Arno DUBOIS (20226379)

{Distributed shared memory systems}

When scaling is more complicated than you thought

Context

MMORPGs

Ultima Online (1997)

World of Warcraft (2004)

https://www.youtube.com/embed/m-Rz5PTMuaw?enablejsapi=1&mute=1

Star Citizen (?)

Shared Memory

Easy to use, hard to manage

Why don't we:

Why don't we:

- “Just add more memory!”

Why don't we:

- “Just add more memory!”
- “Optimize the server” (it's always sh*t, anyway…)

What is it?

1.

One address space

You can ask for a data “somewhere”, don't need to care where and how

2.

Interconnect

Of course, you need a system to connect all this memory

3.

Consistency

Even with the best tech, you have latency. But what if someone is reading the thing you are writing?

IBM zSeries 800


IBM
IBM

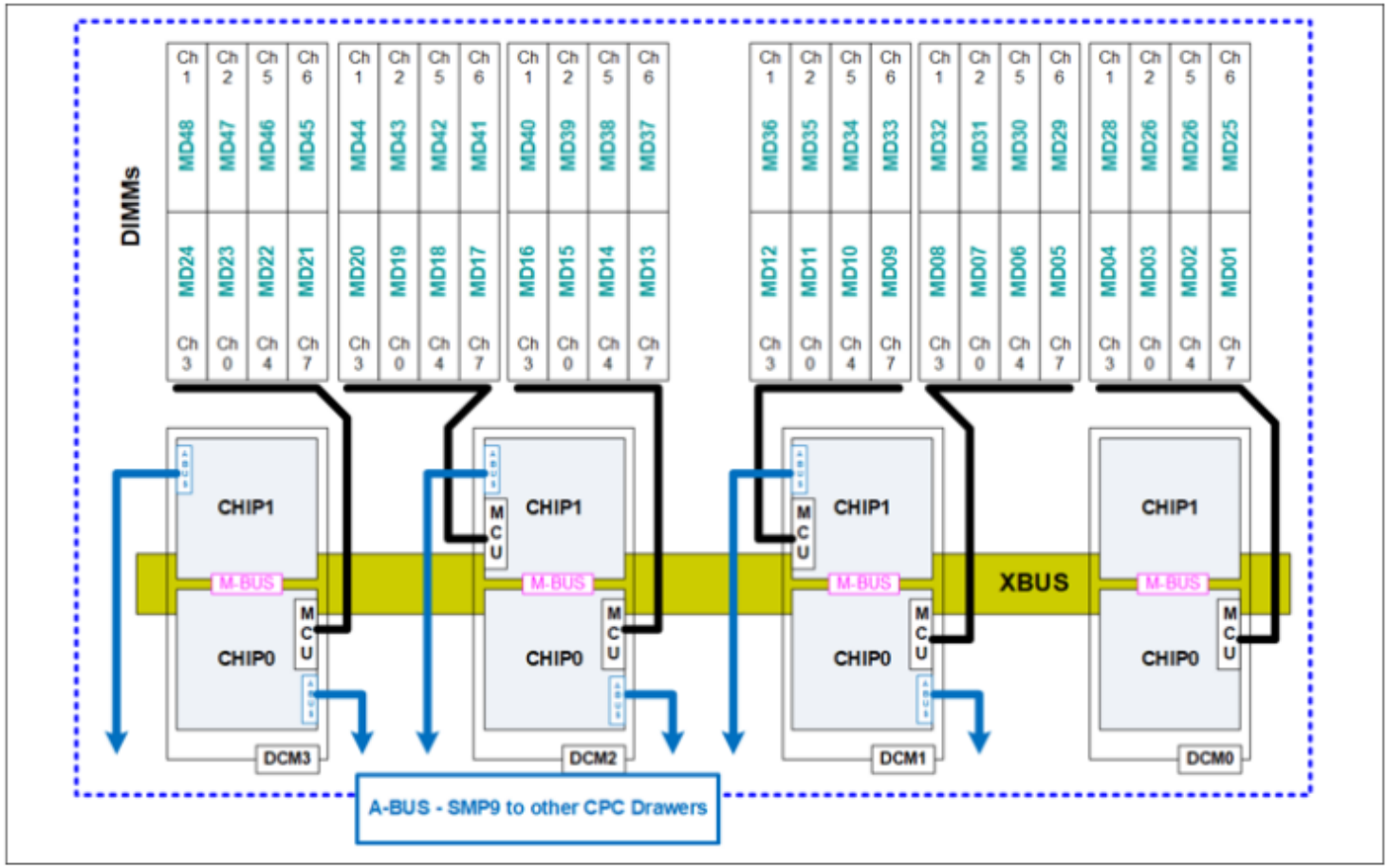
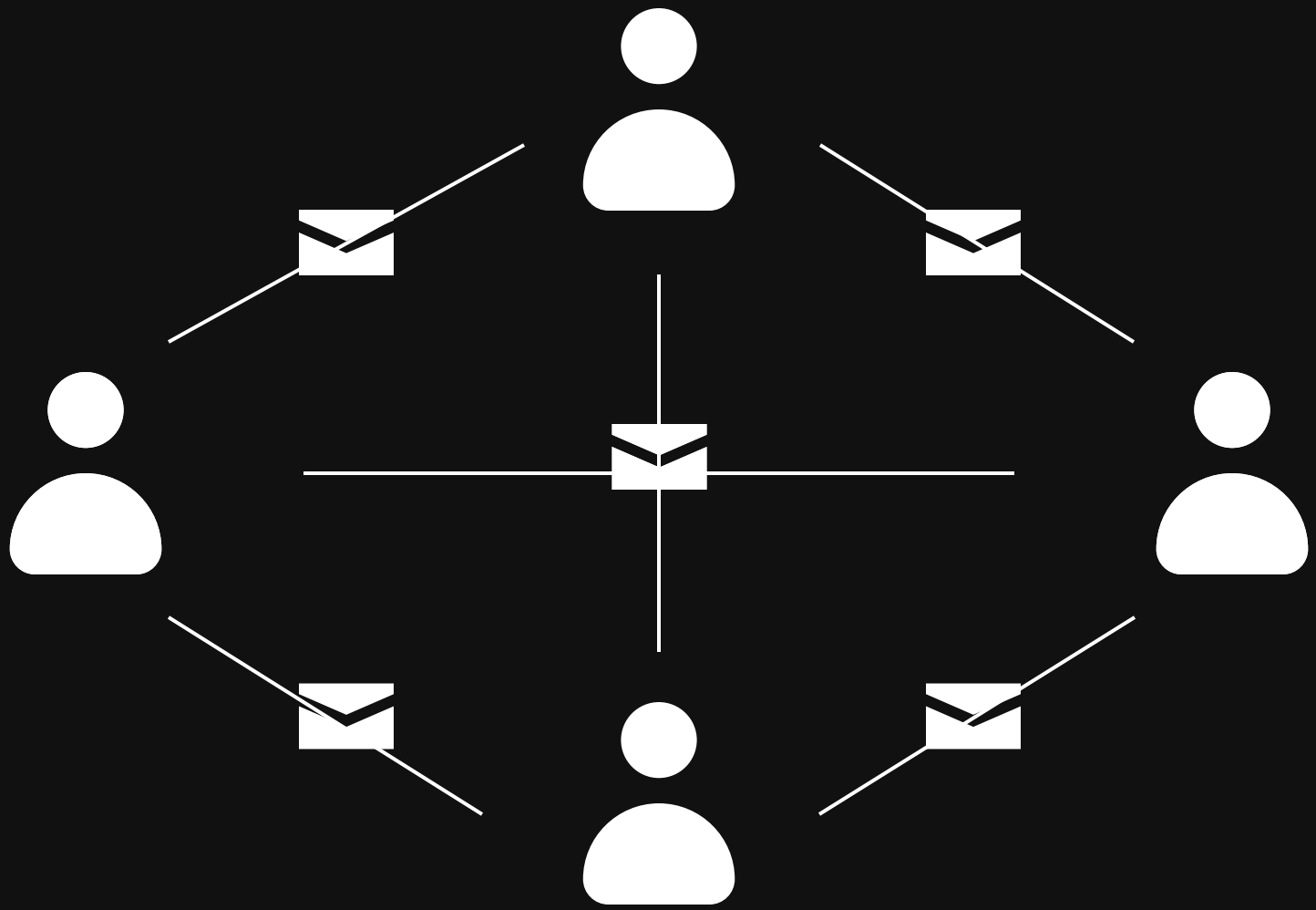


