

CS 744: SNOWFLAKE

Shivaram Venkataraman

Spring 2025

ADMINISTRIVIA

- Midterm on Tuesday!
- Assignment 2 grading
- Project proposal feedback

CLOUD COMPUTING STACK

Machine Learning

SQL

Computational Engines

Scalable Storage Systems

SNOWFLAKE: GOALS

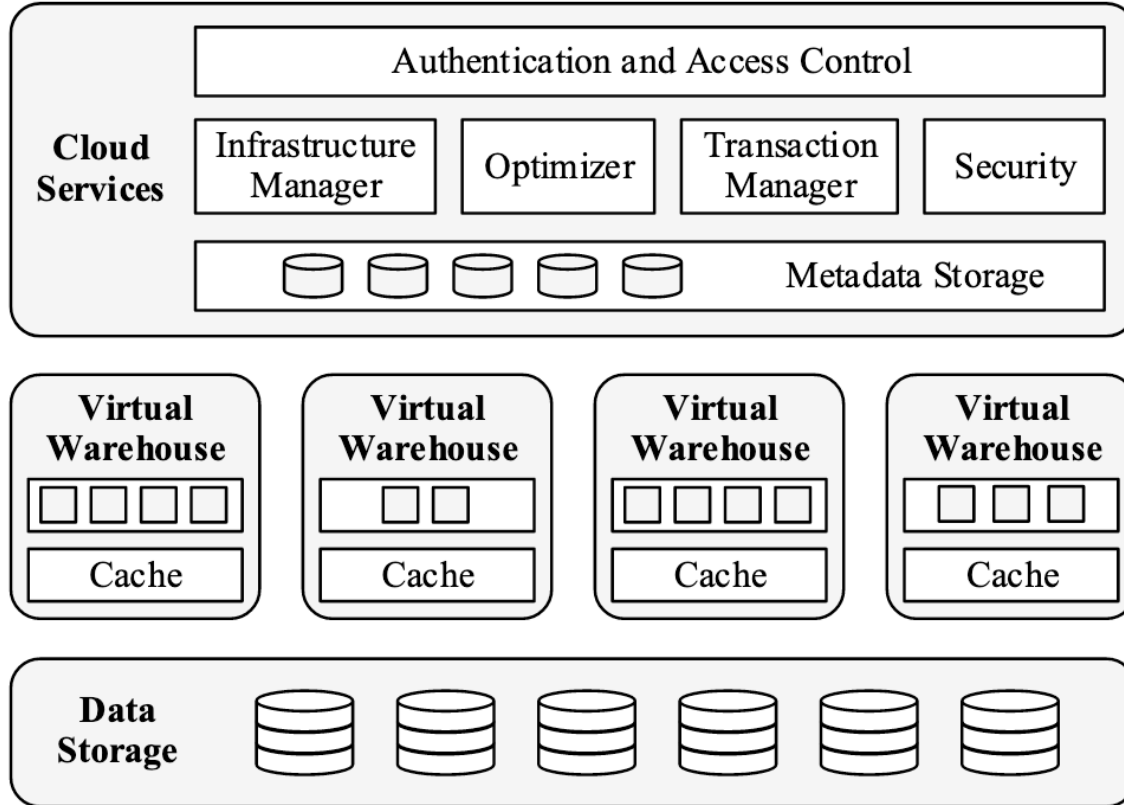
Software-as-a-Service

Elastic

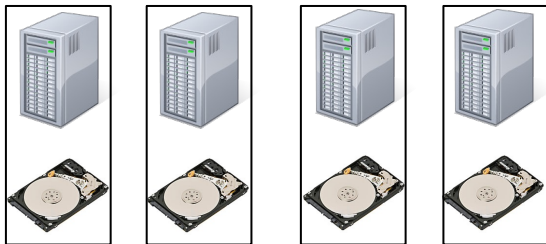
Highly Available

Semi-Structured Data

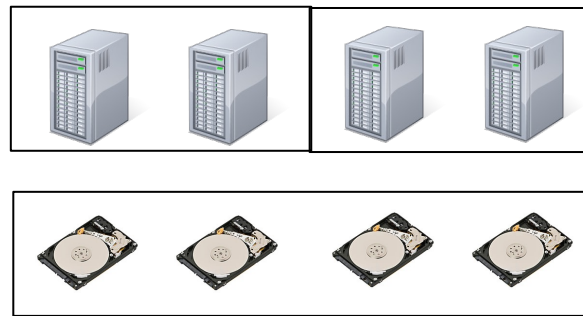
SNOWFLAKE DESIGN



STORAGE VS COMPUTE



Shared Nothing



Multi Cluster, Shared Data

STORAGE: HYBRID COLUMNAR

Alice	32
Bob	22
Eve	24
Victor	27

Alice,32,Bob,22

Alice, Bob, 32,22

Eve,24,Victor,27

Eve,Victor,24,27

Row-oriented

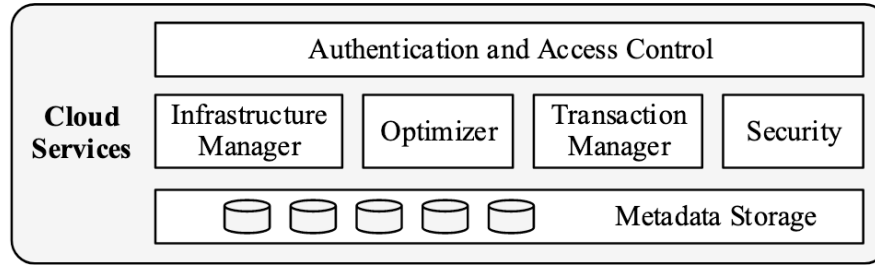
Hybrid Columnar

VIRTUAL WAREHOUSES

Elasticity, Isolation

Local caching, Stragglers

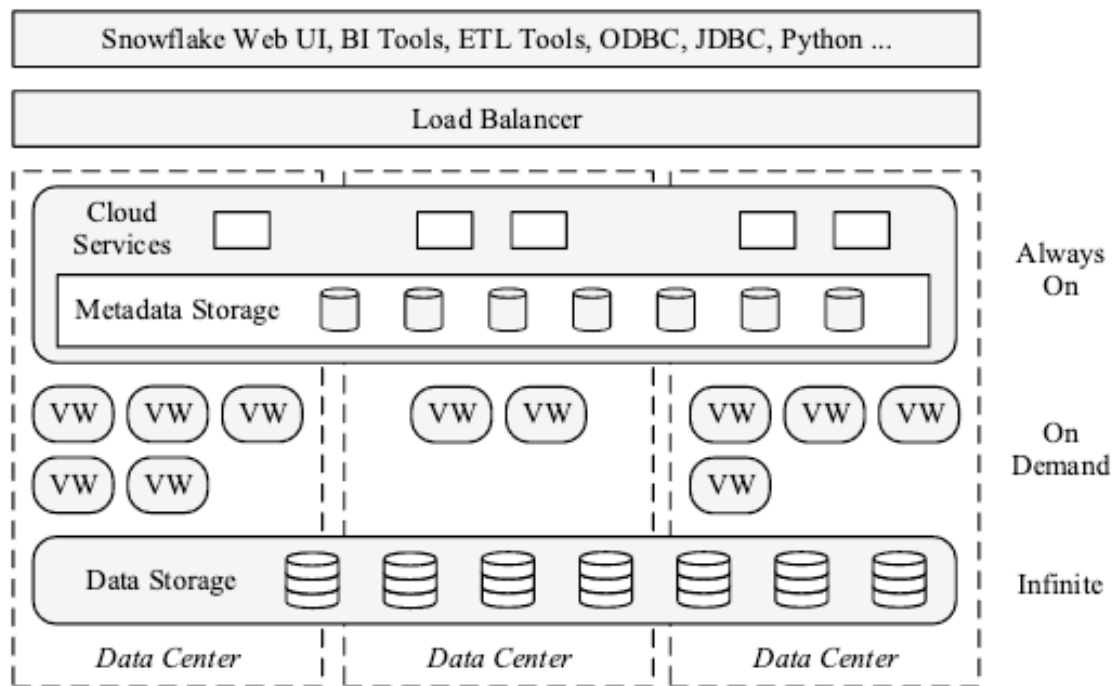
