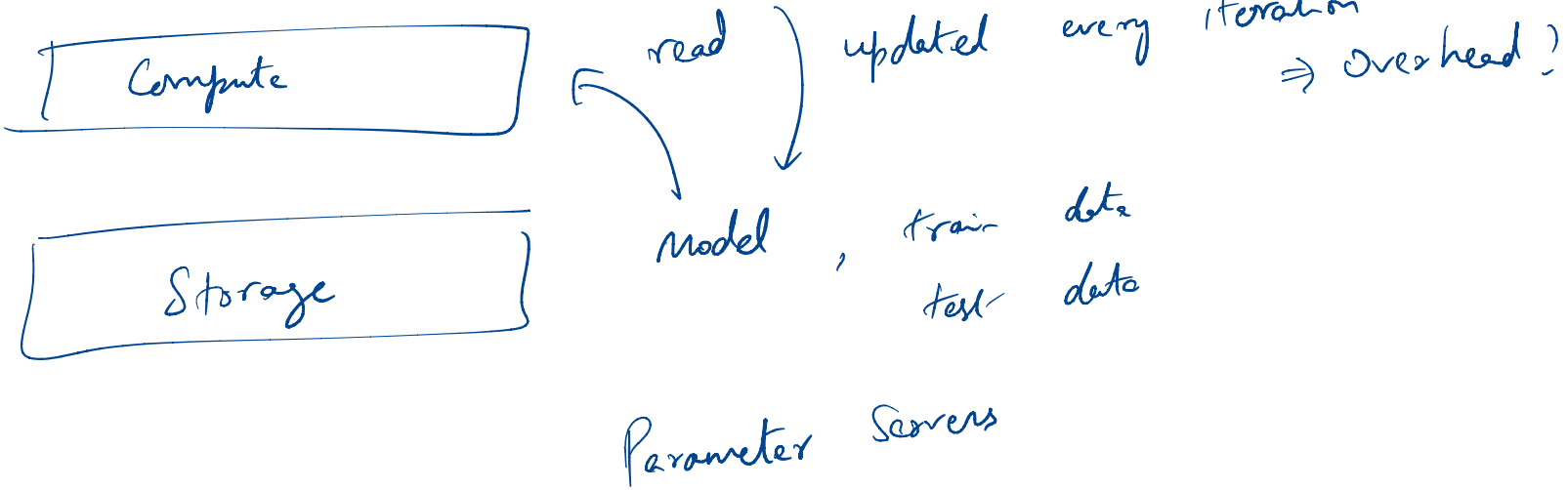
DISCUSSION

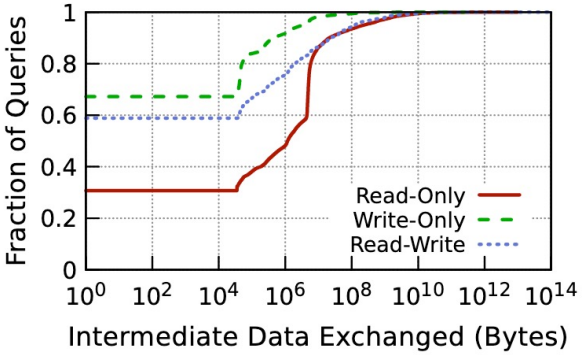
https://forms.gle/buUDM9nRs6Gg9tURA

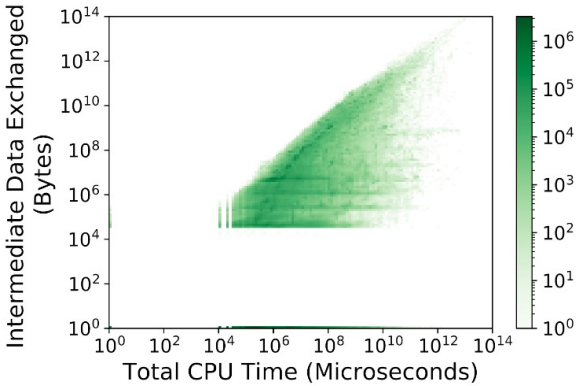
We see how Snowflake leads to the design of an elastic data warehouse. If we were to similarly design an Elastic PyTorch for training how would the design look? What are some design trade-offs compared to existing PyTorch?

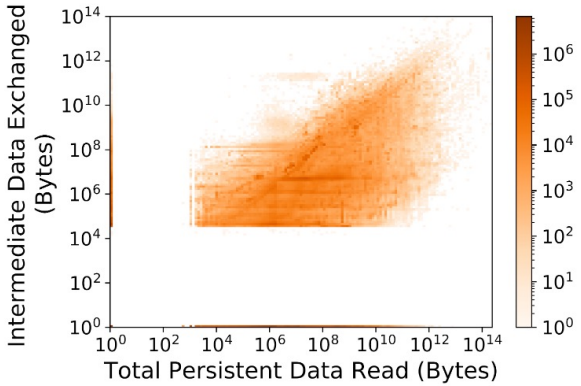
Data Parallel → row-based partitioning

hybrid columnar ?




Fraction of Queries
1
0.8
0.6
0.4
0.2
0
10^0 10^2 10^4 10^6 10^8 10^{10} 10^{12} 10^{14}
Intermediate Data Exchanged (Bytes)
Read-Only
Write-Only
Read-Write


Intermediate Data Exchanged (Bytes)
10^{14} 10^{12} 10^{10} 10^8 10^6 10^4 10^2 10^0
10^0 10^2 10^4 10^6 10^8 10^{10} 10^{12} 10^{14}
Total CPU Time (Microseconds)
10^6 10^5 10^4 10^3 10^2 10^1 10^0


Intermediate Data Exchanged (Bytes)
10^{14} 10^{12} 10^{10} 10^8 10^6 10^4 10^2 10^0
10^0 10^2 10^4 10^6 10^8 10^{10} 10^{12} 10^{14}
Total Persistent Data Read (Bytes)
10^6 10^5 10^4 10^3 10^2 10^1 10^0

NEXT STEPS

Next class: Midterm!