Figure 2-20 CPC drawer memory topology at maximum configuration

IBM SRC2

Messaging

Like letters, but with 0s & 1s



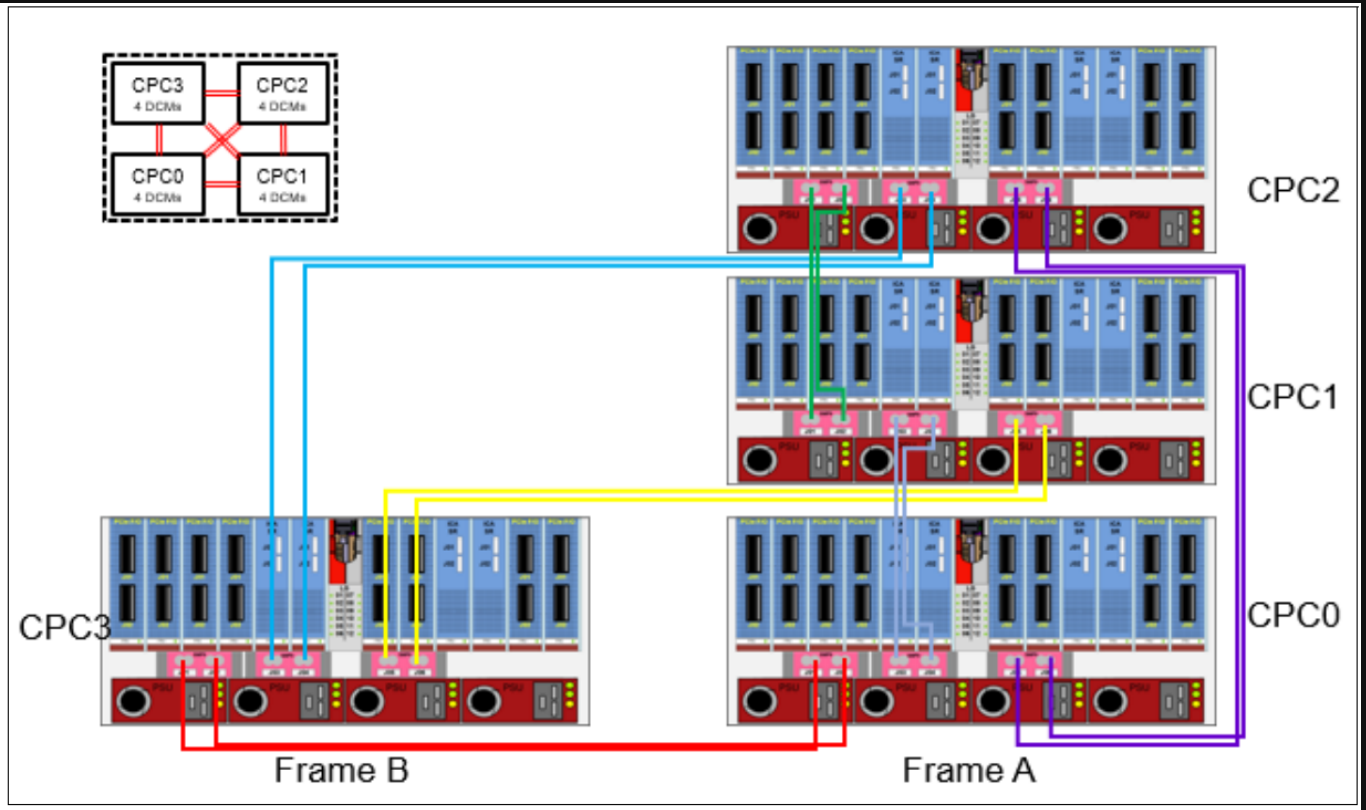
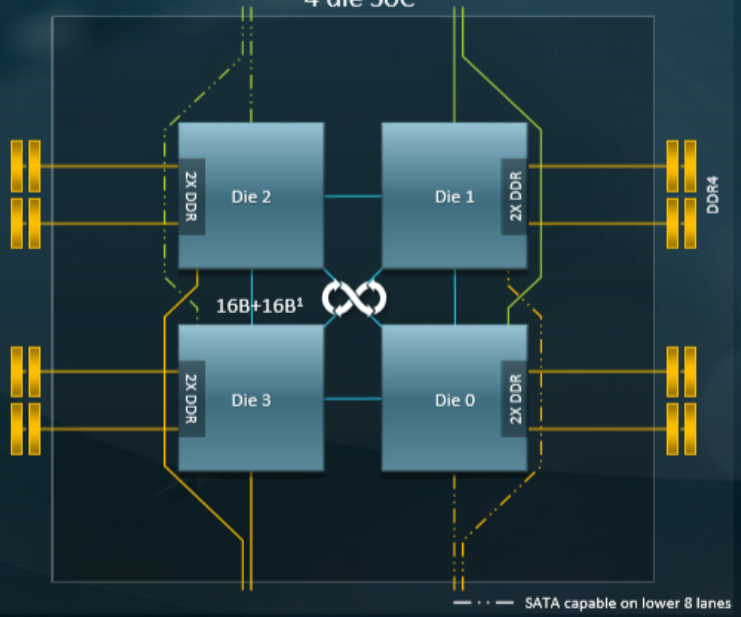


