# Trino For Large Scale ETL

Charles Song (charless@lyft.com)

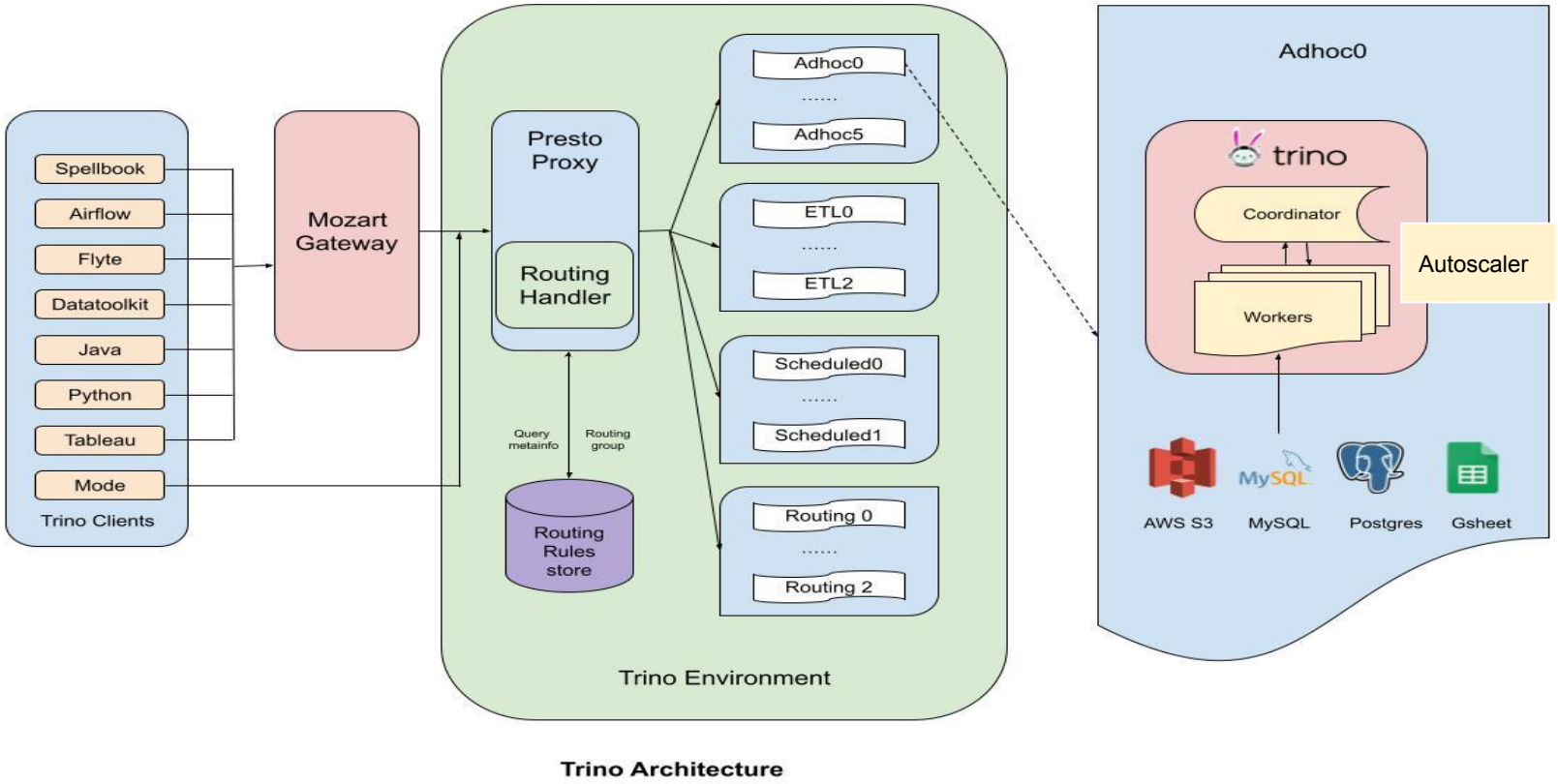
Ritesh Varyani (riteshvaryani@lyft.com)

