4 Myths about in-memory databases busted

Yiftach Shoolman – Co-Founder & CTO @ Redis Labs

@yiftachsh, @redislabsinc

redislabs

Background - Redis

Created by Salvatore Sanfilippo (@antirez)

OSS, in-memory NoSQL k/v database/data-structure engine

1

Fastest growing DB
2013-01 – to date:
DB-Engines

2

Most popular database on containers:
@DevOps.com
& ClusterHQ

3

Top NoSQL databases:
DB-Engines

12

Top tools developers love:
StackShare

Backgroud – Redis Labs

Founded in 2011. HQ in Mountain View CA, R&D in Tel-Aviv IL

The largest commercial company behind OSS Redis

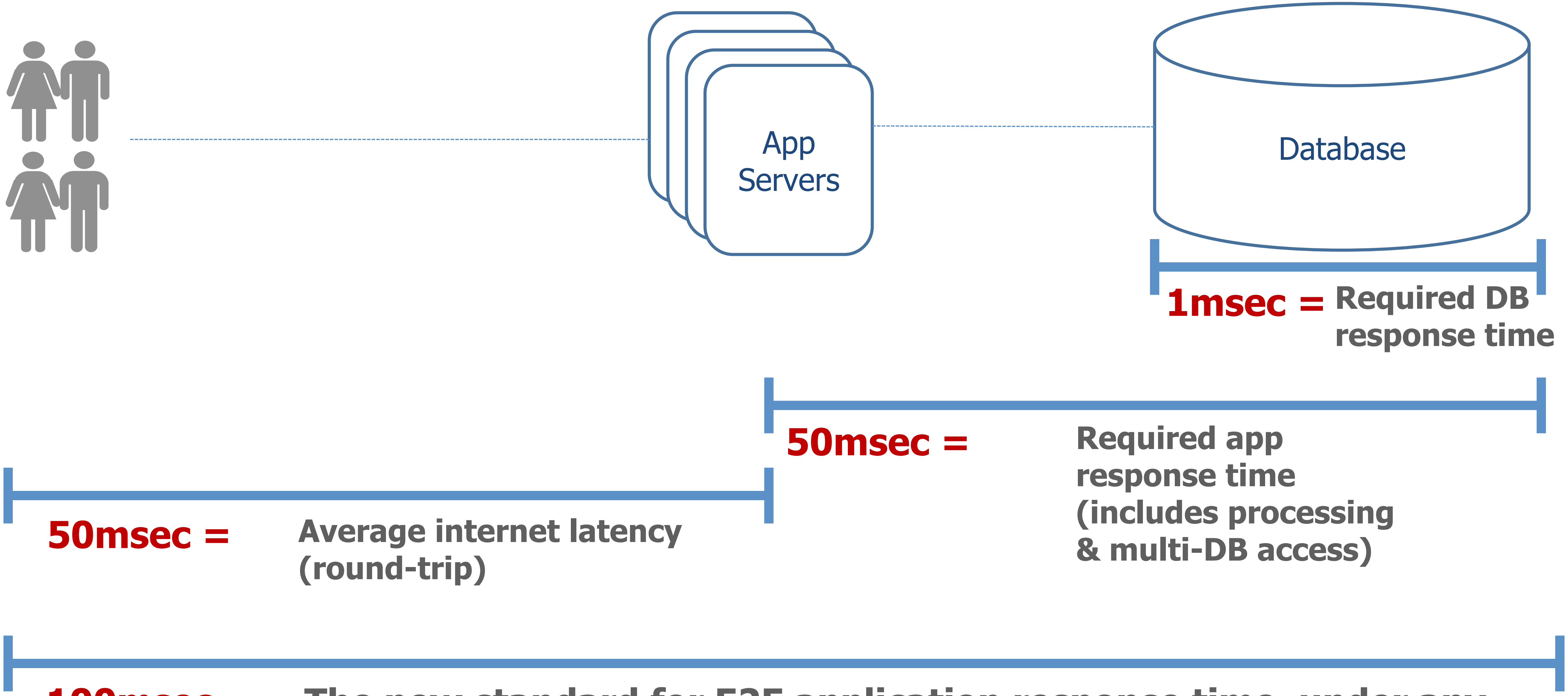
- 5000+ paying customers
- 30,000+ free users
- 100,000+ databases under management
- ±200 new databases/day

Provide enterpsie class Redis deployement

- As a service – Redis Cloud
- On-premises – Redis Labs Enterprise Cluster (RLEC)

$28MM VC funding

Why an in-memory operational DBMS ?



1msec = **Required DB response time**

50msec = **Required app response time (includes processing & multi-DB access)**

50msec = **Average internet latency (round-trip)**

100msec = **The new standard for E2E application response time, under any load**

Why an in-memory analytics DBMS ?

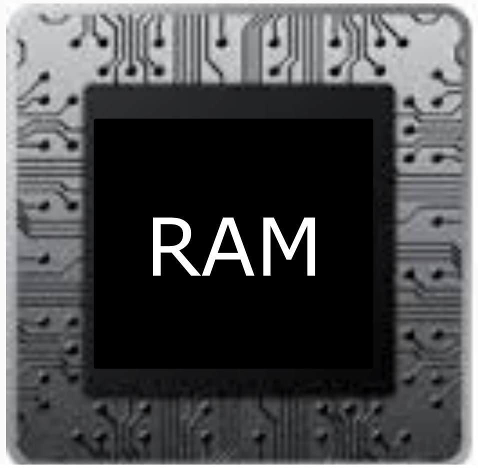
Query time:

Days/Hours

Disk-based

Query time:

Mins/Secs

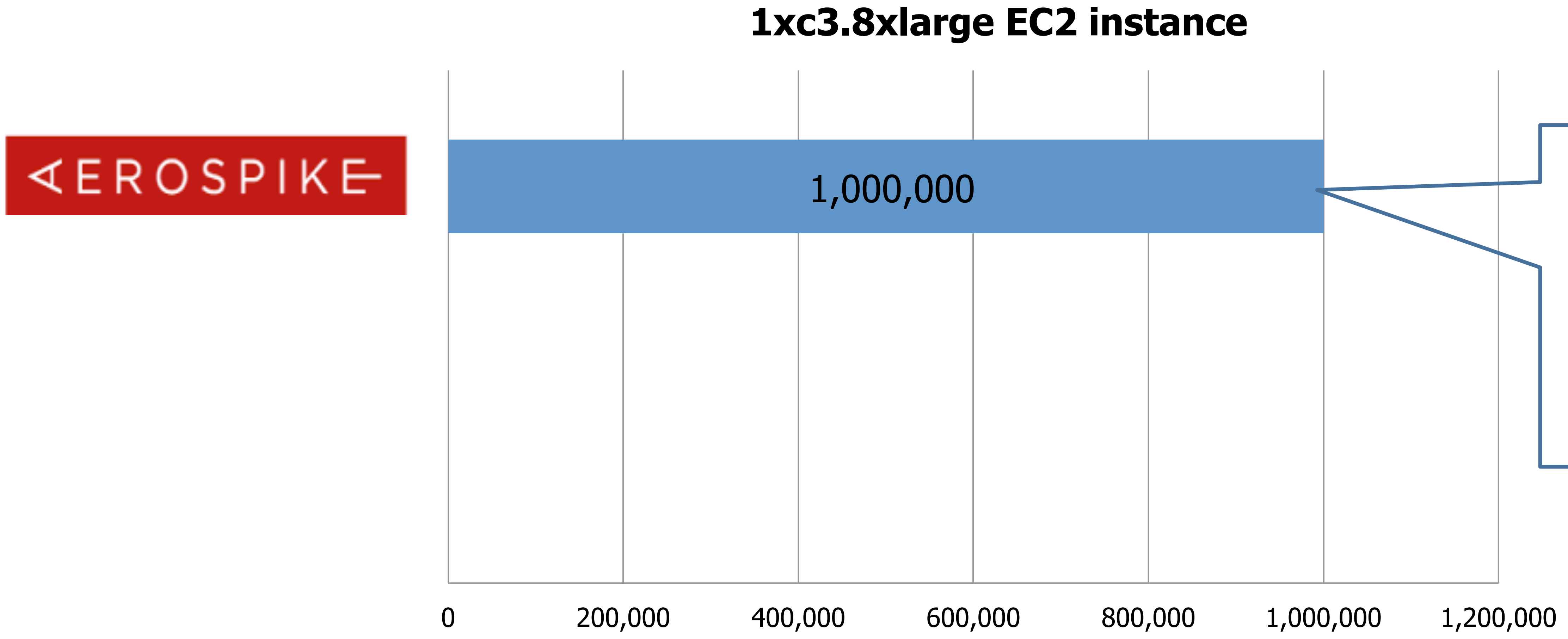


RAM-based

Myth #1

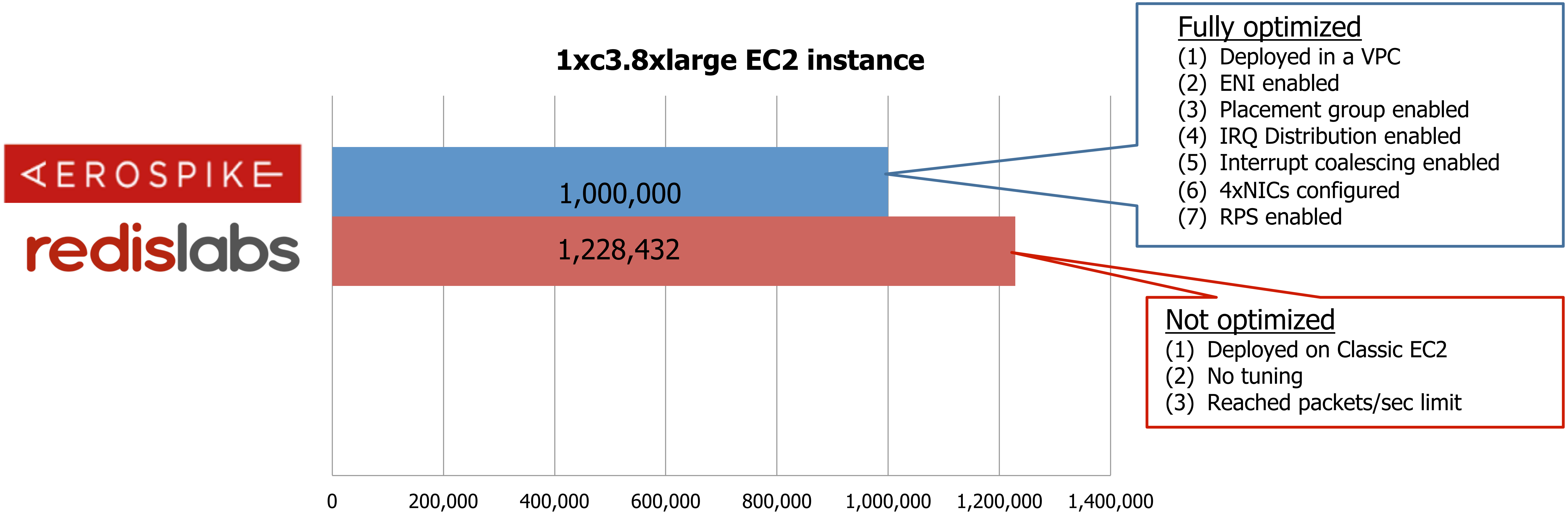
all in-memory databases are equally fast

How many ops/sec can a single EC2 instance do?



(1) Deployed in a VPC
(2) Enhanced Network Interface (ENI) enabled
(3) Placement group enabled
(4) IRQ Distribution enabled
(5) Interrupt coalescing enabled
(6) 4xNICs configured
(7) Receive Packet Steering (RPS) enabled

How many ops/sec can a single EC2 instance do?



