Cloud Data Processing with Cost-Efficient Latency SLOs using Probabilistic Query Performance Predictions

Data Systems Group @ MIT

Markos Markakis

May 10, 2025

Need to Know

Reservations can be made one day in advance, starting at 10:00 am.

09:59:59 → 10:00:00 → 10:00:02 → 10:00:05

Dinner

4:00 PM Dining Room

10:00 PM Dining Room

Notify

SELECT COUNT(*) FROM covers
WHERE date >= '2025-03-01'
AND date <= '2025-03-31'

Total Covers

Month	Days of service	Total covers	Reserved covers	Walkin covers	Waitlist covers	No show covers parties
March	12	84	51	33	0	4 / 2

Average Daily Covers (vs previous month)

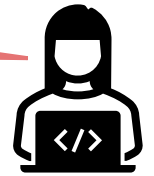
Service	Avg covers	Avg reserved covers	Avg walkin covers	Avg waitlist covers	Avg no show covers	Avg no show res rate
Breakfast	2(-14 ▼)	1(-3 ▼)	1(-11 ▼)	0(0 —)	0(0 —)	33.33%(33.33% ▲)
Lunch	2(-1 ▼)	2(-1 ▼)	0(0 —)	0(0 —)	1(1 ▲)	16.67%(16.67% ▲)
Dinner	7(3 ▲)	4(1 ▲)	3(2 ▲)	0(0 —)	0(0 —)	0%(0% —)

Change in seated diners by week, 2025 vs. 2024

This graph measures the weekly change in seated diners from online reservations for 2025 vs. 2024. Hover over any given date to see how 2025 compares to the respective week in 2024. For example, in the US on the week ending on January 6, 2025, seated diners were up 25% compared to the respective week of the year in 2024.

Global

SELECT...



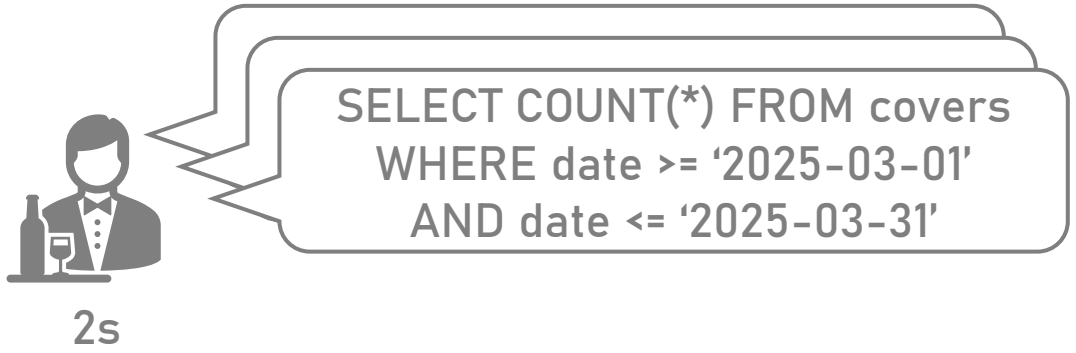
SELECT * FROM
available_tables
WHERE date =
'2025-05-10'

Dinner

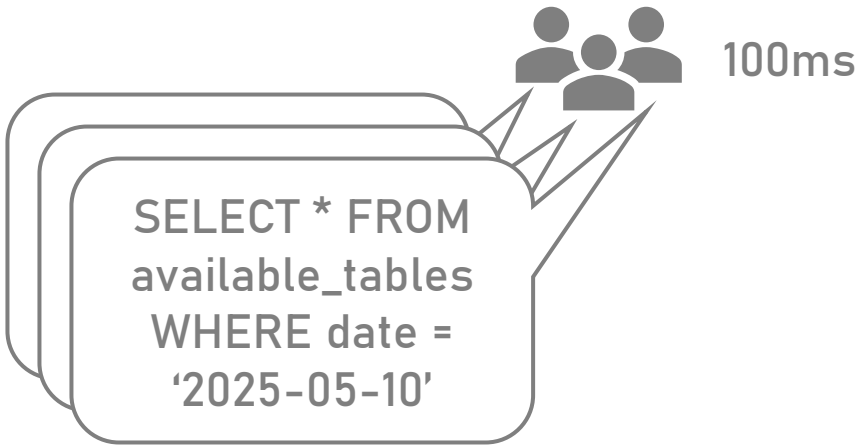
4:00 PM
Dining Room

10:00 PM
Dining Room

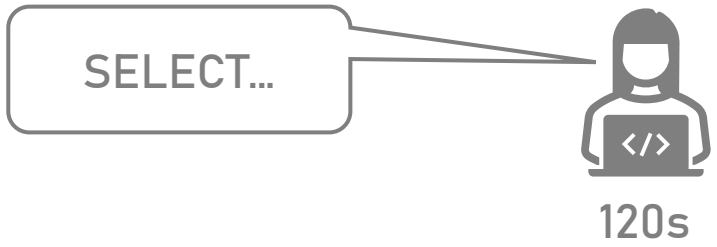
Notify



How can we make them all "fast enough", reliably and cheaply?



100ms



Cloud Data Processing

with Cost-Efficient Latency SLOs

using Probabilistic Query Performance Predictions

Databases are increasingly cloud-hosted

Look under the hood only when you create a database

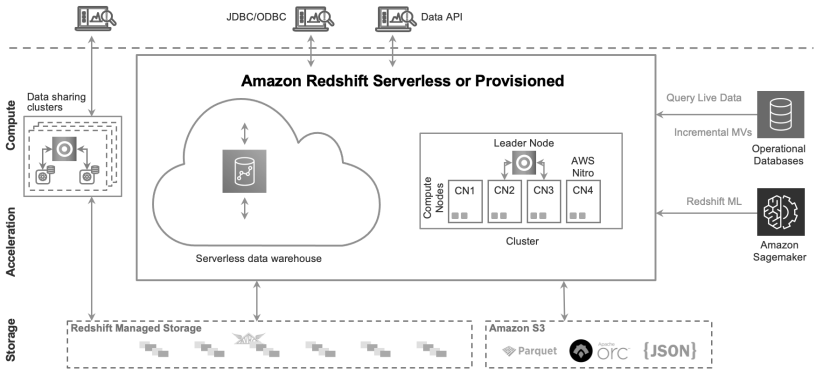
Pay only when you run queries

Run queries when you want

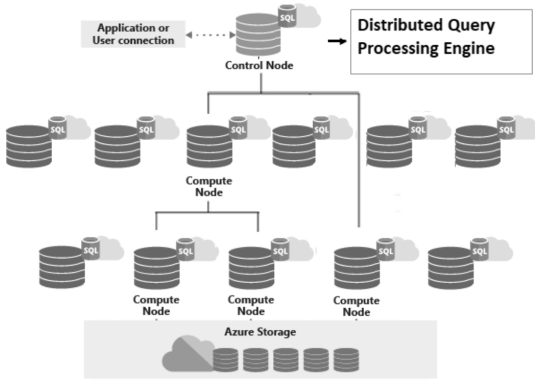
Look under the hood only when you create a database

