Aurora DSQL

Unraveling the pricing!

Farid Nouri Neshat

Freelance Cloud Engineer

https://faridnsh.ninja

Aurora DSQL?

- New serverless dstirbuted database service!
- No capacity limits compared to Aurora Serverless v2
- Always Instantly Available
- No hourly fees
- Cost in us-east-1 is defined $8.00 per 1M Unit of DPUs and $0.33 per GB-month in Storage
- 100,000 DPUs and 1 GB of storage Free tier

DPUS?

> *"Aurora DSQL charges for all database activity using a billing unit called the Distributed Processing Unit (DPU). You can think of a DPU as a measure of how much work the system does to run your SQL workload. This includes compute resources used to execute query logic (e.g., joins, functions, aggregations) as well as the input/output (I/O) required to read from and write to storage. Aurora DSQL tracks this work in real time and bills only for active usage. When your cluster is idle, usage automatically scales to zero and you incur no DPU charges."*

DPUS?

- New trend of AWS pricing!
- Like "Redshift Processing Units" but at least they said it provides 16 GB of memory!
- Like "Aurora Capacity Units" but at least they said "ACU has approximately 2 GiB of memory with corresponding CPU and networking"!
- Like "Timestream Compute Units" but at least they said "TCU comprises 4vCPU and 16 GB of RAM"!
- Like Athena "Data Processing Units" but at least they said "4 vCPU and 16 GB of memory"!
- Like "ElastiCache Processing Units" but at least they said "Reads and writes require 1 ECPU for each kilobyte (KB)"
- Like "OpenSearch Compute Units" but at leas they said "6 GB of RAM"

1. 2016 AWS announced Athena, Query S3 With SQL and pay only $5 per TB scanned! Only limit was 30 minutes query time!
2. (Speculation) Natraully some people abused it to do nested loop cross joins of regpexes or in other words computionally expensive stuff that cost Athena more than what it charged!
3. 2018 AWS annouced Timestream, serverless SQL time series database. Pay $10 per TB scanned, $0.03 per GB stored per month for magnetic storage. S3 is $0.023.
4. (Speculation) Natrurally some people asbused it!
5. 2017-2024, AWS annouces more serverless data tools everything now priced in processing units instead of per GB scanned!
6. 2024 AWS has new CEO, times are tough, some cuts are made to some services! Timestream pricing is changed into processing unit.
7. There's also other distrubted serverless databases priced in "Request Units" without a lot of details(some better than others!)
8. Aurora DSQL pricing is now in DPUs without any details!

What are DPUs in a DPU?

- ComputeDPU
- ReadDPU
- WriteDPU
- MultiRegionWriteDPU

Some interesting limits

- Max query memory: 128 MiB
- Max combined size of the columns used in a primary key: 1 KiB
- Max combined size of the columns in a secondary index: 1 KiB
- Max connection duration: 60 minutes
- Maximum size of a row in a table: 2 MiB
- Maximum connections per cluster: 10,000 connections
- Maximum connection rate per cluster: 100 connections per second
- Maximum connection burst capacity: 11,000 connections
- Connection Refill Rate: 100 connections per second

Schrödinger's Invoice

Before benchmark

Aurora DSQL cost

expensive and not expensive

(superposition)

After benchmark

Aurora DSQL cost

Exact: €0,003 / query

Until you benchmark, your bill is in superposition.

Run benchmarks → collapse the cost function.

Let's do some writes!

What metrics are there?

- ComputeTime
- BytesWritten
- BytesRead
- CommitLatency
- WriteDPU
- MultiRegionWriteDPU
- ReadDPU
- ComputeDPU

Experiment flow

1. Run query that isolate write, read or compute aspects of queries.
2. Check for metrics in Cloudwatch after about 1-2 minutes.
3. Run more queries!
4. Realize it's too slow and make more clusters!
5. Decide to go multi region because only 20 clusters can be in single regions!
6. After making 80 clusters, realize the 20 cluster limit is account wide not regional limit but you got 80 clusters because eventual consistency!

DPU vs Bytes/Time metrics?