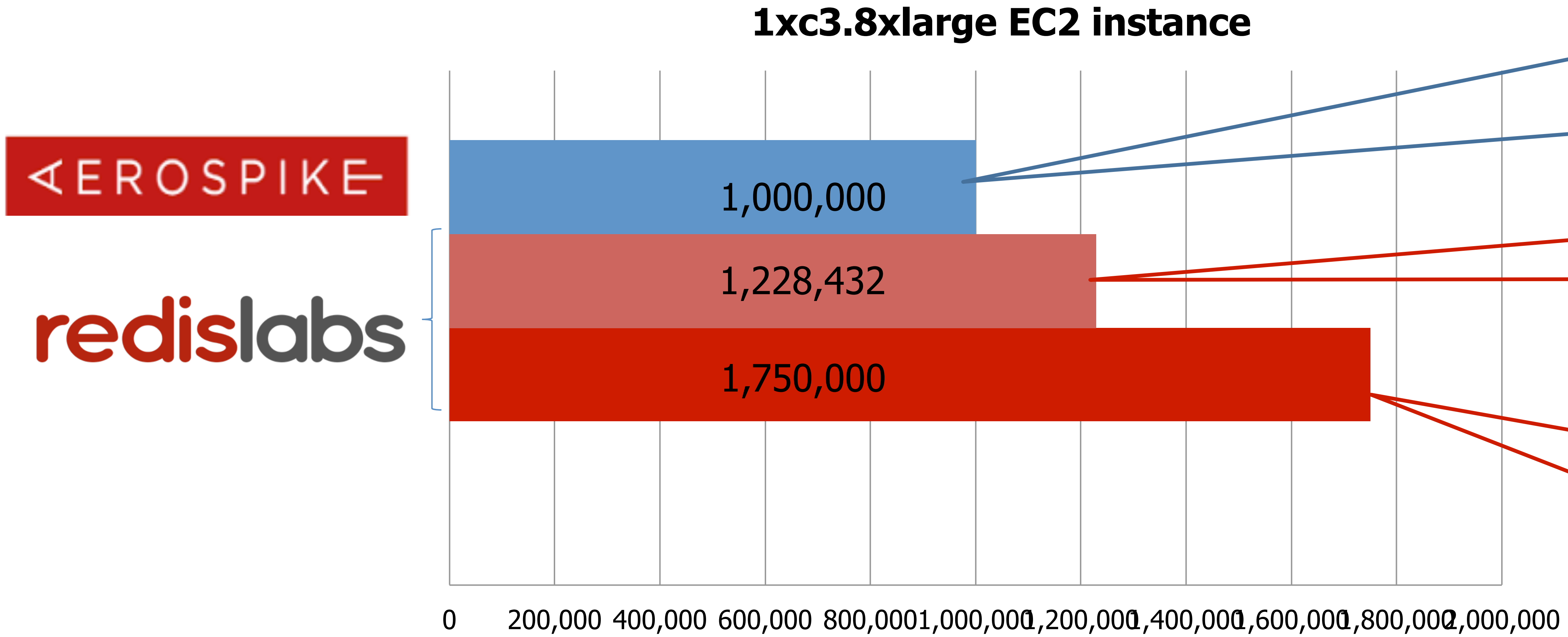
Fully optimized

(1) Deployed in a VPC
(2) ENI enabled
(3) Placement group enabled
(4) IRQ Distribution enabled
(5) Interrupt coalescing enabled
(6) 4xNICs configured
(7) RPS enabled

Not optimized

(1) Deployed on Classic EC2
(2) No tuning
(3) Reached packets/sec limit

How many ops/sec can a single EC2 instance do?



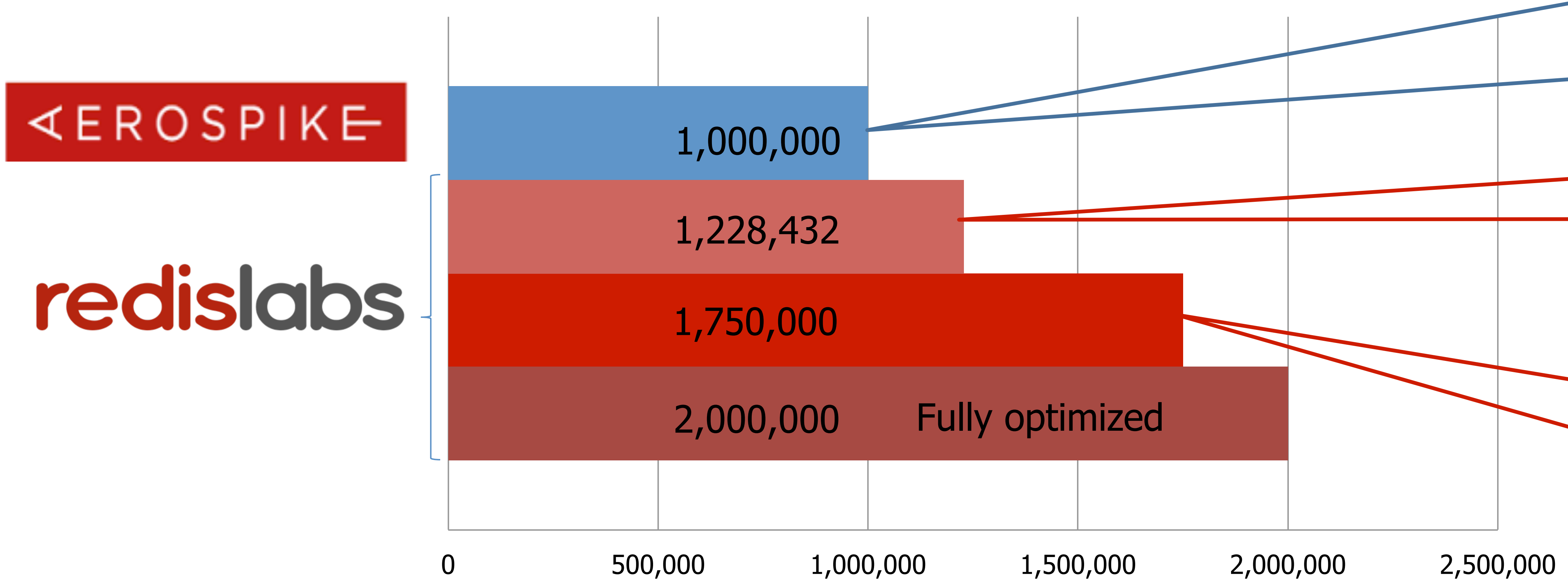
Fully optimized
(1) Deployed in a VPC
(2) ENI enabled
(3) Placement group enabled
(4) IRQ Distribution enabled
(5) Interrupt coalescing enabled
(6) 4xNICs configured
(7) RPS enabled

Not optimized
(1) Deployed on Classic EC2
(2) No tuning
(3) Reached packets/sec limit

Half optimized
(1) Deployed in a VPC
(2) ENI enabled
(3) RPS enabled
(4) Only 1xNIC

How many ops/sec can a single EC2 instance do?

1xc3.8xlarge EC2 instance

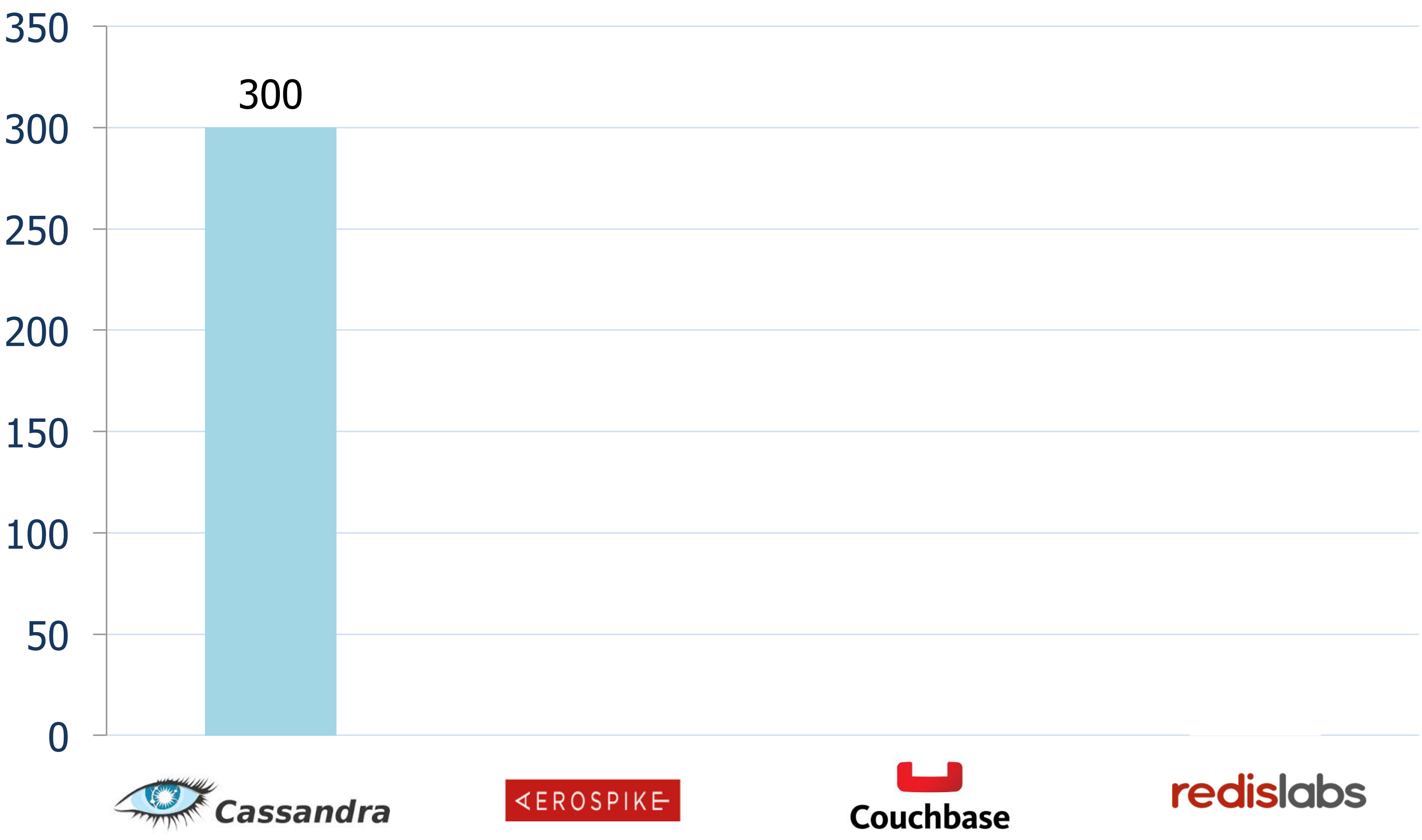


Fully optimized
(1) Deployed in a VPC
(2) ENI enabled
(3) Placement group enabled
(4) IRQ Distribution enabled
(5) Interrupt coalescing enabled
(6) 4xNICs configured
(7) RPS enabled

