Materialize

Planning for scale: CockroachDB @ Materialize

Parker Timmerman, Member of Technical Staff

RoachFest23

Agenda

- What is Materialize?
- Transition to the Cloud
- Architecture Evaluation
- CockroachDB Partnership
- Production and *The Future*
- Lessons Learned

What is Materialize?

- **Operational Data Warehouse**
- You register interest in a set of SQL queries . . .

```
CREATE VIEW interesting_query AS SELECT
...
CREATE DEFAULT INDEX ON interesting_query
```

- . . . and Materialize incrementally updates the results of your query, to return answers within milliseconds of a change to the underlying data:

```
SELECT * FROM interesting_query
SELECT * FROM interesting_query
SELECT * FROM interesting_query
```

<10ms

target query time

Key Materialize Properties

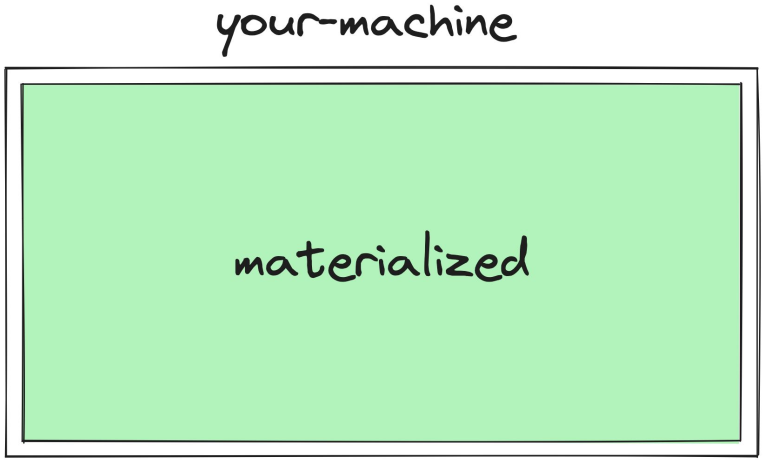
1. **Fast**
   - Target <10ms query time for reads out of an index
2. **Correct**
   - Queries adhere to strict serializability by default

These two properties are in tension!

(Spoiler: CockroachDB helps balance this tension)

Transition to the Cloud

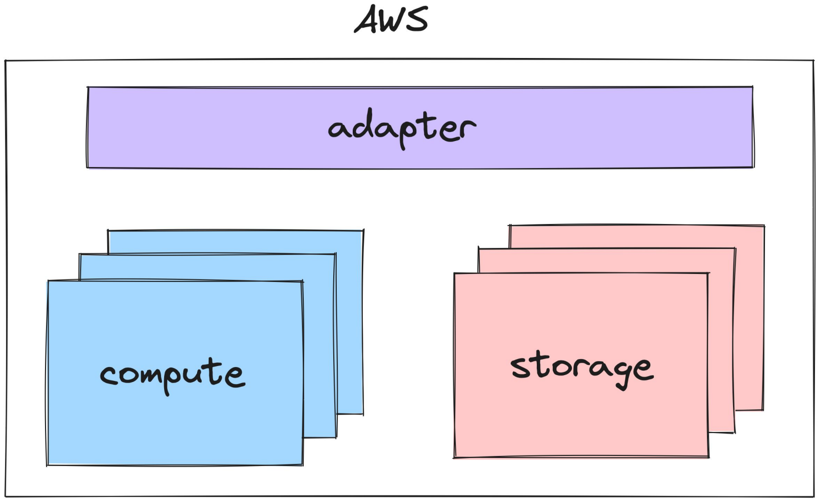
2020 - 2021



- Focus on incremental computation
- Limited to a single node
- No fault tolerance

Transition to the Cloud

2022 - now



- ○ High Availability
- ○ Horizontal Scalability
- ○ Maintain speed and correctness

Planning and Development

- **Design philosophy:** Use as many managed services as possible to decrease operational burden.
  - Amazon EKS for Kubernetes
  - Amazon S3 for blob storage

- What service for the **metadata layer**?
  - Maintains a record of all user objects (tables, views, etc.) and what blobs exist in Amazon S3

Metadata Layer Evaluation

Contenders

- CockroachDB
- FoundationDB
- Amazon RDS/Amazon Aurora
- Amazon Dynamo DB

Requirements

Metadata Layer Evaluation

Contenders

- CockroachDB
- FoundationDB
- Amazon RDS/Amazon Aurora
- Amazon Dynamo DB

Requirements

1. Strong Consistency

Metadata Layer Evaluation

Contenders

- CockroachDB
- FoundationDB
- Amazon RDS/Amazon Aurora
- Amazon Dynamo DB

Requirements

1. Strong Consistency
2. Low Latency (<10ms)

Metadata Layer Evaluation

Contenders

- CockroachDB
- FoundationDB
- Amazon RDS/Amazon Aurora
- ~~Amazon Dynamo DB~~

Requirements

1. Strong Consistency
2. Low Latency (<10ms)

Metadata Layer Evaluation

Contenders

- CockroachDB
- FoundationDB
- Amazon RDS/Amazon Aurora
- ~~Amazon Dynamo DB~~

Requirements

1. Strong Consistency
2. Low Latency (<10ms)
3. Scalability

- Materialize is connection-hungry

Metadata Layer Evaluation

Contenders

- ○ CockroachDB
- ○ FoundationDB
- ○ ~~Amazon RDS/Amazon Aurora~~
- ○ ~~Amazon Dynamo DB~~