Figure 2-10 Maximum CPC drawer SMP connections (rear view)

EPYC™ 7001 SERIES PROCESSORS
4 die SoC



EPYC™ 7002 SERIES PROCESSORS
9 die SoC

EPYC™ 7002 Series is Platform Compatible with EPYC™ 7001 Series to Optimize Ecosystem Deployment
Performance-optimized Die-to-die Infinity Fabric™

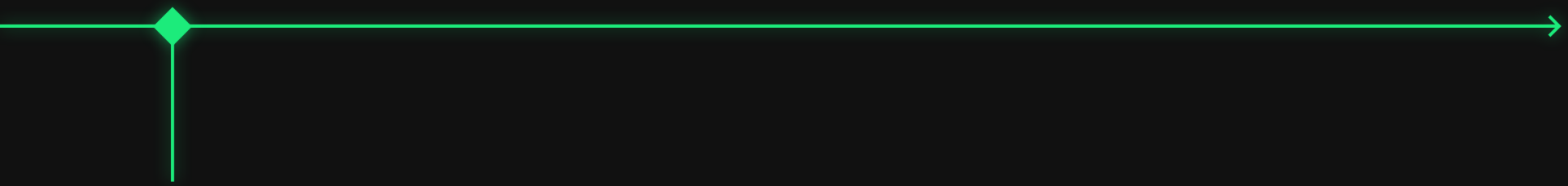
Key:
- Fabric or PCIe
- PCIe
- Die-to-Die Fabric

1: EPYC 7xx1: Die-to-die Infinity Fabric 16B Read + 16B Write / FCLK
2: EPYC 7xx2: Die-to-die Infinity Fabric 32B Read + 16B Write / FCLK