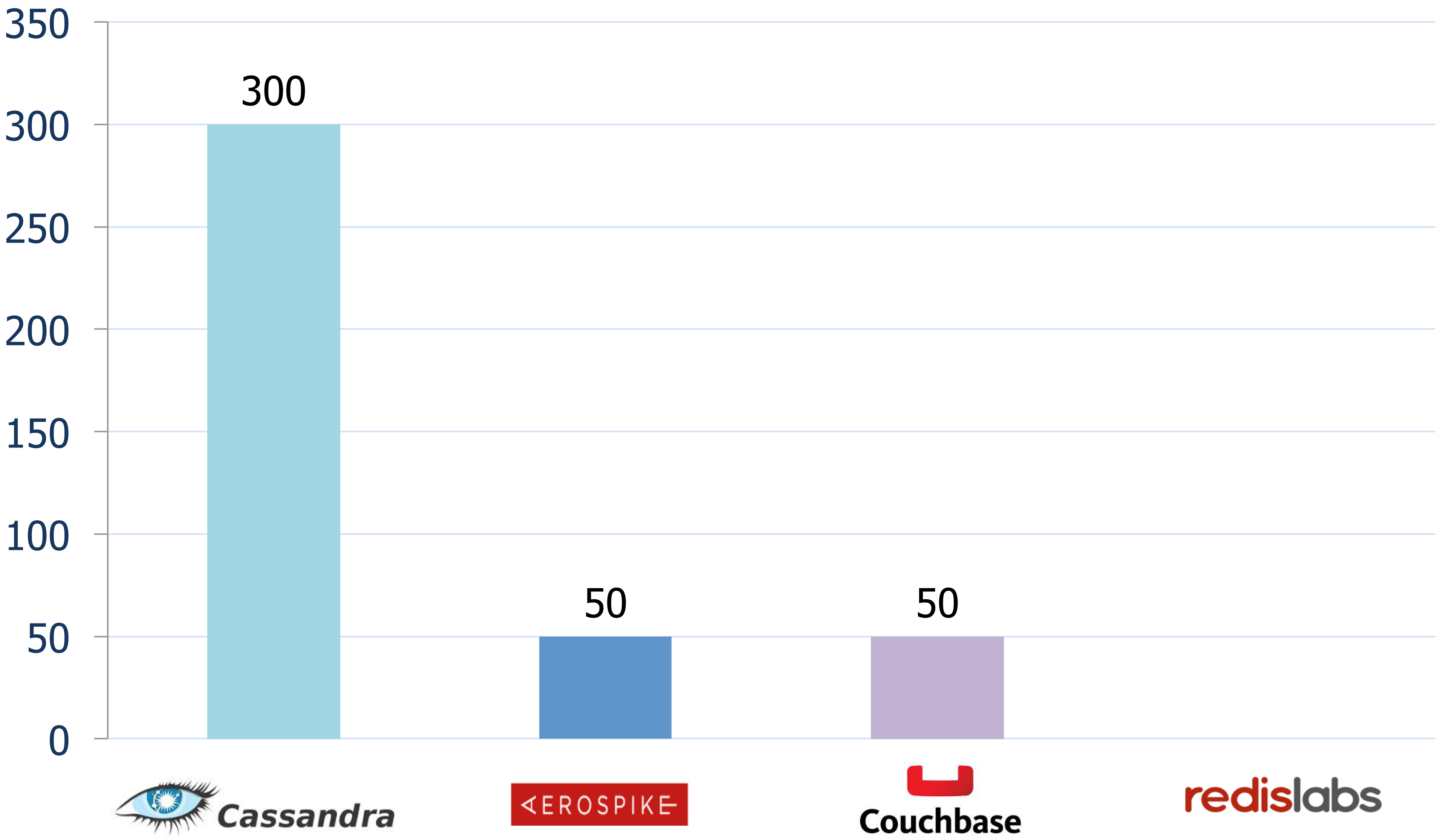
Not optimized
(1) Deployed on Classic EC2
(2) No tuning
(3) Reached packets/sec limit

Half optimized
(1) Deployed in a VPC
(2) ENI enabled
(3) RPS enabled
(4) Only 1xNIC

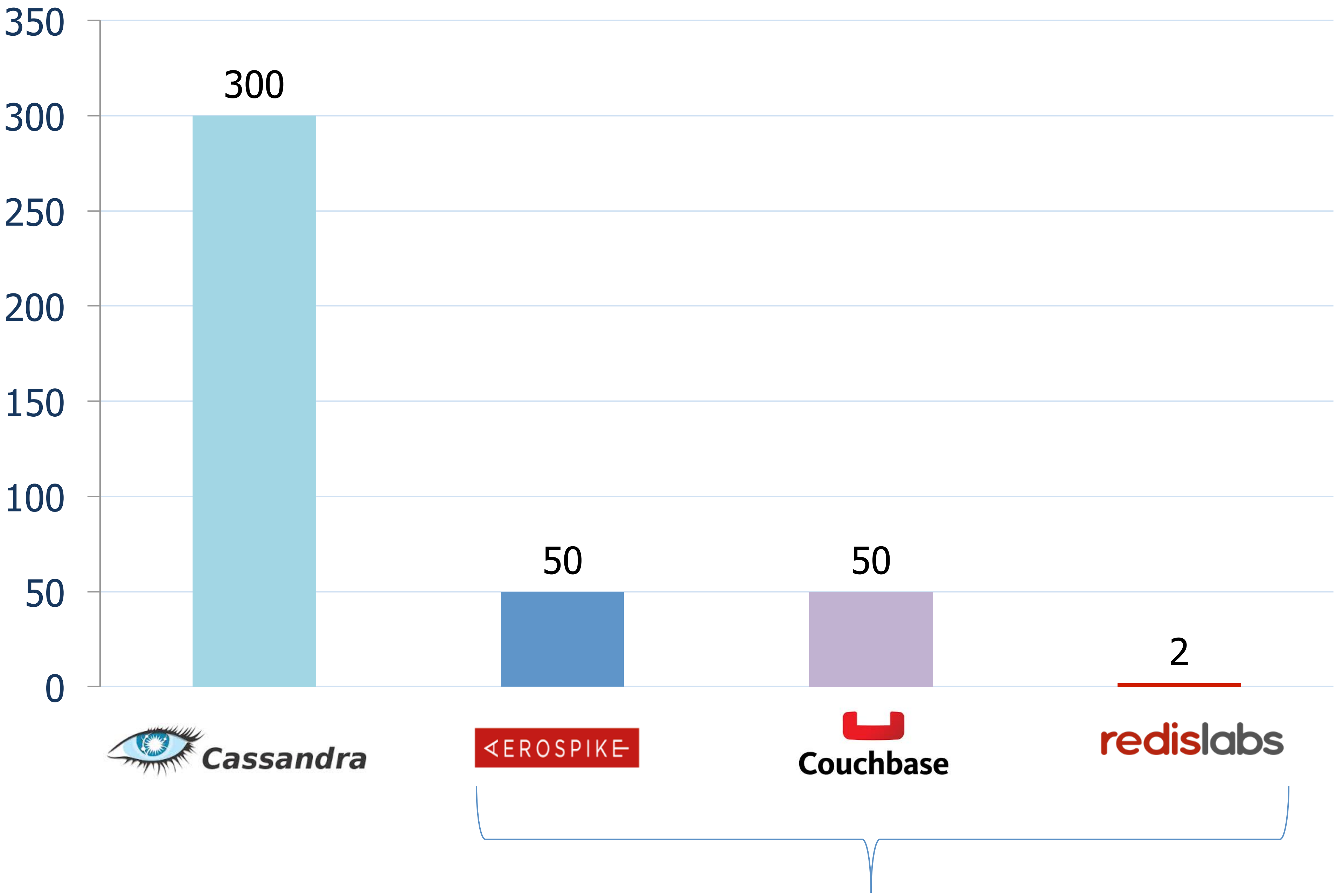
How many servers to get 1M writes/sec on GCE?



How many servers to get 1M writes/sec on GCE?

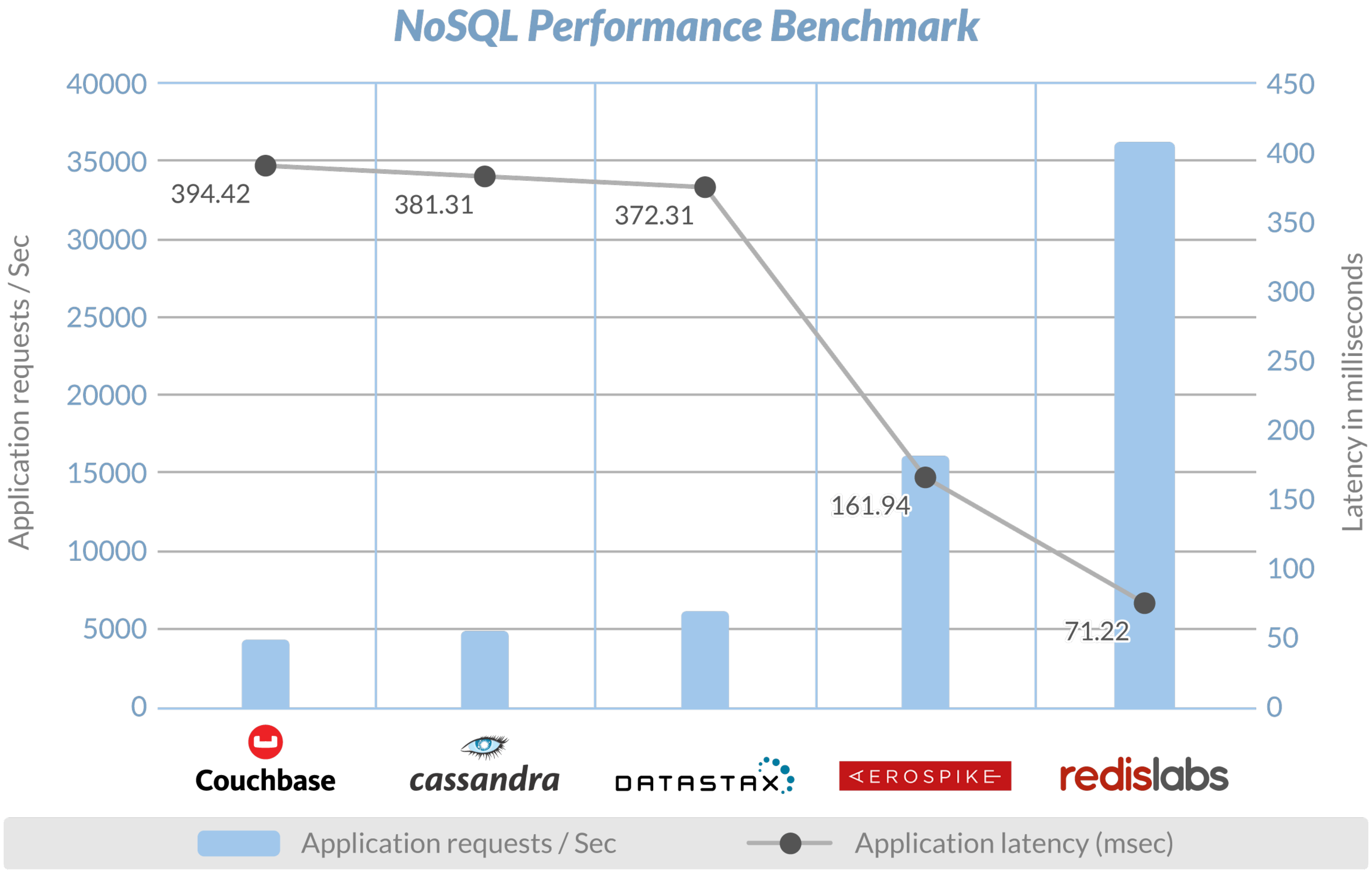


How many servers to get 1M writes/sec on GCE?