Data Platform @ Lyft

# Contents

- Trino @Lyft
- Reliability & Efficiency
  - Autoscaling
  - Replay Framework
- Trino ETL @Lyft
  - ETL Infra and Stats
  - ETL User adoption
  - Challenges and what’s next for ETL

# Trino @Lyft




**Trino Architecture**

# Trino @Lyft

## Scale

- 250k queries/day, 2k identical usernames/day
- 10PB daily read data
- 100TB daily write data
- Up to 500 r6g.16xlarge & 250 c6g.16xlarge EC2 instances (auto-scaled)

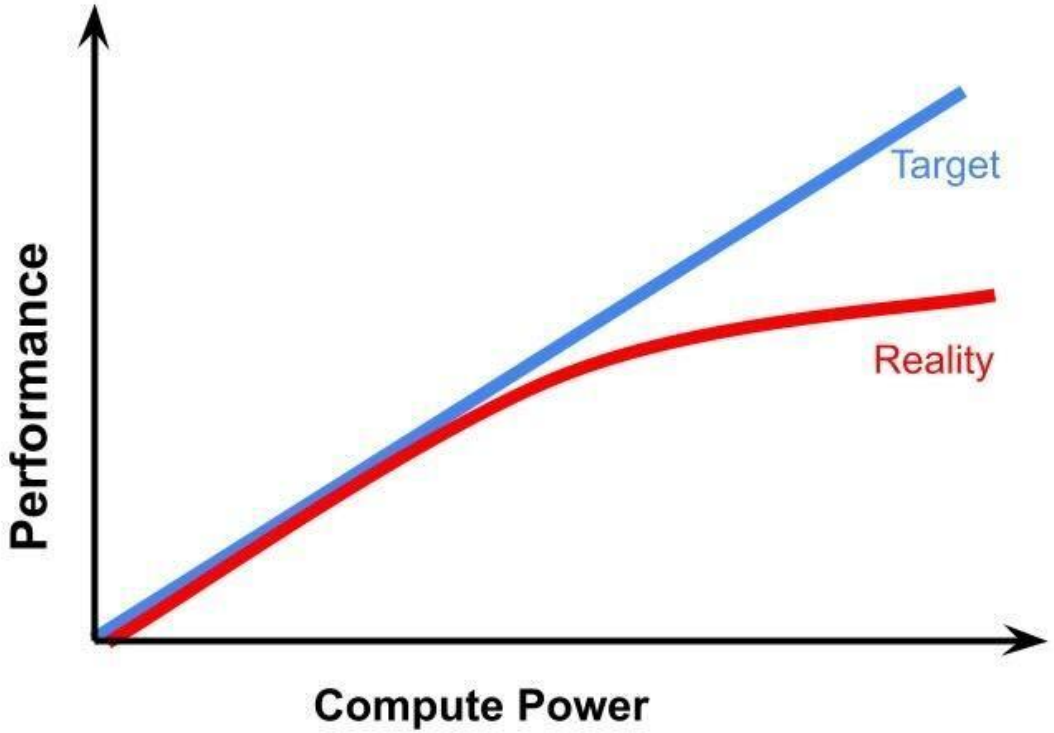
## Operations

- (used to be) ~40 clusters under 15+ routing groups
- 8 different clients

# System Reliability & Efficiency

- Routing Strategy - Concurrency and Queue Management
- Bottleneck and Noise Neighbor
- Coordinator Health
- All Other “Transient” Issues