AMD

Consistency

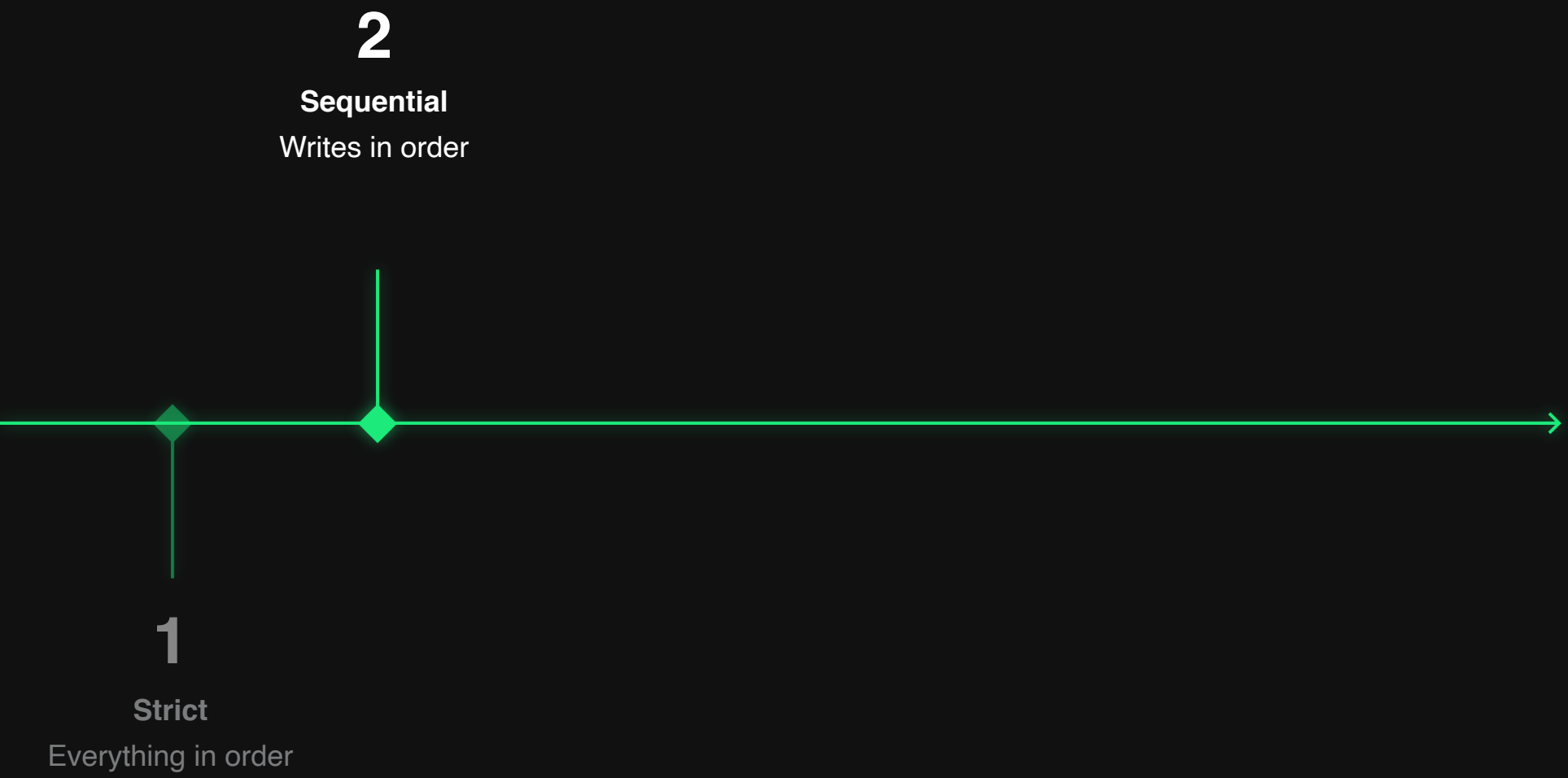
Don't overwrite, don't lose



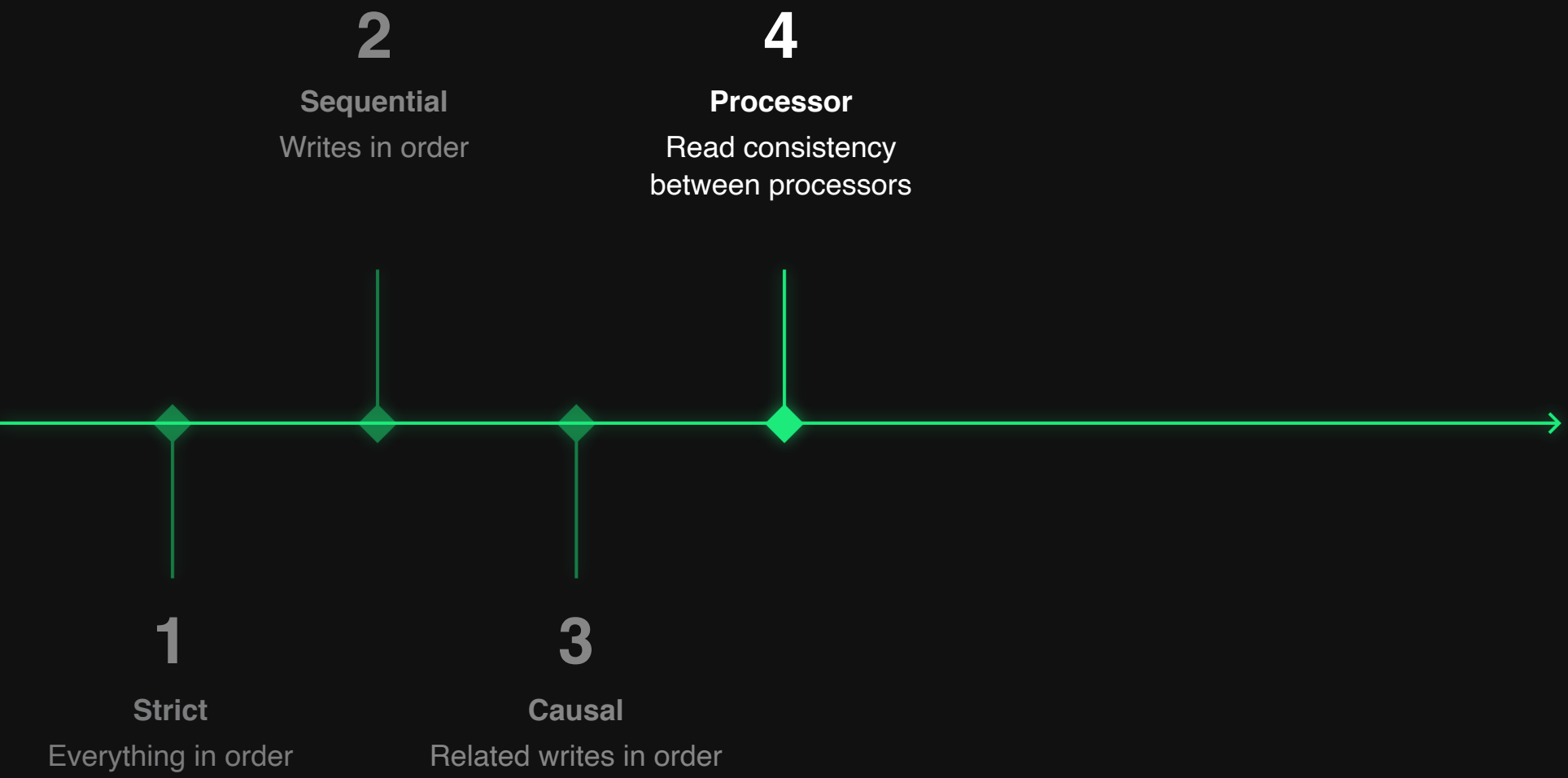
1

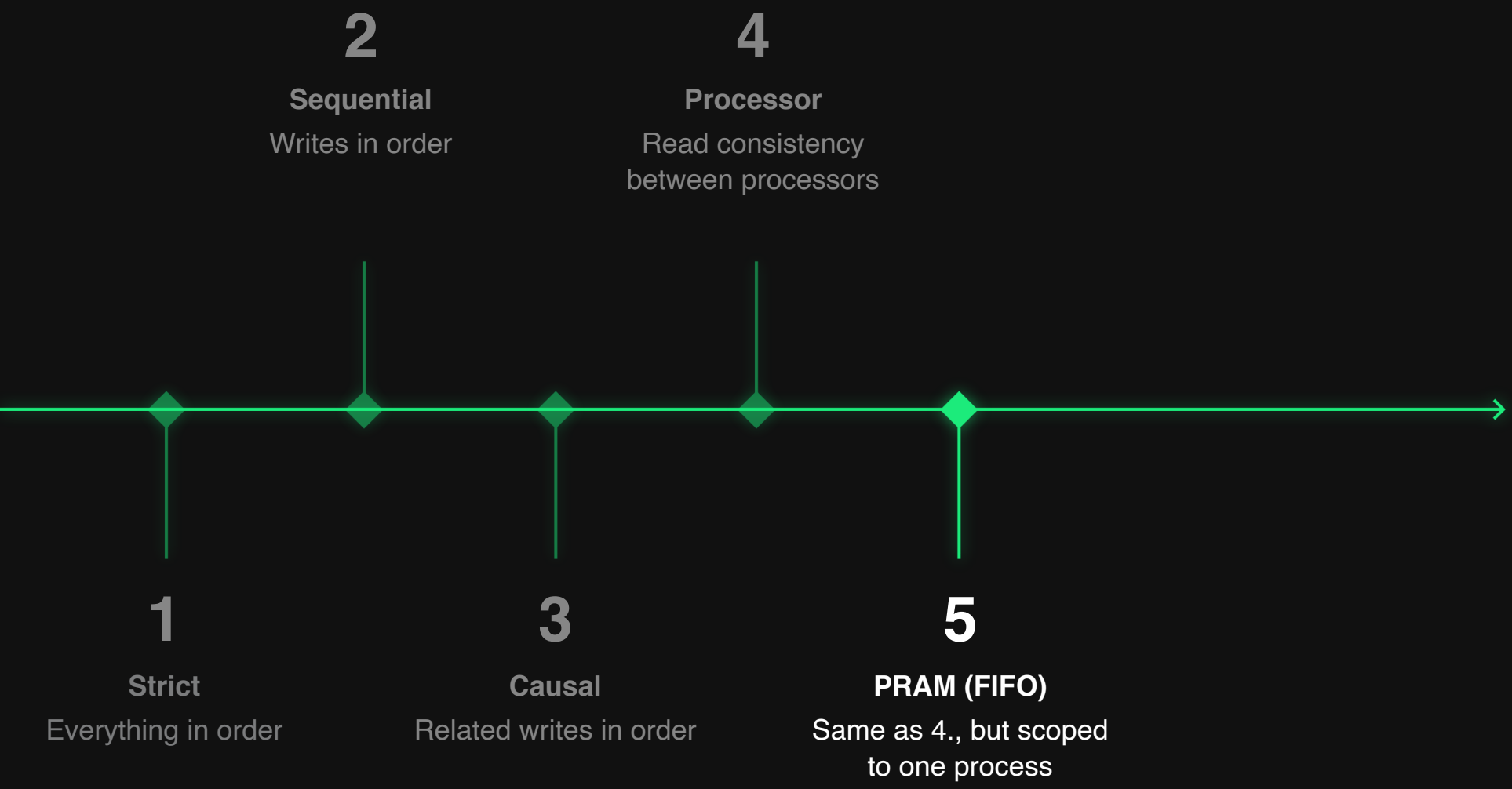
Strict

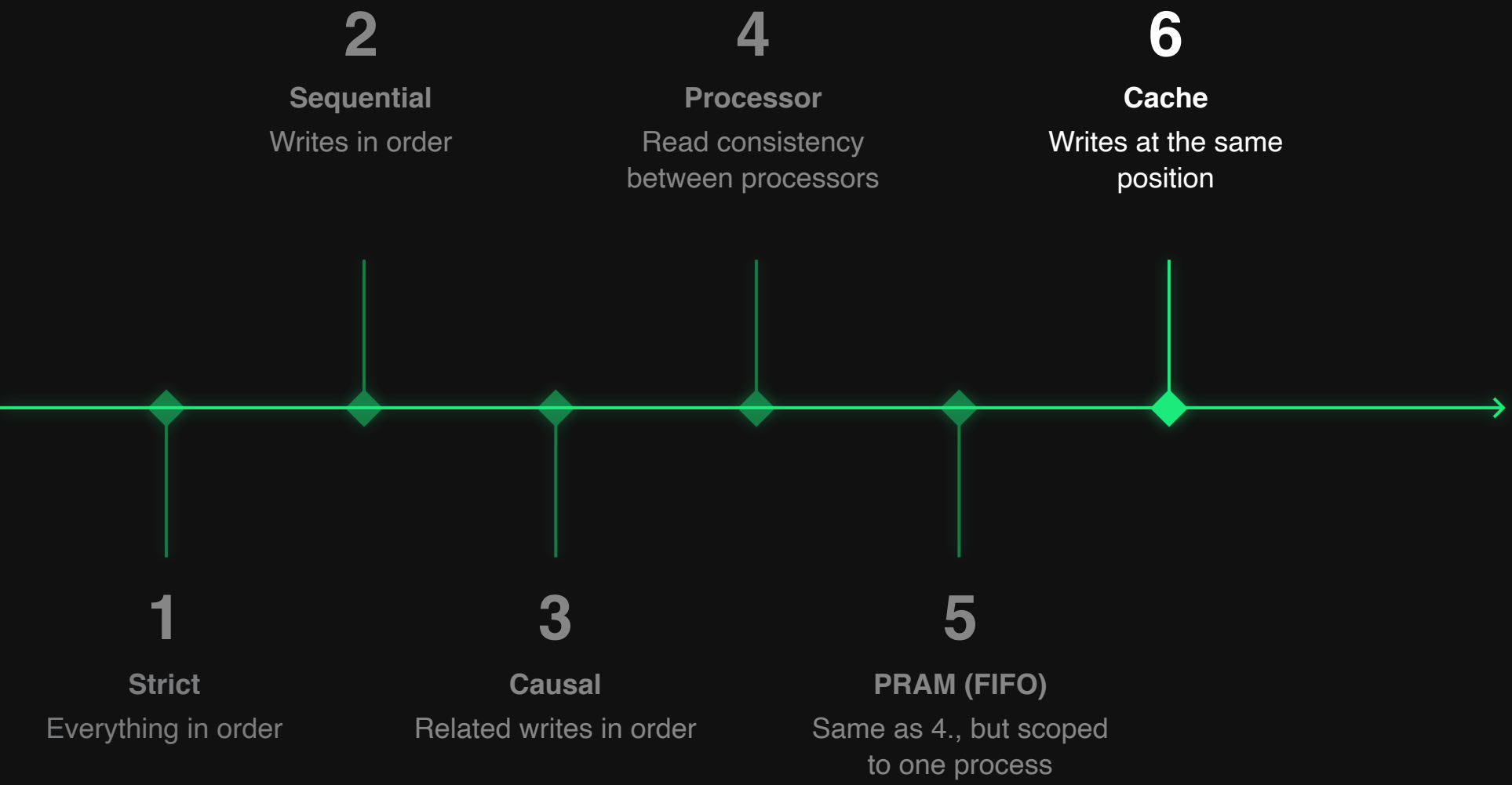
Everything in order

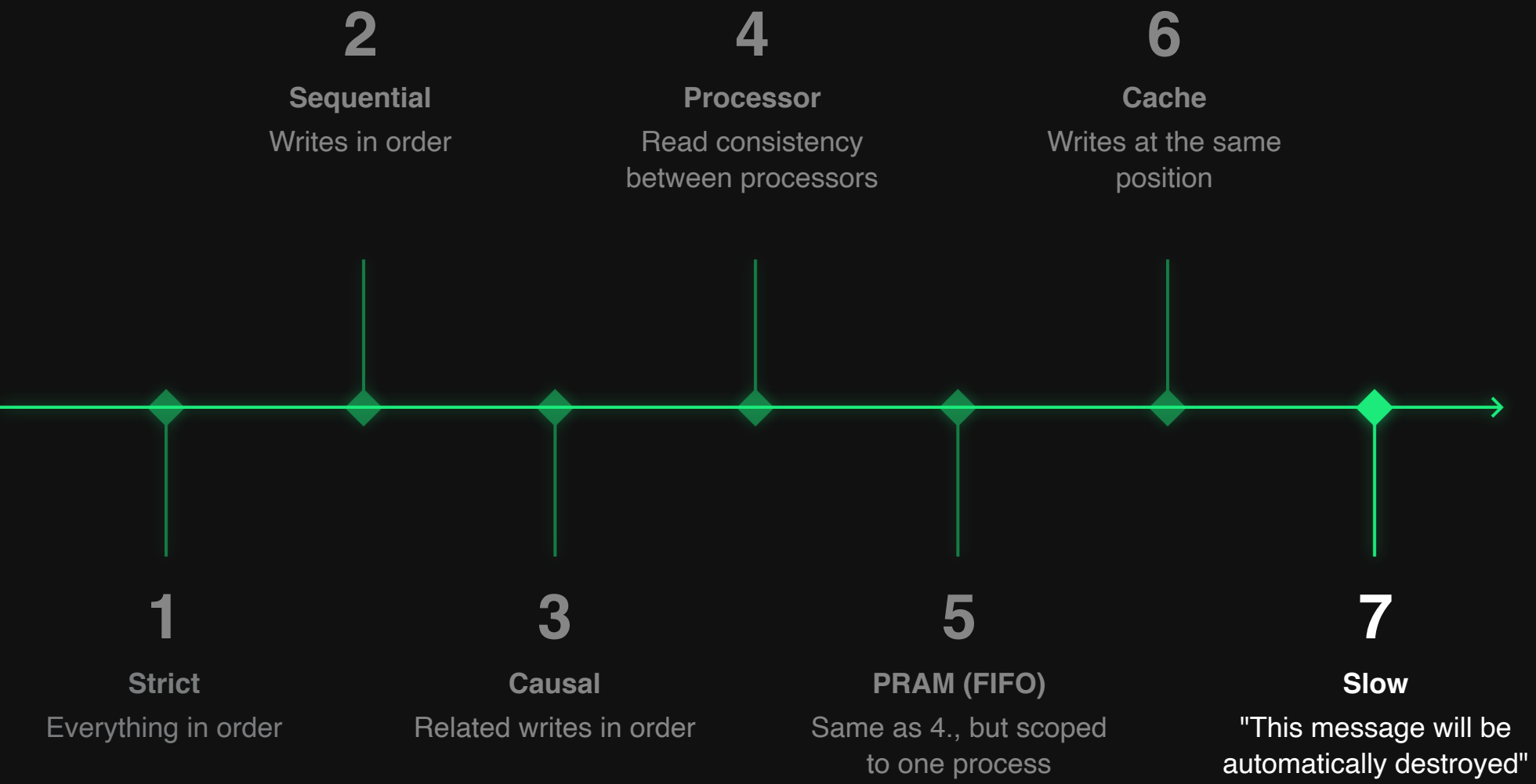

2
Sequential
Writes in order
1
Strict
Everything in order


2
Sequential
Writes in order
1
Strict
Everything in order
3
Causal
Related writes in order


2
Sequential
Writes in order
4
Processor
Read consistency between processors
1
Strict
Everything in order
3
Causal
Related writes in order


2
Sequential
Writes in order
4
Processor
Read consistency between processors
1
Strict
Everything in order
3
Causal
Related writes in order
5
PRAM (FIFO)
Same as 4., but scoped to one process


2
Sequential
Writes in order
4
Processor
Read consistency between processors