…but more abstraction causes latency uncertainty

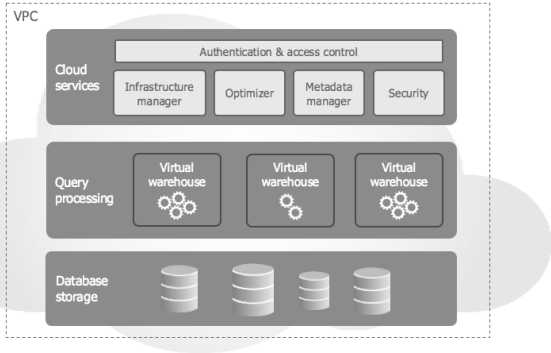
amazon REDSHIFT

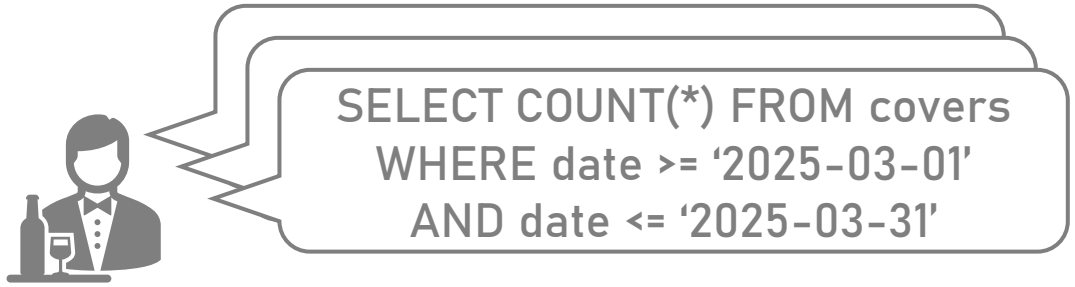


Azure Synapse Analytics



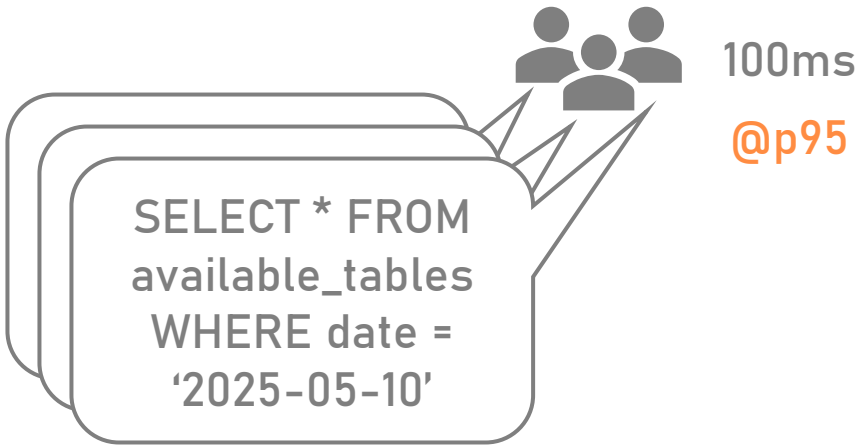
snowflake





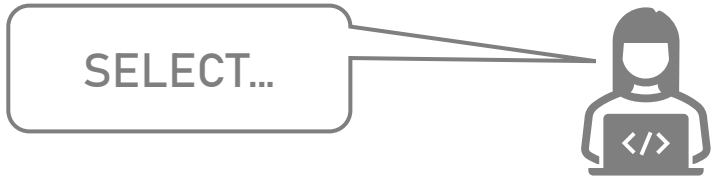
2s

@p95



100ms

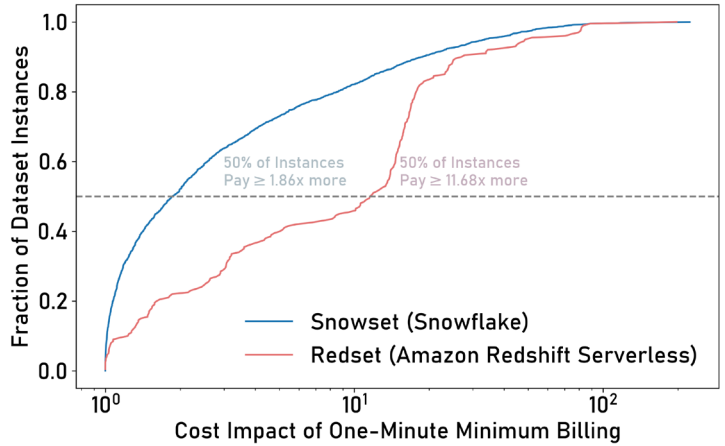
@p95



120s

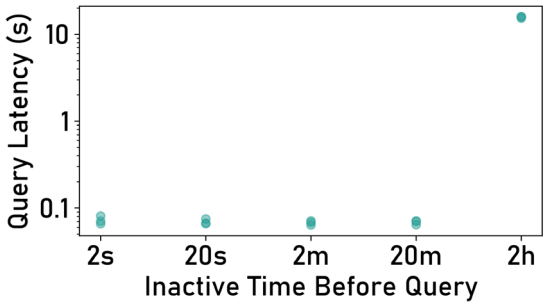
@p95

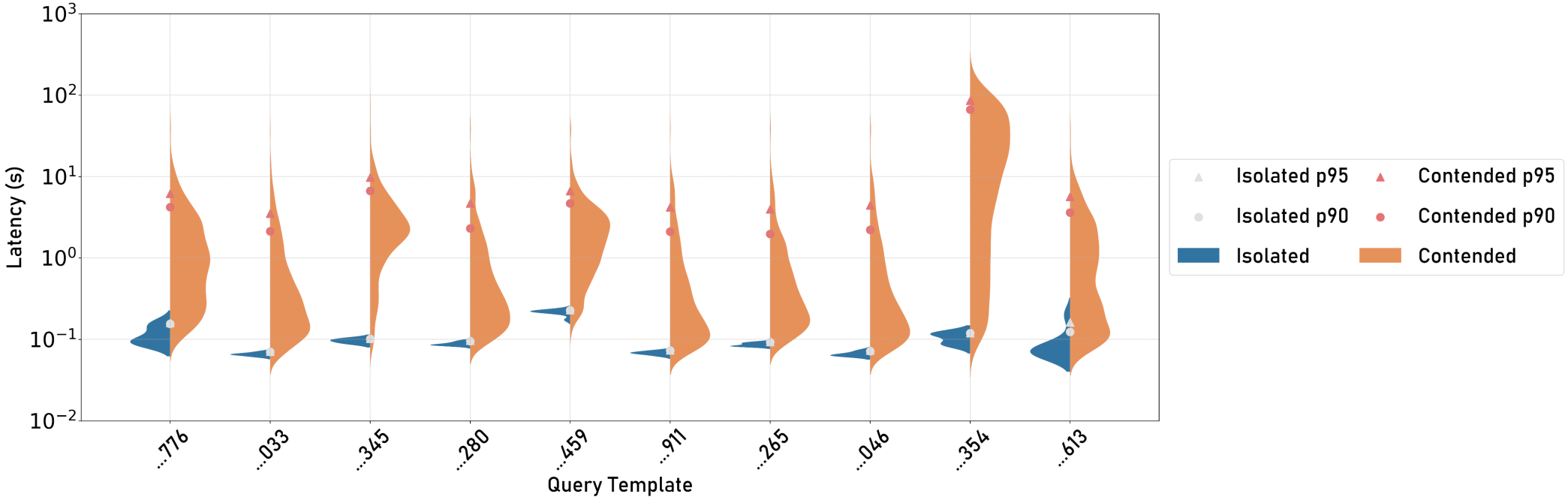
Amazon Redshift Serverless and
Snowflake have 1-minute billing minima