# System Reliability & Efficiency



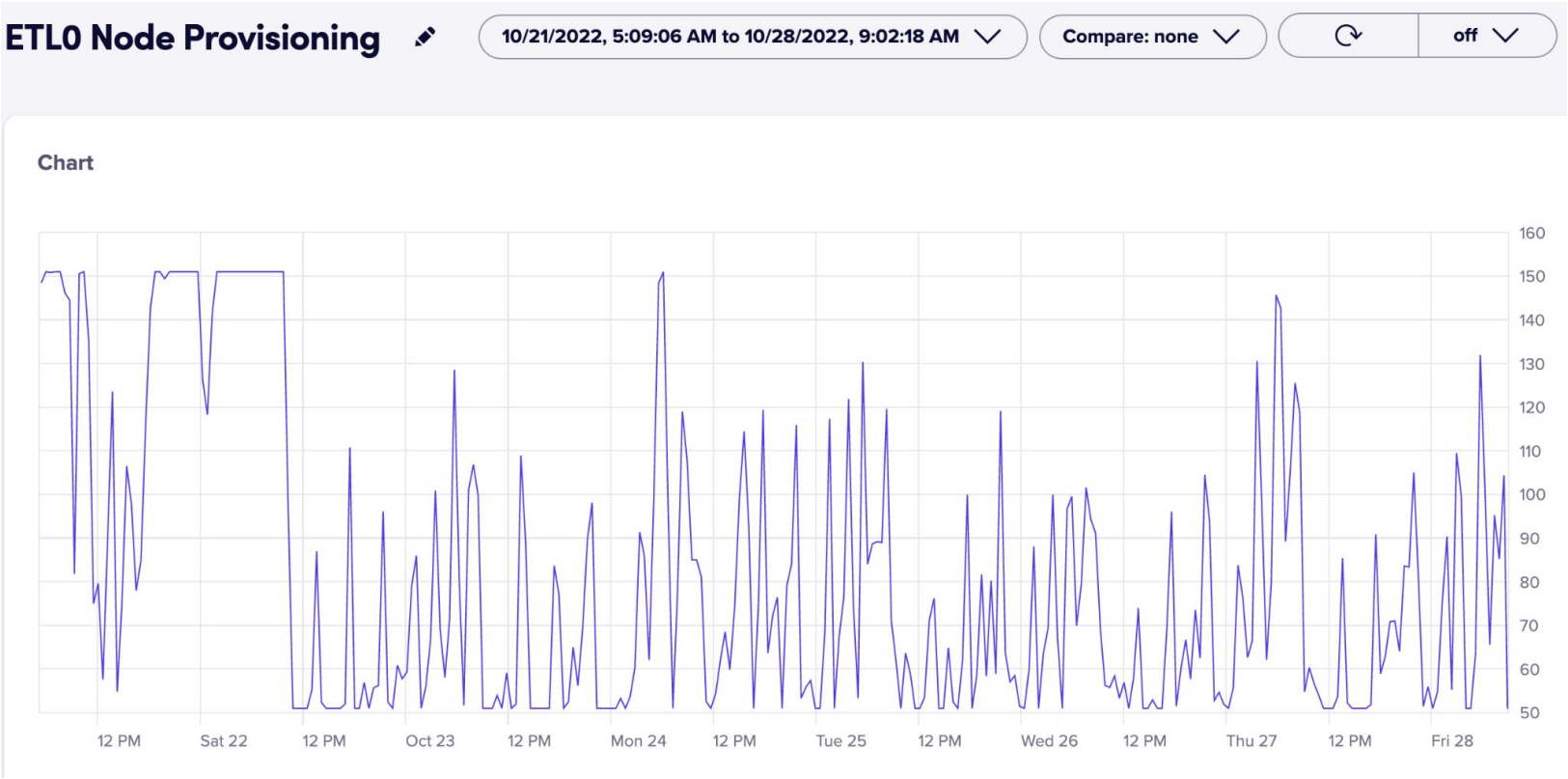
# Autoscaling

- Composite Utilization Score

| cluster_start_time | stop_time | duration | memory_score_stats | | | | cpu_score_stats | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ct | min | avg | med | max | ct | min | avg | med | max |
| 2021-10-08T01:06:43 | 07:21:44 | 06h15m | 376 | 0 | 30 | 30 | 74 | 376 | 0 | 40 | 39 | 74 |
| 2021-10-08T18:05:41 | 07:19:39 | 13h13m | 794 | 0 | 24 | 20 | 67 | 794 | 0 | 41 | 41 | 69 |
| 2021-10-09T18:07:47 | 07:36:41 | 13h28m | 809 | 0 | 24 | 21 | 67 | 809 | 0 | 41 | 41 | 79 |
| 2021-10-10T18:04:45 | 08:36:47 | 14h32m | 870 | 0 | 26 | 23 | 72 | 870 | 0 | 42 | 42 | 70 |

- Decommission and Recommission of workers
  - CloneSet
- Monitoring and Automated Management