6
Cache
Writes at the same position
1
Strict
Everything in order
3
Causal
Related writes in order
5
PRAM (FIFO)
Same as 4., but scoped to one process


2
Sequential
Writes in order
4
Processor
Read consistency between processors
6
Cache
Writes at the same position
1
Strict
Everything in order
3
Causal
Related writes in order
5
PRAM (FIFO)
Same as 4., but scoped to one process
7
Slow
"This message will be automatically destroyed"

Session consistency

1.

Monotonic read

When reading, you will never get an older value

2.

Monotonic write

A write should be finished before the execution of the next one

3.

Read-your-writes

If you read after a write, you should always have the latest value

4.

Writes-follows-reads

When reading, you should not get a value written after it

Session consistency

Go to menti.com and use code:

3354 7336

M

W

w

ol

Go to menti.com and use code: **3354 7336**

Session consistency

1.

Monotonic read

When reading, you will never get an older value

2.

Monotonic write

A write should be finished before the execution of the next one

3.

Read-your-writes

If you read after a write, you should always have the latest value

4.

Writes-follows-reads

When reading, you should not get a value written after it

You're all right!

Take home

1.

Shared memory

To scale beyond the stars, without modifications to your code!

2.

Messages

Like mails, but with data between the systems

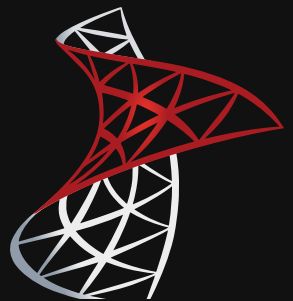
3.

Consistency models

Strict and session models (the latter being more common when using software)

Shared storage systems

Because `{all you want, everything}` is a DB



SQL databases

All the same... Are they?