Pay only when you run queries

…but big gaps can hurt both latency and cost





Run queries when you want

…but risk latency-impacting interference

Look under the hood only when you create a database

…but more abstraction causes latency uncertainty

Pay only when you run queries

…but big gaps can hurt both latency and cost

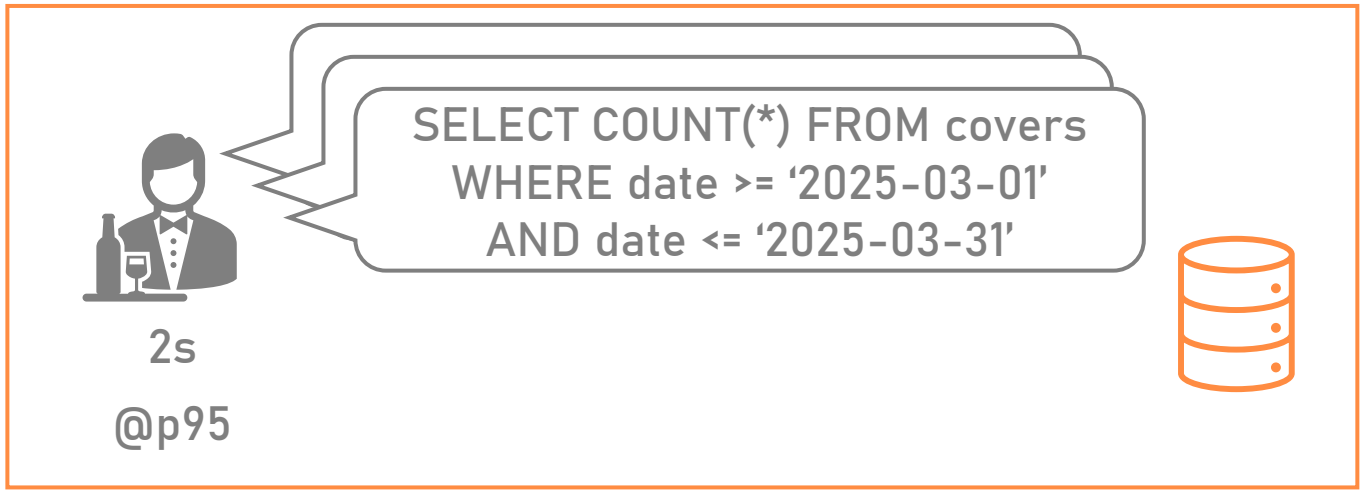