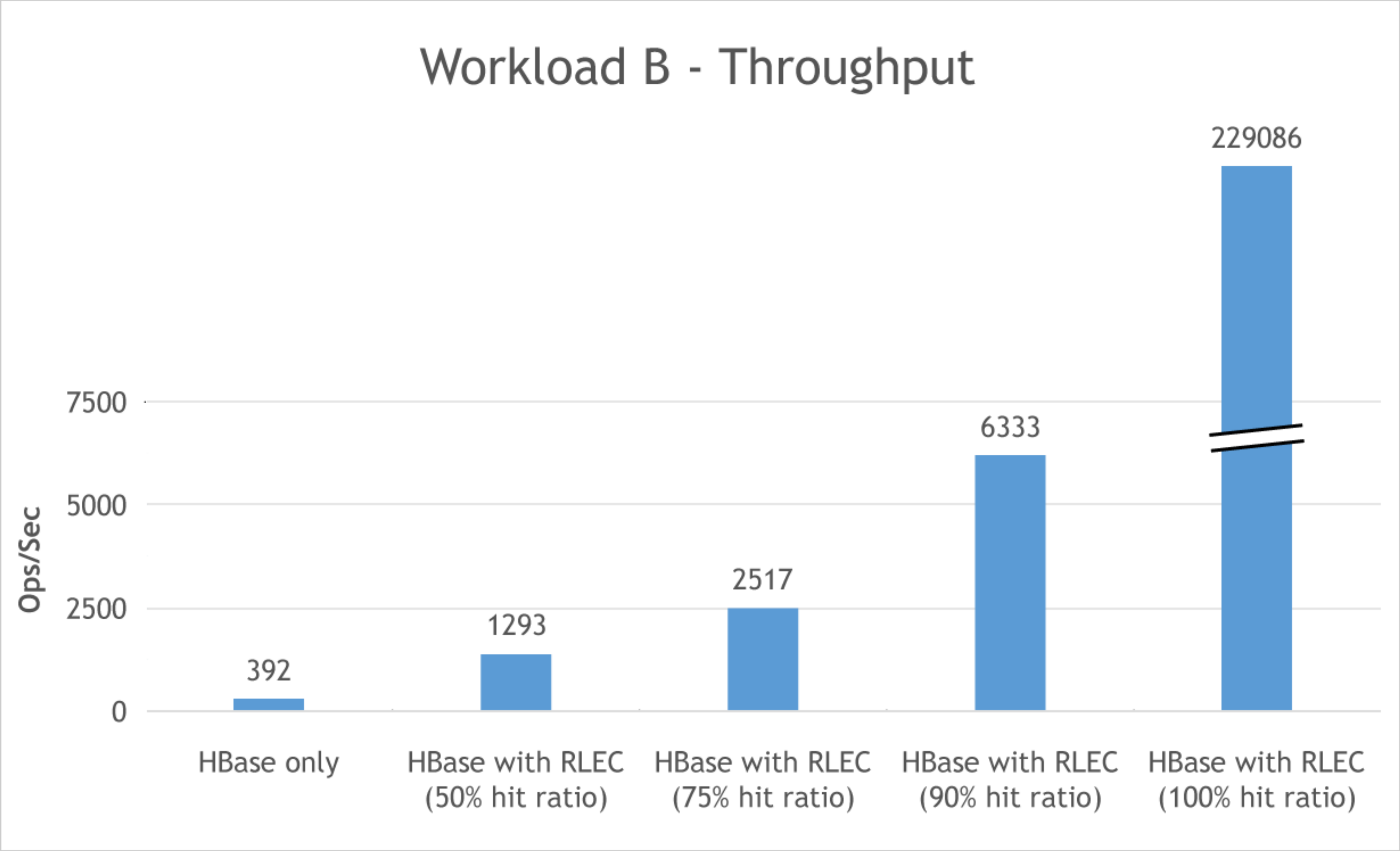


In-memory databases

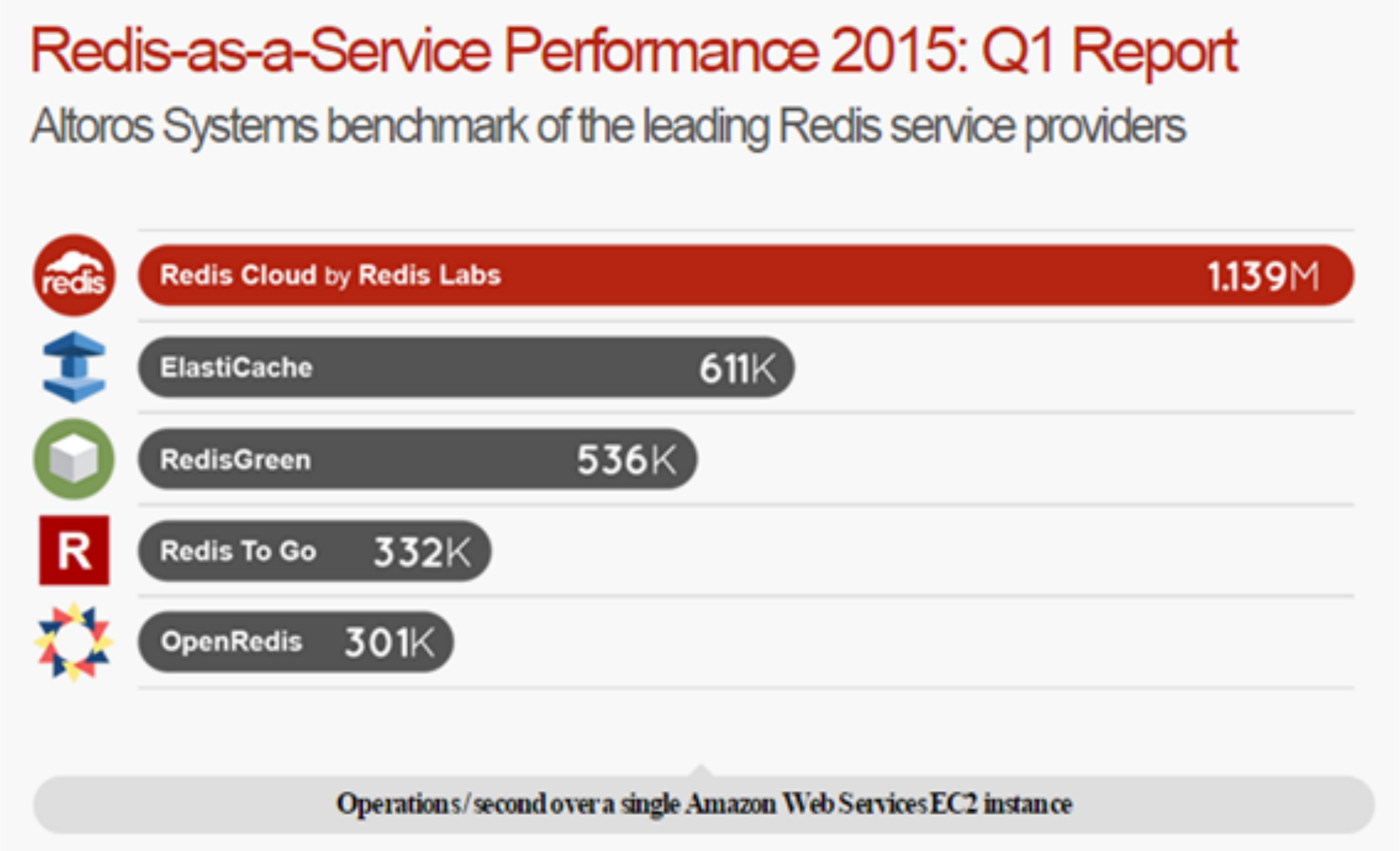
Real-world write intensive app



Hbase+Internal Cache vs. Hbase+Redis



Same Redis core, same HW, different performance



Redis-as-a-Service Performance 2015: Q1 Report

Altoros Systems benchmark of the leading Redis service providers

Operations/second over a single Amazon Web Services EC2 instance

So why aren’t in-memory DBs equally fast?

Most are written in C/C++…. but programming language isn’t the only thing to consider

What affects in-memory DB performance?

(1) Complexity of processing commands

→ How many lines of code per command ? What is the computation complexity (e.g. in Redis most commands are O(1))?

(2) Query efficiency

→ Is it limited to blob queries? Can you query a discrete value?

(3) Pipelining

→ Can you send multiple requests at once to get lower latency and less context switches?

What affects in-memory DB performance?

(4) Protocol efficiency

→ How long it takes to parse a request or to serialize a response

(5) TCP overhead

→ Long-lived (connection pool) vs. short-lived connections

What affects in-memory DB performance?

(6) Single-threaded or multi-threaded architecture

→ Lock-free vs. parallel computing

(7) Shared-nothing (the best) vs. shared-something vs. shared-everything

(8) Built-in acceleration components

Myth #2
A single node is not a cluster

The truth:

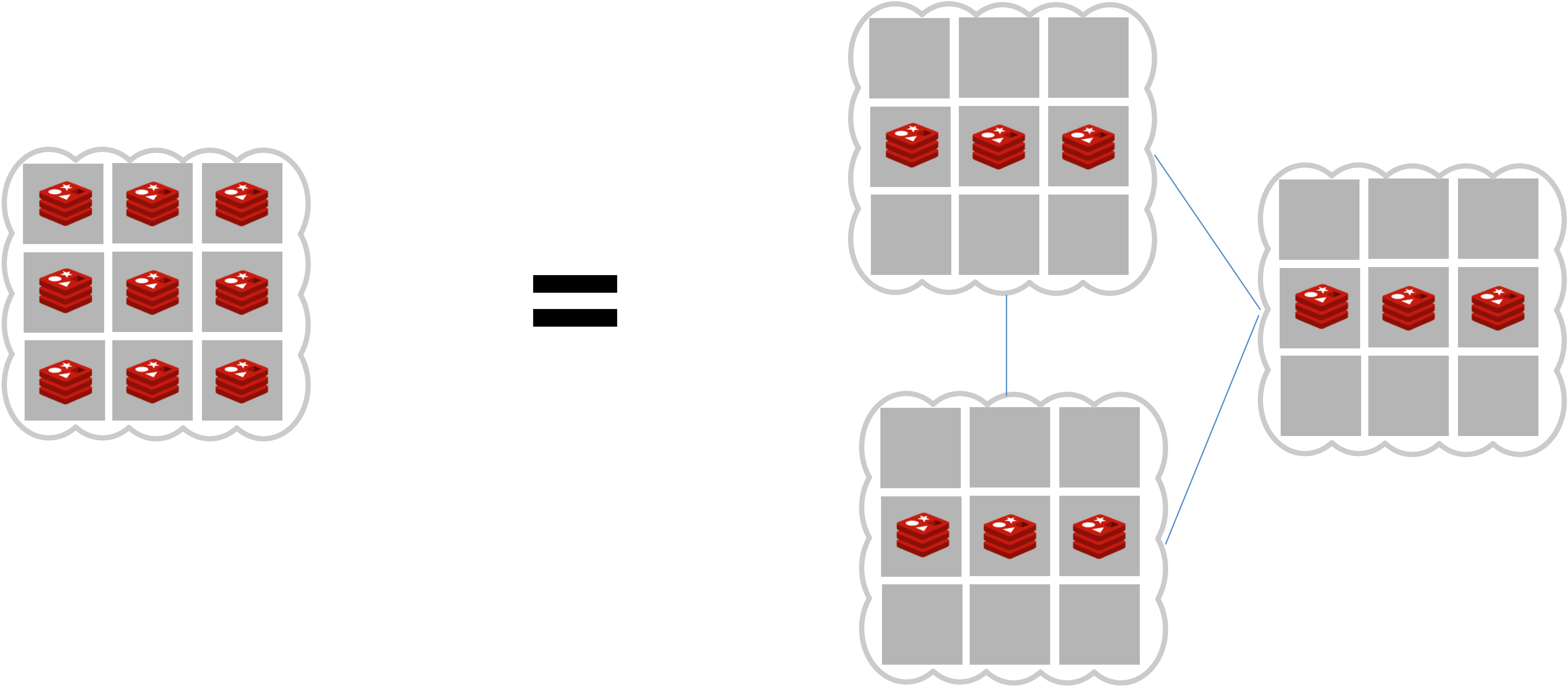
A single node can be a cluster but not a HA cluster