- WriteDPU = 51.20 × MB_Written + ~0.3
- ReadDPU = 1.92 × MB_Read + ~0.024
- ComputeDPU = ComputeTime in seconds!

How do simple inserts do?

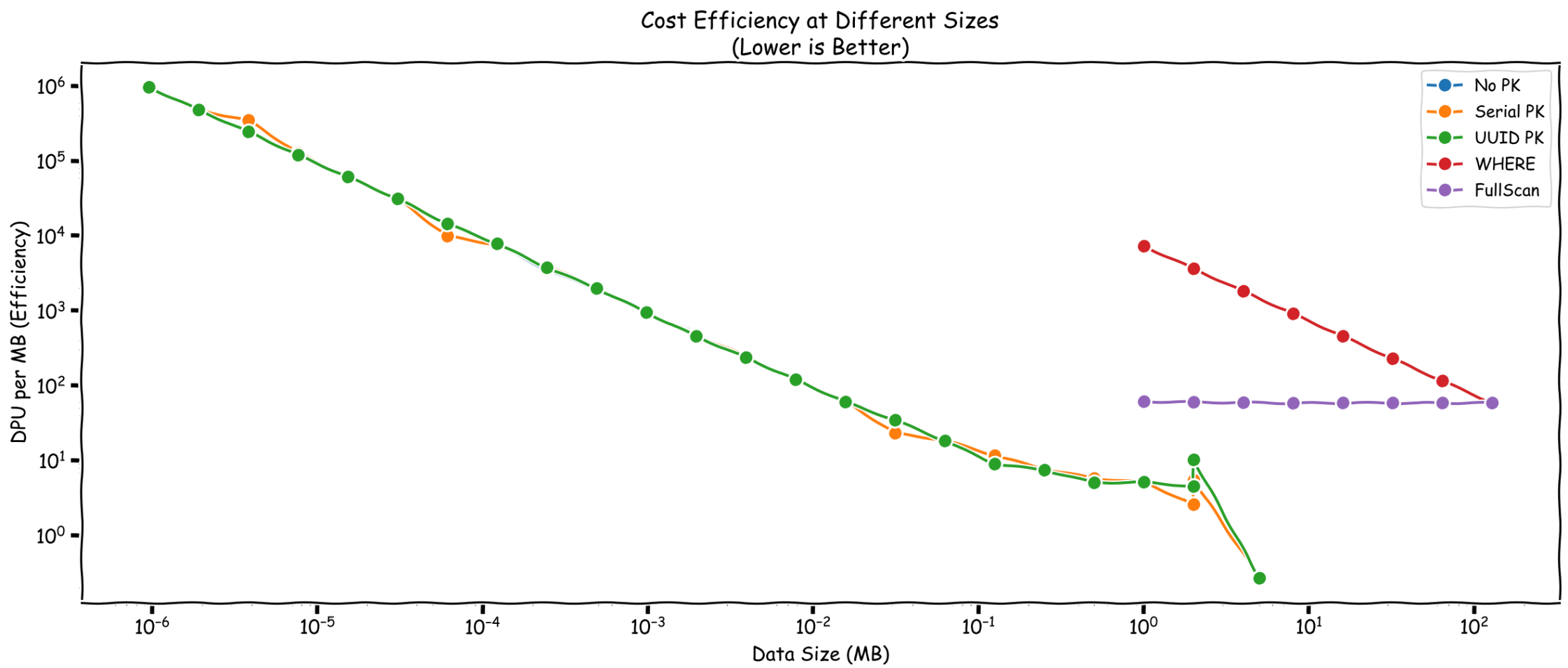
- WriteDPU = 51.20 × Write in MB + 0.7
- ReadDPU = 2.28 × Write in MB + 0.6
- ComputeDPU = 0.42 × Write in MB + 2.2o

