DATA+AI SUMMIT
BY databricks

Benchmarking Data and AI Platforms

Shannon Barrow, Lead Solutions Architect, Databricks
Joe Harris, Sr Software Engineer, Databricks

©2024 Databricks Inc. — All rights reserved

Benchmarking Data and AI Platforms

Joe Harris, Sr Software Engineer

Shannon Barrow, Lead Solutions Architect

©2024 Databricks Inc. — All rights reserved

~~Benchmarking Data and AI Platforms~~

How Much You Bench, Bro?

Shannon Barrow, Lead Solutions Architect

- Joined Databricks in March 2019
- Previously: Principal - Innovation and Thought Leadership, Accenture Applied Analytics
- Despite overarching benchmark discussion, I may:
  - Put extra focus on TPC-DI
  - Put on my Databricks hat for short segments

©2024 Databricks Inc. — All rights reserved

Today's Scope

Focus on LAKEHOUSE and AI Related Benchmarks

Lakehouse/OLAP

- Primarily TPC but others will be mentioned
- Suggestion: view the following benchmarks through the lens of a full end-to-end Lakehouse architecture
- How can an organization get a "full picture" of an end-to-end TCO?

ML/Gen AI

- Highlights and challenges
- Focus on Gen AI
- What to benchmark?
- Lessons learned from Mosaic

©2024 Databricks Inc. — All rights reserved

Why Benchmark?

Level-setting on the Value and Limitations of Benchmarks

Lies, Damned Lies, and Benchmarks

*This is Part 1 of a 5-part series on DW/Analytics Performance Benchmarking and its use in **non-traditional** ways, including DW/Analytics Engineering and Maintenance.*

Part 1. Background

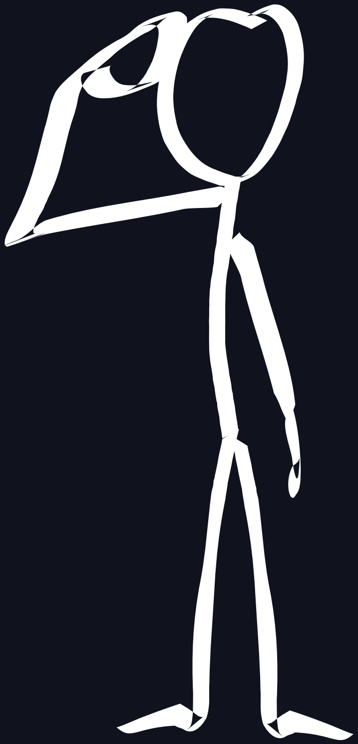
Riffing on an old quote attributable (though apparently not originally) to Mark Twain [1] …

> *There are three kinds of lies: lies, damned lies, and benchmarks [sic]*

We've all seen one-upmanship and "fastest" claims from vendors in the computer technology space using industry-standard benchmarks. From the fastest computers, to the fastest storage systems, to the fastest database systems, there are numerous blogs and papers touting the dominance of one platform or approach over another. In some cases, usually when it suits them, vendors adhere to benchmark specifications and standard reporting requirements. In others, only portions of benchmarks are run and often selectively referenced and co-opted for the desired message. Such is the state of our industry and it will likely not change.

This is the case in the world of Data Warehousing and Analytics. The Transaction Processing Council (TPC) arose out of a need for standard performance metrics and well-defined configuration and pricing guidelines across a number of database vendors and platform offerings. Initially focused on OLTP systems, the organization broadened its work to data warehouse/analytics workloads and other related IT infrastructure areas.

https://rethinkio.com/lies-damned-lies-and-benchmarks/



©2024 Databricks Inc. — All rights reserved

Why Benchmark?

Level-setting on the Value and Limitations of Benchmarks

Level playing field for all platforms

- Standardization and repeatability
  - To conform to the same practices
  - To conform to common industry operations, use cases, input/output, and scale
  - Industry "agreed upon" testing heuristics
- "Official" submissions

Can be hard to believe any results

- Potential for:
  - Cheating
  - Bias
  - Abuse
- Slow pace of modernization

©2024 Databricks Inc. — All rights reserved

Lakehouse Benchmarking

©2024 Databricks Inc. — All rights reserved

TP(P)C - The Ubiquitous Standard

Most prevalent and well-known

Transaction Processing Performance Council

- Formed in 1988
- Benchmarks across multiple domains
  - Decision Support (OLAP)
    - TPC-DI, TPC-H, TPC-DS
  - Transaction Processing (OLTP)
    - TPC-C, TPC-E
  - “Big Data”
    - TPC-HS, TPC-BB
  - Virtualization
    - TPC-V, TPC-HCI
  - Internet of Things (TPC-IOT)
  - AI (TPC-AI)

These are the only ones in scope today

Active Benchmarks Per TPC Website:

Active Benchmarks

Benchmark/Document	Current Version
TPC-C	5.11.0
TPC-DI	1.1.0
TPC-DS	3.2.0
TPC-E	1.14.0
TPC-H	3.0.1
TPCX-AI	1.0.3.1
TPCX-BB	1.6.2
TPCX-HCI	1.1.9
TPCX-HS	2.0.3
TPCX-IOT	2.1.0
TPCX-V	2.1.9

©2024 Databricks Inc. — All rights reserved

Other OLAP Benchmarks

Are TPC Benchmarks The Only Game in Town?

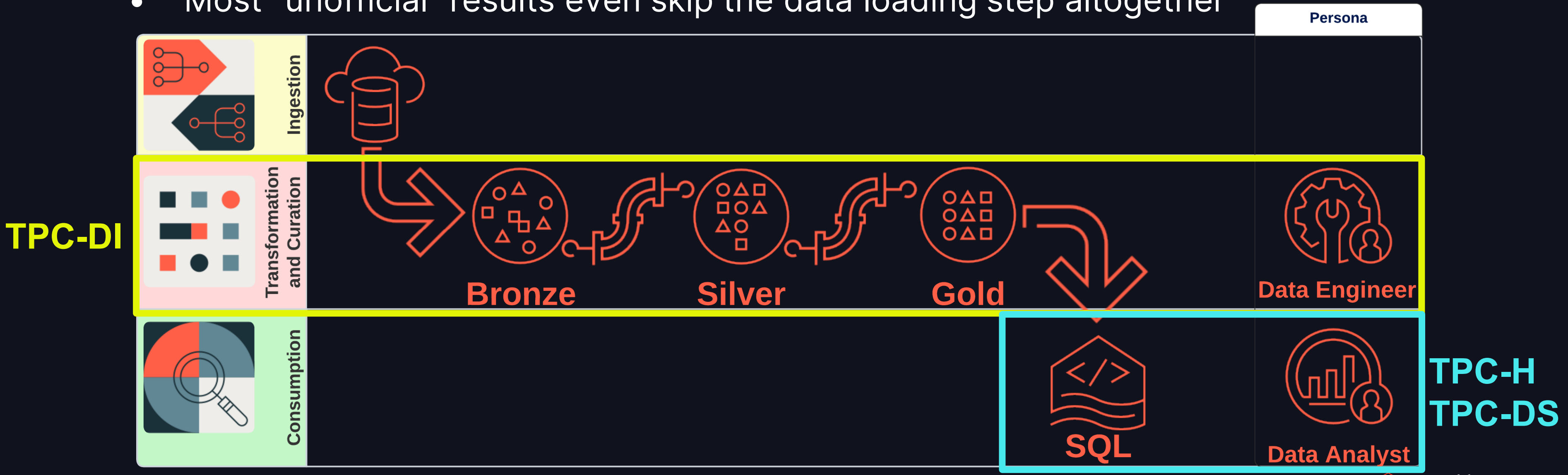
- SSB (Star Schema Benchmark)
  - Designed to measure the performance of databases in a star schema setup
  - Simpler than TPC benchmarks but focused on specific aspects of OLAP querying
- ClickBench, The No-Join Benchmark
  - Focuses on workloads without joins
  - Simulates scenarios common in clickstream analytics