In the new containers/VMs world a cluster is:

A bunch of processes that together look like one big process

A real-world example

A Binary Option platform; 400MB dataset; 1,000,000 ops/sec



A single node, 9 shards cluster

3 nodes, 9 shards cluster

Myth #3

In-memory databases are inconsistent and unreliable

A few facts/questions about consistency

Almost all NoSQL databases (not just in-memory) *ack* the client before commiting to disk

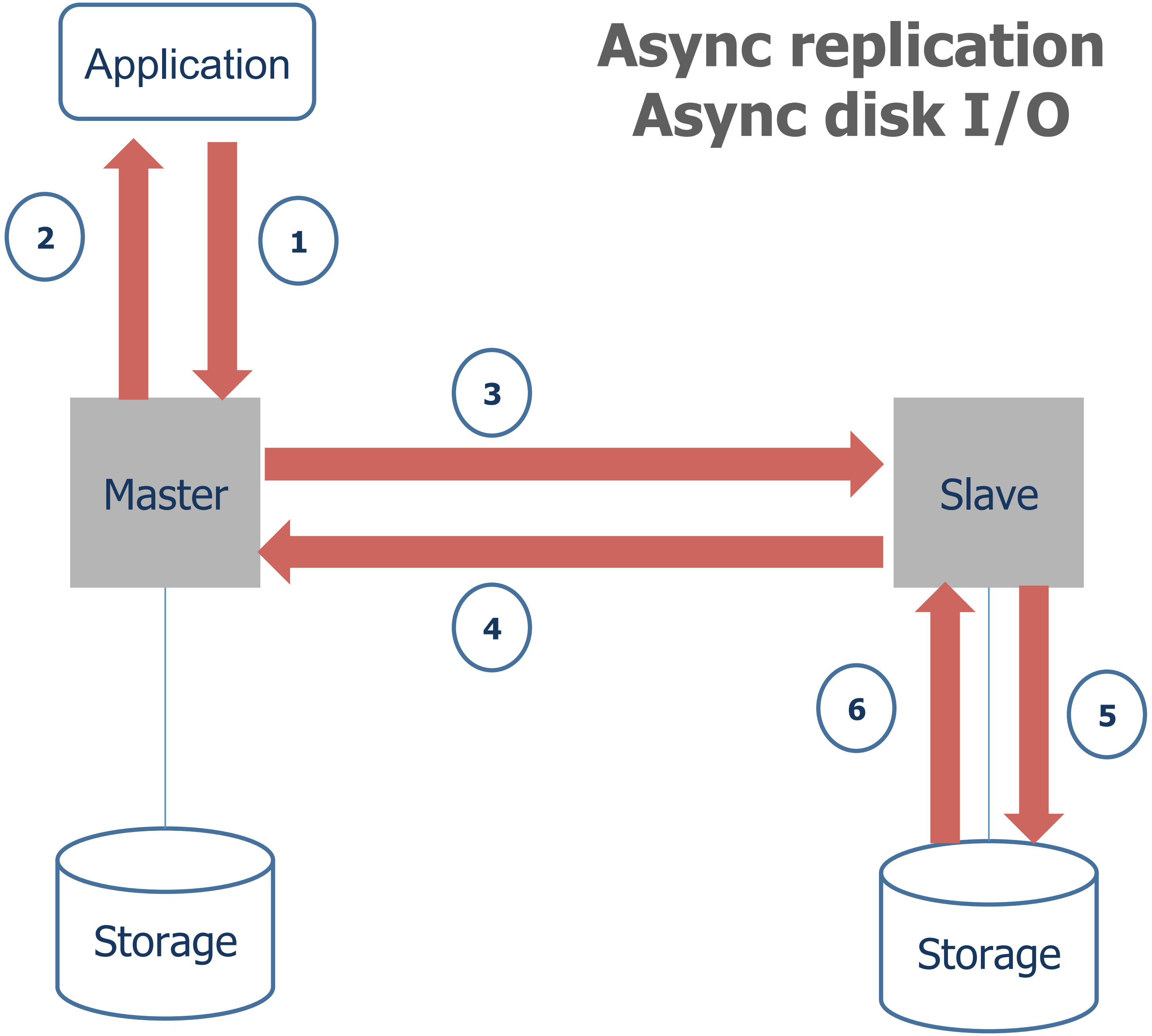
Almost all in-memory databases can commit to disk before they *ack* the client

However, even if you *ack* after everything is committed:

- Is your driver memory buffer persistent and consistent?
- Is your storage system cache persistent and consistent?

Most databases are NOT bulletproof

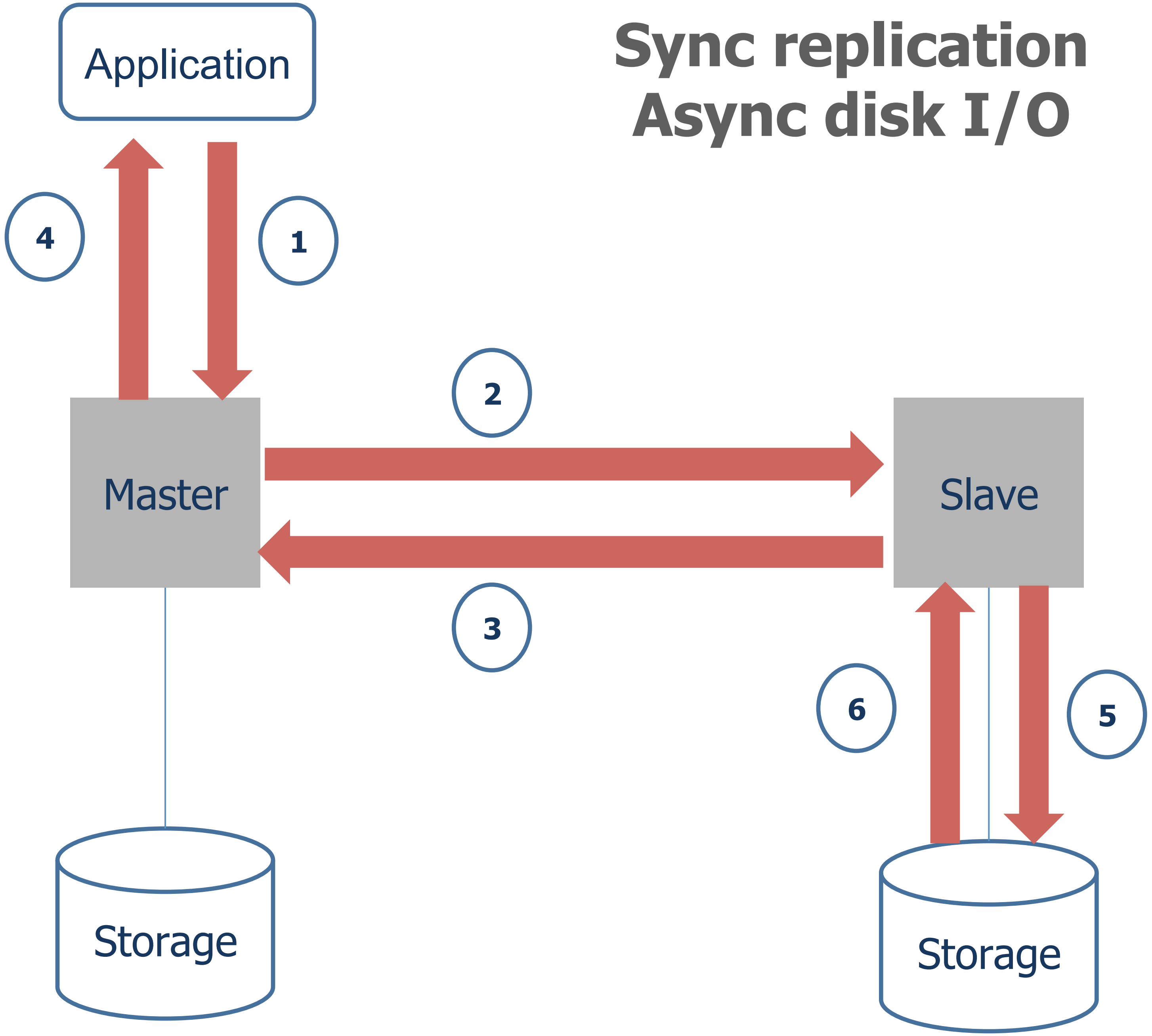
Most in-memory databases are async most of the time