CLOUD SERVICES



Concurrency Control

Pruning

FAULT TOLERANCE



SEMI STRUCTURED DATA

```
{  
  first_name: "john",  
  last_name: "doe",  
  order_id: "1234",  
}
```

Extraction, Flattening operations

```
{  
  first_name: "bucky",  
  last_name: "badger",  
  order_id: "52342",  
  order_date: "3/3/2020",  
}
```

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

SUMMARY, TAKEAWAYS

Snowflake

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

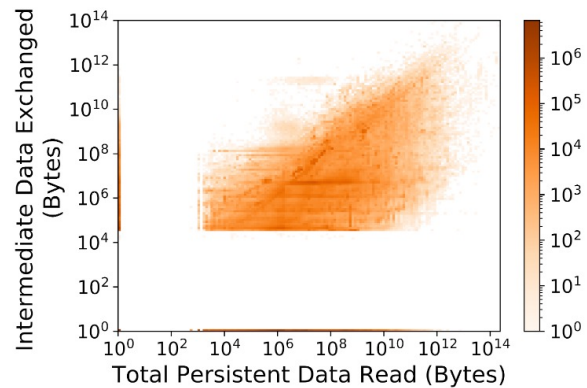
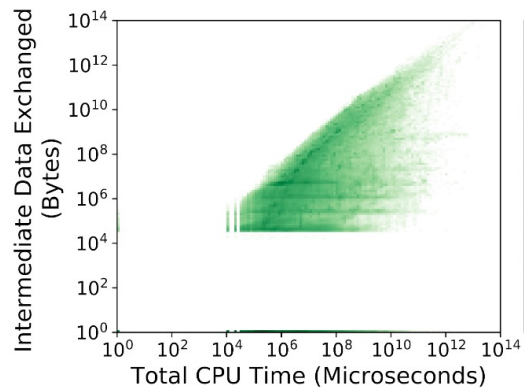
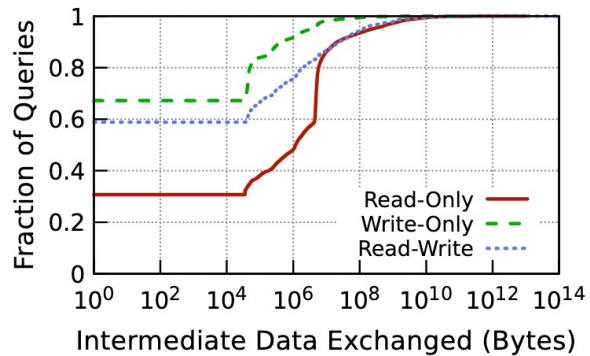
- Hybrid columnar storage format
- Elastic compute with virtual warehouses
- Pruning, semi-structured optimizations, fault tolerant



DISCUSSION

<https://forms.gle/KjsqAajFrVyyMi4s8>

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!