©2024 Databricks Inc. — All rights reserved

Lakehouse Focus - 10k Foot View

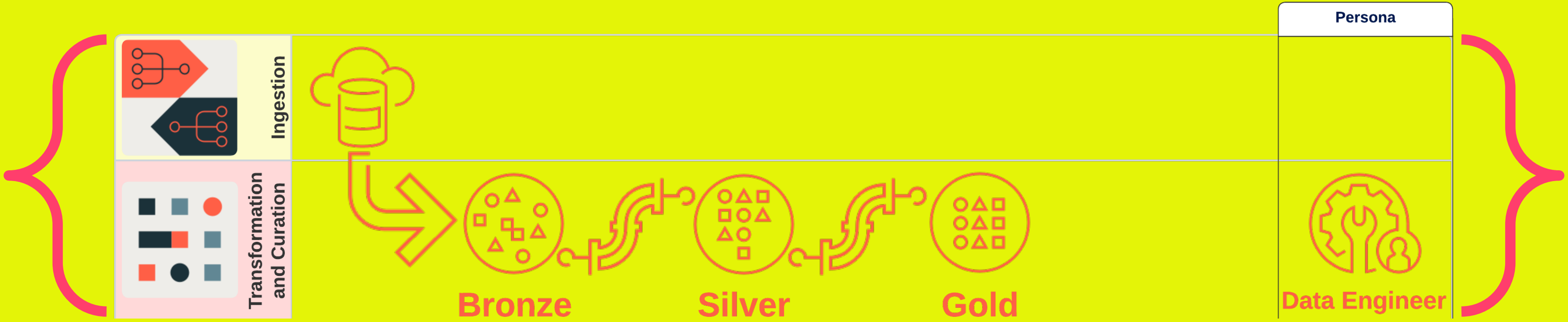
TPC fragments the Lakehouse Architecture into separate benchmarks

- There is no SQL consumption in TPC-DI
- No transformations in TPC-H or TPC-DS
  - Most "unofficial" results even skip the data loading step altogether



©2024 Databricks Inc. — All rights reserved

ETL: TPC-DI

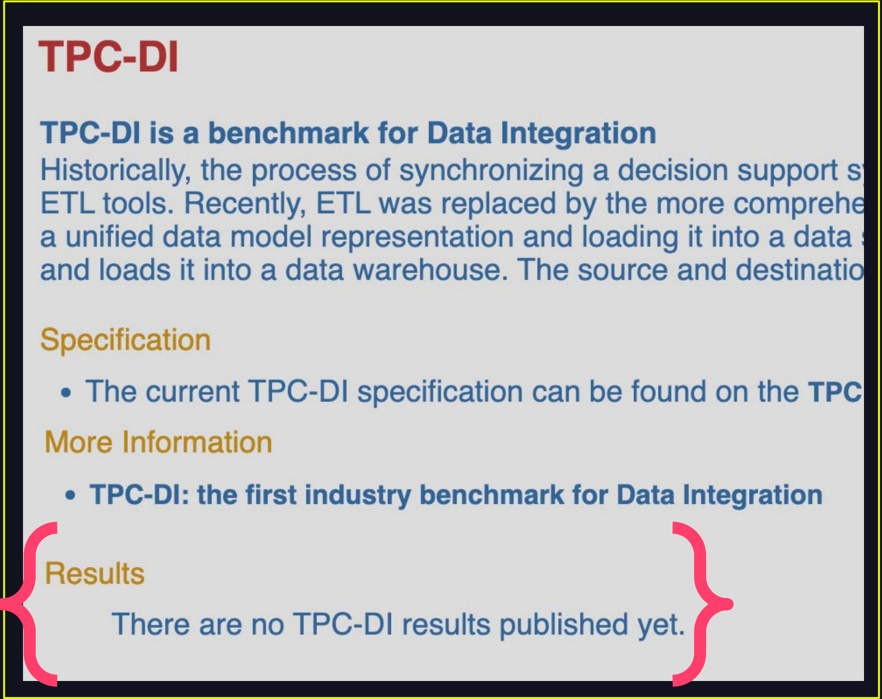


©2024 Databricks Inc. — All rights reserved

TPC-DI: Data Integration

The Ingestion and ETL One

- ZERO official submissions
  - Was Databricks first to code it?
- I originally presented **completed** benchmark at DAIS 2022
  - Not **submitted** (not for lack of trying)
- Extremely short TL;DR:
  - ZERO code given
  - Ingest: TXT, CSV, XML
  - Transform: based upon 100+ pages of business rules
  - Load: all 3 medallion layers


TPC-DI

TPC-DI is a benchmark for Data Integration

Historically, the process of synchronizing a decision support s
ETL tools. Recently, ETL was replaced by the more comprehe
a unified data model representation and loading it into a data
and loads it into a data warehouse. The source and destinatio

Specification

- The current TPC-DI specification can be found on the **TPC**

More Information

- **TPC-DI: the first industry benchmark for Data Integration**

Results

There are no TPC-DI results published yet.

©2024 Databricks Inc. — All rights reserved

Is the TPC-DI Valuable?

A Frustrating Benchmark that hides some real valuable insight

The best official ETL benchmark available

The worst official ETL benchmark available

©2024 Databricks Inc. — All rights reserved

Is the TPC-DI Valuable?

A Frustrating Benchmark that hides some real valuable insight

The best official ETL benchmark available

- Robust even though built for legacy DWs
- Business rules make for realistic test
  - Though it suffers DQ issues with data generator at higher scale factors
- Flexibility in how rules are coded
  - Allows practitioners to optimize to their platform

The worst official ETL benchmark available

- Is anyone aware of another “official” ETL benchmark?
- No ”official” submittals
- Scoring metrics are confusing and do not even allow for cloud platforms
- No provided code means it is extremely frustrating to attempt this benchmark
  - Made worse by long, confusing business rules

©2024 Databricks Inc. — All rights reserved

Excuse Me While I Digress…

I will speak longer on TPC-DI than originally planned

Why?