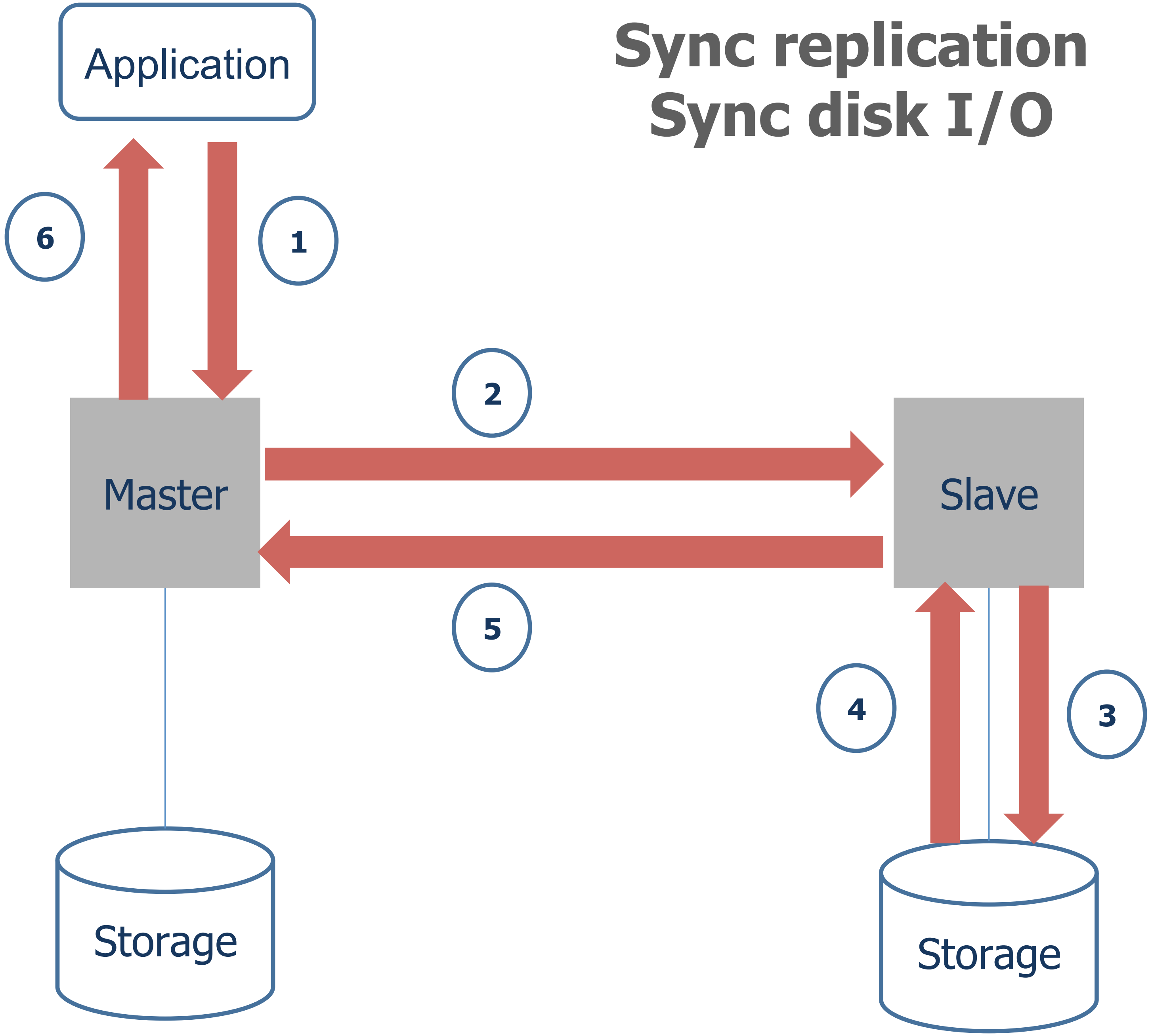
Async replication
Async disk I/O

Some of them can partially sync

Sync replication
Async disk I/O

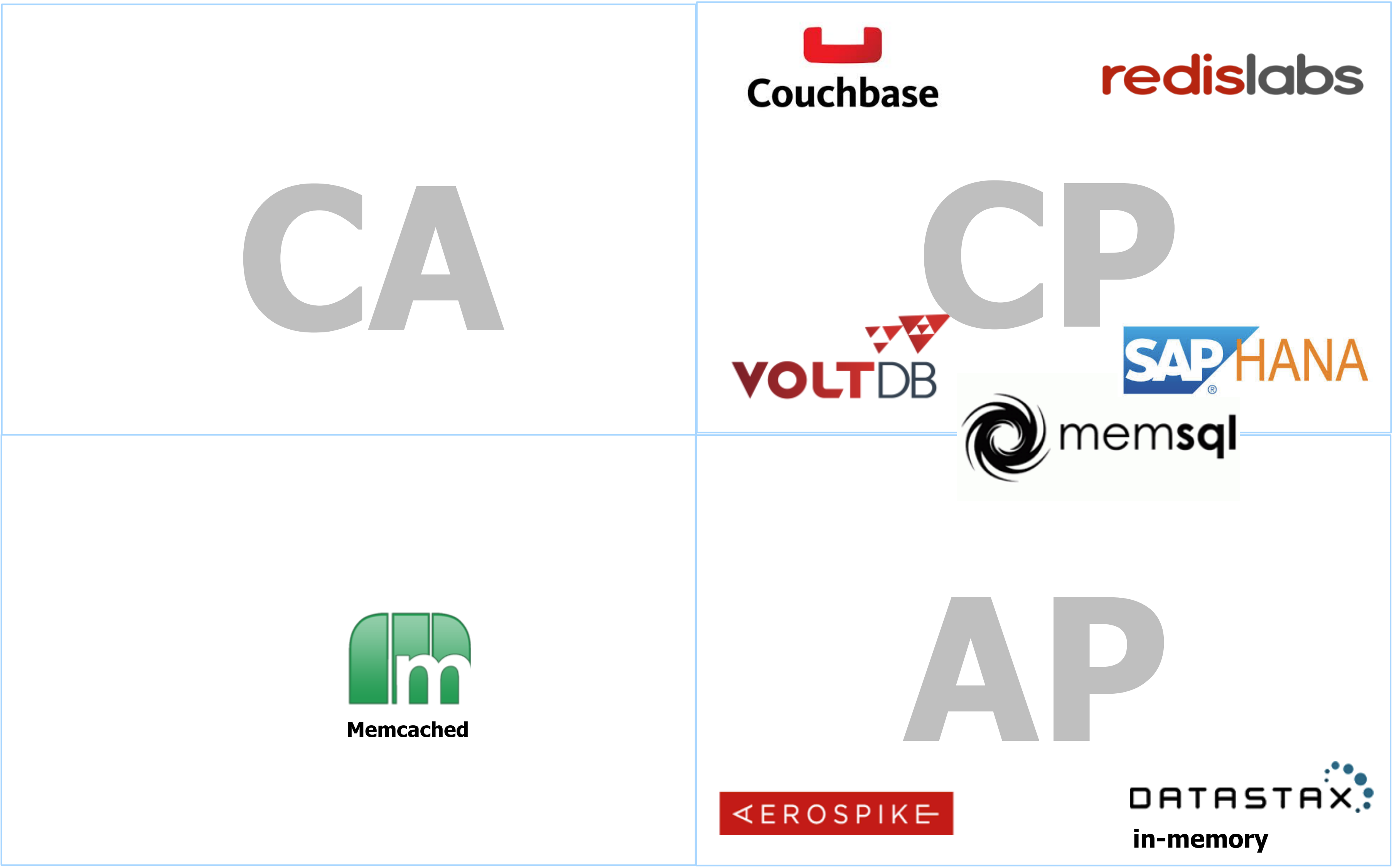


A few of them fully sync



Sync replication
Sync disk I/O

CAP and in-memory databases



CA

CP

Couchbase

redislabs

VOLTDB

SAP HANA

memsql

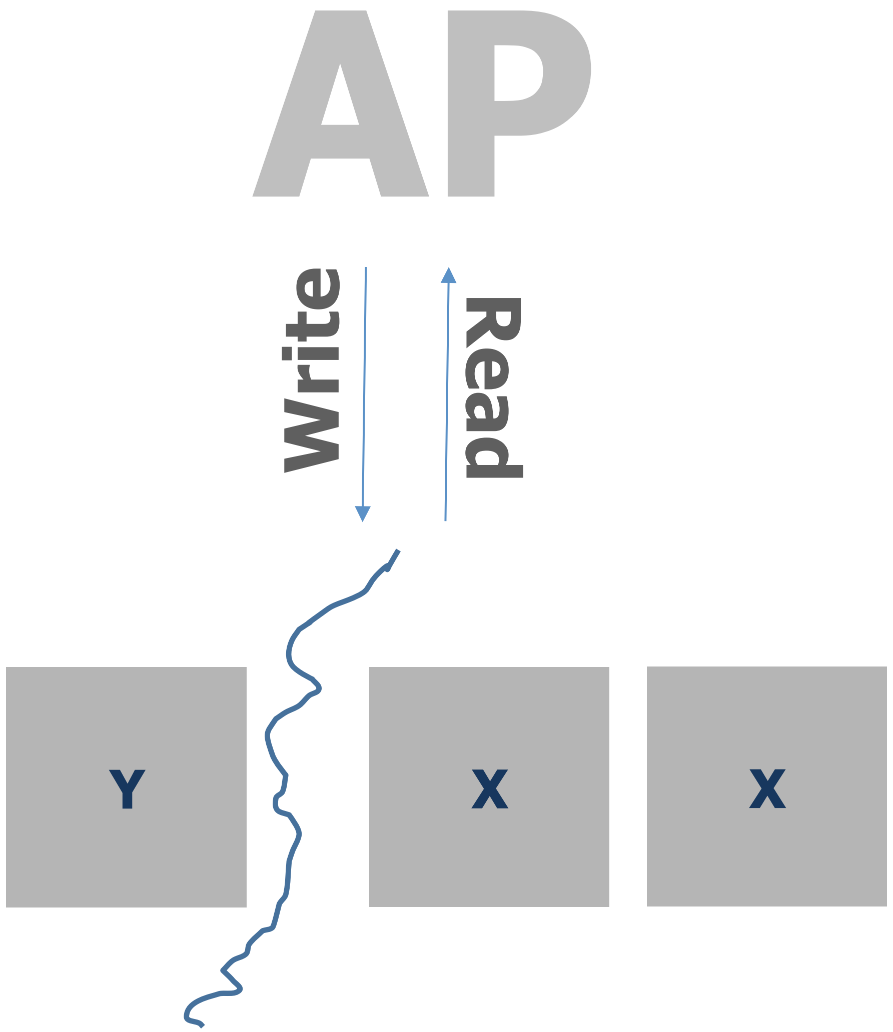
AP

Memcached

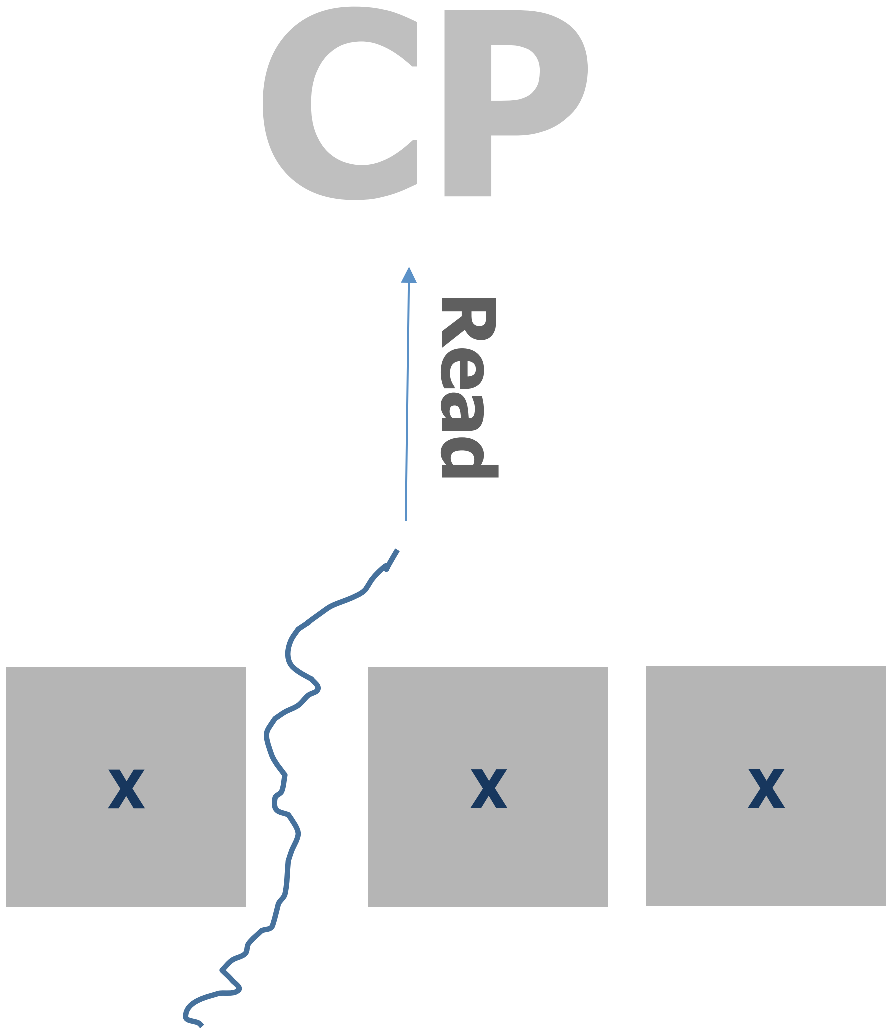
AEROSPIKE

DATASTAX in-memory

Behavior during network splits



Inconsistency



Full-consistency

Are in-memory databases reliable?

Redis Labs facts:

- Provisioned 100s of TBs of RAM
- 500+ node failure events → 1 failure every 2 days
- ~30 complete data-center outages → 1 outage every month
- Users with high availability (HA) features enabled haven't lost a single byte of data

Myth #4

In-memory databases are expensive

Which one costs more?

