# CS 744: SNOWFLAKE

Shivaram Venkataraman

Fall 2020

# ADMINISTRIVIA

- Assignment 1 grades out!
- Assignment 2 by mid-week
- Midterm this week!

- Project Proposal Peer review

# AEFIS FEEDBACK

How has your experience been reading papers?

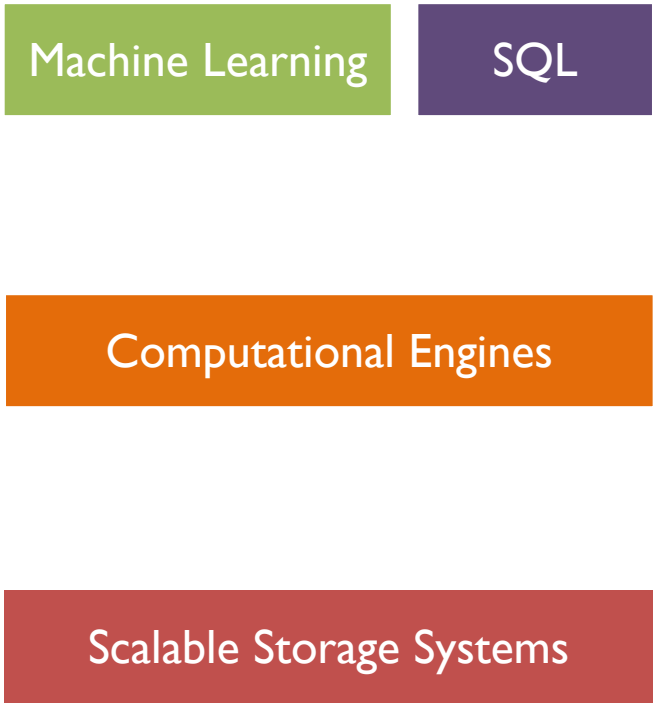
Are the lectures useful for learning?

How are the discussion groups? Did you get to know students in the class? Would it help to have the same group each time?

Anything else we could improve for the second half?