A) Because me

2) Because everybody

D) Because Joe Abandoned Us

©2024 Databricks Inc. — All rights reserved

2 Years of TPC-DI on Databricks

From Initial Implementation to Scorching Performance Today

This is a slide from the DAIS 2022 Session in which we announced the TPC-DI had been finally implemented

Photon price per billion rows:

$1.51

Traditional Notebook Workflow Results

Performance Dashboard

- The TPC-DI has a rather confusing benchmark algorithm
- Simplified: TCO approach based on cost per row processed

These were the best performing combinations with On-Demand Pricing:

Run Time (minutes)	Worker	Total Costs	Price per Billion Rows	Photon	Graviton
36.4	m6gd.8xlarge	$23.28	$1.44	No	Yes
24.0	m6gd.4xlarge	$24.47	$1.51	Yes	Yes

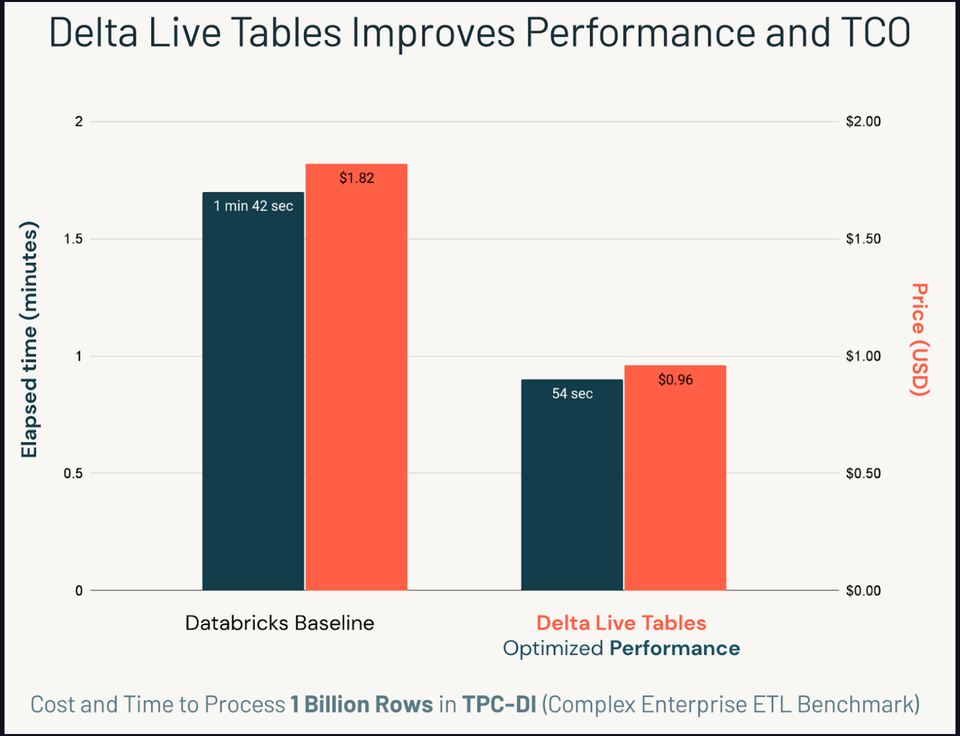
- **SPOT** instances drops this price to as low as **85 CENTS!**

©2024 Databricks Inc. — All rights reserved

2 Years of TPC-DI on Databricks

From Initial Implementation to Scorching Performance Today

In April 2023, we published a blog, How We Performed ETL on One Billion Records For Under a Dollar, to tout the power and TCO of Delta Live Tables on this benchmark.



Photon price per billion rows:

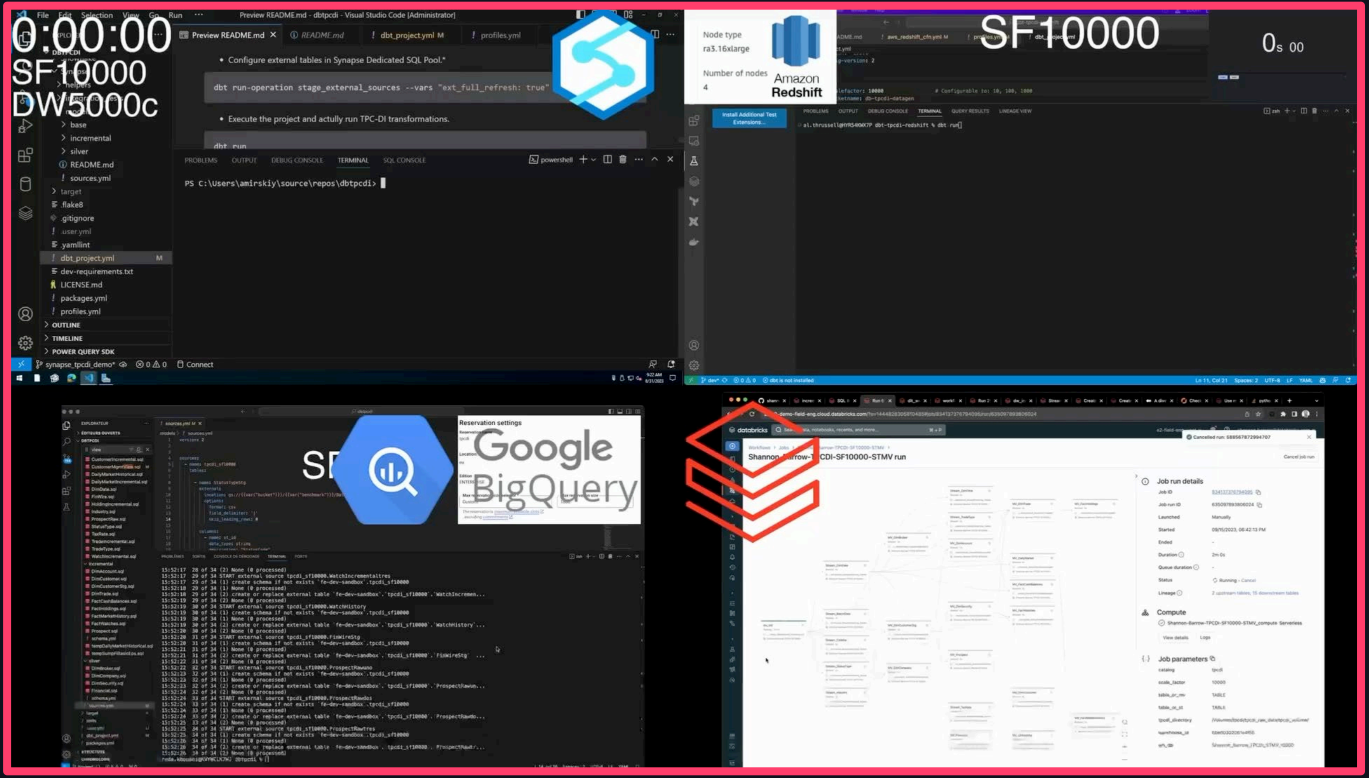
$0.96

©2024 Databricks Inc. — All rights reserved

2 Years of TPC-DI on Databricks

From Initial Implementation to Scorching Performance Today

This video compiled in September 2023 compares a dbt implementation against CDW competitors