Run queries when you want

…but risk latency-impacting interference

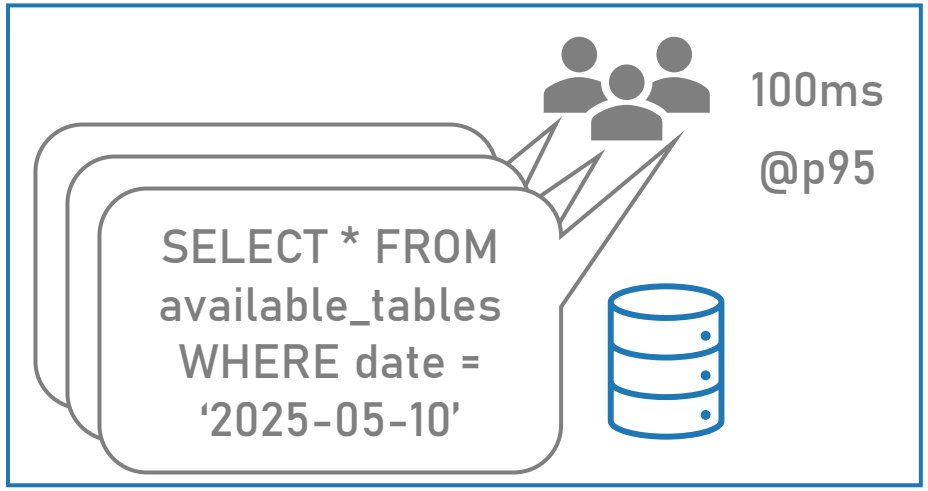
Cloud Data Processing

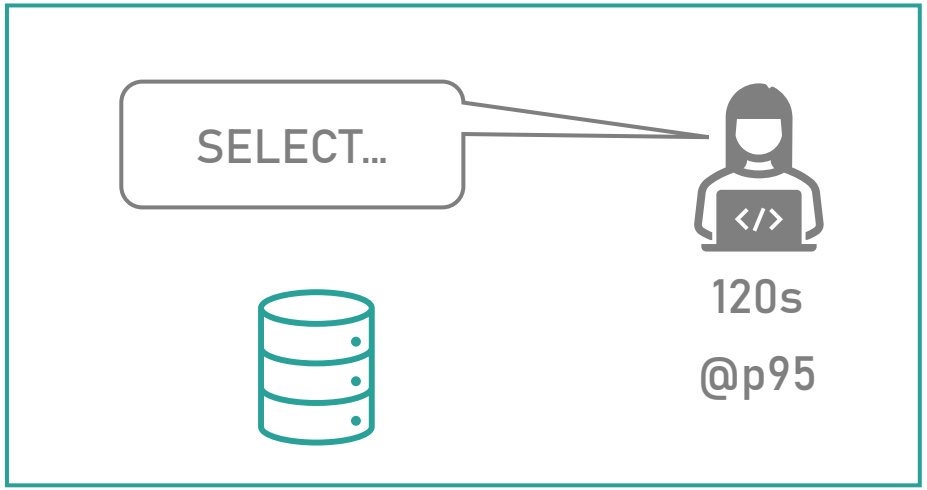
with Cost-Efficient Latency SLOs

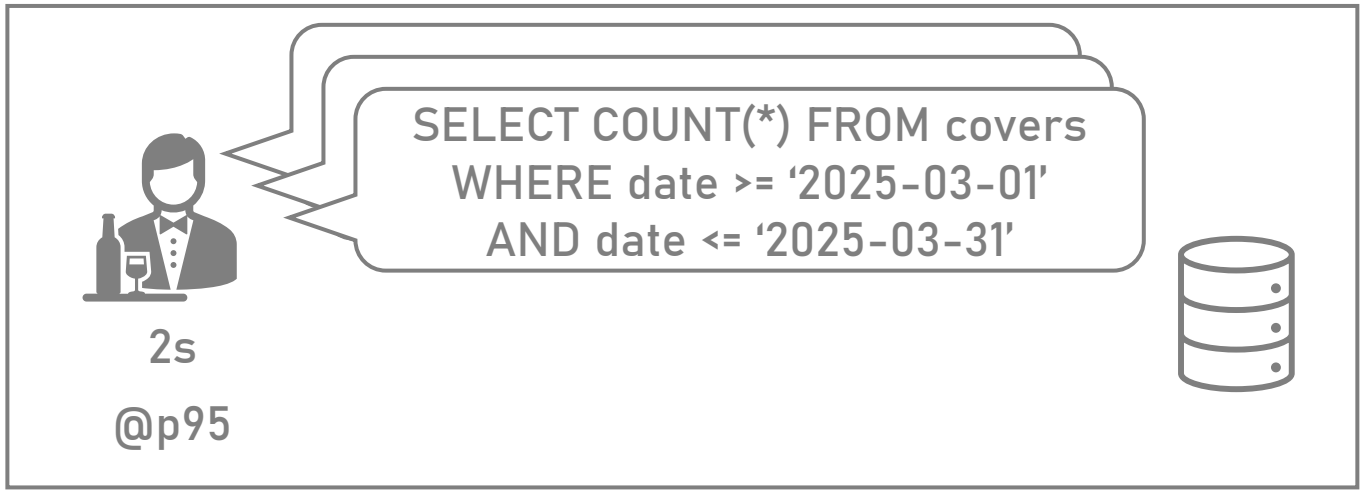
using Probabilistic Query Performance Predictions

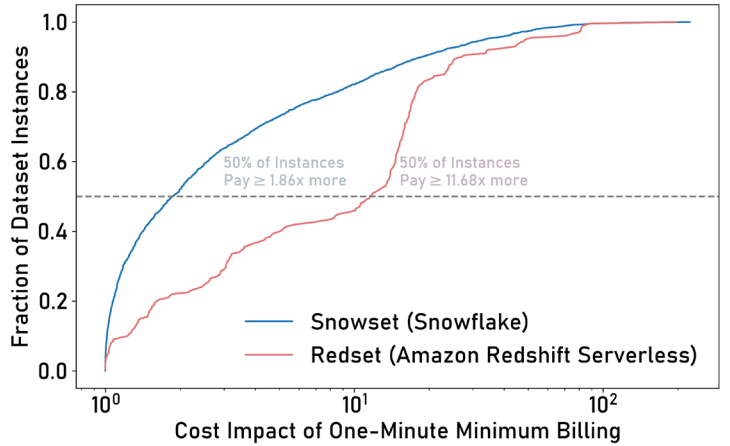


Why not just isolate the workloads?