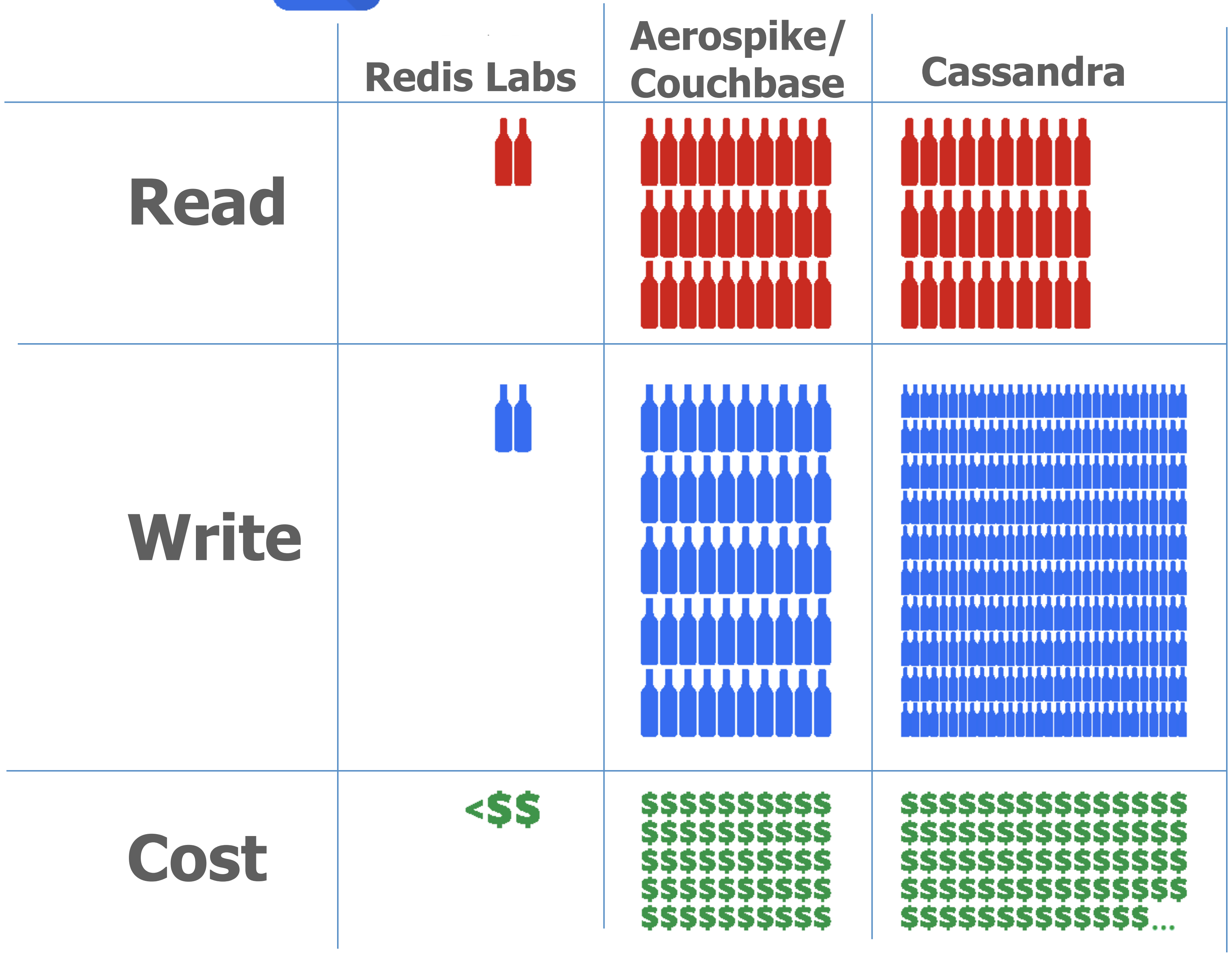
Real-world use case:

- **500+GB**
- **400K writes/sec**
- **1500 reads/sec**
- **37.5KB average object size**

	Others
1.5Gbps	120Gbps
No extra work at app level	Tons of work at app level
6-node cluster	**30+ node cluster**

Which one costs more (2)?

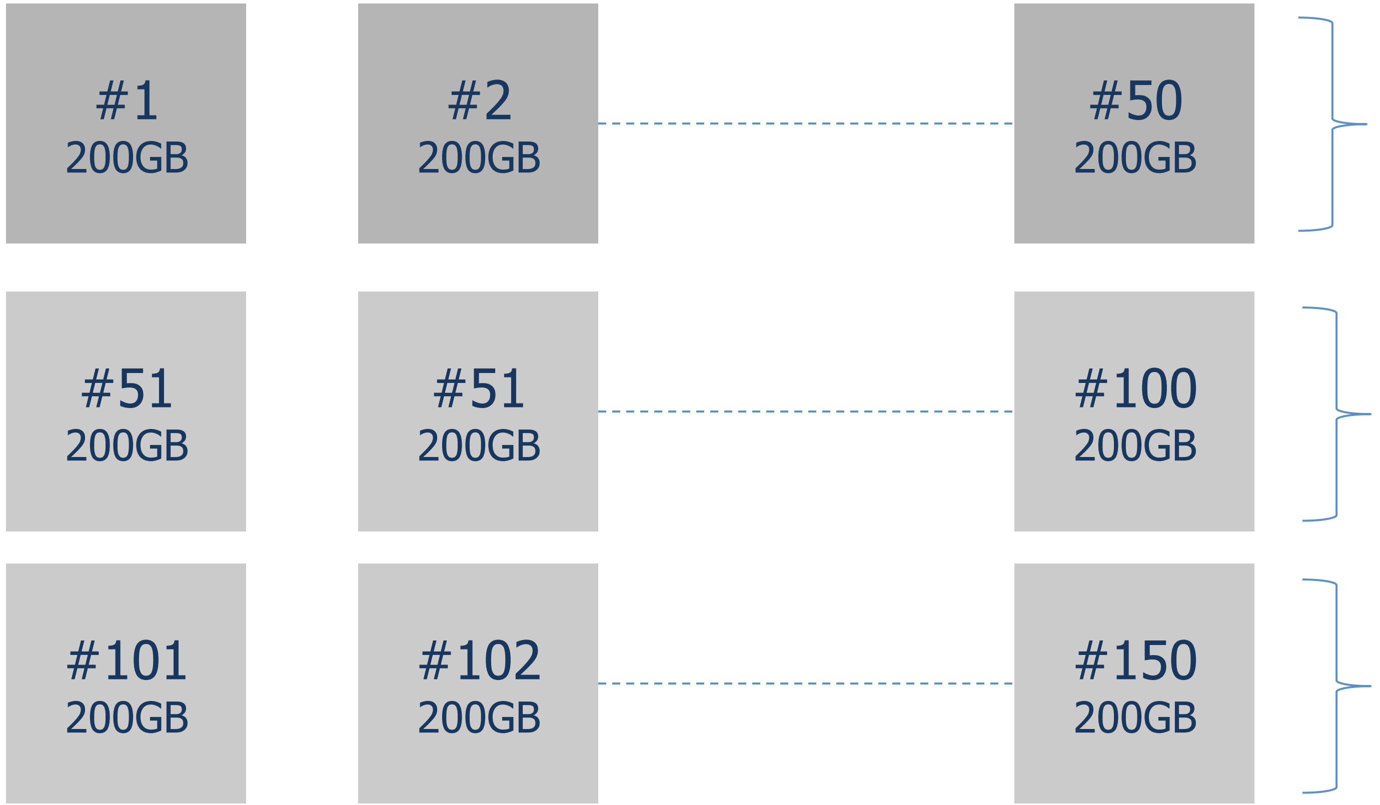
Serving Over 1,000,000 Ops/s
with Google Compute Engine



	Redis Labs	Aerospike/ Couchbase	Cassandra
Read			
Write			
Cost	<$$	\$\$\$\$\$\$\$\$\$\$ \$\$\$\$\$\$\$\$\$\$ \$\$\$\$\$\$\$\$\$\$ \$\$\$\$\$\$\$\$\$\$ \$\$\$\$\$\$\$\$\$\$	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$ \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$ \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$ \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$ \$\$\$\$\$\$\$\$\$\$\$\$\$...

Sometimes in-memory can be very expensive

HA deployment of 10TB in-memory dataset on EC2



#1 200GB

#2 200GB

#50 200GB

50 x r3.8xlarge instances

#51 200GB

#51 200GB

#100 200GB

1st replica for HA

#101 200GB

#102 200GB

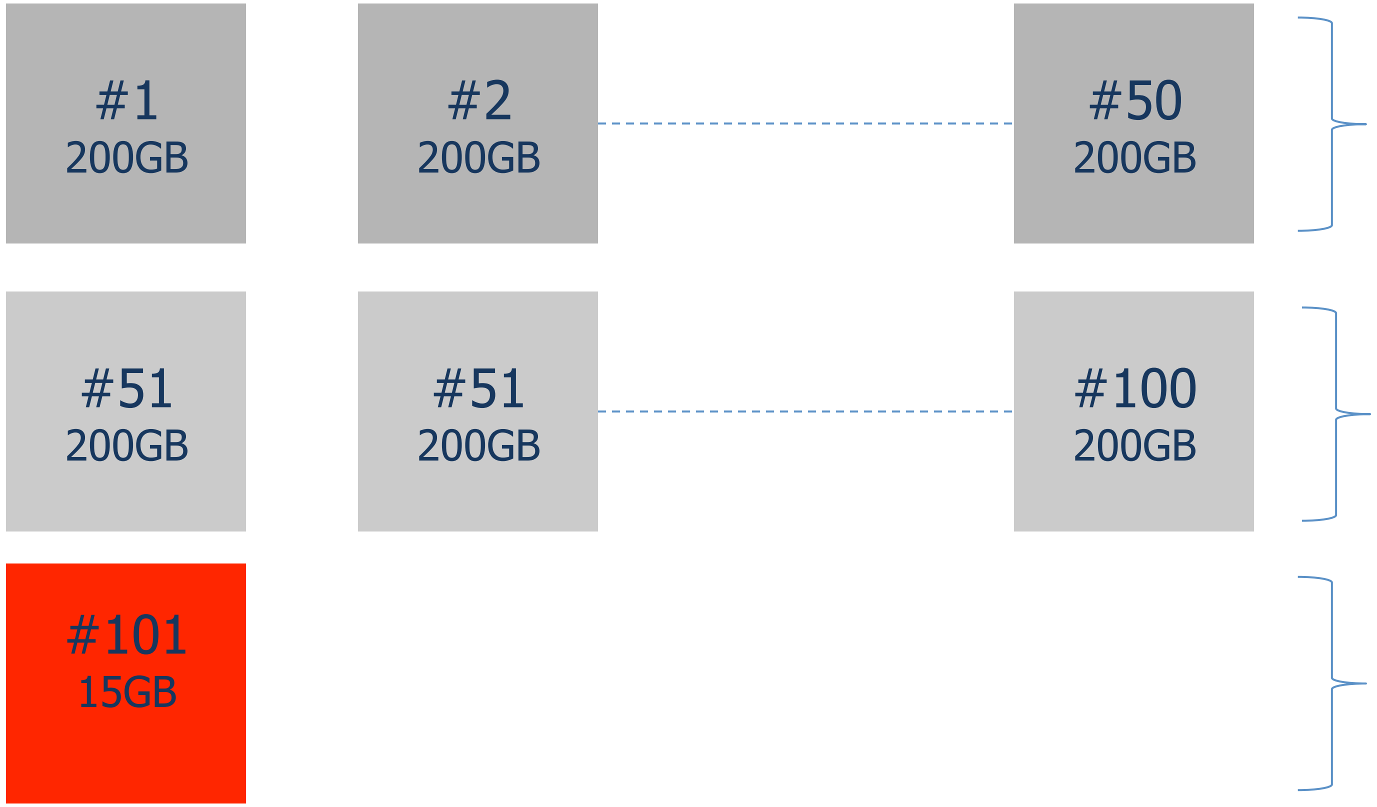
#150 200GB

2nd replica for quorum

Total cost (reserved instances) = $2,132,250/yr

Do we really need 2 replicas?