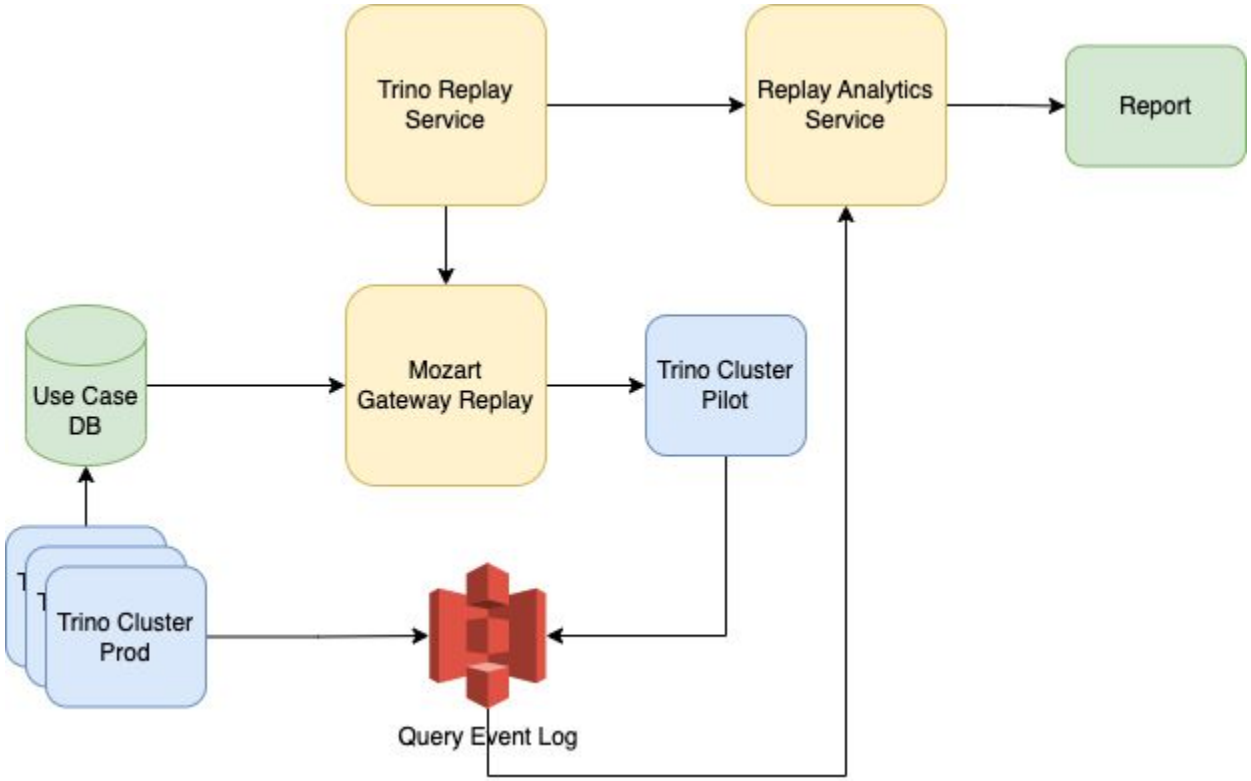
# Autoscaling



# Replay Framework

39

# Replay Framework



# Trino ETL @Lyft

- GA'd July 2022
- Moved all of Trino to AWS Graviton from AWS Intel in H1 2021 (~10% savings)
- Current Trino version: 365
- Push towards deprecating Hive and moving to Trino and Spark
- High demand for faster development iterations/ ANSI-SQl compatible compared to Hive for testing new DAGs and modifying existing ones
- Resiliency built into orchestration layer for ETL
- Create, Insert, DQ, Promote
- Swap partition, insert-overwrite and insert-append with best effort rollback modes developed

# ETL Infra @Lyft

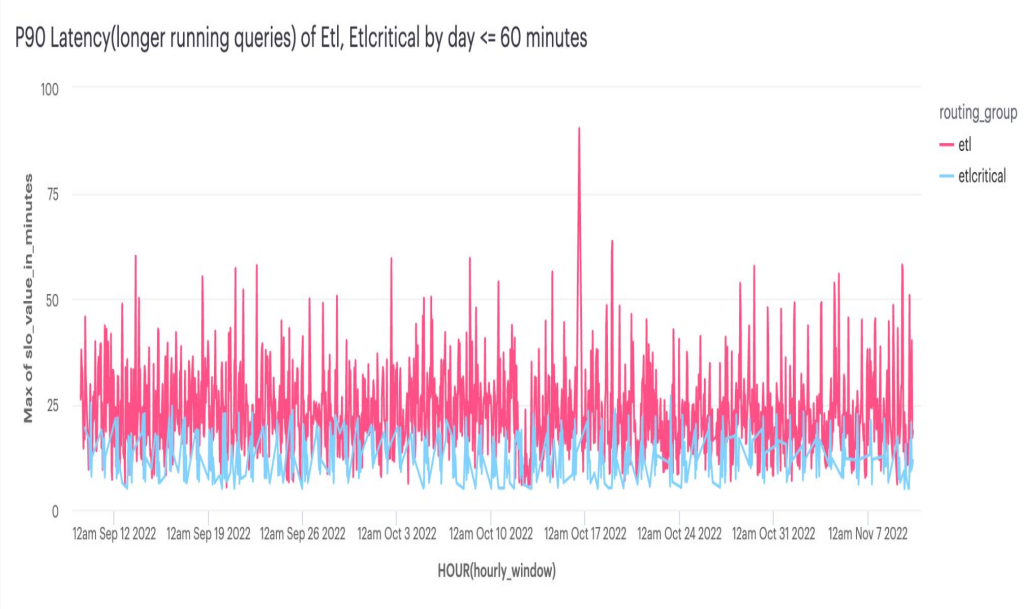
- Separate backends for different use cases:
  - Production ETL DAGs
  - High priority TO Core-Concepts use cases (soon to be consolidated with above)
  - ETL backfill in production (successfully run backfills of rides data of 1 year)
  - ETL testing and DAG development
- Every DAG gets their own resource group
- Within a Trino cluster, we use weighted tiering for preferring high priority DAGs over others
- 2 hour overall runtime limit for queries
- Best practices involving right use of broadcast joins, query sharding and scaling writers for ETL

# ETL Stats

- 2.5PB daily read data
- 60TB daily write data
- 480 unique DAGs
  - Ride data analytics, experimentation platform, localizations, TBS , Vaulting, Privacy and GDPR teams
- 60000 queries per day across different ETL clusters
- P90 latency ~25 minutes

