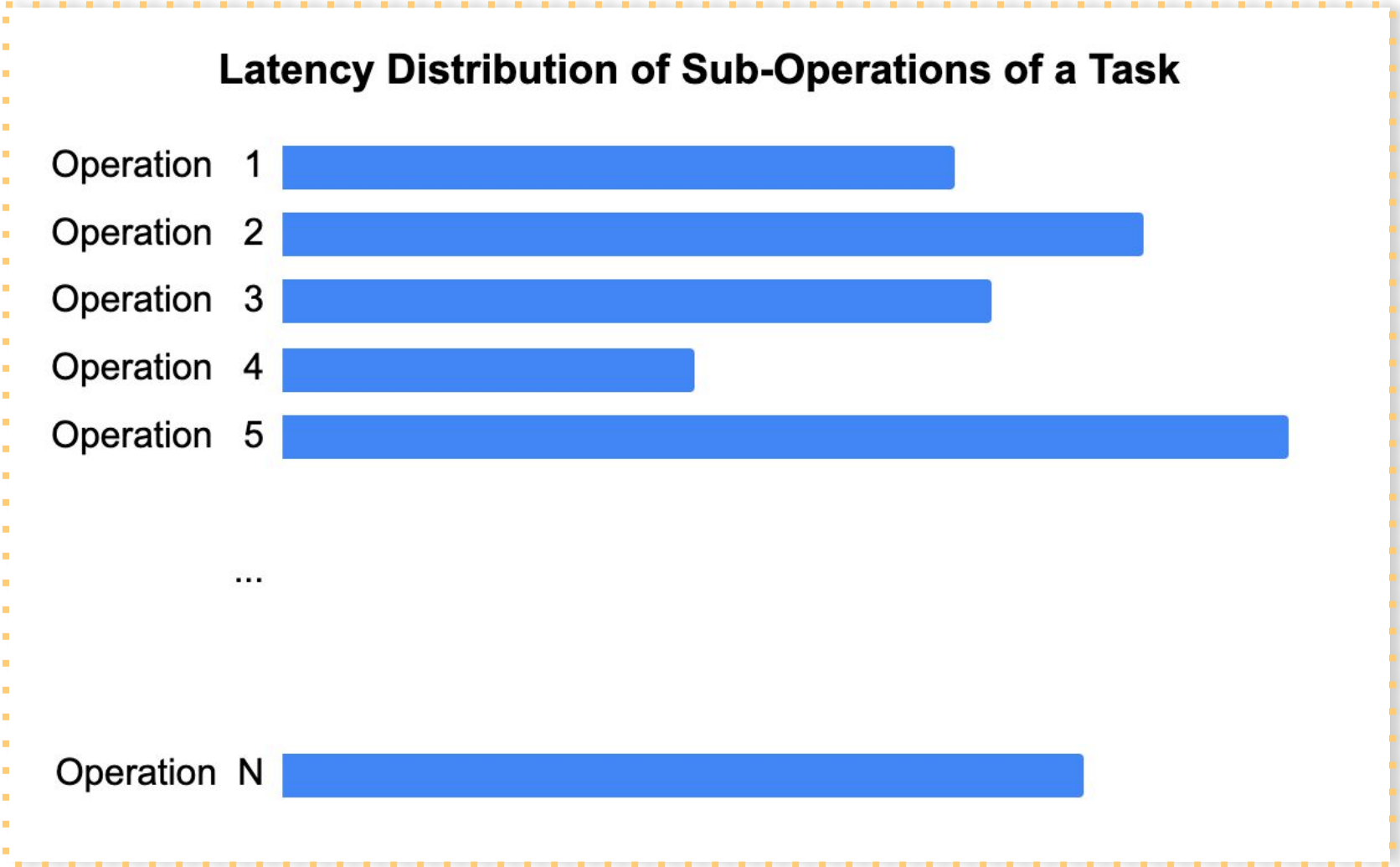
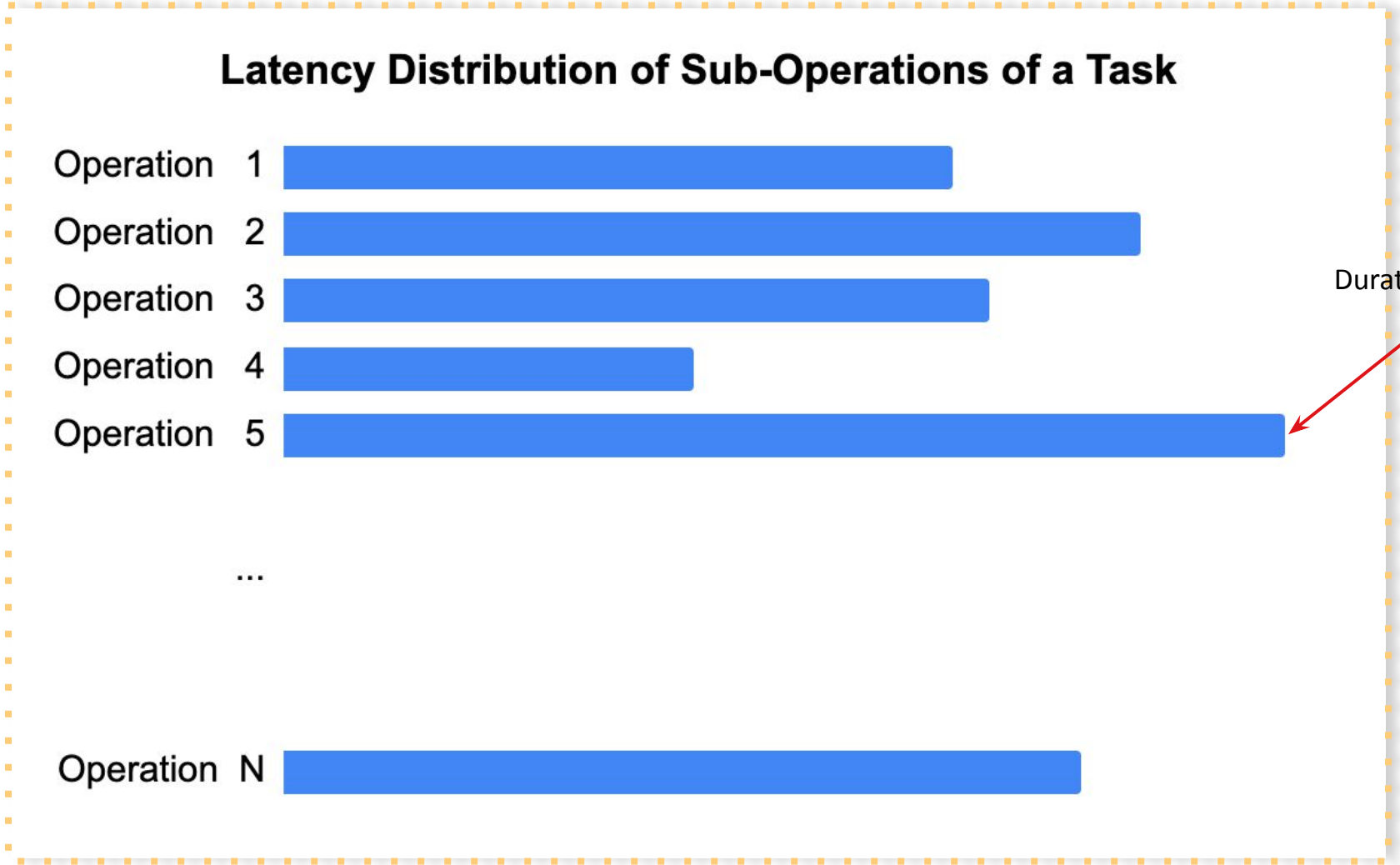
Scaling Aerospike