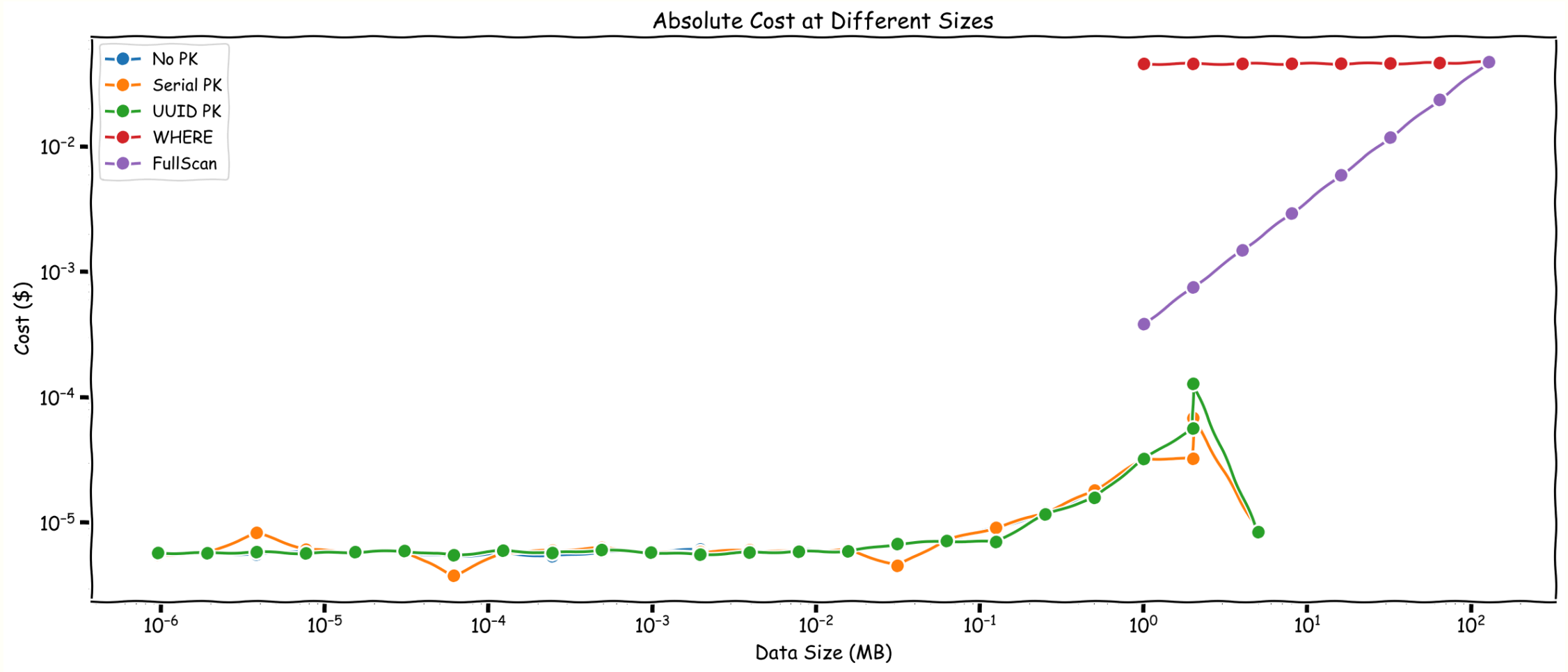
Reads?

