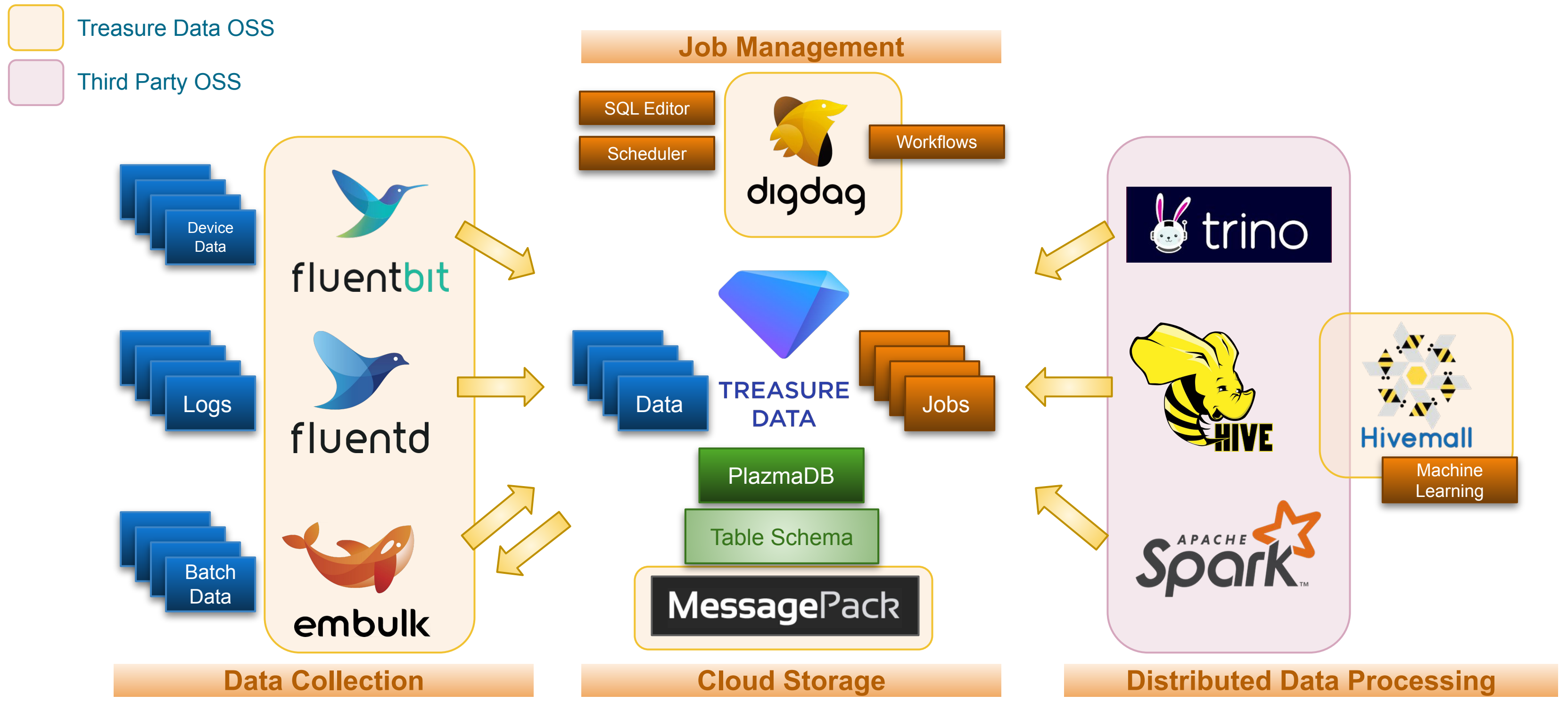
TREASURE DATA

Journey of Migrating Millions of Queries on The Cloud

Taro L. Saito, Naoki Takezoe, Yukihiro Okada, Takako Shimamoto, Dongmin Yu, Suprith Chandrashekharachar, Kai Sasaki, Shohei Okumiya, Yan Wang, Takashi Kurihara, Ryu Kobayashi, Keisuke Suzuki, Zhenghong Yang, Makoto Onizuka