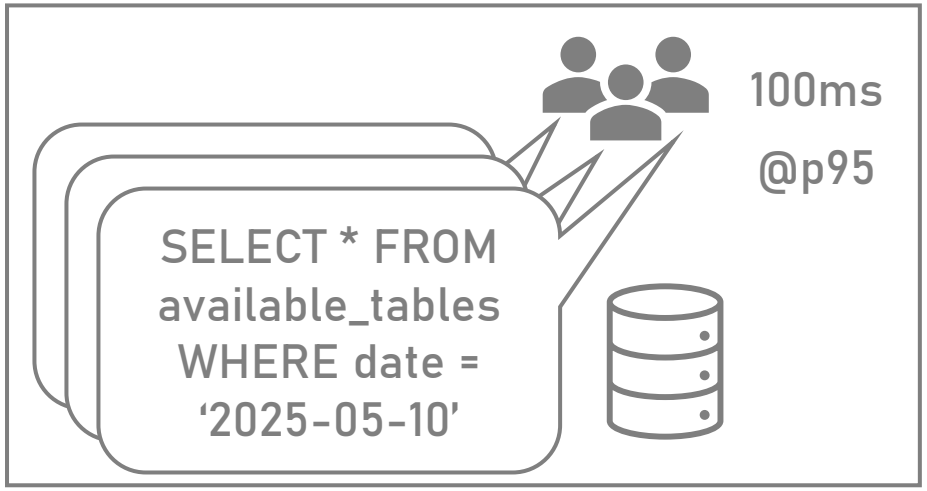


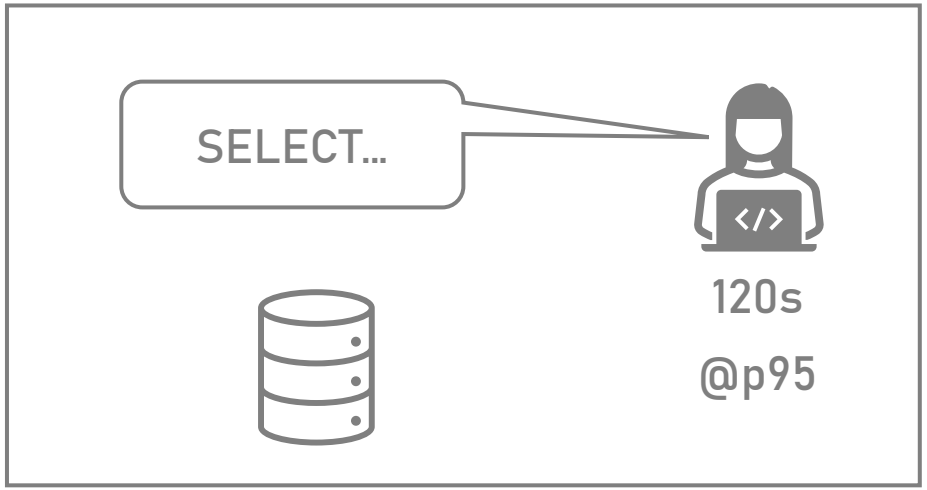


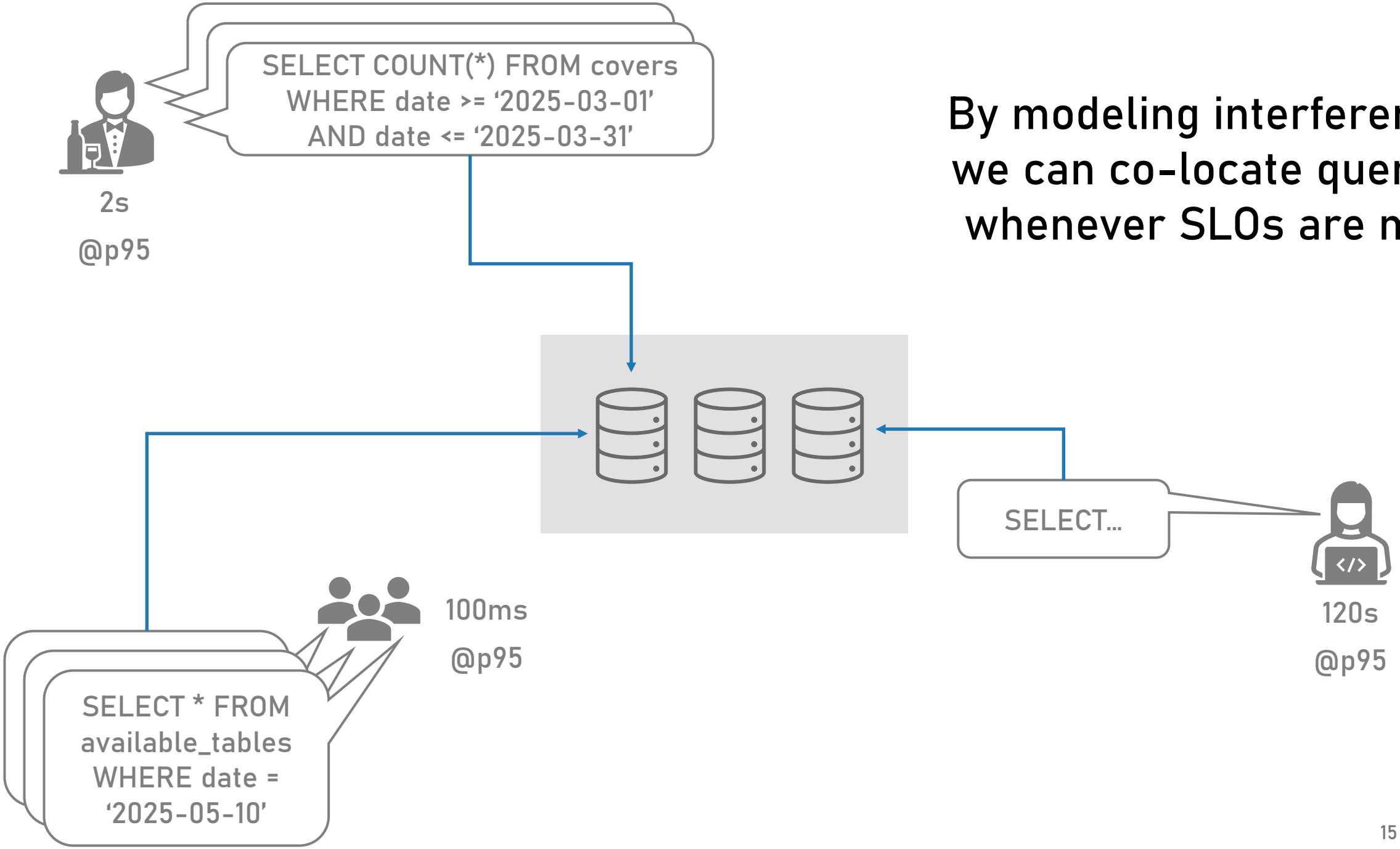


Why not just isolate the workloads?