Meet the Team

Hadoop

COO

Hive

S3

Cassandra

NoSQL DB

Hadoop distribution

Hadoop distribution

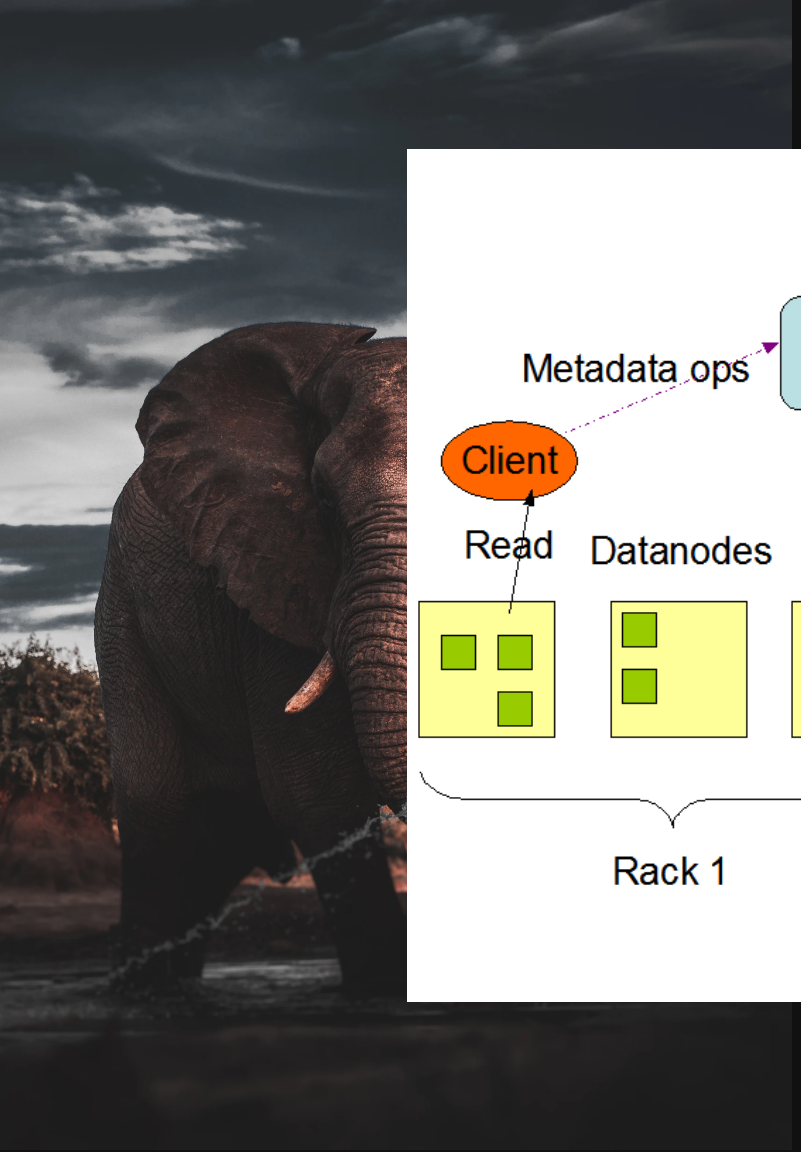
- NameNodes

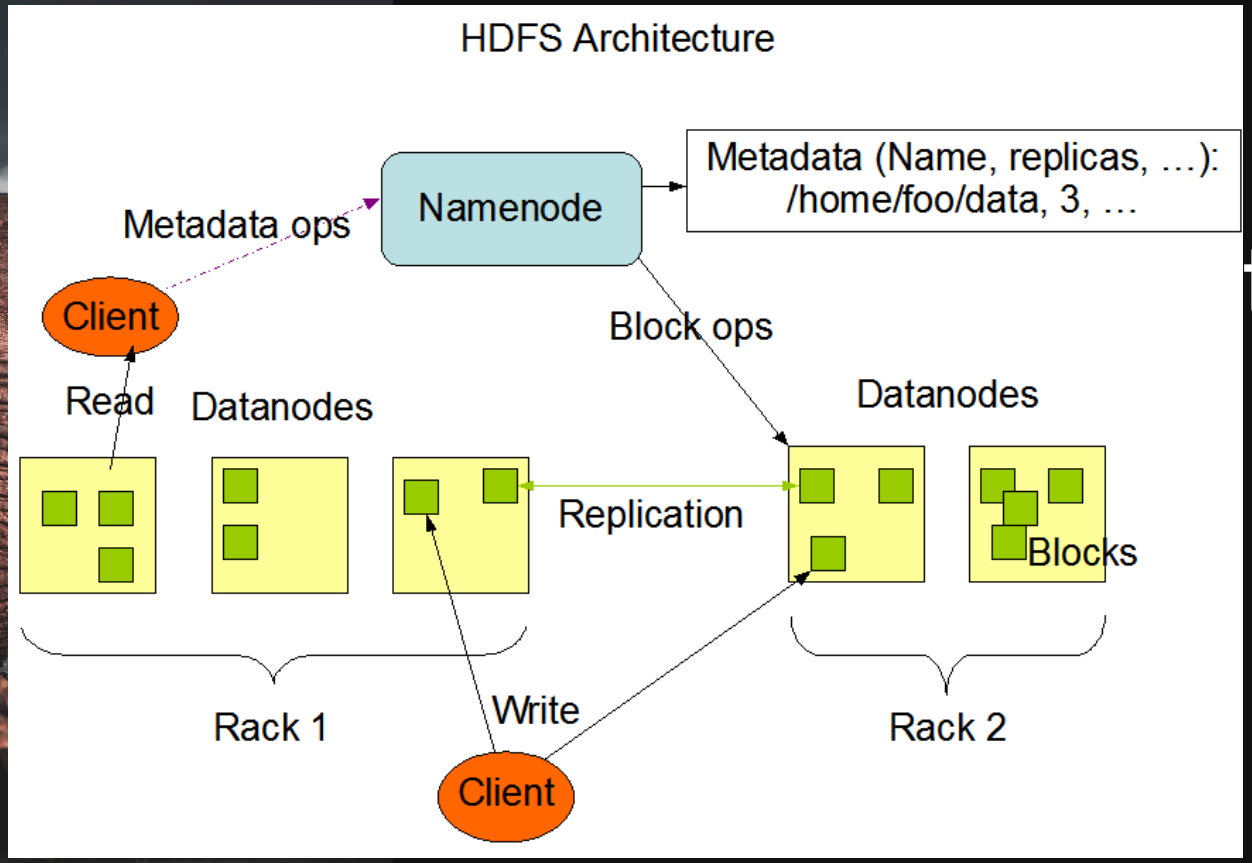
Hadoop distribution

- NameNodes
- DataNodes

Hadoop distribution

- NameNodes
- DataNodes
- Write-once-read-many




HDFS Architecture
Namenode
Metadata (Name, replicas, …):
/home/foo/data, 3, …
Metadata ops
Client
Block ops
Read
Datanodes
Datanodes
Replication
Blocks
Rack 1
Write
Rack 2
Client

tion

Hive

Hive

- Key-Value store

Hive

- Key-Value store
- Custom consistency

Hive

- Key-Value store
- Custom consistency
- RAFT quorum protocol

What is RAFT

```
Election safety: at most one leader can be elected in a given term.
Leader append-only: a leader can only append new entries to its logs
(it can neither overwrite nor delete entries).
Log matching: if two logs contain an entry with the same index and
term, then the logs are identical in all entries up through the given
index.
Leader completeness: if a log entry is committed in a given term then
it will be present in the logs of the leaders since this term
State machine safety: if a server has applied a particular log entry
to its state machine, then no other server may apply a different
command for the same log.
```