DBTest ‘22 on June 17th

Treasure Data is an enterprise customer data platform (CDP) on the cloud



Upgrading query engine is always tough

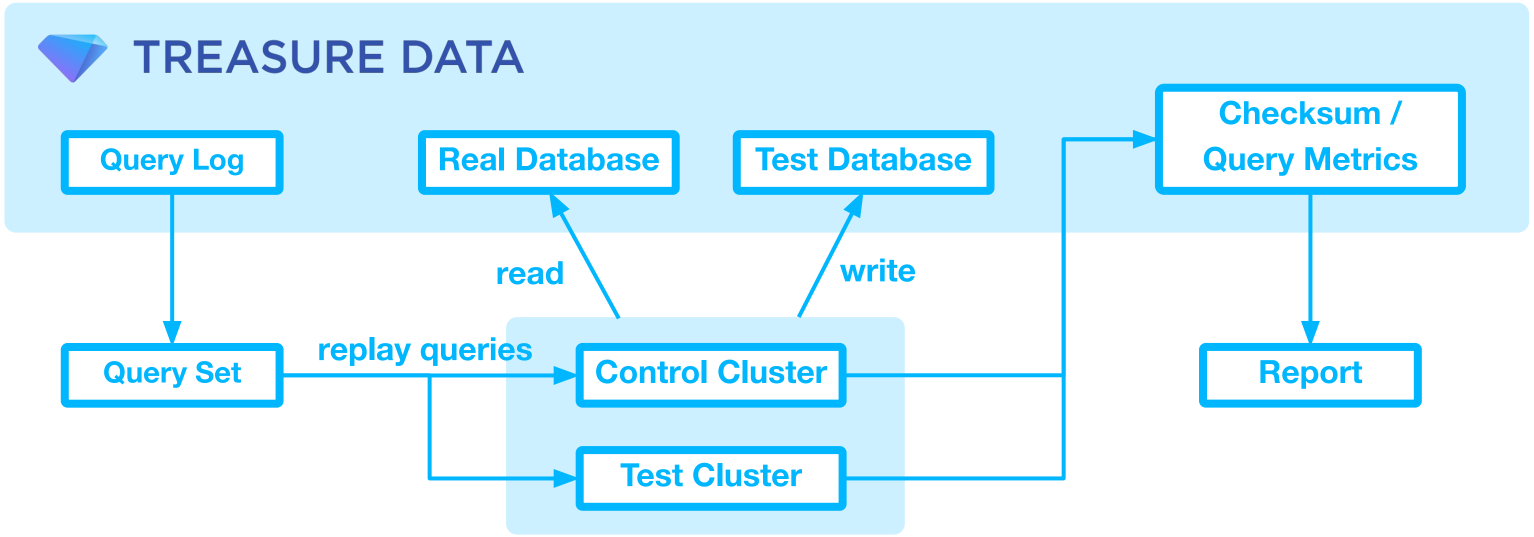
- **Various customer use cases**
  - There are many edge cases in queries, data, and combination of both
    - General benchmark and test cases are not enough
  - Need to minimize customer frustration caused by upgrading
    - Keep backward compatibility as much as possible
    - Notify customers of incompatible queries and how to fix them in advance if we will break compatibility

- **Activeness of OSS development**
  - In particular, Trino development is super active
    - Monthly or more frequent release with hundreds of commits
    - No stable versions
  - But staying at the same version so long is also painful
    - Cannot use new features and optimizations unless backporting
    - Backporting will get harder over time

TREASURE DATA

Query simulator

Test using production data and queries with security and safety



- **Security:** Don’t show customer data and query results
- **Safety:** Don’t cause any side-effect on customer data

Challenges in query simulation

- **Query simulation takes very long time**
    - Very large number of queries need to be tested
    - Not only time, but also cost of test clusters
        - **We need to make query simulation faster**

- **Result verification is not straightforward**
    - Many false positives and duplications
    - Result analysis tends to rely on personal knowledge
        - **We need to make result verification easier**

TREASURE DATA

How we can make query simulation faster?