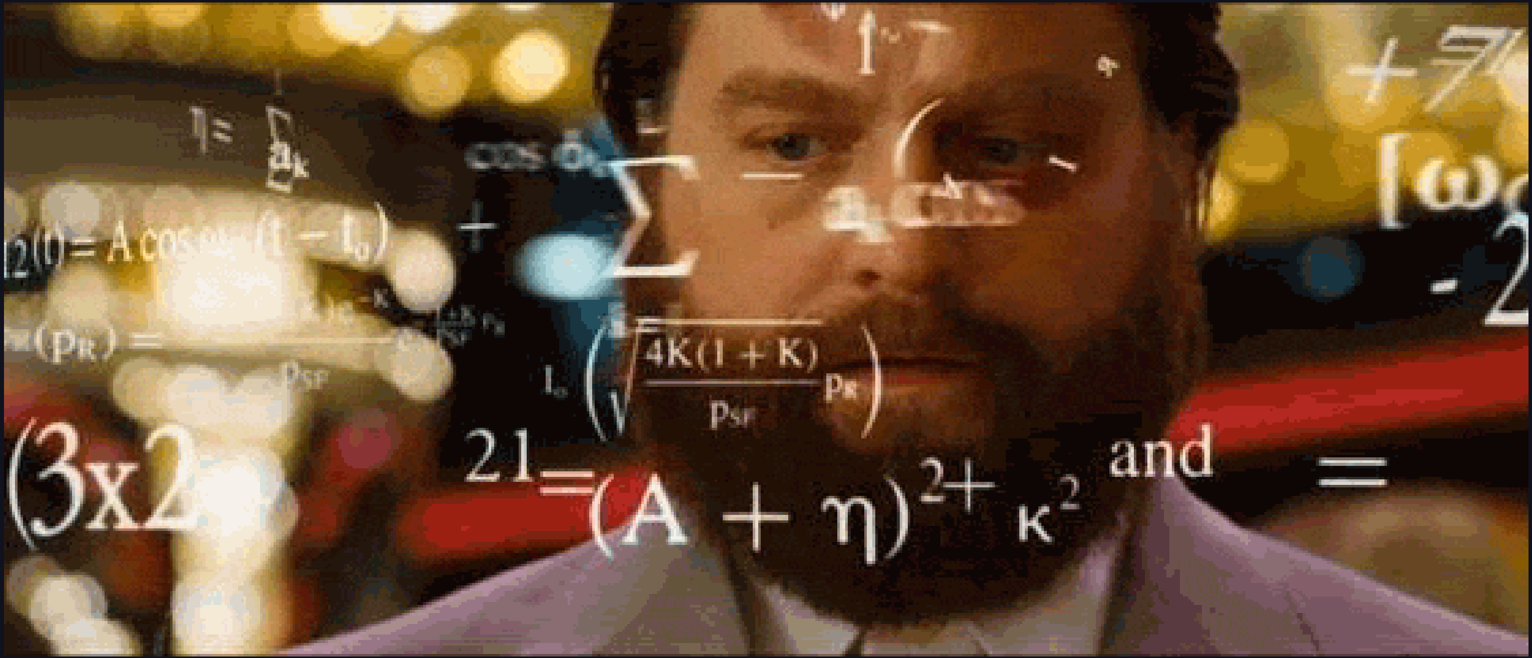


Photon price per billion rows: $0.73

©2024 Databricks Inc. — All rights reserved

A Prominent CDW Was missing…

The Truth is Out There…



©2024 Databricks Inc. — All rights reserved

A Prominent CDW Was missing…

Some can handle large file sizes, others can't

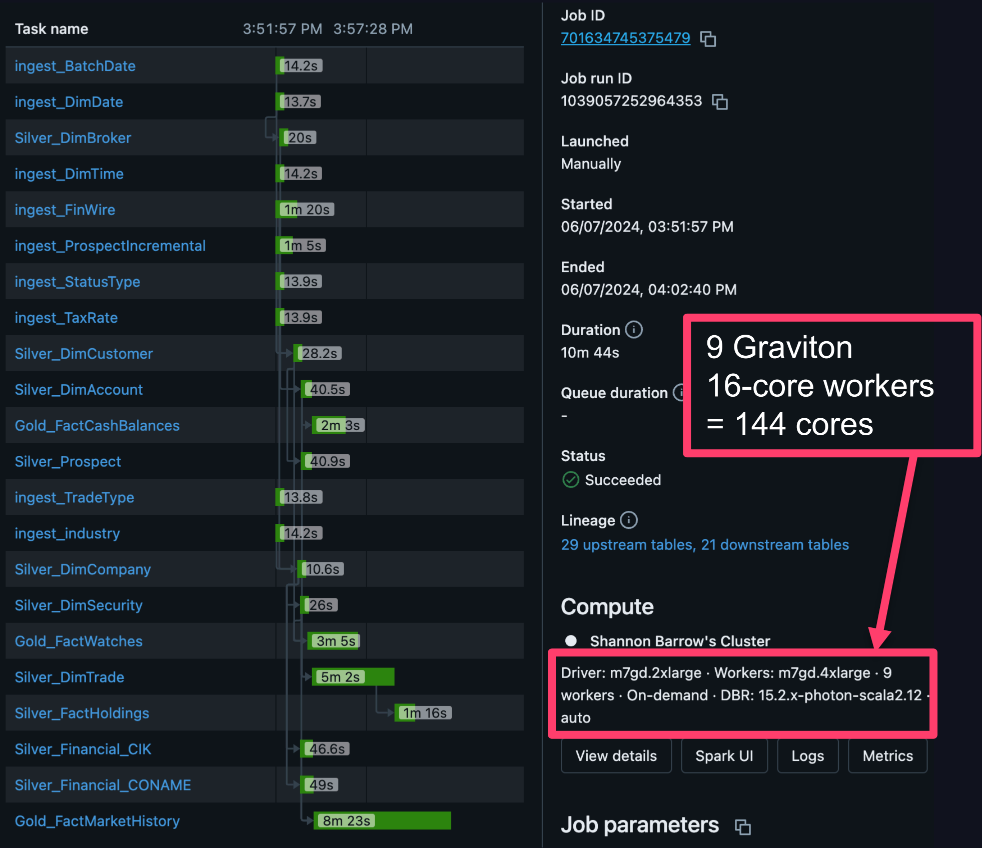
- We tried benchmarking the other CDW but found it wholly intractable at larger scale factors since it is the only one that is unable to split raw files natively
- We weren't the only ones to notice

©2024 Databricks Inc. — All rights reserved

2 Years of TPC-DI on Databricks

From Initial Implementation to Scorching Performance Today

- Last month (May 2024) we expanded the benchmark to test non-DWs– which required moving from dbt
- Since AWS has been improving EMR over the last few years this became the obvious first choice for non-dbt tests