Efficient HA deployment of 10TB in-memory dataset on EC2



#1 200GB

#2 200GB

#50 200GB

#51 200GB

#51 200GB

#100 200GB

#101 15GB

50 x r3.8xlarge instances

1 replica for HA

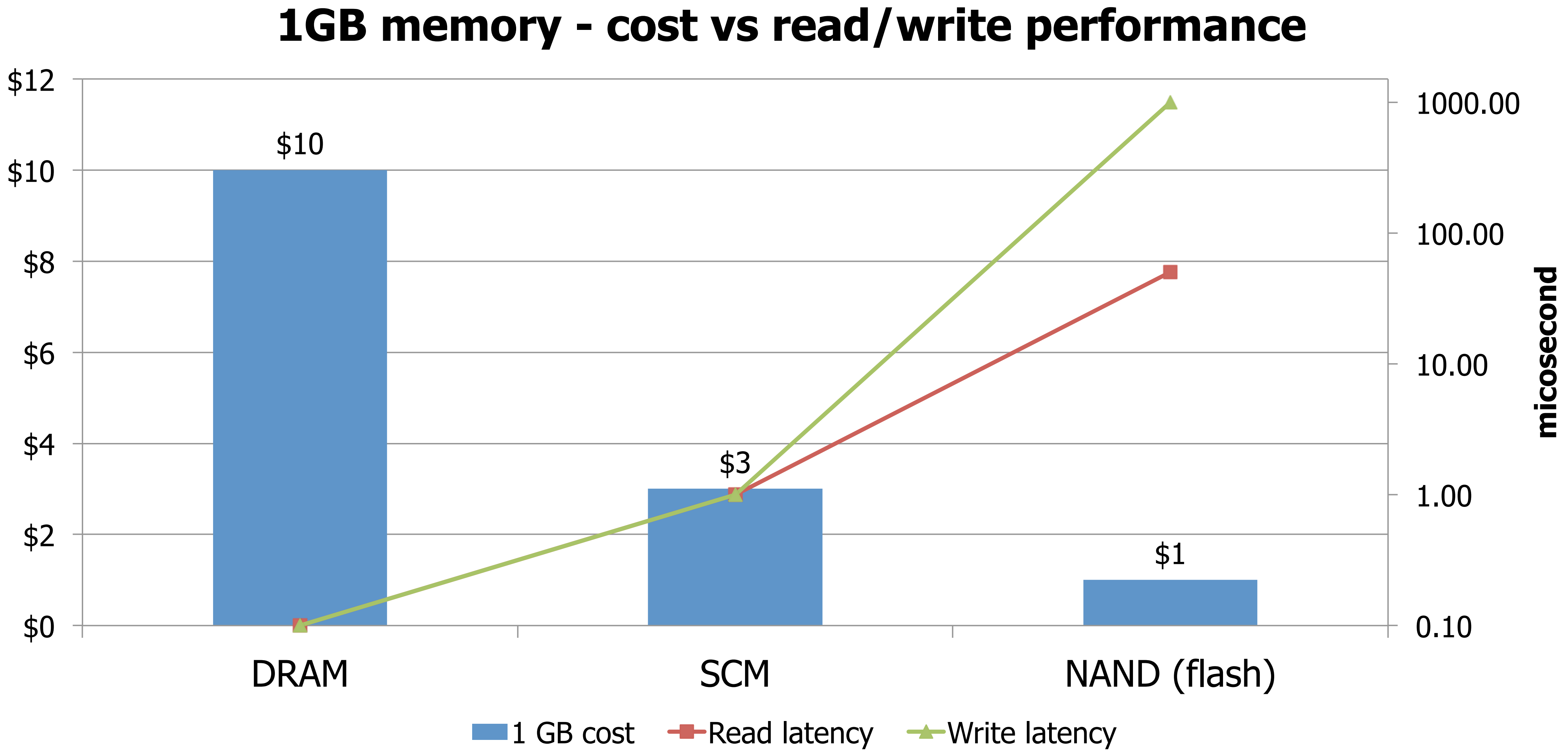
1 quorum server

Total cost (reserved instances) = $1,421,500/yr

Savings = $710,750/yr

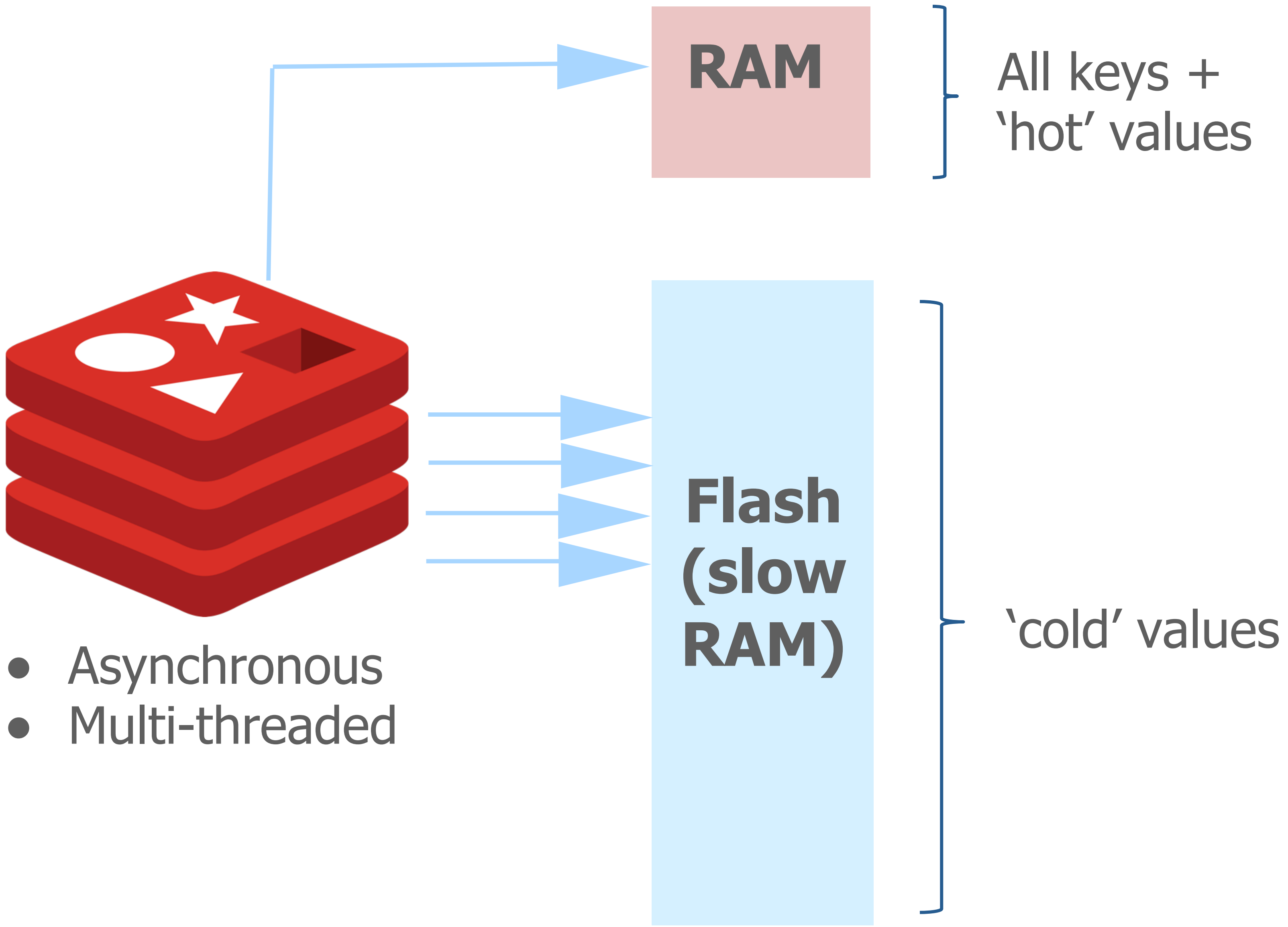
Can we save more?

Price/performance of memory technologies



Redis on Flash

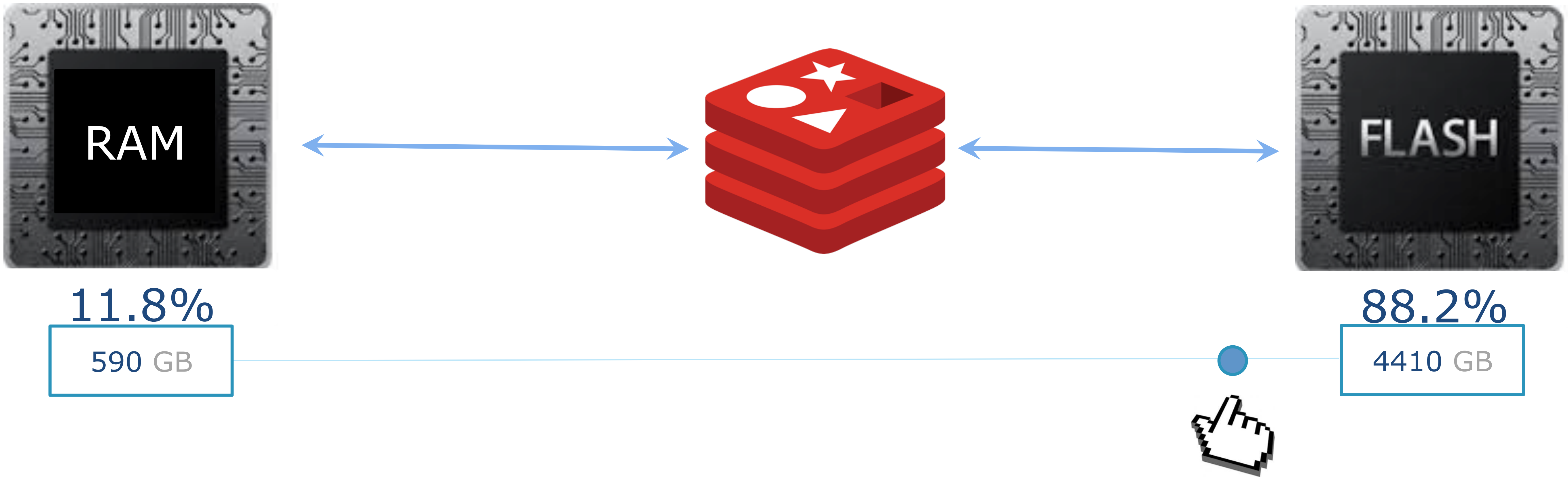
Flash used as a RAM extender and NOT as persistent storage



- Asynchronous
- Multi-threaded

How to achieve optimal price/performance

By dynamically setting RAM/Flash ratio



11.8%

590 GB

88.2%

4410 GB

Single server performance - 10% in RAM / 90% in Flash

RAM Hits Ratio	Ops/Sec	Latency
	Low latency scenarios	
100%	1.35M	1.00 msec
80%	340K	1.07 msec
50%	200K	0.96 msec
20%	160K	1.00 msec
	High throughput scenarios	
100%	2.00M	2.40 msec
80%	671K	6.20 msec
50%	483K	10.00 msec
20%	366K	14.50 msec

**IBM P8 with IBM FlashSystem*

10TB Redis deployment on EC2

	Redis (on RAM) 2 replicas	Redis (on RAM) 1 replicas	Redis on Flash 1 replica
Instance type	r3.8xlarge	r3.8xlarge	i2.8xlarge
# of instances	150	100	10
RAM	30TB	20TB	2TB
Flash	-	-	64TB
Persistent storage (EBS)	150TB	100TB	80TB
1yr costs (reserved instances)	**$2,132,250**	**$1,421,500**	**$318,090**
Yearly savings	-	$710,750	$1,814,160
Savings %	-	33.33%	**85.08%**

Summary

4 myths about in-memory databases busted

All in-memory databases are NOT equally fast

You can create a single node in-memory cluster

In-memory databases can be consistent and reliable

With the right technology, in-memory databases are not expensive

Thank you

Click to get more info about
Redis, Redis Labs, Redis Cloud and RLEC

redislabs