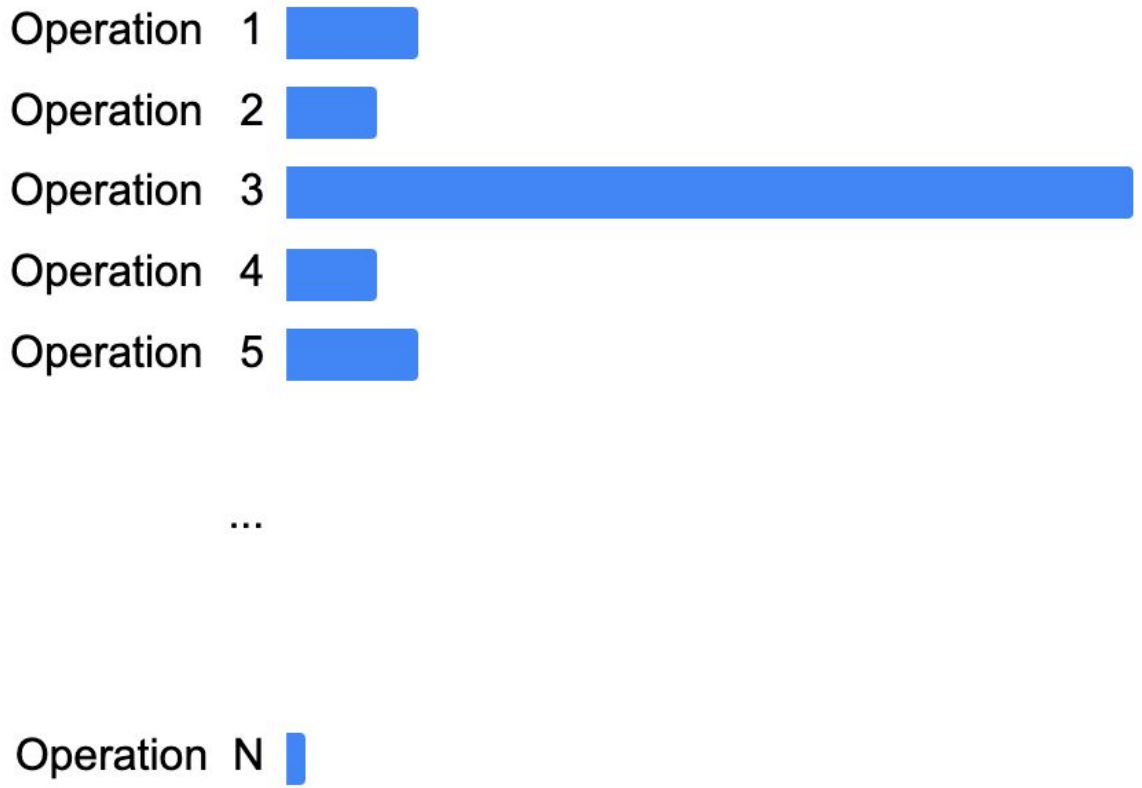
Lessons for Data-Intensive Application Developers