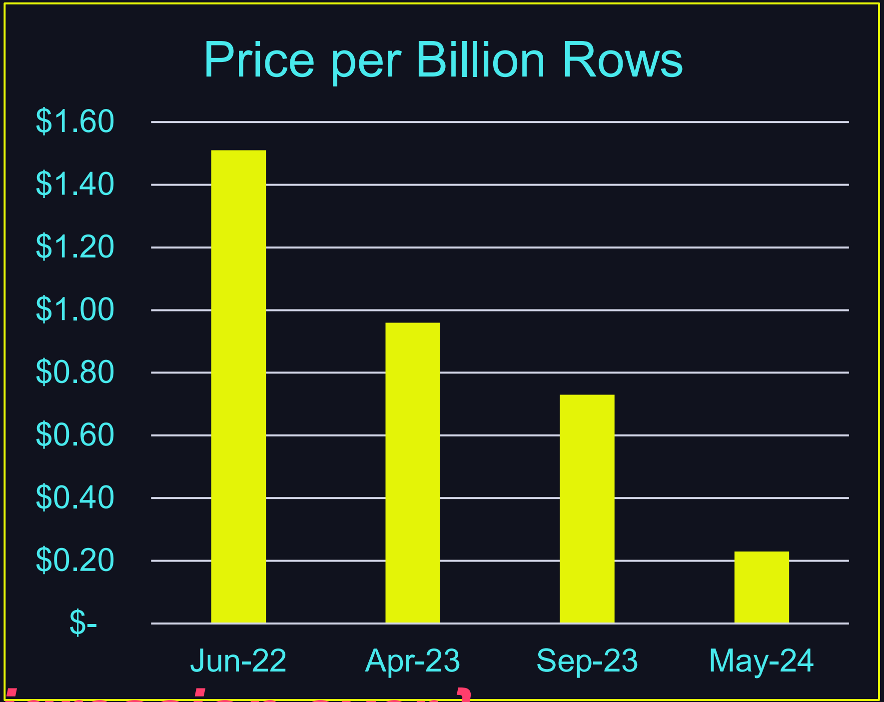


©2024 Databricks I

2 Years of TPC-DI on Databricks

From Initial Implementation to Scorching Performance Today

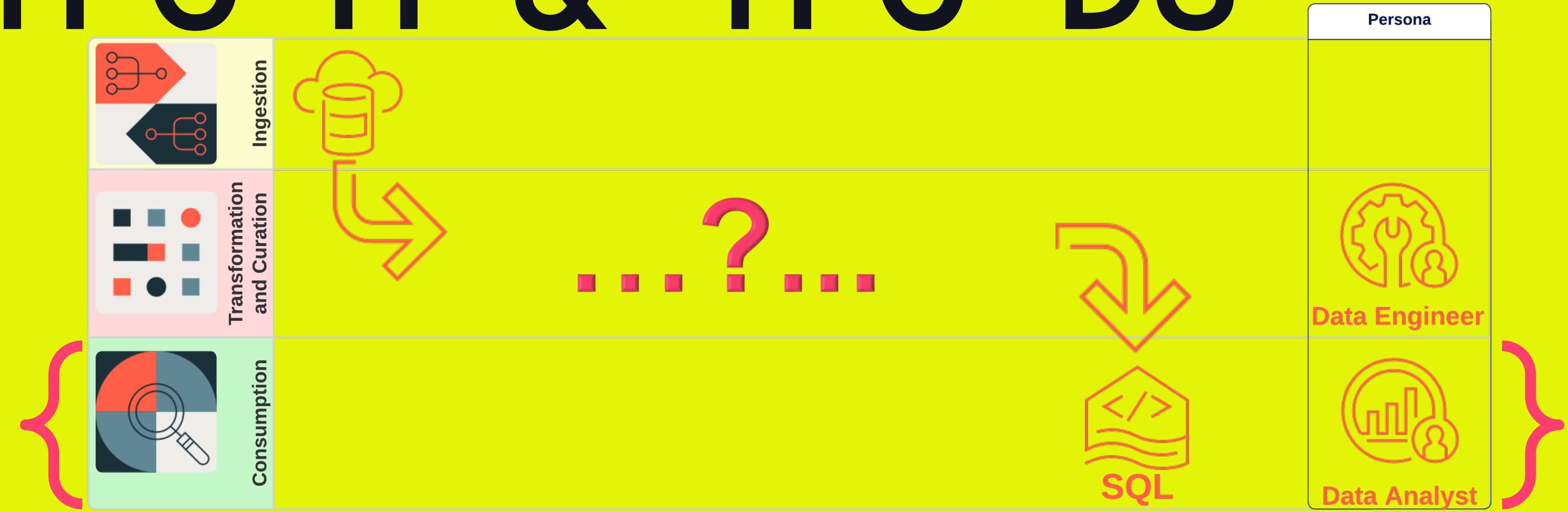
- 2.2x faster on ¼ the cores as 2 years ago!
  - 24 minutes down to 10.75 minutes
  - 576 cores down to 144 cores
- Improvements from:
  - PHOTON shifting into overdrive
  - Gradual code and orchestration improvements
    - No code is provided - optimize code to match the platform
  - Newer generation VMs
  - Other platform enhancements
- 1 year ago: "A billion rows for under a dollar"
- Today: as low as 20¢ on spot (27¢ on-



{ Databricks Digression over }

©2024 Databricks Inc. — All rights reserved

SQL : TPC-H & TPC-DS



©2024 Databricks Inc. — All rights reserved

So TPC-H… this still a thing?

The “OG” OLAP benchmark

- Released in April 1999 to “fix” issues with TPC-D
- However, the following year TPC moved to develop a new decision support benchmark to better reflect modern OLAP implementations
- In January 2012, the TPC-DS was released – begging the question why the TPC-H is still used by organizations

©2024 Databricks Inc. — All rights reserved

How is TPC-H constructed?

With the “Easy Button”

- Inmon-style DW model with 1 very large table (lineitem) and 7 smaller tables
- All tables contain DATE and STRING columns that are joined using numeric business keys
- Low Query Complexity: 22 queries with only 1 LEFT JOIN, simple aggregates and subqueries, no nested CTEs, and predicates applied directly to large tables
- Easier Tuning Complexity: often “super-tuned” and each query gets a perfectly covering index
- Does not require a sophisticated optimizer: needs join reordering and predicate pushdown

©2024 Databricks Inc. — All rights reserved

TPC-DS: The Popular Kid in School

The one we all fight over

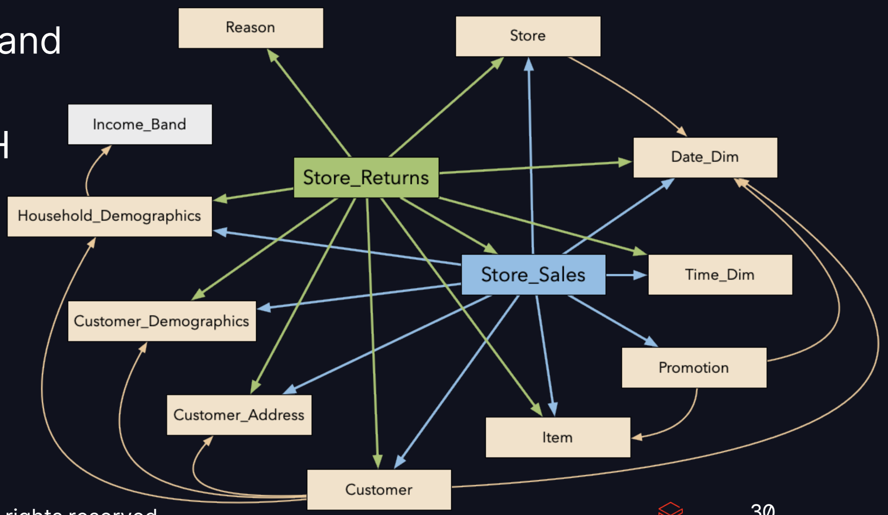
- First published in 2012 to counter aging TPC-H's limitations and tackle current OLAP trends
- A benchmark that is often misused/misrepresented by skipping one or more of the 3 components designed to ensure operational considerations aren't forgotten for the sake of over-indexing on this benchmark's SQL queries
    - Load Test, Throughput Test, and Data Maintenance Test
    - A multi-cloud data warehouse platform even publishes a highly-tuned, preloaded TPC-DS dataset in all deployed warehouses for users to consume
- Still valuable in a vacuum when results can be trusted and validated

©2024 Databricks Inc. — All rights reserved

How is TPC-DS constructed

Retailer selling goods via 3 different distribution channels: Store, Catalog, Internet

- Based on Kimball dimensional modeling (The Data Warehouse Toolkit)
  - Replaced TPC-H 3NF approach with hybrid approach between 3NF and star schema, or a “multiple snowflake schema”
- Significantly more complicated than TPC-H
  - Heavy on advanced SQL features/functions and lopsided filters
  - 99 queries compared to meager 22 in TPC-H
- 4 query classes:
  - pure reporting queries
  - pure ad-hoc queries
  - iterative OLAP queries
  - extraction or data mining queries