- Reduce the number of queries by clustering by query signature
- Reduce the amount of data by narrowing table scan ranges
- Test only specific queries (by period, running time, query type, etc)

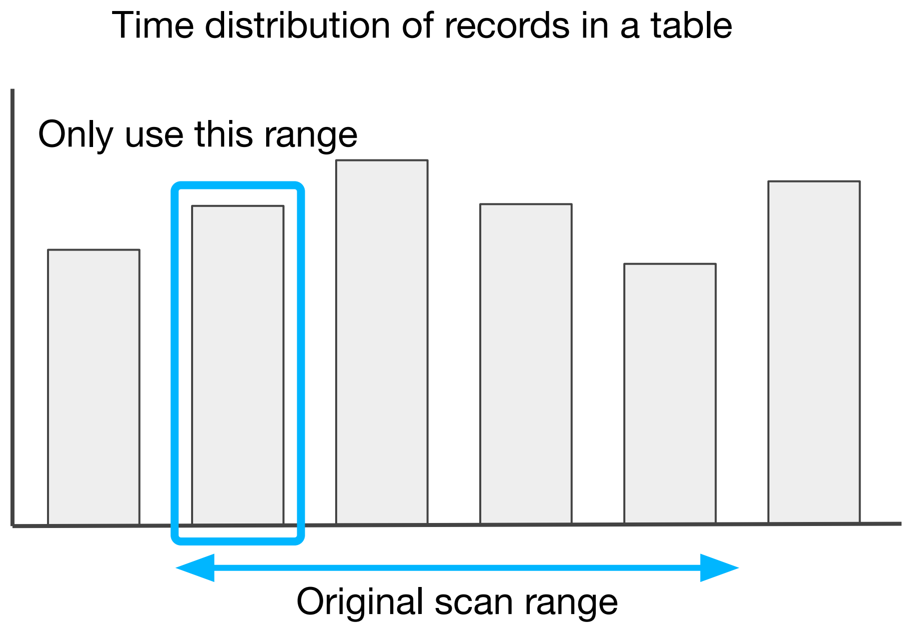
Clustering queries by query signature

Reduce 90% of queries a day need to be tested

Query signature	Corresponding SQL statements
S(T)	SELECT ... FROM ...
S[*](T)	SELECT * FROM ... (select all columns)
G(S(T))	SELECT ... FROM ... GROUP BY
S(LJ(T, T))	SELECT ... FROM .. LEFT JOIN ...
WS[A(a,S(T))]	WITH a AS SELECT .. (define aliases to queries)
O(S(T))	SELECT ... ORDER BY
CT(S(T))	CREATE TABLE AS SELECT ...
I(S(T))	INSERT INTO ... SELECT ...
E(S(T))	SELECT distinct ... FROM ... (duplicate elimination)
U(S(T),S(T))	SELECT ... UNION ALL SELECT ...

Narrowing scan ranges

Use only x% of total records by adding a time range predicate



```
SELECT time, parh, user_agent
FROM access
```

```
SELECT time, path, user_agent
FROM (
  SELECT time, path, user_agent
  FROM access
)
WHERE time >= from AND time < to
```

TREASURE DATA

Choose options depending on the purpose

- **For checking query compatibility**
  - Group by query signature and narrow scan ranges
- **For checking performance differences**
  - Test only long-running queries without scan range narrowing
- **For checking detailed behavior of particular queries**
  - Test only specified queries without grouping by query signature and scan range narrowing

Challenges in query simulation

- **Query simulation takes very long time**
  - Very large number of queries need to be tested
  - Not only time, but also cost of test clusters
    - **We need to make query simulation faster**

- **Result verification is not straightforward**
  - Many false positives and duplications
  - Result analysis tends to rely on personal knowledge
    - **We need to make result verification easier**