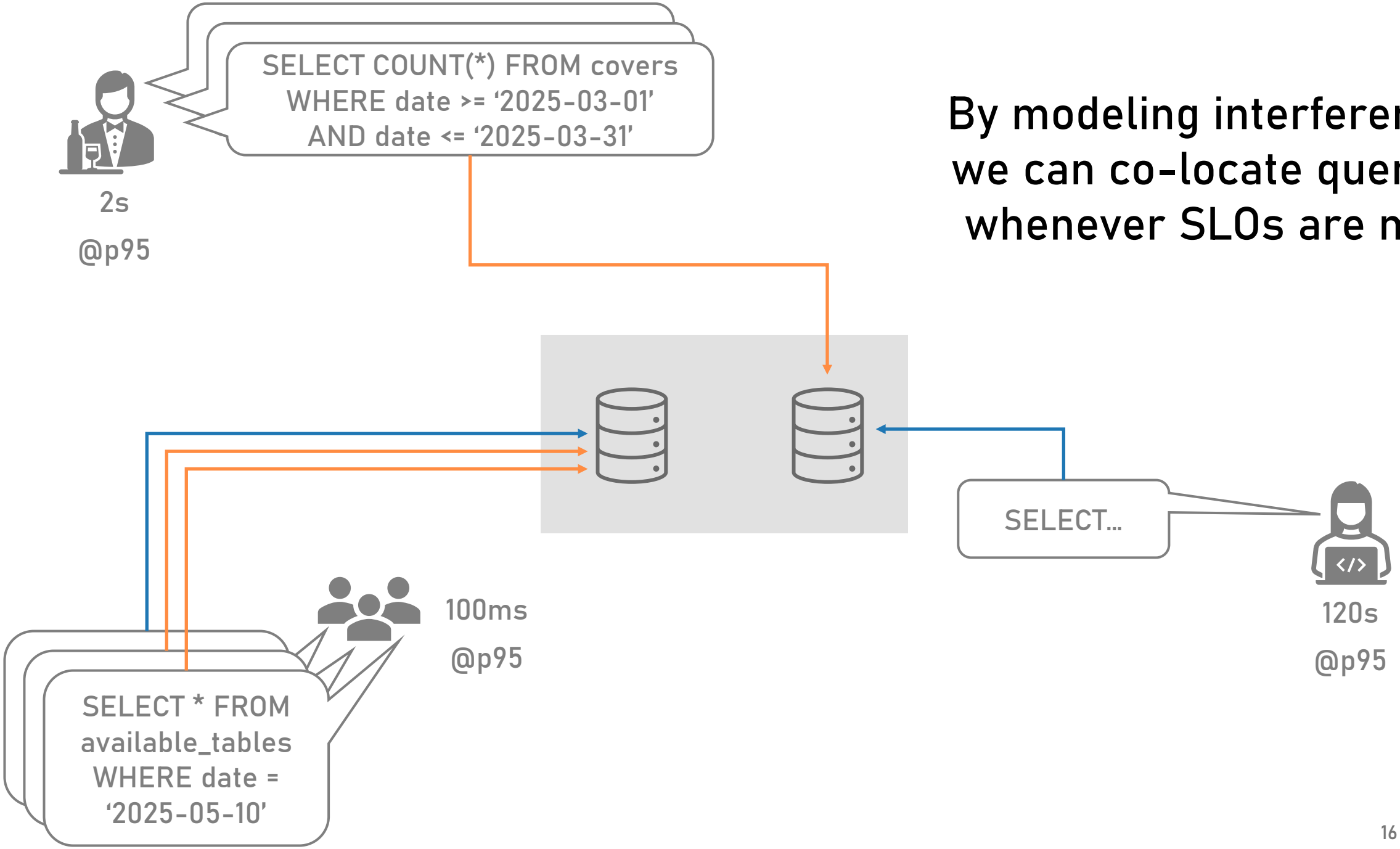
Costly







By modeling interference, we can co-locate queries whenever SLOs are met



By modeling interference, we can co-locate queries whenever SLOs are met

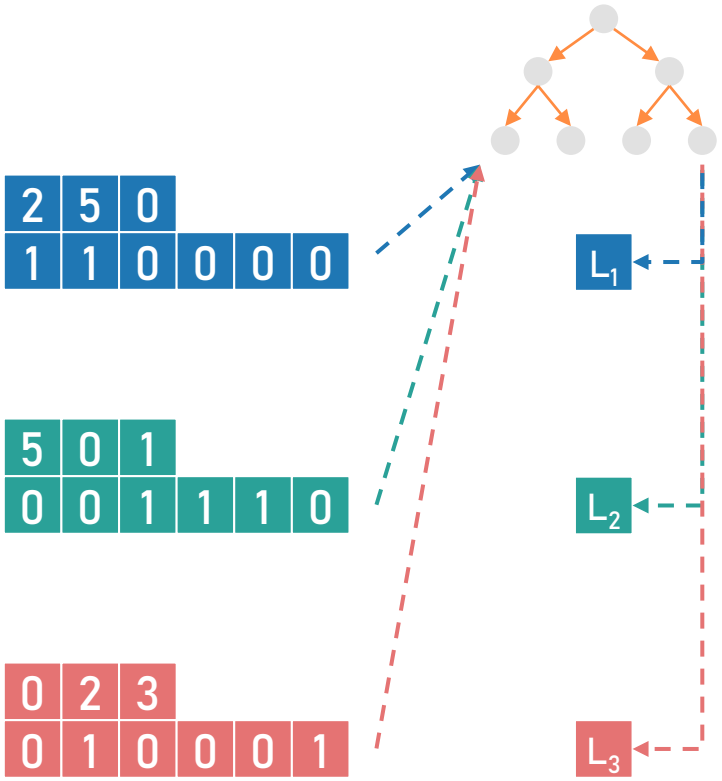
Cloud Data Processing
with Cost-Efficient Latency SLOs
using Probabilistic Query Performance Predictions