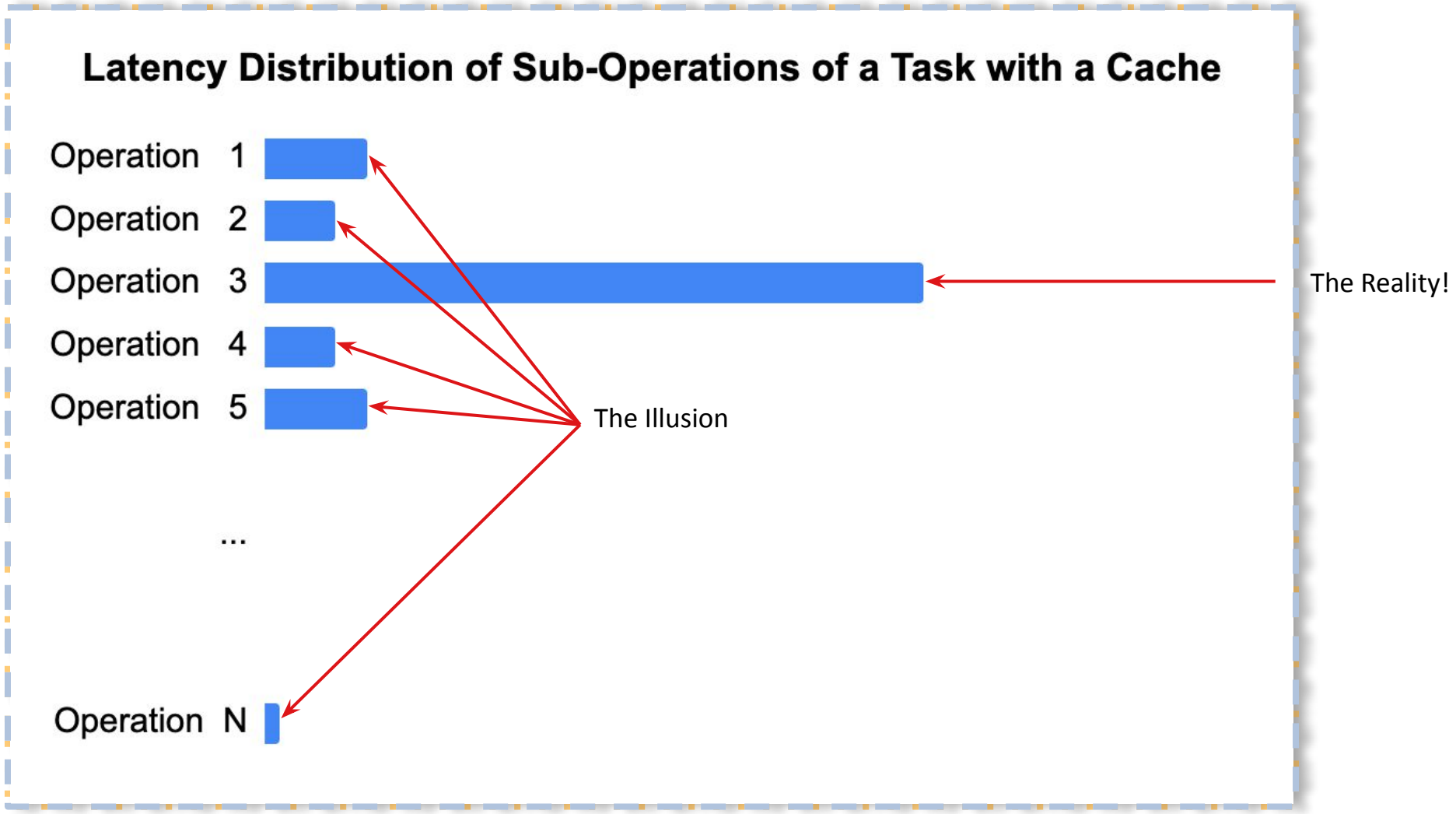
AEROSPIKE
{ developers }

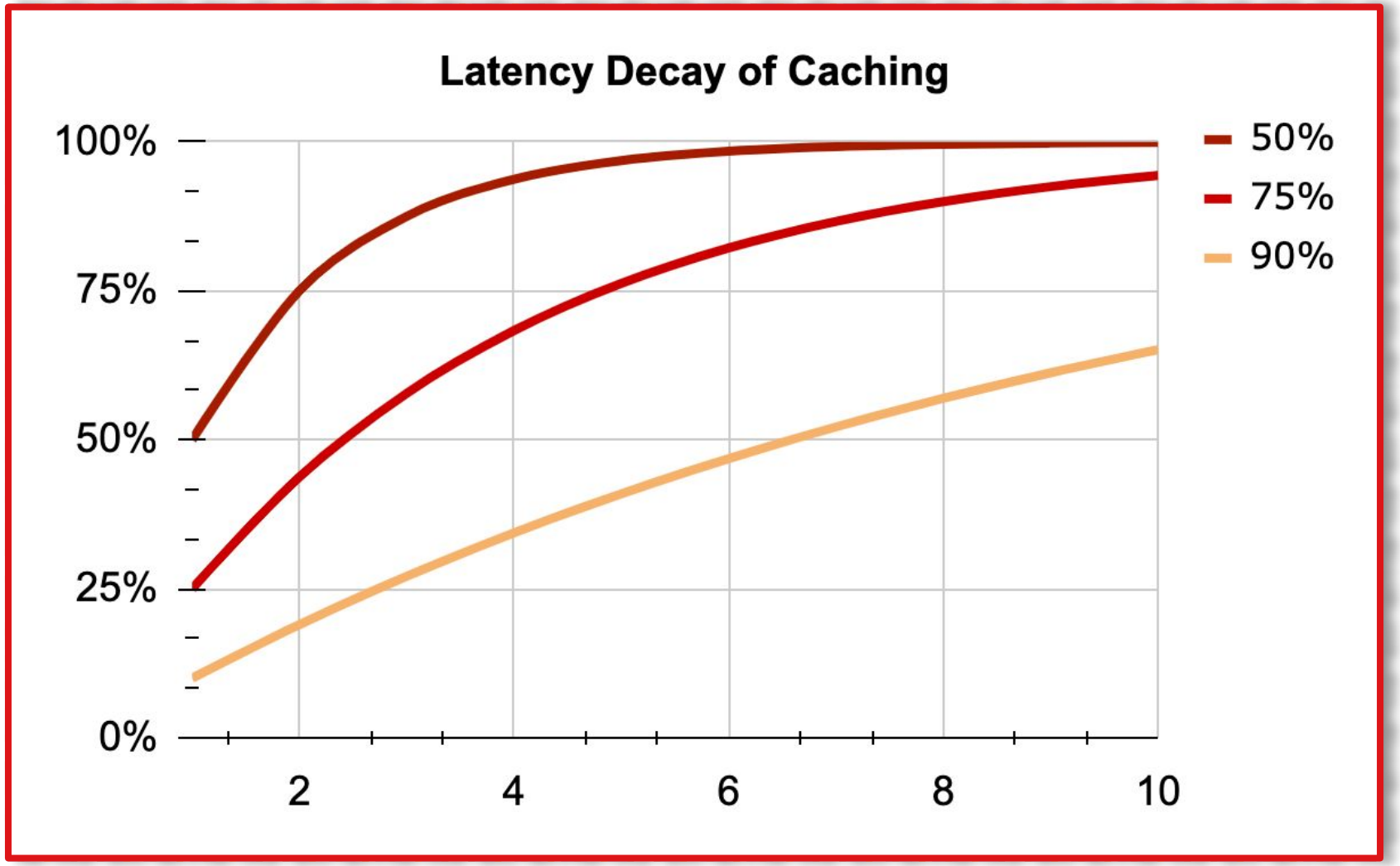




Latency Distribution of Sub-Operations of a Task with a Cache




Latency Distribution of Sub-Operations of a Task with a Cache
Operation 1
Operation 2
Operation 3
Operation 4
Operation 5
...
Operation N
The Illusion
The Reality!