Hive

- Key-Value store
- Custom consistency
- RAFT quorum protocol

Hive

- Key-Value store
- Custom consistency
- RAFT quorum protocol
- Sharding

Cassandra

Cassandra

- Key-Value store

Cassandra

- Key-Value store
- No quorum

Cassandra

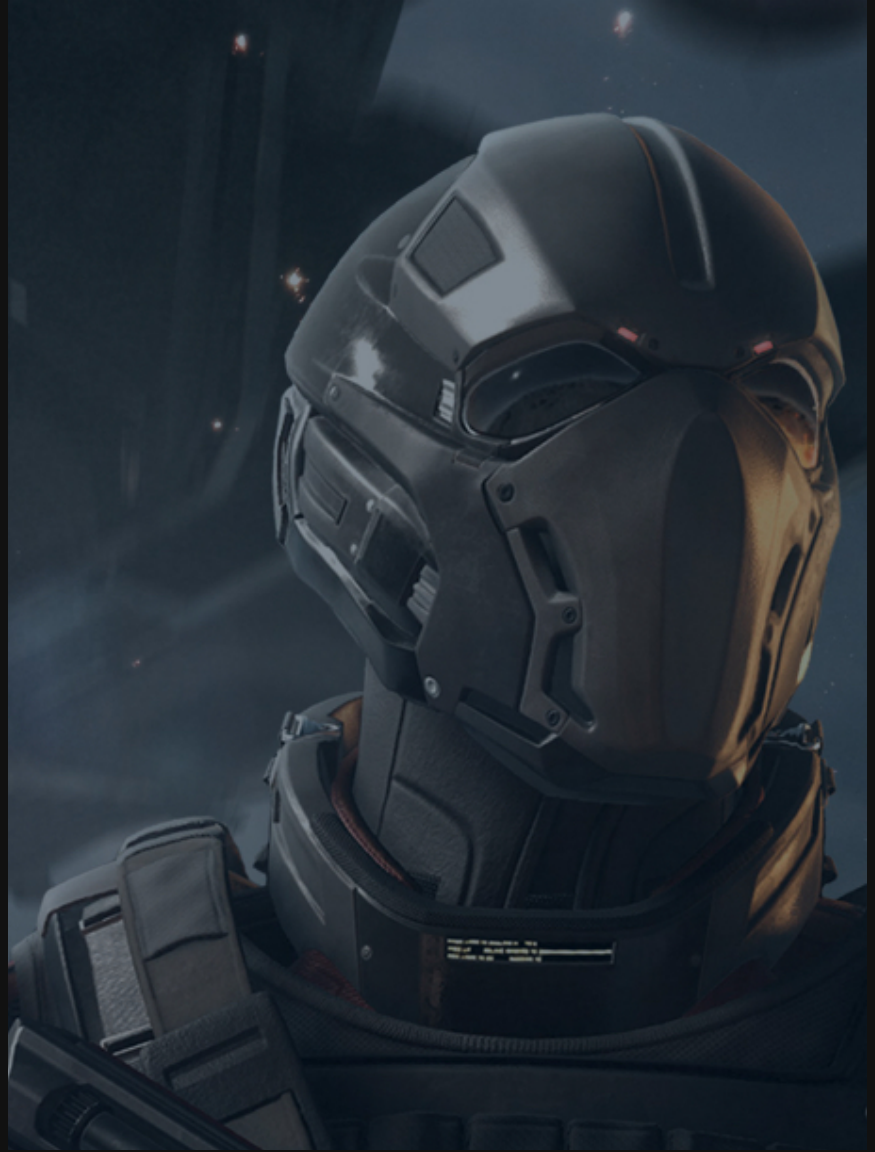
- Key-Value store
- No quorum
- Custom consistency

Cassandra

- Key-Value store
- No quorum
- Custom consistency
- Position-aware

Use case

Star Citizen



https://www.youtube.com/embed/rvKS70FDZV8?mute=1&enablejsapi=1

Server meshing

Classic MMORPGs	Star Citizen
1 shard is 1 server	1 shard is connected to other servers
1 server is the owner	All the data are distributed everywhere
You connect to the server	You connect to the "replication layer"
1 shard is one server, still	True meshing

To what extent will it work?

$well$

And if there is a problem?

$$problems\ doesn't\ exists$$

Comparison with EVE Online

...the end

See you on the other side!

https://robertsspaceindustries.com/roadmap/release-view

Take home

1.

Databases distribution

How database are splitting themselves, their limits (SQL vs. others)

2.

How to create a distributed system

Those are real-world examples, so we know they works

3.

Do you really need one?

Sometime, having a big server with good optimizations is all you need

Media credits

https://www.reddit.com/r/gaming/comments/batdve/an_original_world_of_warcraft_blade_server/
https://www.youtube.com/m-Rz5PTMuaw
https://en.wikipedia.org/wiki/IBM_Z
https://www.blosc.org/posts/blosc2-meets-rome/
https://www.youtube.com/rvKS70FDZV8

References

https://en.wikipedia.org/wiki/Distributed_shared_memory
https://en.wikipedia.org/wiki/IBM_Z
https://www.redbooks.ibm.com/abstracts/sg248951.html
https://www.ibm.com/support/pages/system/files/inline-files/IBM%20Shared%20Memory%20Communications%20Version%202_2.pdf
https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html
https://papers.s3.fr-par.scw.cloud/hive.pdf
https://robertsspaceindustries.com/comm-link/transmission/18397-Server-Meshing-And-Persistent-Streaming-Q-A
https://www.eveonline.com/news/view/a-history-of-eve-database-server-hardware