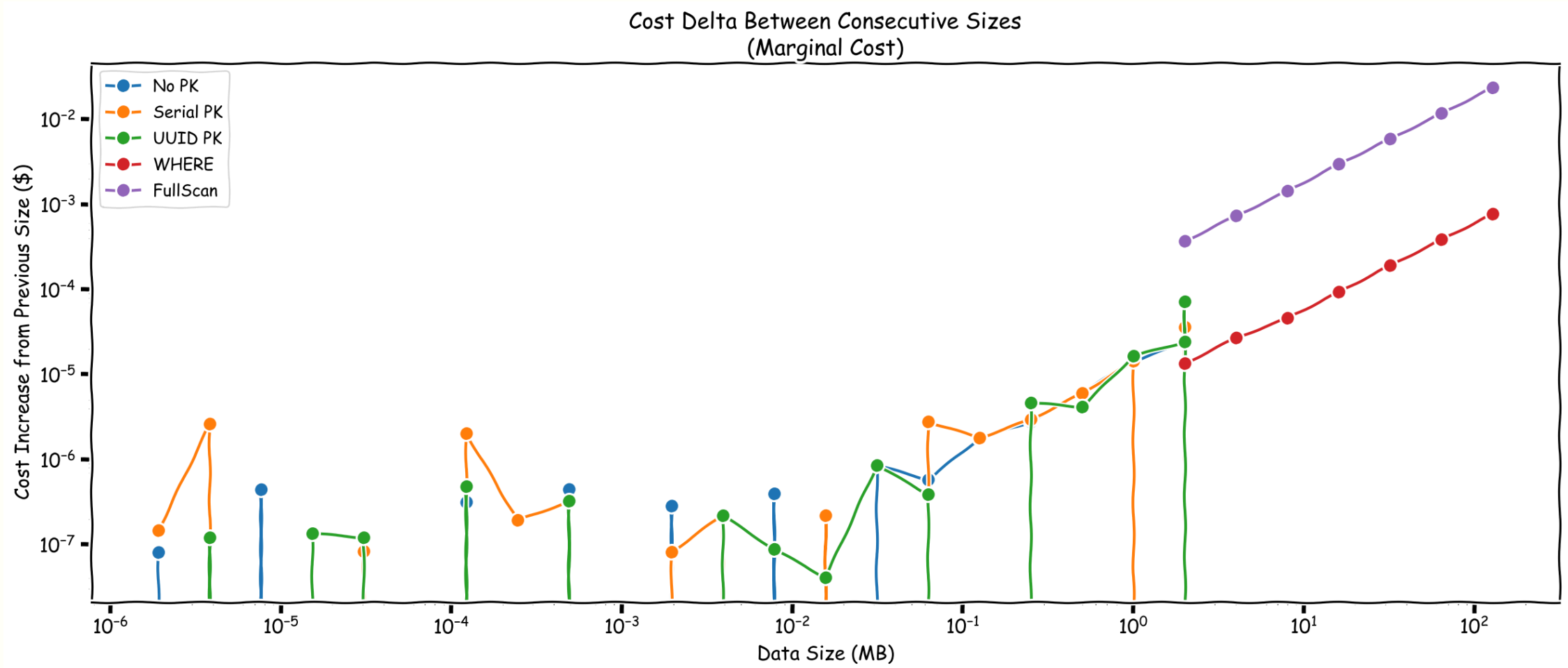
Scan Type	BytesRead	Read/Data Ratio	BytesWritten	ReadDPU	WriteDPU	ComputeDPU
No PK	2.03 MB	2.03×	0 MB	3.91	0.00	1.26
Serial PK	2.03 MB	2.03×	0 MB	3.92	0.00	1.28
UUID PK	2.03 MB	2.03×	0 MB	3.92	0.00	1.26
WHERE	385.03 MB	385×	127.99 MB	739.41	6,554.75	6.00