Latency Decay of Caching
100%
75%
50%
25%
0%
2
4
6
8
10
50%
75%
90%

Aerospike NoSQL Database

Performant

- Serves data from disk with speed similar to a Cache.
- Handles high throughput.
- Scales From GB to PB.
- Predictable performance.

Highly Available

- Resilient.
- Self-healing Capability.
- Multi-DC Replication.
- Rack/AZ Awareness.

Efficient

- Low cost.
- Use resource efficiently.
- High Availability with RF=2.
- Reduce CO2 emissions!

Interoperable

- Key/value, document, graph.
- Kubernetes Operator.
- AWS Graviton compatible.
- Kafka, Spark, Pulsar, Trino, Elasticsearch integration.

AEROSPIKE { developers }

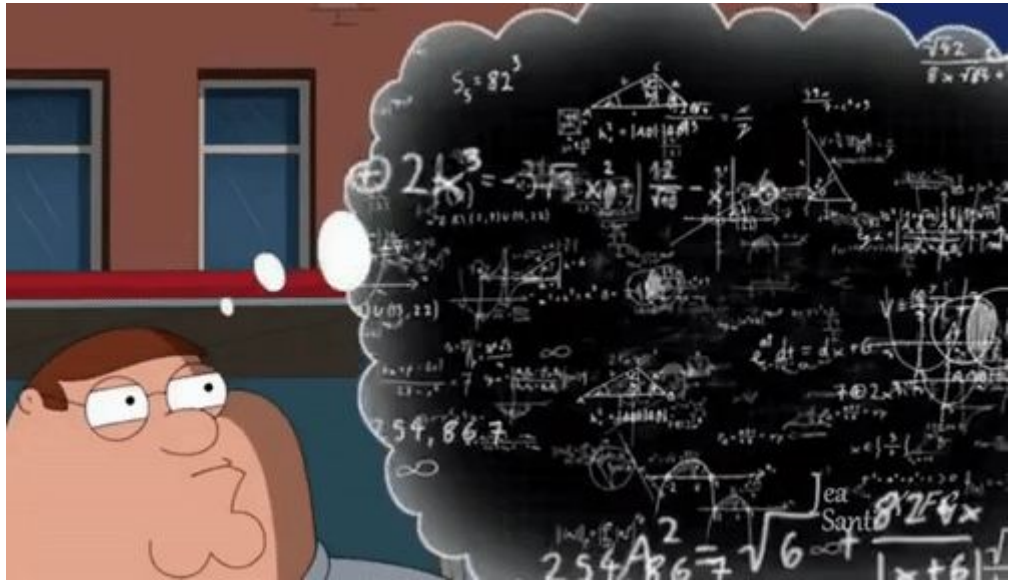
Software Design

- Software design is a practice in prioritisation of competing concerns.
- When prioritising, some trade-offs have to be made.
- Trade-offs cause imperfections!
- Two solutions may solve the same problem, but they have different imperfection.

Example: High-Availability Trade-Off/Imperfection

When building a Highly Available distributed system, would you design it to be:

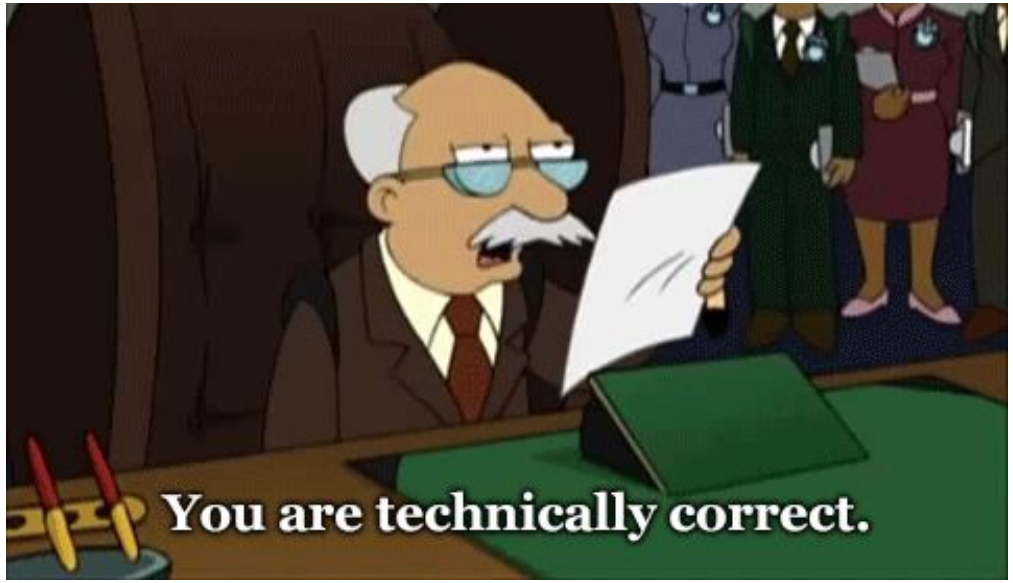
1. Efficient during failures, but only effective during normal times.
2. Efficient during normal times, but only effective during failures.