Queries fight for resources like CPU and Memory

Q1
Operators: 2xO1, 5xO2
Tables accessed T1, T2

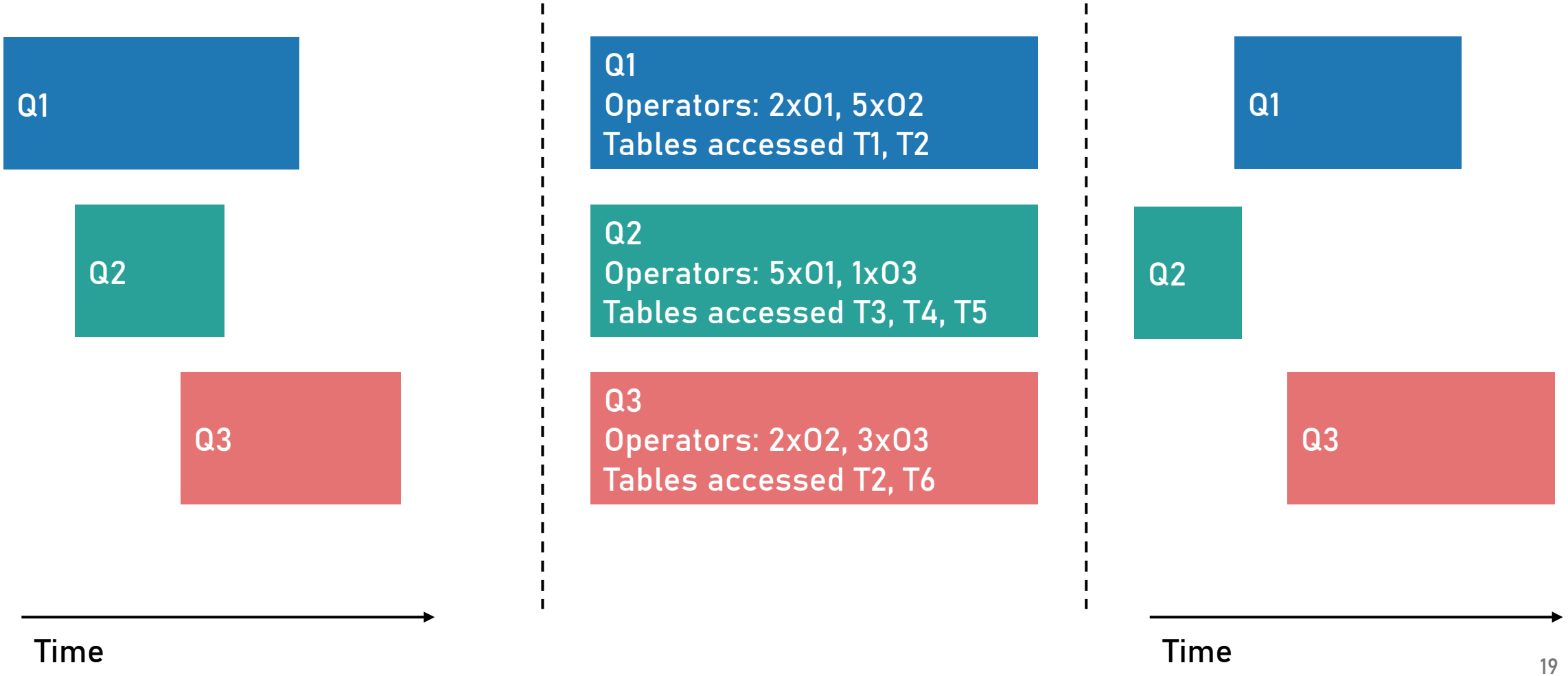
Q2
Operators: 5xO1, 1xO3
Tables accessed T3, T4, T5

Q3
Operators: 2xO2, 3xO3
Tables accessed T2, T6

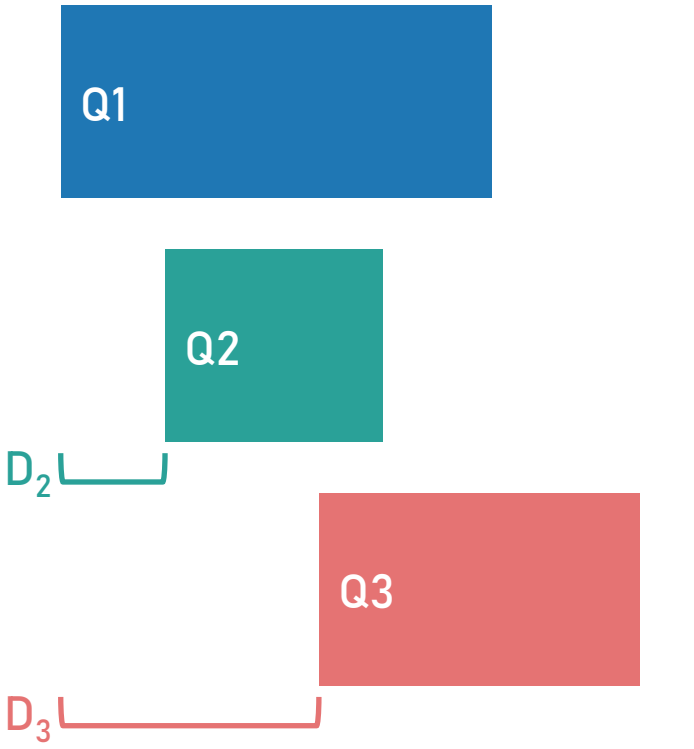


Queries fight for resources like CPU and Memory

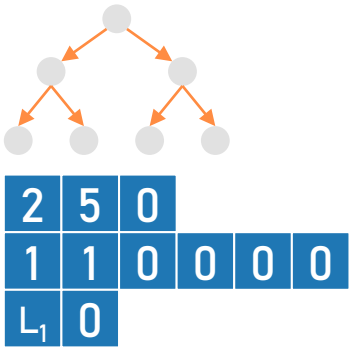
…but the latency impact depends on relative timing



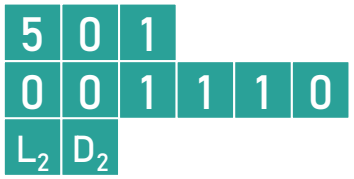
We can use arrival time differences as features



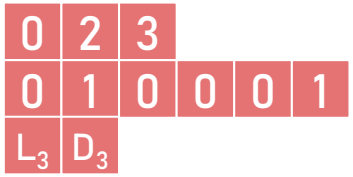
Q1
Operators: 2xO1, 5xO2
Tables accessed T1, T2