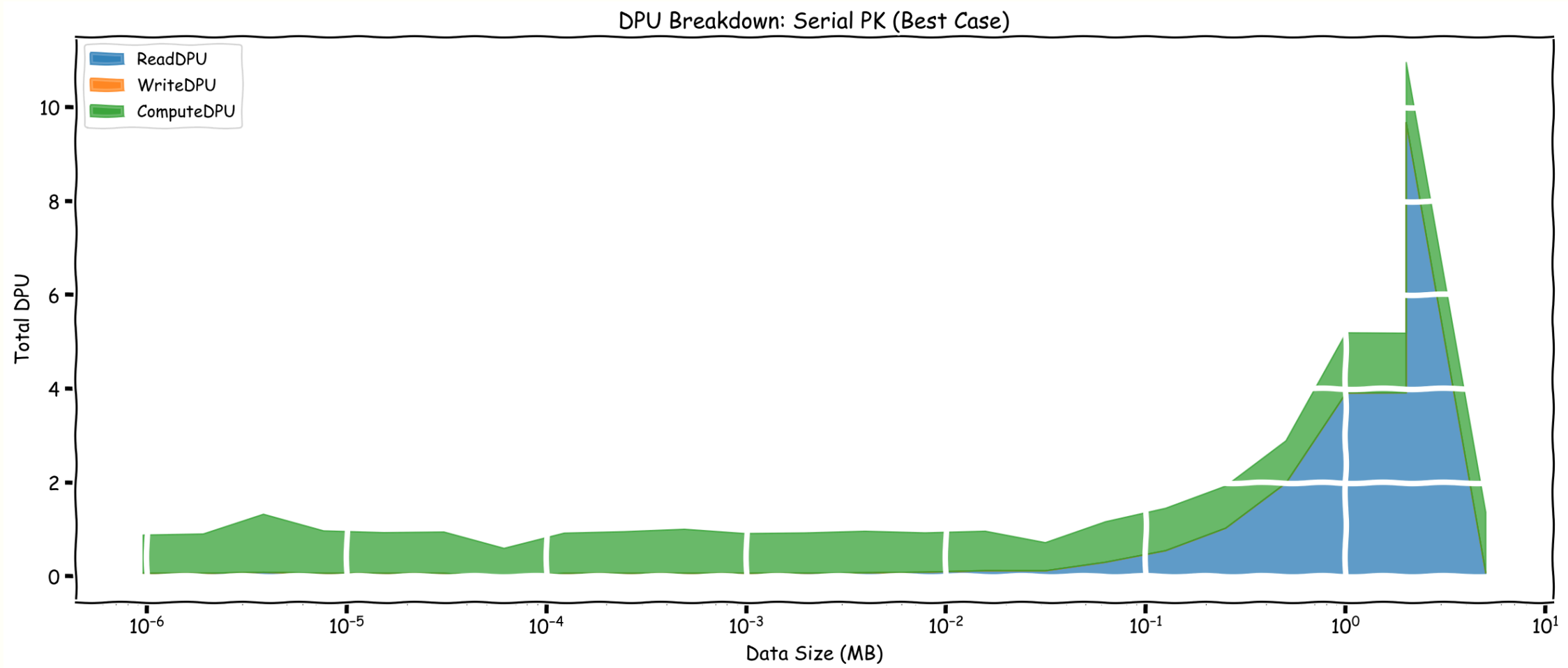
At 10 MB Data Size:

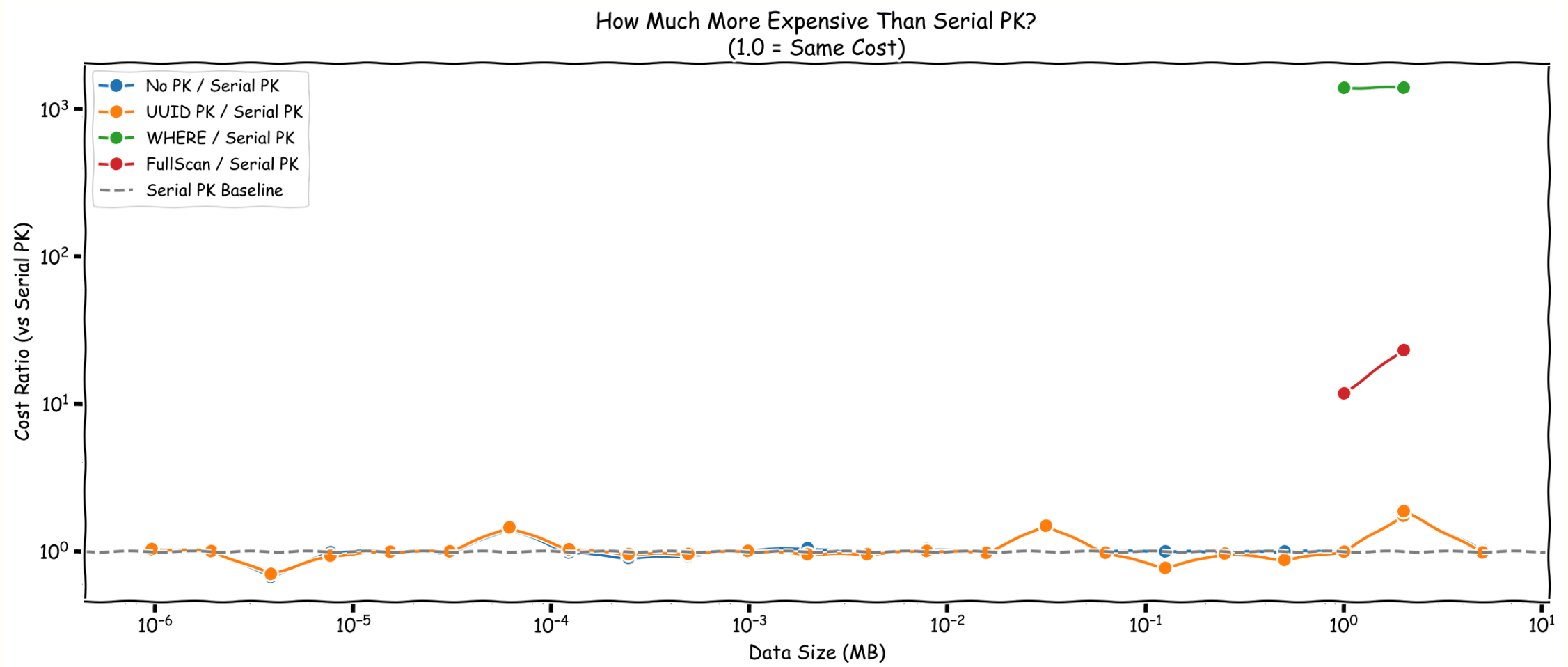
Scan Type	BytesRead	Read/Data Ratio	BytesWritten	ReadDPU	WriteDPU	ComputeDPU
No PK	0.03 MB	0.01×	0 MB	0.08	0.00	1.32
Serial PK	0.03 MB	0.01×	0 MB	0.08	0.00	1.30
UUID PK	0.03 MB	0.01×	0 MB	0.08	0.00	1.27
WHERE	392.03 MB	49×	127.99 MB	752.84	6,554.70	6.46