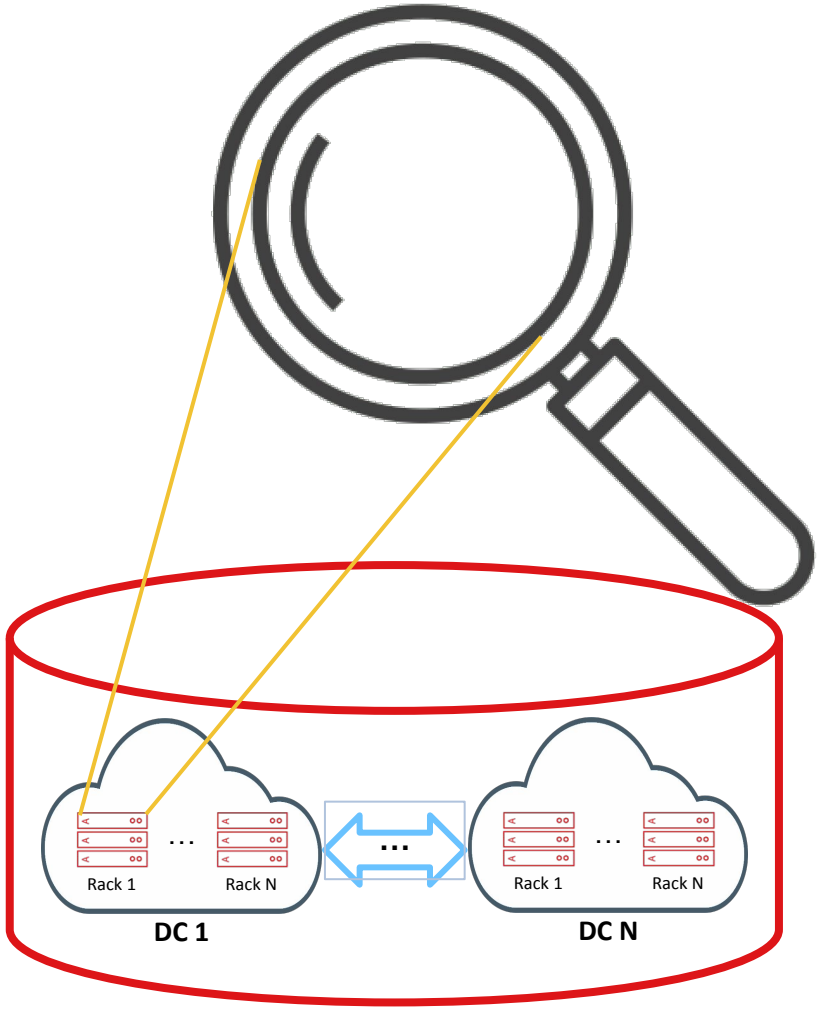
“It Depends!” 🤷

Let’s have a look at AWS SLAs:

- EC2 instance level uptime during a month: 99.5% (Uptime: 716:20, Downtime: 3:36)
- EC2 region level uptime during a month: 99.99% (Uptime: 719:56, Downtime: 0:04)