# Monitoring (SLOs)

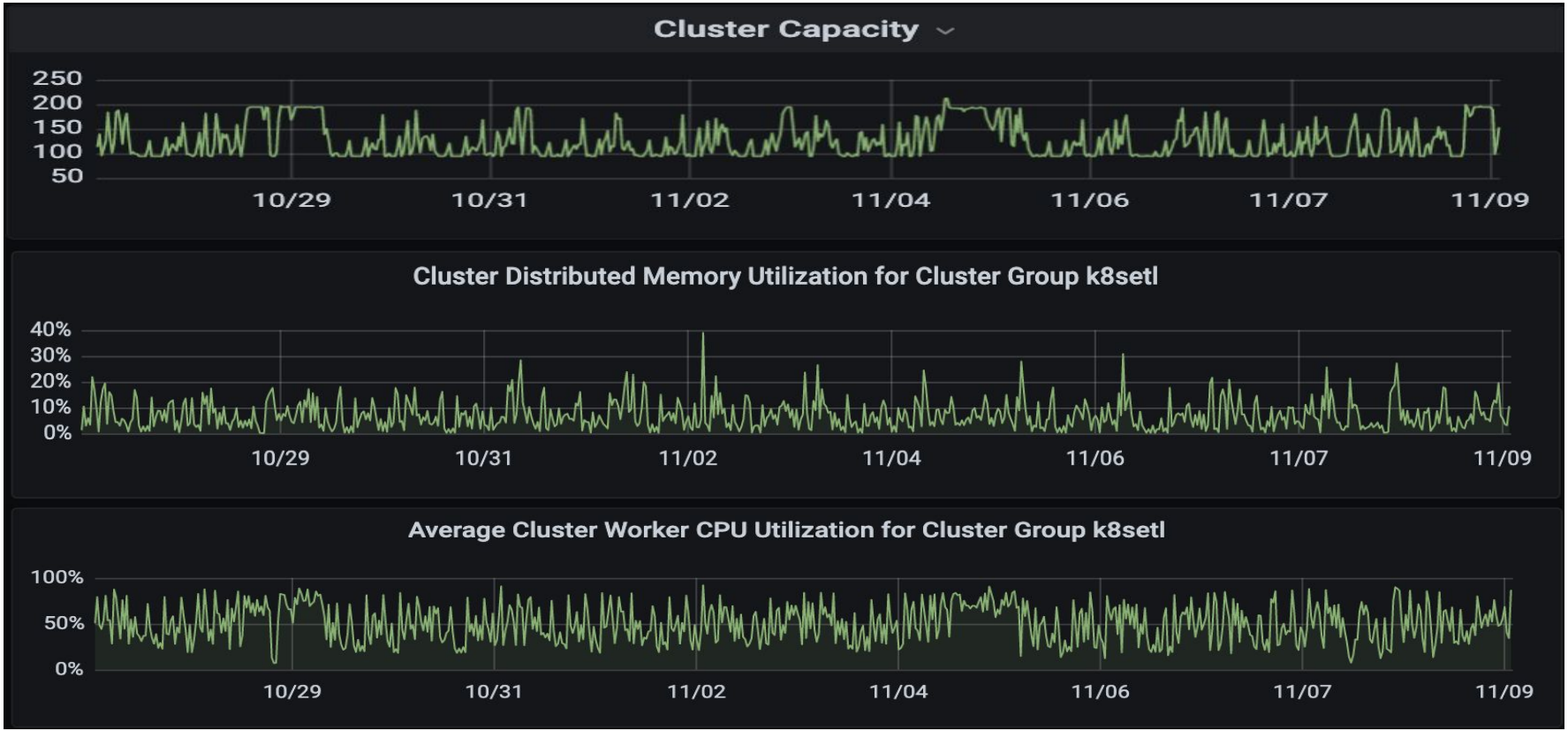
- SLOs established for ETL et al. workloads tracked weekly
  - Cluster Availability
  - Query Reliability
  - Query Success Rate
  - Query Latencies