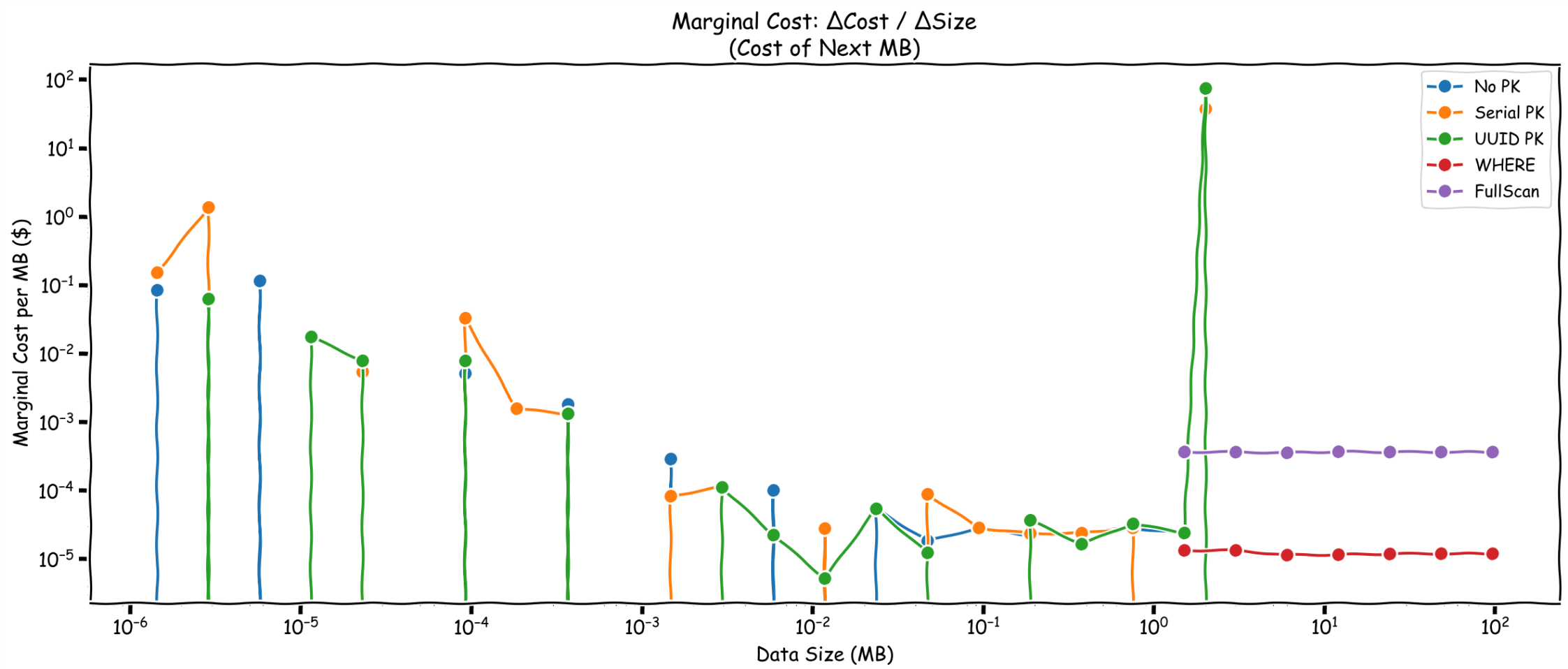

Cost Efficiency at Different Sizes
(Lower is Better)
DPU per MB (Efficiency)
Data Size (MB)
No PK
Serial PK
UUID PK
WHERE
FullScan


Absolute Cost at Different Sizes
No PK
Serial PK
UUID PK
WHERE
FullScan
Cost ($)
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Data Size (MB)


Cost Delta Between Consecutive Sizes
(Marginal Cost)
No PK
Serial PK
UUID PK
WHERE
FullScan
Cost Increase from Previous Size ($)
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Data Size (MB)


DPU Breakdown: Serial PK (Best Case)
ReadDPU
WriteDPU
ComputeDPU
Total DPU
0
2
4
6
8
10
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
Data Size (MB)


How Much More Expensive Than Serial PK?
(1.0 = Same Cost)
No PK / Serial PK
UUID PK / Serial PK
WHERE / Serial PK
FullScan / Serial PK
Serial PK Baseline
Cost Ratio (vs Serial PK)
10^3
10^2
10^1
10^0
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^0
10^1
Data Size (MB)


Marginal Cost: ΔCost / ΔSize
(Cost of Next MB)
Marginal Cost per MB ($)
Data Size (MB)
No PK
Serial PK
UUID PK
WHERE
FullScan

Compute or Complex computations?

- 1 second compute = 1 DPU
- This comes to $0.0288 per hour, somewhere between t4g.small(2GB) and t4.micro(1GB) instances!
- With only 128 MB of memory available to queries.
- In Aurora Capacity Units(ACU) for Aurora Serverless, 256 MB would cost $0.02.

Operation Type	Avg Throughput	Avg ComputeDPU
Cross Join	2.6M rows/sec	7.31 DPU
Sort	664K rows/sec	5.26 DPU
Regex (100 rows)	200 rows/sec	5.0 DPU

I looked for a real world application that I can test it with!