Requirements

1. Strong Consistency
2. Low Latency (<10ms)
3. Scalability

- Materialize is connection-hungry

Metadata Layer Evaluation

Contenders

- ○ CockroachDB
- ○ FoundationDB
- ○ ~~Amazon RDS/Amazon Aurora~~
- ○ ~~Amazon Dynamo DB~~

Requirements

1. Strong Consistency
2. Low Latency (<10ms)
3. Scalability
   - Materialize is connection-hungry
4. Managed Solution

Metadata Layer Evaluation

Contenders

- CockroachDB
- ~~FoundationDB~~
- ~~Amazon RDS/Amazon Aurora~~
- ~~Amazon Dynamo DB~~

Requirements

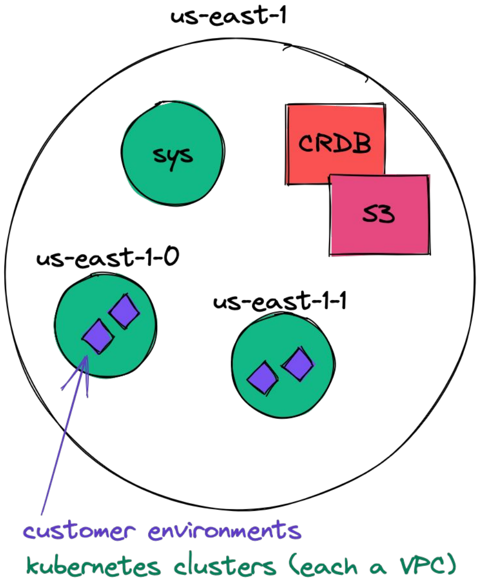
1. Strong Consistency
2. Low Latency (<10ms)
3. Scalability
   - Materialize is connection-hungry
4. Managed Solution

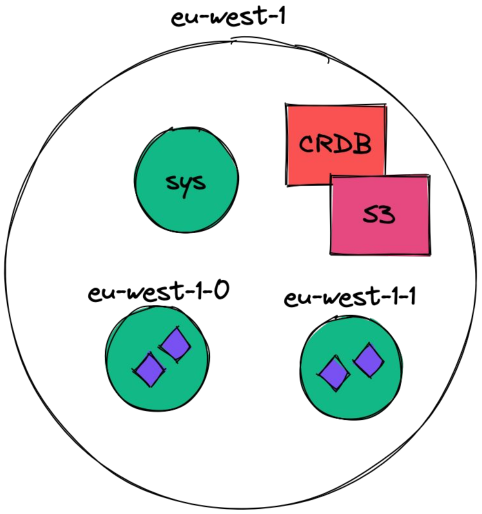
Persist

- Maps durable storage semantics to ones that are native to timely dataflow (partial time-varying collections).
- Uses CockroachDB to provide *consensus*.

```
INSERT INTO consensus (shard, sequence_number, data)
SELECT <shard_id>, <new_sequence_num>, <new_data>
WHERE (SELECT sequence_number FROM consensus
       WHERE shard = <shard_id>
       ORDER BY sequence_number DESC LIMIT 1) = <expected_seq_num>;
```

High-Level Cloud Architecture





Cockroach Labs Partnership

- Established a Design Partnership
    - Reviewed application design for working with serializable isolation
    - Added key enhancements to CockroachDB Dedicated
        - Amazon CloudWatch metrics
        - SSO
        - Fine-grained access controls
        - SCIM provisioning

- Monthly leadership syncs
- Weekly tech team syncs
- Joint Slack Channel

Production and The Future

- Materialize entered private preview in **October 2022**
- Production-ready on **February 1, 2023**
  - Many customers powering production use cases on Materialize
  - Accepting free trials for qualified users
- Scale
  - Largest cluster is six 8vCPU/32GB memory nodes at ~50% CPU usage
  - ~2,000 outstanding CockroachDB connections
  - ~2,000 `INSERT` statements per second
- Prometheus metrics on our Grafana dashboards
- CockroachDB will scale easily with us for many years to come

Performance Optimizations

- Minimize the garbage collection interval
- One-phase commit fast path
  - `EXPLAIN ANALYZE (VERBOSE)` is your friend here
- Consider unique data version identifiers so you can cache data outside of CockroachDB without losing consistency

Final Thoughts

- Building on managed services has successfully minimized operational burden
  - All managed service instances (Amazon EKS, CockroachDB) are shared by multiple customers to further reduce burden.
- The choice of CockroachDB for the metadata layer has served us well
  - Performance promise is real: low single-digit millisecond transaction commits
  - Scalability promise is real: Sizing up a CockroachDB cluster is just three clicks in the web UI
  - Team has been quick to sand down rough edges

Thank You!