Q2
Operators: 5xO1, 1xO3
Tables accessed T3, T4, T5

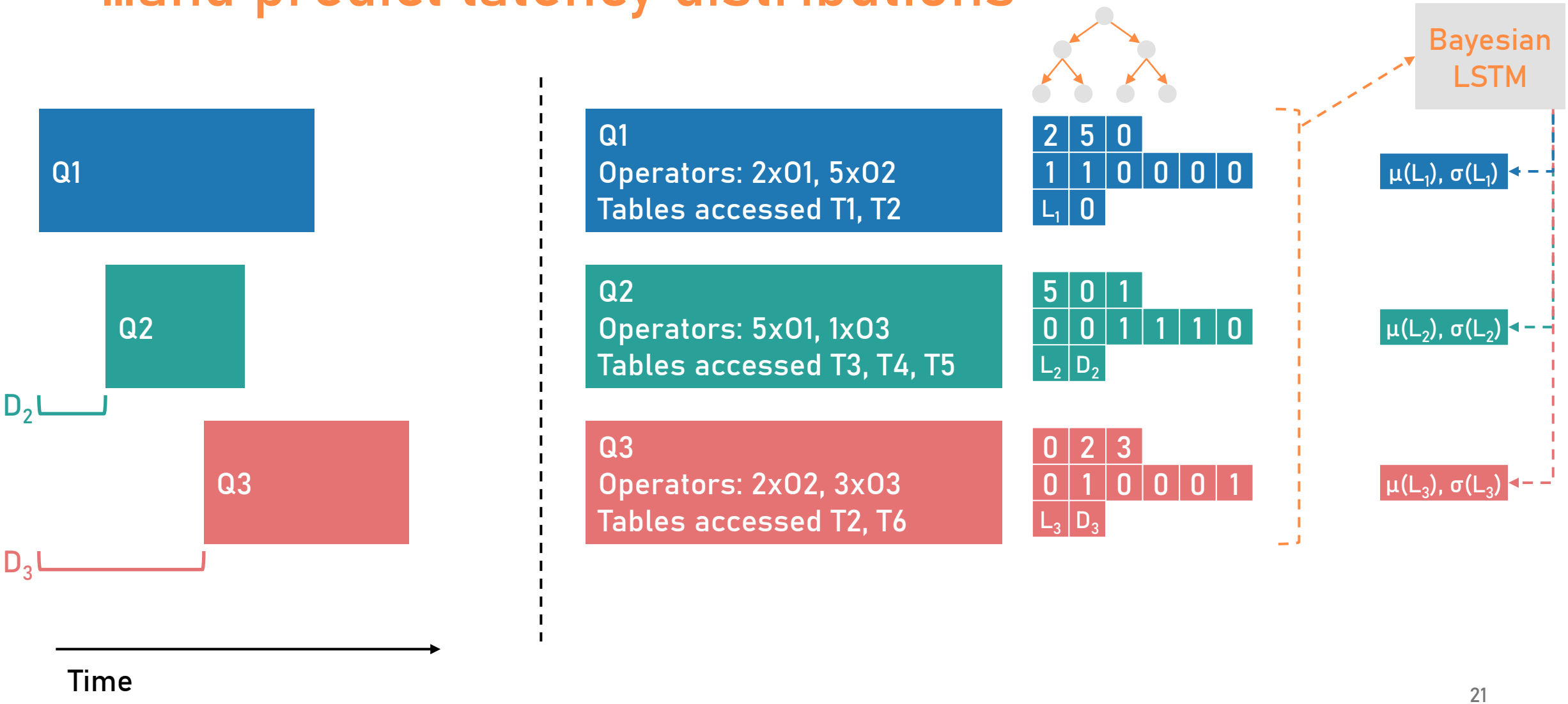


Q3
Operators: 2xO2, 3xO3
Tables accessed T2, T6



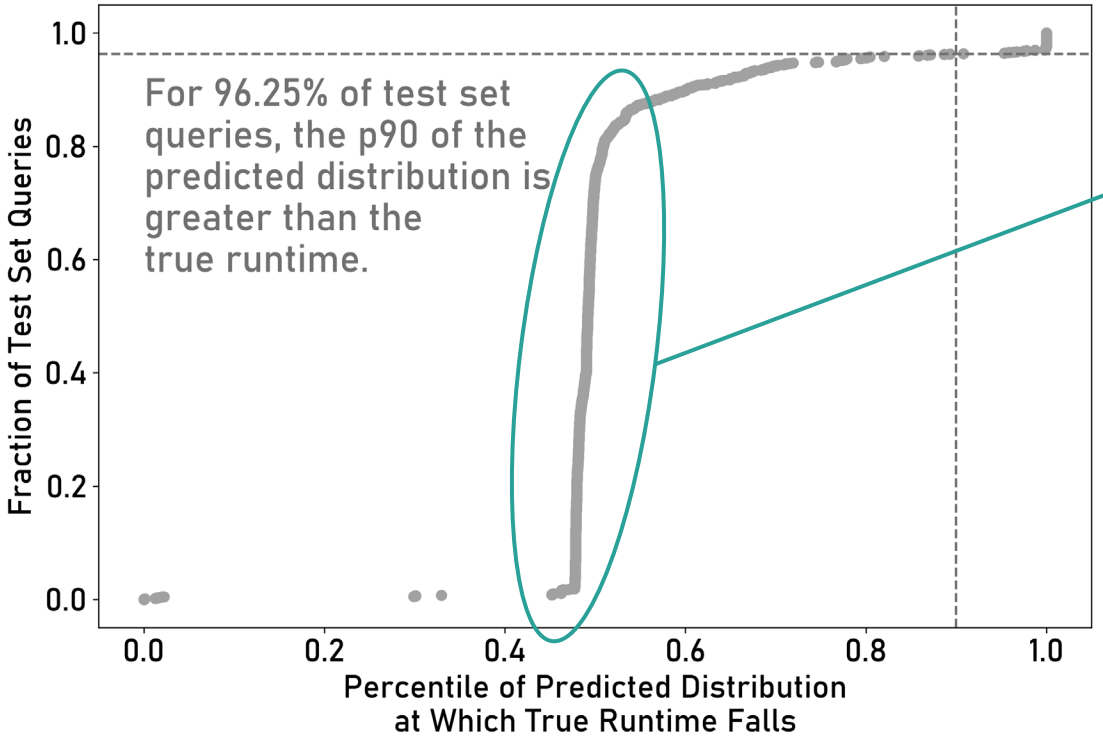
We can use arrival time differences as features

…and predict latency distributions



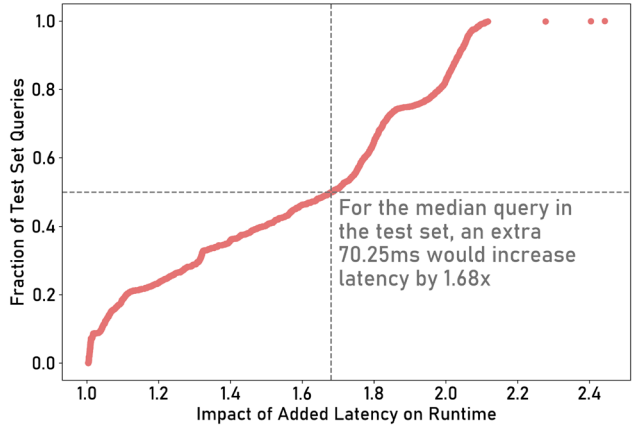
The predicted distributions capture the tail okay

…but some important questions remain



1. High concentration near the mean = predicted distributions are very wide, what better type of distribution to assume?

2. Median inference time is 40.13ms but the maximum can be 70.25ms, how can we bring this down?



Thank you!

Ziniu Wu

Tim Kraska