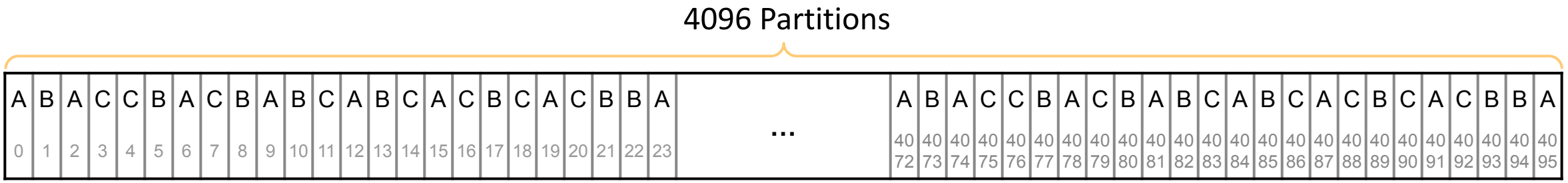
Demystifying Aerospike Architecture



AEROSPIKE
{ developers }

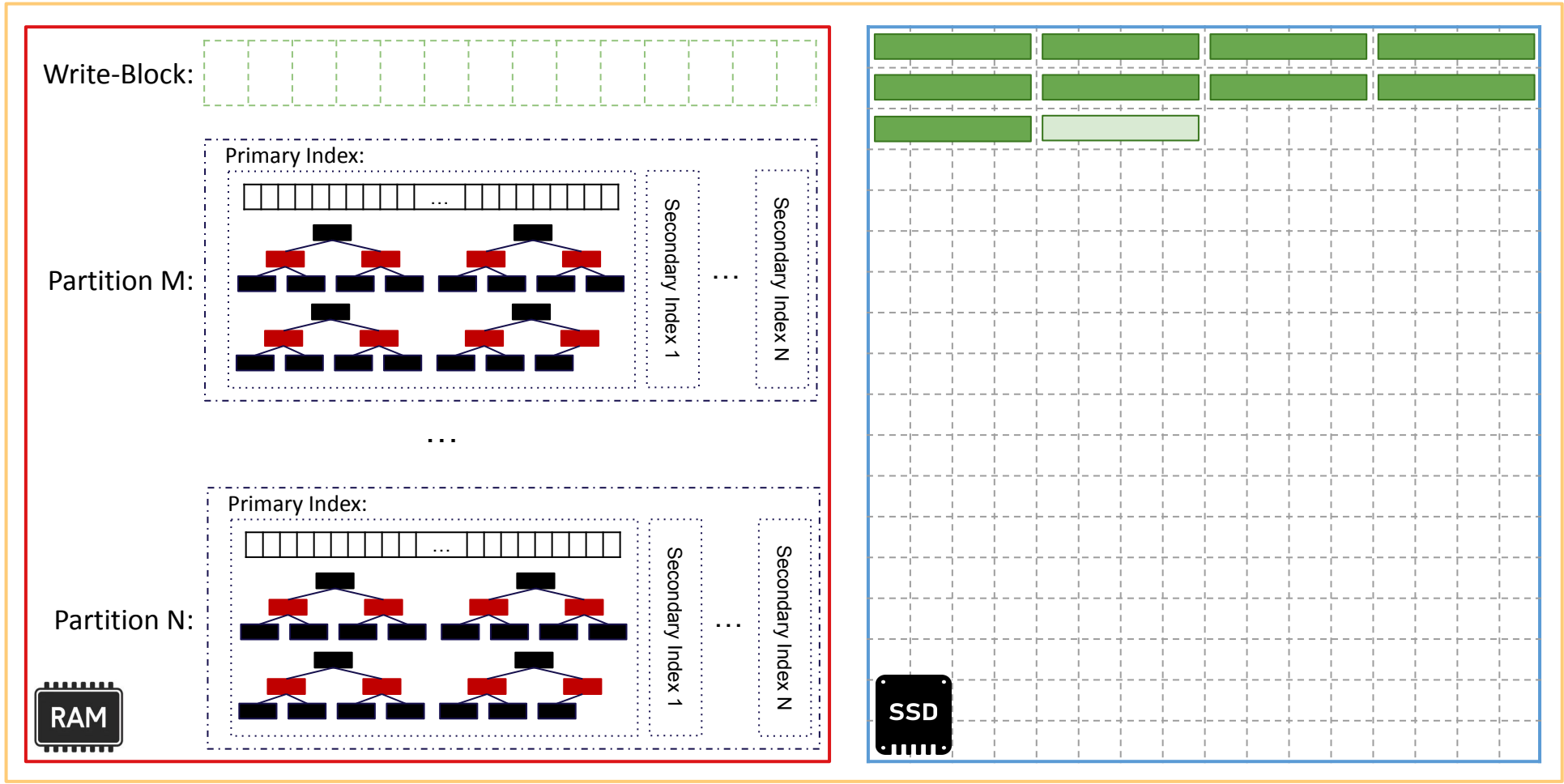
Data Distribution

Nodes: A, B, C



AEROSPIKE { developers }

Node Architecture



AEROSPIKE { developers }

Write-Block Magic

- A Write-Block gets flushed to the disk:
    i. Overwritten In-place: every second.
    ii. Immutably: When it doesn't have enough space for a new record.
- Write-Block guarantees both durability and fast access on disk. databases write the data on the disk twice:
    i. Durability (eg. OpLog, WriteAheadLog, CommitLog, Journal)
    ii. Fast access (Some kind of B-Tree or B+Tree)
- Some other benefits of Write-Block:
    i. Because Write-Block size is fixed, recovering and reusing space is simple.
    ii. Disk would never need to be defragged.

The Result

AEROSPIKE
{ developers }

3717	168938	409285	31599	0	0	0	0	0	0	609821
3718	170325	412446	31916	0	0	0	0	0	0	614689
3719	169268	409149	31565	0	0	0	0	0	0	609983
3720	170264	413315	31813	0	0	0	0	0	0	615390
3721	169174	410702	31699	0	0	0	0	0	0	611578
3722	170306	412384	31911	0	0	0	0	0	0	614599
3723	169260	409527	31647	0	0	0	0	0	0	610433
3724	169933	412725	31859	0	0	0	0	0	0	614518
3725	169062	408494	31547	0	0	0	0	0	0	609101

Collected Keys Stat [Max: 50000000, Save Ratio: 25%, # Processed: 944294266, # Collected: 50000000, Full Percent: 100%, Duration: 00:19:52]

Task Thread	Write 92 (92)	Read(Hit %50) 118 (118)	Update 25 (25)	Expression R 0 (0)	Expression W 0 (0)	Batch W (10) 0 (0)	Search 0 (0)	UDF 0 (0)	UDF Aggregate 0 (0)	Total TPS
3726	170410	413650	31900	0	0	0	0	0	0	615960
3727	169153	410968	31782	0	0	0	0	0	0	611903
3728	169329	412434	31587	0	0	0	0	0	0	613350
3729	170238	413316	31824	0	0	0	0	0	0	615378
3730	169457	411910	31705	0	0	0	0	0	0	613072
3731	170438	414912	31909	0	0	0	0	0	0	617259
3732	168993	410619	31695	0	0	0	0	0	0	611310
3733	170389	414060	31874	0	0	0	0	0	0	616321
3734	169510	412119	31667	0	0	0	0	0	0	613295
3735	168426	408124	31506	0	0	0	0	0	0	608057
3736	169997	414269	31840	0	0	0	0	0	0	616105
3737	169349	412395	31687	0	0	0	0	0	0	613432
3738	170238	412416	31776	0	0	0	0	0	0	614430
3739	168712	410046	31624	0	0	0	0	0	0	610382
3740	169826	411726	31814	0	0	0	0	0	0	613367
3741	169214	410018	31783	0	0	0	0	0	0	611014
3742	169388	409524	31716	0	0	0	0	0	0	610627
3743	169947	410886	31806	0	0	0	0	0	0	612640
3744	169278	409546	31550	0	0	0	0	0	0	610373
3745	169901	411822	31862	0	0	0	0	0	0	613585
3746	169241	410132	31664	0	0	0	0	0	0	611038
3747	169209	411079	31709	0	0	0	0	0	0	611999
3748	169572	409387	31613	0	0	0	0	0	0	610569
3749	169108	408406	31423	0	0	0	0	0	0	608939
3750	170351	412168	31593	0	0	0	0	0	0	614110

Collected Keys Stat [Max: 50000000, Save Ratio: 25%, # Processed: 948532714, # Collected: 50000000, Full Percent: 100%, Duration: 00:19:52]

Task Thread	Write 92 (92)	Read(Hit %50) 118 (118)	Update 25 (25)	Expression R 0 (0)	Expression W 0 (0)	Batch W (10) 0 (0)	Search 0 (0)	UDF 0 (0)	UDF Aggregate 0 (0)	Total TPS
3751	169177	410913	31545	0	0	0	0	0	0	611635
3752	169214	412940	31670	0	0	0	0	0	0	613825
3753	170174	415264	31903	0	0	0	0	0	0	617342
3754	169710	412713	31652	0	0	0	0	0	0	614075
3755	168959	411995	31622	0	0	0	0	0	0	612575

| Collected Keys Stat [Max: 50000000, Save Ratio: 25%, # Processed: 948532714, # Collected: 50000000, Full Percent: 100%, Duration: 00:19:52]