©2024 Databricks Inc. — All rights reserved

Feature Comparison: TPC-H & TPC-DS

Easier to consume cheat sheet

Feature	TPC-H	TPC-DS
Data Model	Simpler schema, uses Inmon style DW model	Complex schema, Kimball style dimensional model.
Schema	1 very large table (lineitem) 7 smaller tables	6 fact tables (3 _sales, 3 _returns) 18 dimension tables
Data Types	All tables contain DATE and STRING columns. Tables are joined using numeric business keys	Fact tables use only INTEGER and NUMERIC columns. Only dimension tables use TIMESTAMP and STRING Tables are joined using numeric surrogate keys
Query Complexity	22 queries: Low complexity • Only uses 1 LEFT JOIN • Only uses simple aggregates • Subqueries are simple, no nested CTEs • Predicates applied directly to large tables	99 queries: High complexity • 9 queries use LEFT JOIN, 3 use a cross join • Complex aggregates, 15 queries use window functions • Complex nested CTEs used in most queries • Predicates applied only to dimension tables
Tuning Complexity	Easier for vendors to tune • Often "super-tuned" with perfect indexes • Does not require a sophisticated optimizer:	Harder for vendors to tune • Optimizing a specific query can make others slower • Requires a sophisticated query optimizer: must be

Is the TPC-H Valuable?

Best for: Ad -Hoc Manual Benchmarking

Easy Peasy Man

- Simpler schema, easier to understand and manage.
- Fewer benchmark queries and they are easy to understand
- Tables contain DATE and STRING columns that are used as predicates

Too Simple and Easy to Shortcut. Been replaced!

- Less realistic, not representative of more complex modern data warehousing needs.
- Simple queries do no reflect hyper-complex real world queries from tools like Tableau and dbt
- Simple schema does not reflect best practices such as SCD type 2.

©2024 Databricks Inc. — All rights reserved

Is the TPC-DS Valuable?

Best for: Vendor Supported POC Evaluations

Most modern of the Common SQL Benchmarks

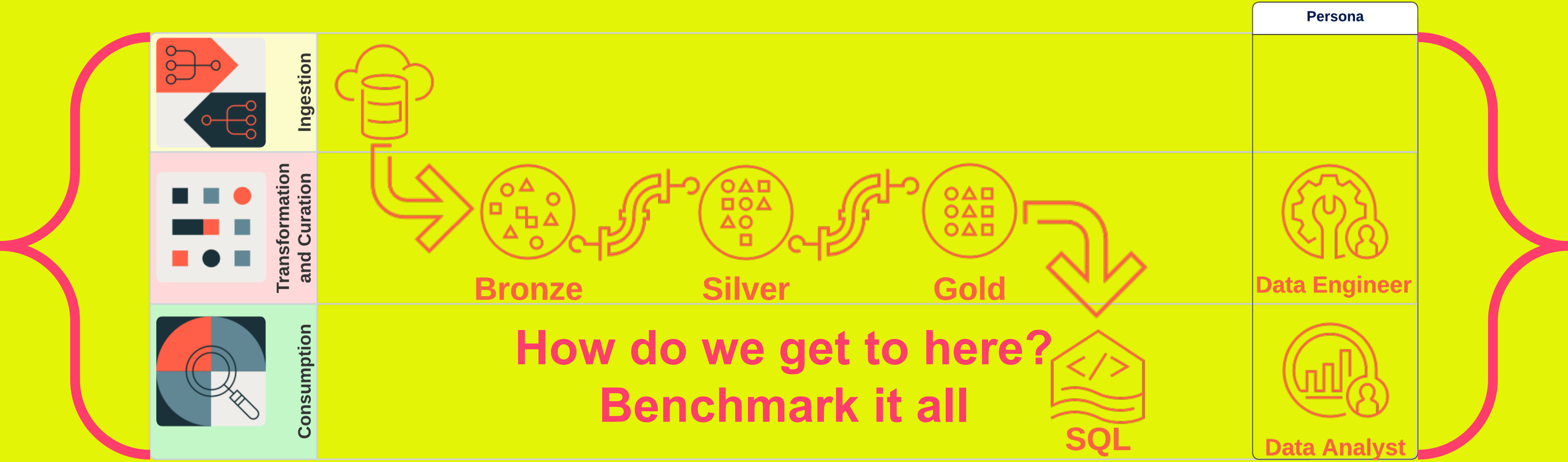
- Complex, realistic schema that better mimics enterprise data warehouses.
- Covers a broad spectrum of query types, SQL operators, and complex joins.
- Requires a sophisticated optimizer, testing more capabilities.

Complexity & Popularity Result in Missed Stages

- Higher complexity in setup and longer time to implement and tune.
- Many complex queries can make the results hard to evaluate.
- Can require significant resources to fully utilize and understand performance implications.

©2024 Databricks Inc. — All rights reserved

...TPC - LH?



©2024 Databricks Inc. — All rights reserved

The State of "Lakehouse" Benchmarks

How can the industry do better?

No Real "Official" Lakehouse Benchmark

- Each benchmark focuses on a portion of end-to-end Lakehouse platform
  - Favors bias and "shortcuts" to improve performance
- Reveals flaws in keeping benchmarks current
  - Example: TPC-DI has no way to calculate its benchmarked metric for cloud platforms

LHBench: Berkeley white paper implementing a Lakehouse benchmark on EMR

- Composed of 4 tests:
  - TPC-DS
  - TPC-DS Refresh
  - Merge Microbenchmark
  - Large File Count
- Pattern appears sound and it's a great start - but not "official"
  - Community may want to move beyond TPC-DS as the core of the benchmark.

©2024 Databricks Inc. — All rights reserved

Thought Experiment: Cluster TPC-DI

How costly is optimizing and what can be learned to "balance" a Lakehouse benchmark?

- Modified for OPTIMIZE on all fact tables
- Adjusting for cluster start times, it takes 44% longer
  - $6.42 On-Demand nodes ($4.92 on spot)
- Despite tuning the tables this price is still less than:
  - Half the price of EMR
  - 1/3 the price of Big Query
  - Over 15x cheaper than any other CDW we

©2024 Databricks Inc. — All rights reserved

Thought Experiment: Cluster TPC-DI

How costly is optimizing and what can be learned to “balance” a Lakehouse benchmark?

- Modified for OPTIMIZE on all fact tables
- Adjusting for cluster start times, it takes 44% longer

Is this worth it?