Applications
Machine Learning
SQL

# CLOUD COMPUTING STACK



# SNOWFLAKE: GOALS

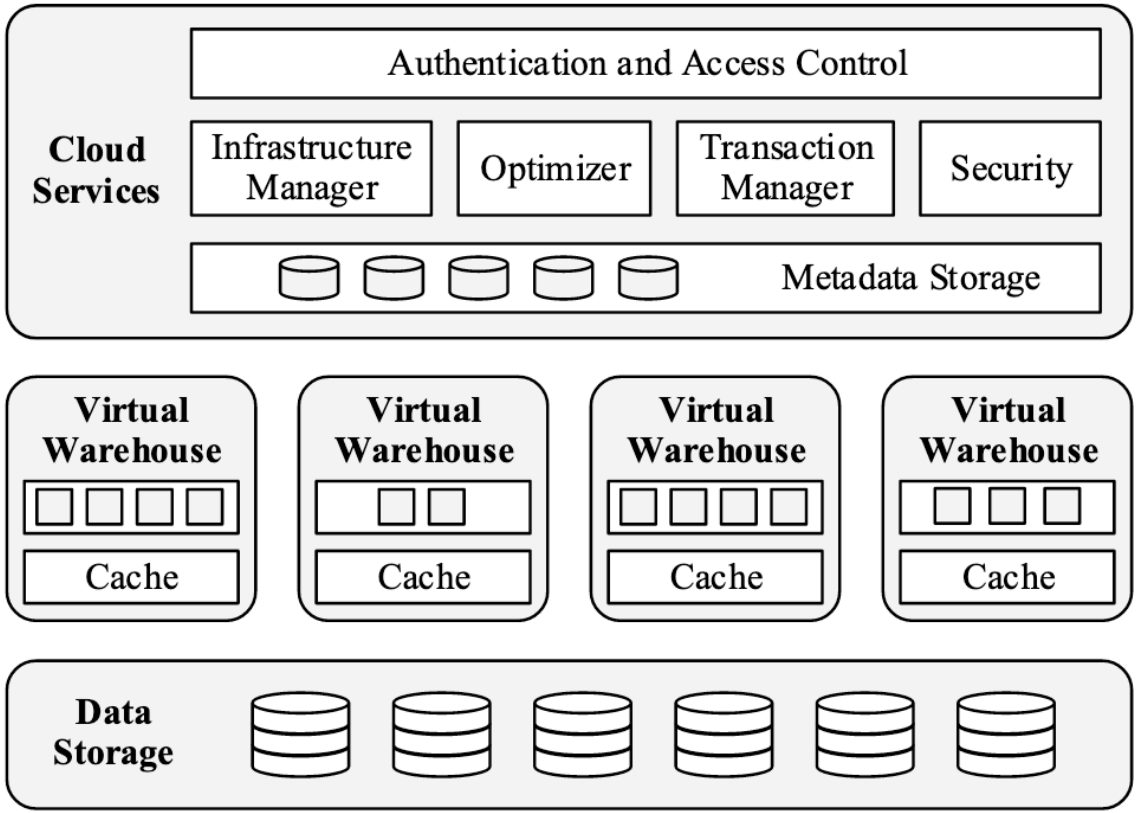
Software-as-a-Service

Elastic

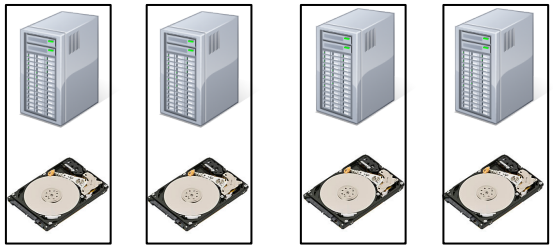
Highly Available

Semi-Structured Data

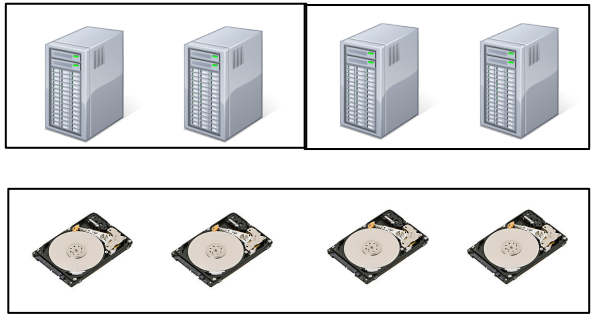
# SNOWFLAKE DESIGN



# STORAGE VS COMPUTE



Shared Nothing



Multi Cluster, Shared Data

# STORAGE: HYBRID COLUMNAR

| | |
|---|---|
| Alice | 32 |
| Bob | 22 |
| Eve | 24 |
| Victor | 27 |

Alice,32,Bob,22

Eve,24,Victor,27

Row-oriented

Alice, Bob, 32,22

Eve, Victor,24,27

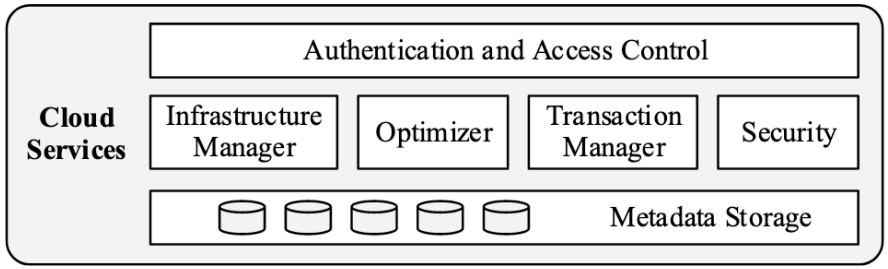
Hybrid Columnar

# VIRTUAL WAREHOUSES

Elasticity, Isolation

Local caching, Stragglers

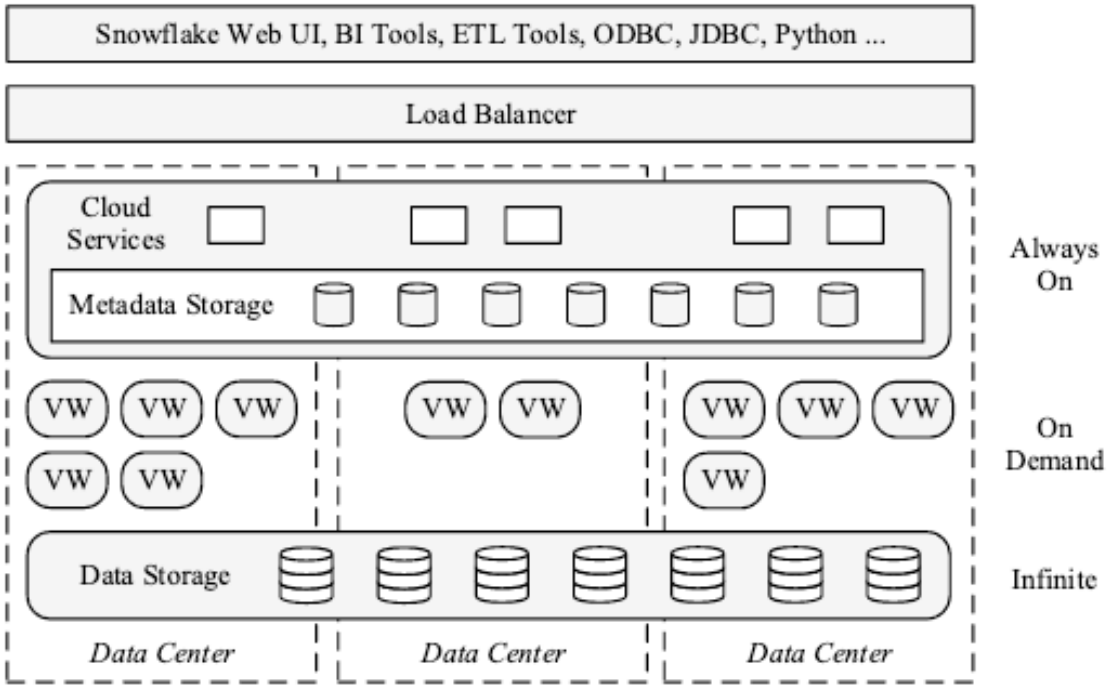
# CLOUD SERVICES




Concurrency Control

Pruning

# FAULT TOLERANCE



# SEMI STRUCTURED DATA

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