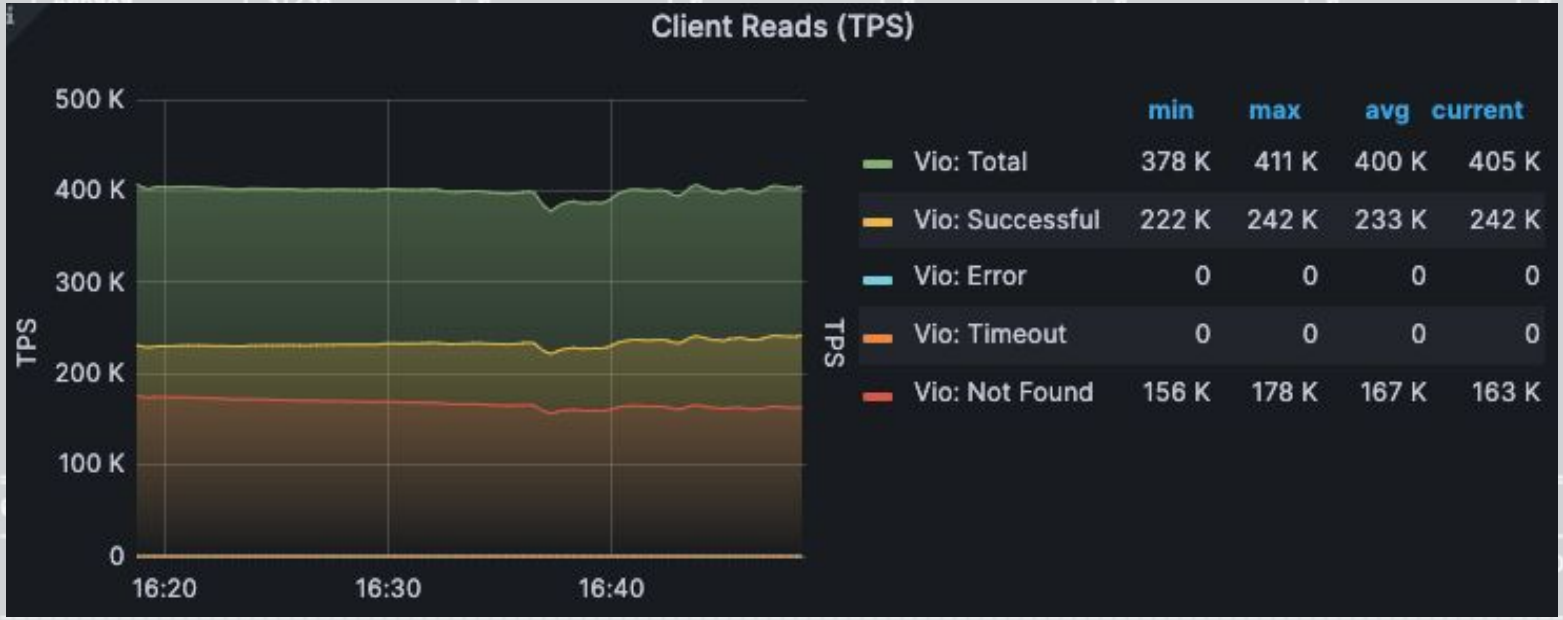
Task Thread	Write 92 (92)	Read(Hit %50) 118 (118)	Update 25 (25)	Expression R 0 (0)	Expression W 0 (0)	Batch W (10) 0 (0)	Search 0 (0)	UDF 0 (0)	UDF Aggregate 0 (0)	Total TPS
3751	169177	410913	31545	0	0	0	0	0	0	611635
3752	169214	412940	31670	0	0	0	0	0	0	613825
3753	170174	415264	31903	0	0	0	0	0	0	617342
3754	169710	412713	31652	0	0	0	0	0	0	614075
3755	168959	411995	31622	0	0	0	0	0	0	612575
3756	169822	412695	31715	0	0	0	0	0	0	614231
3757	167586	405881	31239	0	0	0	0	0	0	604708
3758	169165	408816	31478	0	0	0	0	0	0	609457
3759	167812	[illegible]								606040
3760	168579									611314
3761	169507									615037
3762	168594									609991
3763	167871									607945
3764	168965									608794
3765	168376									607397
3766	168465									607699
3767	169156									613200
3768	168979									612248
3769	168569									612119
3770	169993									615309
3771	169285									615056
3772	168996									613380
3773	170455									617685
3774	168645									610878
3775	168915									612950

| Collected Keys Stat [Max: 50

Task Thread	Write 92 (92)	[illegible]							F Aggregate (0)	Total TPS
3776	170083	414252	31869	0	0	0	0	0	0	616201
3777	168736	410862	31572	0	0	0	0	0	0	611171
3778	169046	413283	31713	0	0	0	0	0	0	614042
3779	169445	412084	31702	0	0	0	0	0	0	613231
3780	170375	415090	31955	0	0	0	0	0	0	617420
3781	169635	412633	31757	0	0	0	0	0	0	614024
3782	169516	412774	31710	0	0	0	0	0	0	614000
3783	170857	416498	31923	0	0	0	0	0	0	619279
3784	169384	410922	31565	0	0	0	0	0	0	611870
3785	169500	412789	31691	0	0	0	0	0	0	613985
3786	169731	413755	31805	0	0	0	0	0	0	615288
3787	170378	414514	31899	0	0	0	0	0	0	616789
3788	169255	412681	31663	0	0	0	0	0	0	613600
3789	169308	412446	31723	0	0	0	0	0	0	613478

| Collected Keys Stat [Max: 50000000, Save Ratio: 25%, # Processed: 948532714, # Collected: 50000000, Full Percent: 100%, Duration: 00:19:52]

Task Thread	Write 92 (92)	Read(Hit %50) 118 (118)	Update 25 (25)	Expression R 0 (0)	Expression W 0 (0)	Batch W (10) 0 (0)	Search 0 (0)	UDF 0 (0)	UDF Aggregate 0 (0)	Total TPS
3751	169177	410913	31545	0	0	0	0	0	0	611635
3752	169214	412940	31670	0	0	0	0	0	0	613825
3753	170174	415264	31903	0	0	0	0	0	0	617342
3754	169710	412713	31652	0	0	0	0	0	0	614075
3755	168959	411995	31622	0	0	0	0	0	0	612575
3756	169822	412695	31715	0	0	0	0	0	0	614231
3757	167586	405881	31239	0	0	0	0	0	0	604708
3758	169165	408816	31478	0	0	0	0	0	0	609457
3759	167812	[illegible]								606040
3760	168579									611314
3761	169507									615037
3762	168594									609991
3763	167871									607945
3764	168965									608794
3765	168376									607397
3766	168465									607699
3767	169156									613200
3768	168979									612248
3769	168569									612119
3770	169993									615309
3771	169285									615056
3772	168996									613380
3773	170455									617685
3774	168645									610878
3775	168915									612950