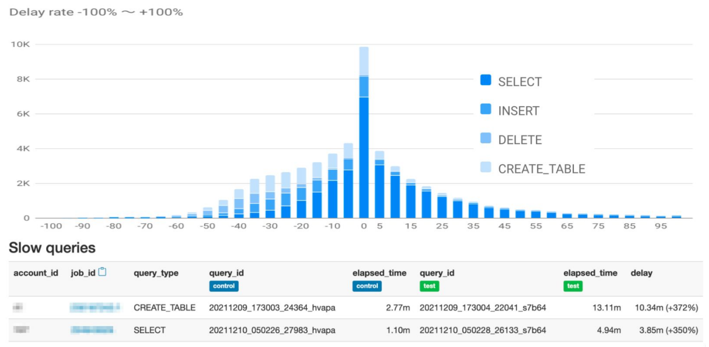
How we can make result verification easier?

- Exclude uncheckable queries as much as possible
- Generate a human-readable report
- Assistance tools for investigation

TREASURE DATA

Reporting for easier result verification

- **List problematic queries**
  - Differences in query results, errors, performance, resource usage, scan ranges, worker distribution, etc
- **Exclude uncheckable queries**
  - Non-deterministic queries
  - Failed queries with the same (or similar) error on both versions
- **Finally, check remaining queries by human**
  - Reduced more than 90% of queries need to be checked
  - In addition, suggest potential cause of different results



Slow queries

account_id	job_id	query_type	query_id control	elapsed_time control	query_id test	elapsed_time test	delay
[redacted]	[redacted]	CREATE_TABLE	20211209_173003_24364_hvapa	2.77m	20211209_173004_22041_s7b64	13.11m	10.34m (+372%)
[redacted]	[redacted]	SELECT	20211210_050226_27983_hvapa	1.10m	20211210_050228_26133_s7b64	4.94m	3.85m (+350%)

TREASURE DATA

Assistance tools for investigation

trino-compatibility-checker
https://github.com/takezoe/trino-compatibility-checker

Run the same query on multiple versions of Trino using docker and compare query results to identify the version that introduced the incompatibility

```
✅ 317: Right(37a6259cc0c1dae299a7866489dff0bd)
❌ 350: Left(java.sql.SQLException: Query failed (#20210526_154140_00004_yzz4q): Multiple entries with same
key: @38f546e: null=expr and @38f546e: null=expr)
✅ 334: Right(37a6259cc0c1dae299a7866489dff0bd)
❌ 342: Left(java.sql.SQLException: Query failed (#20210526_154251_00003_2km75): Multiple entries with same
key: @63f11e0f: null=expr and @63f11e0f: null=expr)
✅ 338: Right(37a6259cc0c1dae299a7866489dff0bd)
❌ 340: Left(java.sql.SQLException: Query failed (#20210526_154338_00002_2xtvz): Multiple entries with same
key: @76615946: null=expr and @76615946: null=expr)
❌ 339: Left(java.sql.SQLException: Query failed (#20210526_154358_00002_jdnni): Multiple entries with same
key: @4aca599d: null=expr and @4aca599d: null=expr)
```

TREASURE DATA

Trino bugs found by query simulation

- #8027 'Multiple entries with same key' error on duplicated grouping of literal values
- #19764 Missing shallowEquals() implementation for SampledRelation
- #10861 Query fails if IS NOT NULL is used for information_schema
- #10937 Predicate push down doesn't work outside the scope of sub query
- #11259 TRY should handle invalid value error in cast VACHAR as TIMESTAMP
- #12199 Fix query planning failure on multiple subqueries due to IllegalStateException at ScopeAware.scopeAwareComparison()

Also, we found many bugs in our target version that had been already fixed in the latest version so we could backported them

TREASURE DATA

Future work

- **More support for investigation**
    - After finding bugs, compatibility/performance issues, the root cause investigation is still tough
    - Some tools or automations for drill-down investigation will be helpful

- **More efficient query simulation**
    - Better workload compression (e.g. exclude more queries that won’t improve the test coverage)
    - Isolate simulation traffic from other production components (e.g. pseudo reproduction of real data by synthetic data)

TREASURE DATA