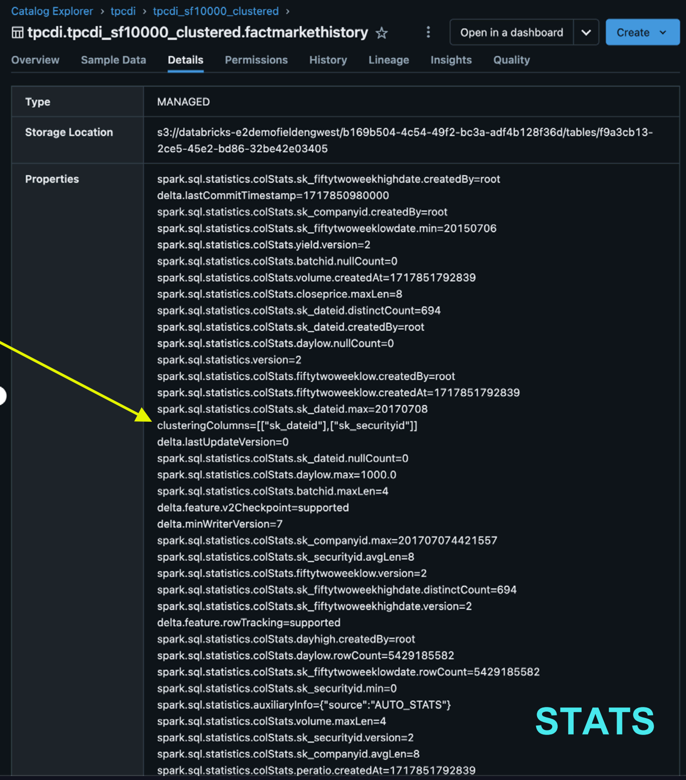
The answer is always the same: if consumption savings are greater than than the costs to optimize the data

DATA+AI SUMMIT

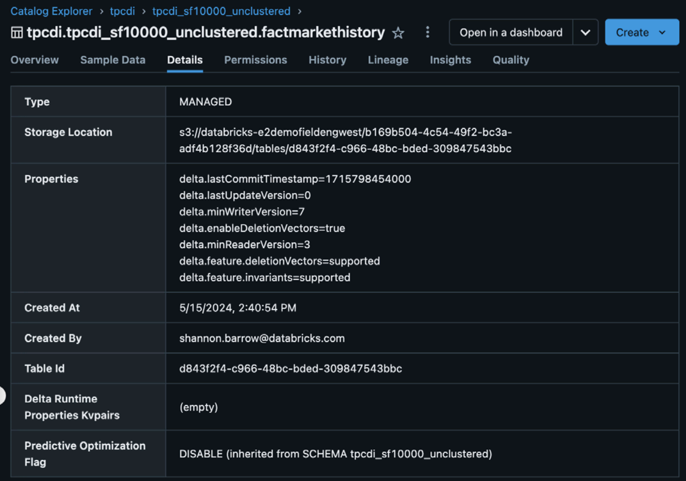
Thought Experiment: Cluster TPC-DI

How costly is optimizing and what can be learned to "balance" a Lakehouse benchmark?

- Autostats feature: Stats on Write!
- Leverages Liquid Clustering



STATS



NO STATS

Thought Experiment: Cluster TPC-DI

How costly is optimizing and what can be learned to “balance” a Lakehouse benchmark?

- Autostats feature: Stats on Write!
- Leverages Liquid Clustering
- Point lookup ad-hoc type query

```sql
set use_cached_result = false;
select * from factmarkethistory
where sk_securityid = 42949723972 and sk_dateid = 20170401
```

Raw results

	sk_securityid	sk_companyid	sk_dateid
1	42949723972	19700316303537	20170401

Optimized

Wall-clock duration		
Total wall-clock duration	350 ms	
Scheduling	35 ms	10%
Waiting for compute	0 ms	
Waiting in queue	35 ms	
Running	315 ms	90%
Optimizing query & pruning files	155 ms	
Executing	160 ms	
Start time	2024-06-09 21:44:17.640 -04:00	
End time	2024-06-09 21:44:17.990 -04:00	
Result fetching by client	49 ms	
Aggregated task time		
Tasks total time	39 ms	
Tasks time in Photon	82 %	
IO		
Rows returned	1	
Rows read	1	
Bytes read	94.19 MB	
Bytes pruned	170.86 GB	
Bytes read from cache	100 %	
Bytes written	0 bytes	
Files & partitions		
Files read	2	
Files pruned	2,731	
Partitions read	0	

.35s vs 10.2s

30x improvement

Not Optimized

Wall-clock duration		
Total wall-clock duration	10 s 235 ms	
Scheduling	43 ms	0%
Waiting for compute	0 ms	
Waiting in queue	43 ms	
Running	10 s 192 ms	100%
Optimizing query & pruning files	139 ms	
Executing	10 s 53 ms	
Start time	2024-06-09 21:43:35.606 -04:00	
End time	2024-06-09 21:43:45.841 -04:00	
Result fetching by client	45 ms	
Aggregated task time		
Tasks total time	3.59 m	
Tasks time in Photon	98 %	
IO		
Rows returned	1	
Rows read	1	
Bytes read	2.57 GB	
Bytes pruned	716.18 MB	
Bytes read from cache	62 %	
Bytes written	0 bytes	
Files & partitions		
Files read	448	
Files pruned	155	
Partitions read	0	

©2

Thought Experiment: Cluster TPC-DI

How costly is optimizing and what can be learned to “balance” a Lakehouse benchmark?

- Autostats feature: Stats on Write!
- Leverages Liquid Clustering
- Point lookup ad-hoc type query
- BI-like Query using Dimensional filtering and dynamic file pruning

```sql
set use_cached_result = false;
with cte as (
    select distinct sk_securityid
    from tpcdi.tpcdi_sf10000_clustered.dimsecurity sec
    join tpcdi.tpcdi_sf10000_clustered.dimcompany comp on comp.sk_companyid = sec.sk_companyid
    join tpcdi.tpcdi_sf10000_clustered.industry ind  on ind.in_name = comp.industry
    join tpcdi.tpcdi_sf10000_clustered.statustype status_comp on status_comp.st_name = comp.status
    join tpcdi.tpcdi_sf10000_clustered.statustype status_sec on status_sec.st_name = sec.status
    where
        ind.in_id = 'WU' -- Retail (Apparel)
        and status_comp.st_id = 'ACTV' -- Only active companies
        and status_sec.st_id = 'ACTV' -- Only active securities
        and comp.iscurrent  -- is the current version of SCD Type 2 records
        and comp.stateprov = 'AL' -- alternative location by State or Province
        and comp.sprating = 'AAA' -- Only the BEST S&P rating available
        and sec.iscurrent -- is the current version of SCD Type 2 records
        and sec.exchangeid = 'NASDAQ' -- ONLY NASDAQ Exchange
        and sec.issue = 'COMMON' -- ONLT common issue stock, not preferred
)
select trade.*
from tpcdi.tpcdi_sf10000_clustered.dimtrade trade
join tpcdi.tpcdi_sf10000_clustered.dimdate dd on trade.sk_createdateid = dd.sk_dateid
join cte on cte.sk_securityid = trade.sk_securityid
where dd.fiscalyearid = 2015 -- all trades for 1 year
```

results

tradeid	sk_brokerid	sk_createdateid	sk_createtimeid	sk_closedateid
605927611	10243683	20141105	132425	201411
661288248	35482844	20150122	84217	201503
660721160	35482844	20150121	133310	201503
606908399	10243683	20141106	222758	201411
596186927	21148125	20141022	205013	201412
596418065	21148125	20141023	44220	201412
589507618	20760652	20141013	113621	201410
590191010	20760652	20141014	103835	201410
551946739	34101145	20140821	164009	201408
653527702	38231555	20150111	105646	201503
654098412	38231555	20150112	61227	201503
554992464	34101145	20140825	232223	201408
631338585	8063934	20141211	63130	201412
517391481	27507523	20140704	31143	201407

Compute: Serverless Shared Endpoint

Query Source: New query

	Optimized	Not Optimized
Wall-clock duration		
Total wall-clock duration	2 s 522 ms	10 s 629 ms
Scheduling	36 ms (1%)	46 ms (0%)
Waiting for compute	0 ms	0 ms
Waiting in queue	36 ms	46 ms
Running	2 s 486 ms (99%)	10 s 583 ms (100%)
Optimizing query & pruning files	417 ms	309 ms
Executing	2 s 69 ms	10 s 274 ms
Start time	2024-06-09 20:43:16.652 -04:00	2024-06-09 20:51:39.158 -04:00
End time	2024-06-09 20:43:19.174 -04:00	2024-06-09 20:51:49.787 -04:00
Result fetching by client	54 ms	220 ms
Aggregated task time		
Tasks total time	13.82 s	4.40 m
Tasks time in Photon	94 %	99 %
IO		
Rows returned	94	94
Rows read	74,987,490	1,282,212,389
Bytes read	3.92 GB	50.13 GB
Bytes pruned	65.58 GB	1.57 GB
Bytes read from cache	100 %	82 %
Bytes written	0 bytes	0 bytes
Files & partitions		
Files read	93	471
Files pruned	1,022	11
Partitions read	0	3

20x task time improvement!