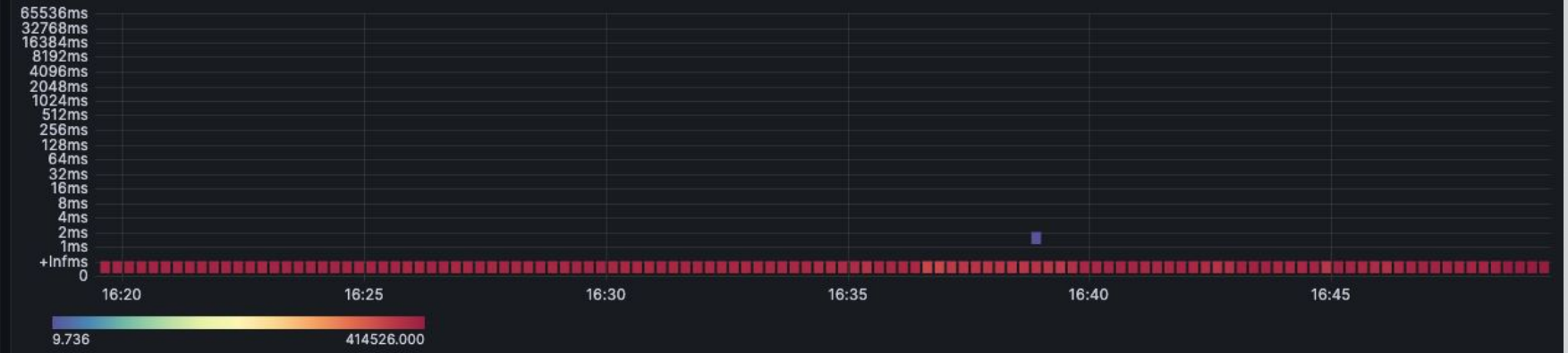
| Collected Keys Stat [Max: 50

Task Thread	Write 92 (92)								F Aggregate (0)	Total TPS
3776	170083	414252	31869	0	0	0	0	0	0	616201
3777	168736	410862	31572	0	0	0	0	0	0	611171
3778	169046	413283	31713	0	0	0	0	0	0	614042
3779	169445	412084	31702	0	0	0	0	0	0	613231
3780	170375	415090	31955	0	0	0	0	0	0	617420
3781	169635	412633	31757	0	0	0	0	0	0	614024
3782	169516	412774	31710	0	0	0	0	0	0	614000
3783	170857	416498	31923	0	0	0	0	0	0	619279
3784	169384	410922	31565	0	0	0	0	0	0	611870
3785	169500	412789	31691	0	0	0	0	0	0	613985
3786	169731	413755	31805	0	0	0	0	0	0	615288
3787	170378	414514	31899	0	0	0	0	0	0	616789
3788	169255	412681	31663	0	0	0	0	0	0	613600
3789	169308	412446	31723	0	0	0	0	0	0	613478

| Collected Keys Stat [Max: 50000000, Save Ratio: 25%, # Processed: 948532714, # Collected: 50000000, Full Percent: 100%, Duration: 00:19:52]

Task Thread	Write 92 (92)	Read(Hit %50) 118 (118)	Update 25 (25)	Expression R 0 (0)	Expression W 0 (0)	Batch W (10) 0 (0)	Search 0 (0)	UDF 0 (0)	UDF Aggregate 0 (0)	Total TPS
3751	169177	410913	31545	0	0	0	0	0	0	611635
3752	169214	412940	31670	0	0	0	0	0	0	613825

p95 ms
1
p99 ms
1
p99.9 ms
1

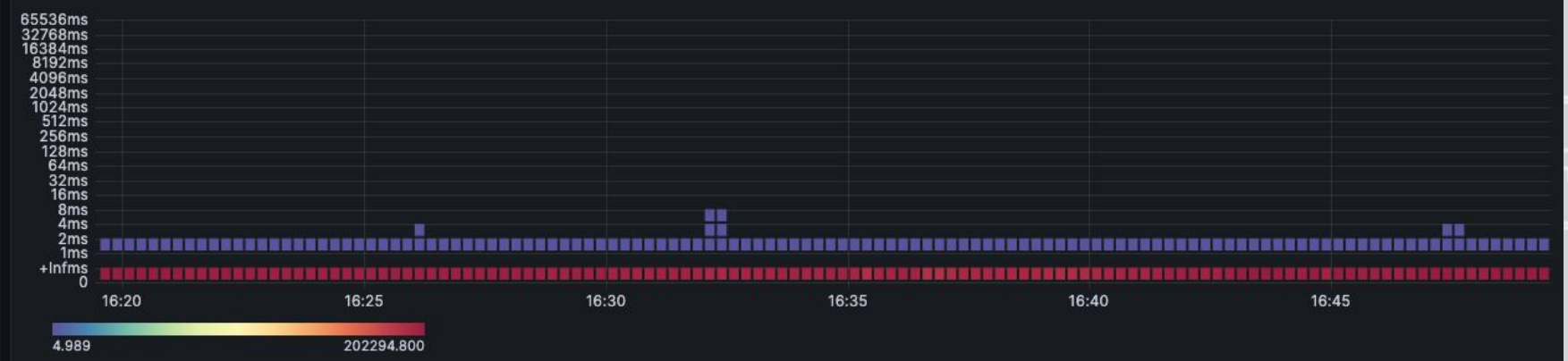


write

Percentiles write

p95 ms
1
p99 ms
1
p99.9 ms
1

Latency write



3776	169516	412774	31710	0	0	0	0	0	0	614000
3777	170857	416498	31923	0	0	0	0	0	0	619279
3778	169384	410922	31565	0	0	0	0	0	0	611870
3779	169500	412789	31691	0	0	0	0	0	0	613985
3780	169731	413755	31805	0	0	0	0	0	0	615288
3781	170378	414514	31899	0	0	0	0	0	0	616789
3782	169255	412681	31663	0	0	0	0	0	0	613600
3783	169308	412446	31723	0	0	0	0	0	0	613478

Example* of Performant and Efficient

Name	Instance ID	Instance state	Instance type
aerolab4-Demo_1	i-04a18f4d5c3849854	Running	i4i.4xlarge
aerolab4-Demo_2	i-0d8dd3bc198e5eea3	Running	i4i.4xlarge
aerolab4-Demo_3	i-06d83bfea16d28e76	Running	i4i.4xlarge
aerolab4-Demo_4	i-0a3ceaba775d77b00	Running	i4i.4xlarge
aerolab4-Perseus_1	i-097d76344a2fbf4e3	Running	c5.9xlarge

Instance Name	vCPU Count	Memory (GB)	Storage (GB)
i4i.4xlarge	16	128	3750

Node Count	RF	Capacity (TB)	Throughput
4	2	3	600 K TPS

Region	AWS Annual Reserved Cost
London	$36,282
Ireland	$34,483
Ohio	$31,203

* This example can be presented to you as a live demo upon request.

AEROSPIKE { developers }

Thank you :)

Developer Hub
AEROSPIKE
https://developer.aerospike.com

GitHub
https://github.com/aerospike-examples

Documentation
AEROSPIKE
https://docs.aerospike.com

AEROSPIKE
{ developers }