Extraction operation

Flattening

Infer types, Pruning

# 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)

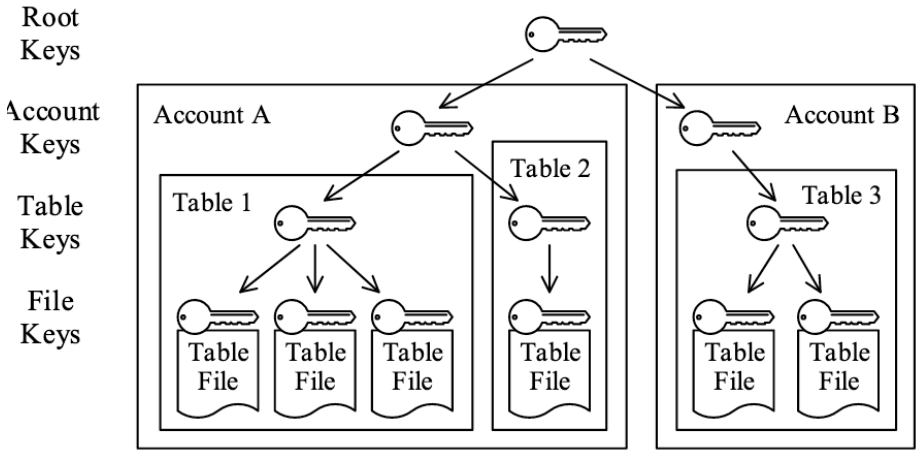
Undo accidental deletes

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

# AEFIS FEEDBACK

# DISCUSSION

https://forms.gle/ZFosdUnizXYABAE86

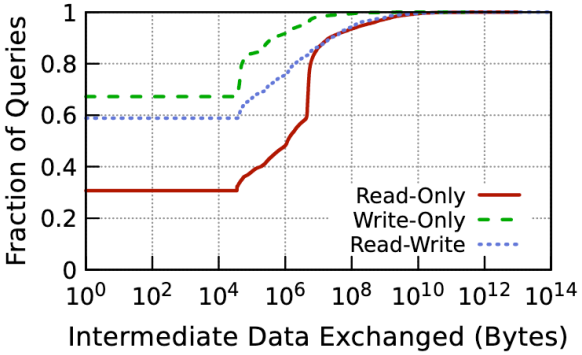
We see how Snowflake leads to the design of an elastic data warehouse. If we were to similarly design an Elastic PyTorch for training how would the design look? What are some design trade-offs compared to existing PyTorch?

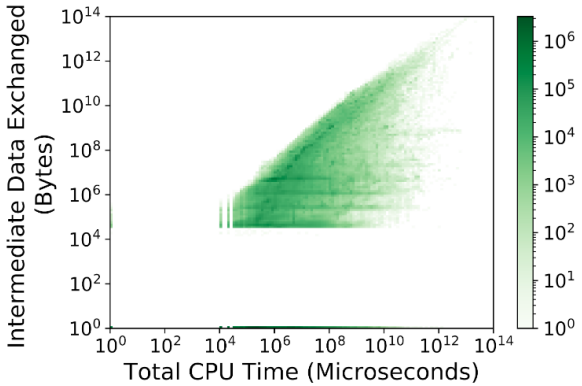
# NEXT STEPS

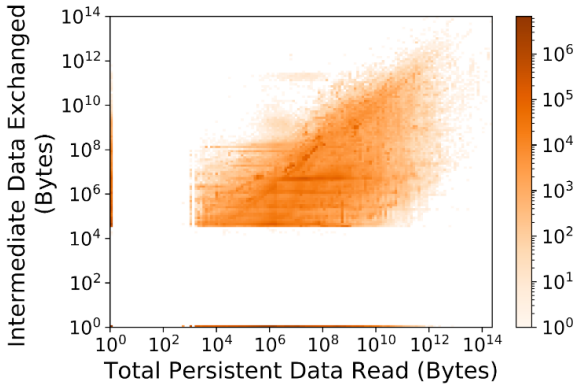
Next class: Midterm!

AEFIS feedback

Project proposal peer feedback assignments


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)
$10^6$ $10^5$ $10^4$ $10^3$ $10^2$ $10^1$ $10^0$