# Trino ETL User Adoption @Lyft

- Migration off of Hive
  - Custom shadow framework
    - Transpiler framework/ Interpolation of prod schemas and tables with shadow ones for writes
  - Data quality framework
    - Correctness, completeness checks between data sets
  - Manual migration
- Service level tiering experience
  - Mostly able to control with queue limit backpressure and weighted resource groups
- (Relevant) Documentation and Best Practices
- Huge dent in ETL runtimes for high number of use cases
  - Overall ETL DAG runtimes reduced anywhere from 30-90% from Hive to Trino

# Autoscaling aiding during predictable high traffic times



# Challenges for ETL @Lyft

- Challenges
  - Slow rollouts
  - Ensuring reliability for queries in shared tenancy model and changes to data models
  - Accounting for organic growth
  - My query is slow! (query optimization and cluster tuning)

# 365 Upgrade challenges: 10839/10841

**Ritesh Varyani** ⛔ 9 months ago

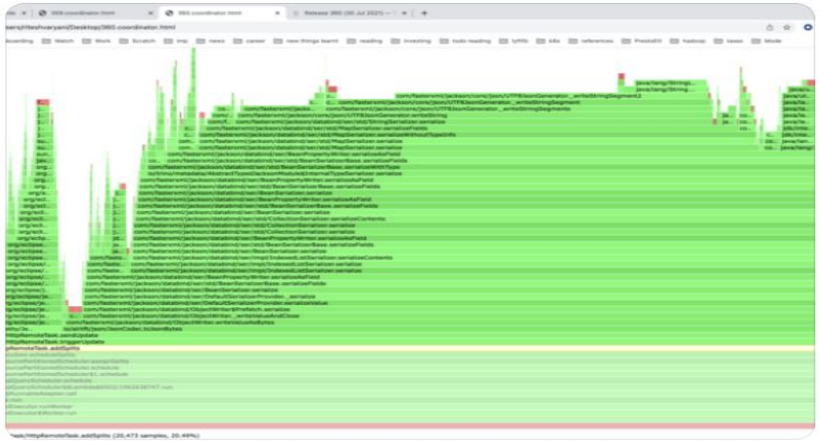
there is a big diff on `addSplits` on coordinator in 359 vs 360

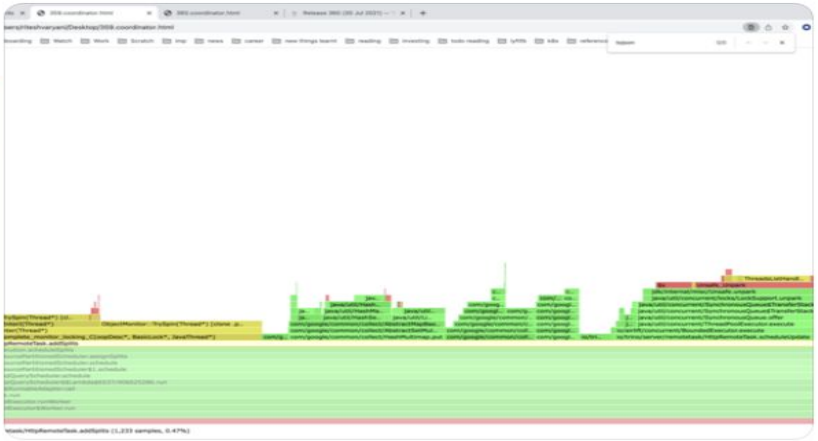
`359 -> 0.47%` is spent in `addSplits`

`360 -> 20.49%` is spent in `addSplits`

**Ritesh Varyani** ⛔ 9 months ago

2 files





# What’s next for ETL @Lyft

- What’s next?
  - Focus on reliability for organic adoption
  - Enable cost based optimizer with Stats collection
  - Sharding at orchestration layer to break down queries
  - Replay framework for writes
  - (Faster) upgrades
    - Enable fault tolerant execution
    - Tardigrade!

# Thank you! Questions?