Aurora DSQL Cannot Power Gitea Without Major Refactoring

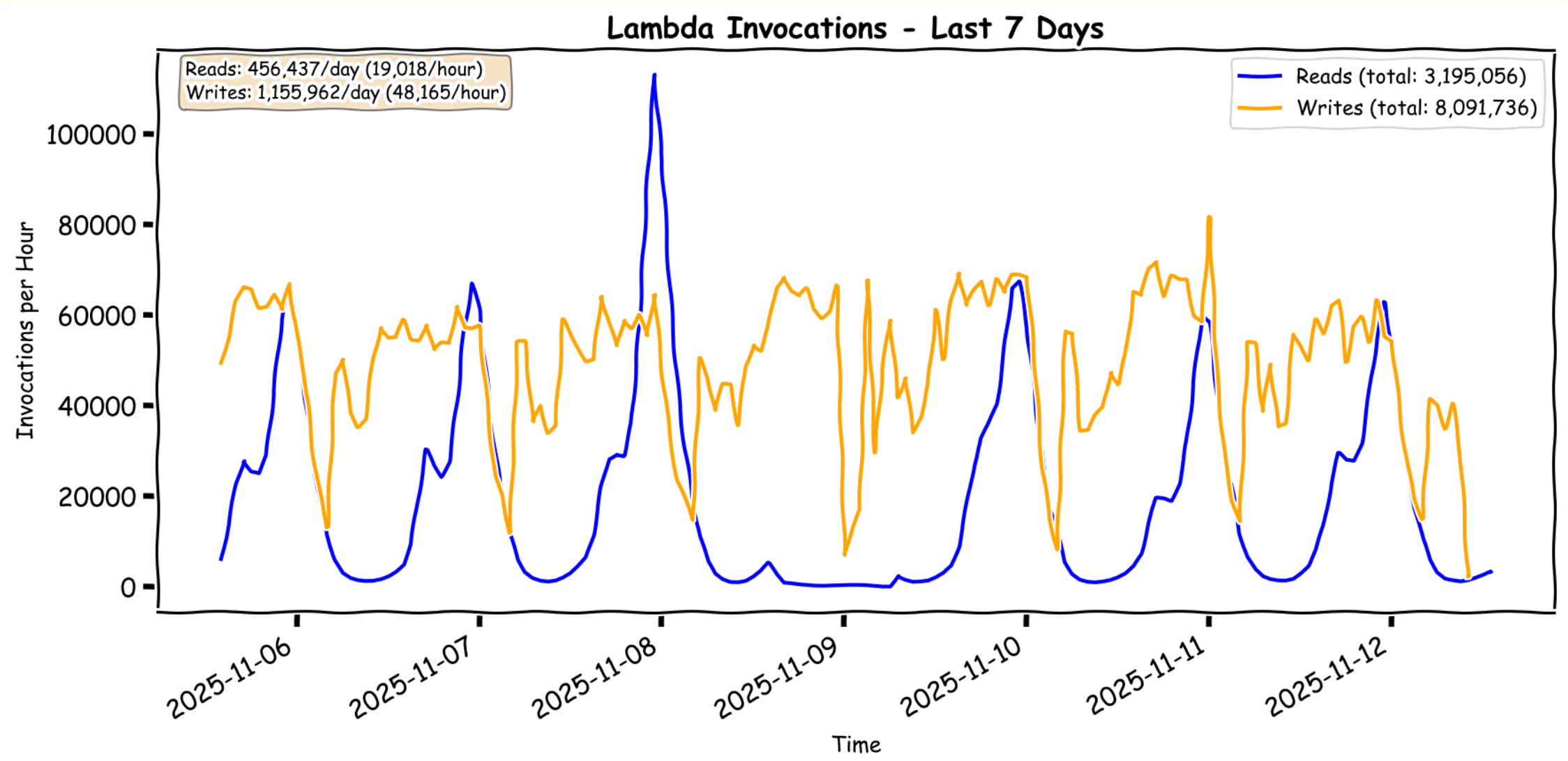
Critical compatibility barriers block Metabase deployment

Why Aurora DSQL fails as a Directus backend

Aurora DSQL Cannot Power Strapi CMS

Amazon Aurora DSQL is fundamentally incompatible with Strapi CMS despite being

My client's application!



My client's application!

- $46 for queries at $2.3 per million queries
- $61.6 for writes at $1.1 per million queries
- Current costs are $100 monthly for vanila RDS, ignoring storage costs!

Comparison with DyanmoDB?

- 1KB write in DSQL: $28.4 per million writes
- 1KB write in dyanmodb: $1.25 per million transactional writes!
- 4KB READ in DSQL: $6 per million reads
- 4KB READ in dyanmodb: $0.25 per million transactional reads!