Thought Experiment: Cluster TPC-DI

How costly is optimizing and what can be learned to “balance” a Lakehouse benchmark?

- How do we balance the Query Load based on the TPC-DI optimize added latency?
- Back of the napkin math…
  - Conservatively assume 2x performance gains for SQL
  - Assume 5 minutes longer to optimize ETL (at 10k scale factor)– need to save at 5 minutes in queries
  - $5 = \frac{NonOptimizedTablesQueryTime}{2}$
  - Therefore if there is 2x improvement in the SQL times and we need to make up 5 minutes, we need approximately 10 minutes of non-optimized tables query time
    - 5 minutes on optimized tables

©2024 Databricks Inc. — All rights reserved

AI Benchmarking

https://harvard-edge.github.io/cs249r_book/contents/benchmarking/benchmarking.html

©2024 Databricks Inc. — All rights reserved

Why Benchmark in AI/ML?

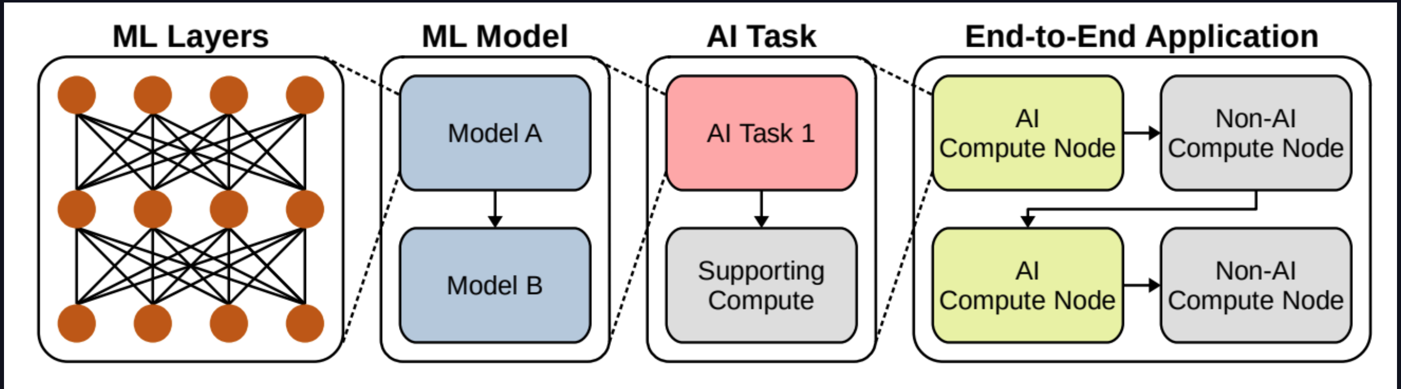
- Standardized methods allow us to quantitatively know the capabilities of different models, software, and hardware enabling fair comparisons across different solutions.
- Allow ML developers to measure the inference time, memory usage, power consumption, and other metrics that characterize a system.
- Goals and Objectives:
    - Performance assessment
    - Resource evaluation
    - Validation and verification
    - Competitive analysis
    - Credibility
    - Regulation and Standardization

©2024 Databricks Inc. — All rights reserved

What to Benchmark in AI/ML?

How does one benchmark something so subjective?

- 3 primary categories:
  - Hardware/System
  - Model
  - Data
- Granularity:
  - Micro
  - Macro
  - End to End
- Training vs Inference



https://harvard-edge.github.io/cs249r_book/contents/benchmarking/benchmarking.html

©2024 Databricks Inc. — All rights reserved

In an LLM Not Far Away...

©2024 Databricks Inc. — All rights reserved

Keeping Pace and Choosing Wisely

Benchmarks are rapidly created and deprecated, what can Mosaic's Gauntlet teach us?

- According to Stanford's 2024 AI INDEX REPORT
  - 15 benchmarks were deprecated in 2023 alone - many of which were less than 4 years old
  - 18 new benchmarks were added in 2023
- The "Mosaic Evaluation Gauntlet" (blog)
  - Evaluated 39 public benchmarks split across 6 core competencies
  - In order to prioritize the metrics that are most useful for research tasks across model scales, we tested the benchmarks using a series of

Group 1: Well-behaved metrics robust to few-shot settings

These benchmarks reliably ordered models by training scale and monotonically improved at any number of shots. We believe that these benchmarks can provide a reliable evaluation signal for models in this range.

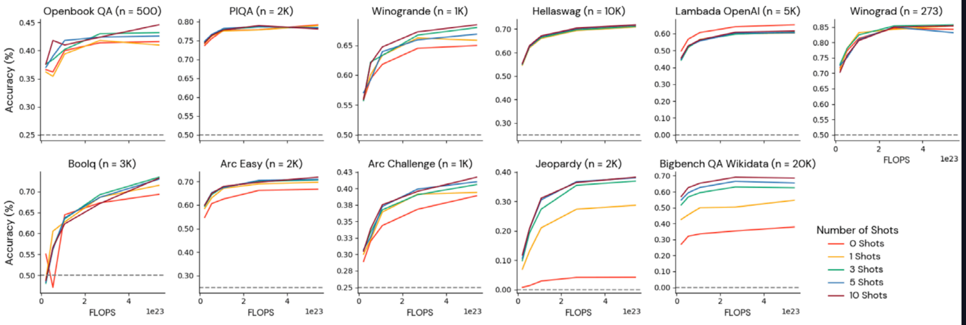


Figure 2: Monotonically improving benchmarks. These benchmarks include popular tasks like Lambada, BoolQ, Arc, and Hellaswag.

©2024 Databricks Inc. — All rights reserved

Challenges and Trends

Human evaluation is “in”

- Practitioners are growing incredibly skeptical about Academic Benchmarks
  - Habitual issues overfitting models to existing benchmarks
  - MMLU, HumanEval, Hellaswag are bona fide benchmarks but model creators' game the system for models to do well on them
- Accordingly, practitioners today tend to prefer evaluating their LLM options by human preference in the real-world - like LMSYS
  - The HAI Stanford Report even points out “human evaluation is in” (Chapter 2)
  - LMSYS: Allows users to vote on the better response based on a prompt they provide to the LLMs - the user is blind to the choice of the models they're given).

©2024 Databricks Inc. — All rights reserved

Q&A

©2024 Databricks Inc. — All rights reserved