BlowFish:
Dynamic Storage-Performance Tradeoff in Data Stores

Anurag Khandelwal, Rachit Agarwal, Ion Stoica


High-Throughput Data Stores

High-Throughput Data Stores

Key-Value Stores

BigTable

amazon DynamoDB

redis

High-Throughput Data Stores

Key-Value Stores

BigTable

amazon DynamoDB

redis

NoSQL Stores

HYPERTABLE

High-Throughput Data Stores

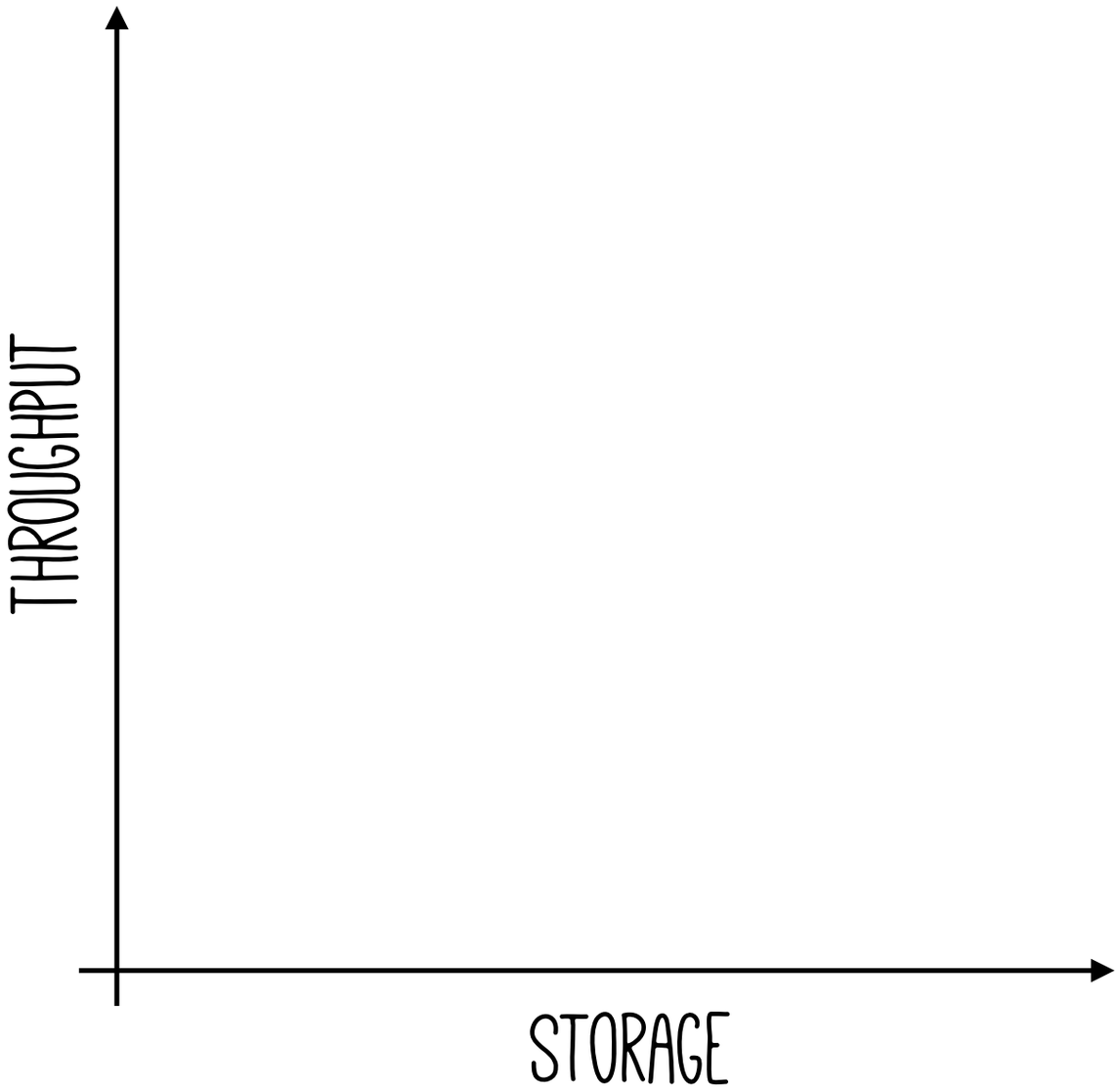
Key-Value Stores

NoSQL Stores

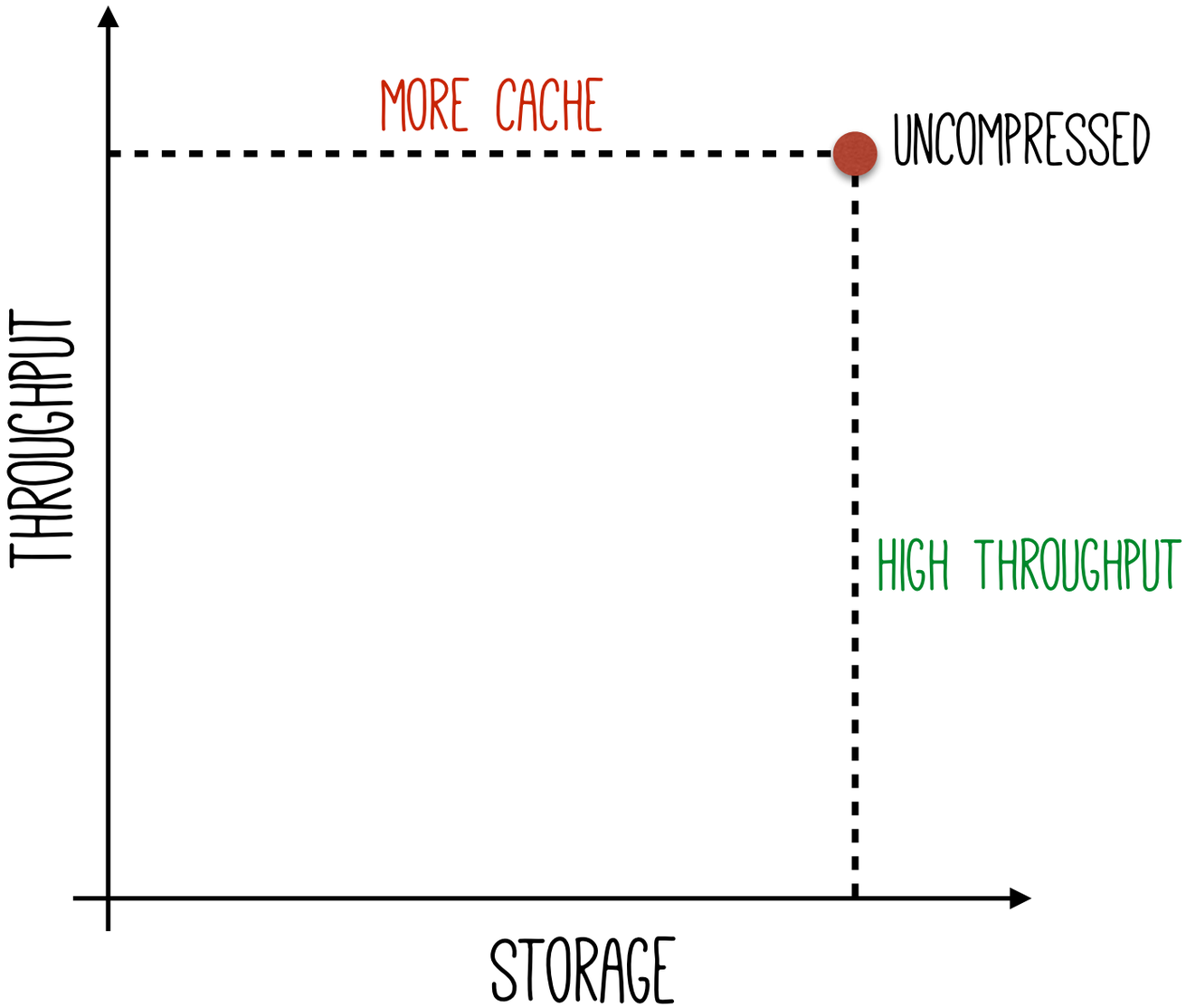
High Throughput: Queries/Second

Existing Data Stores

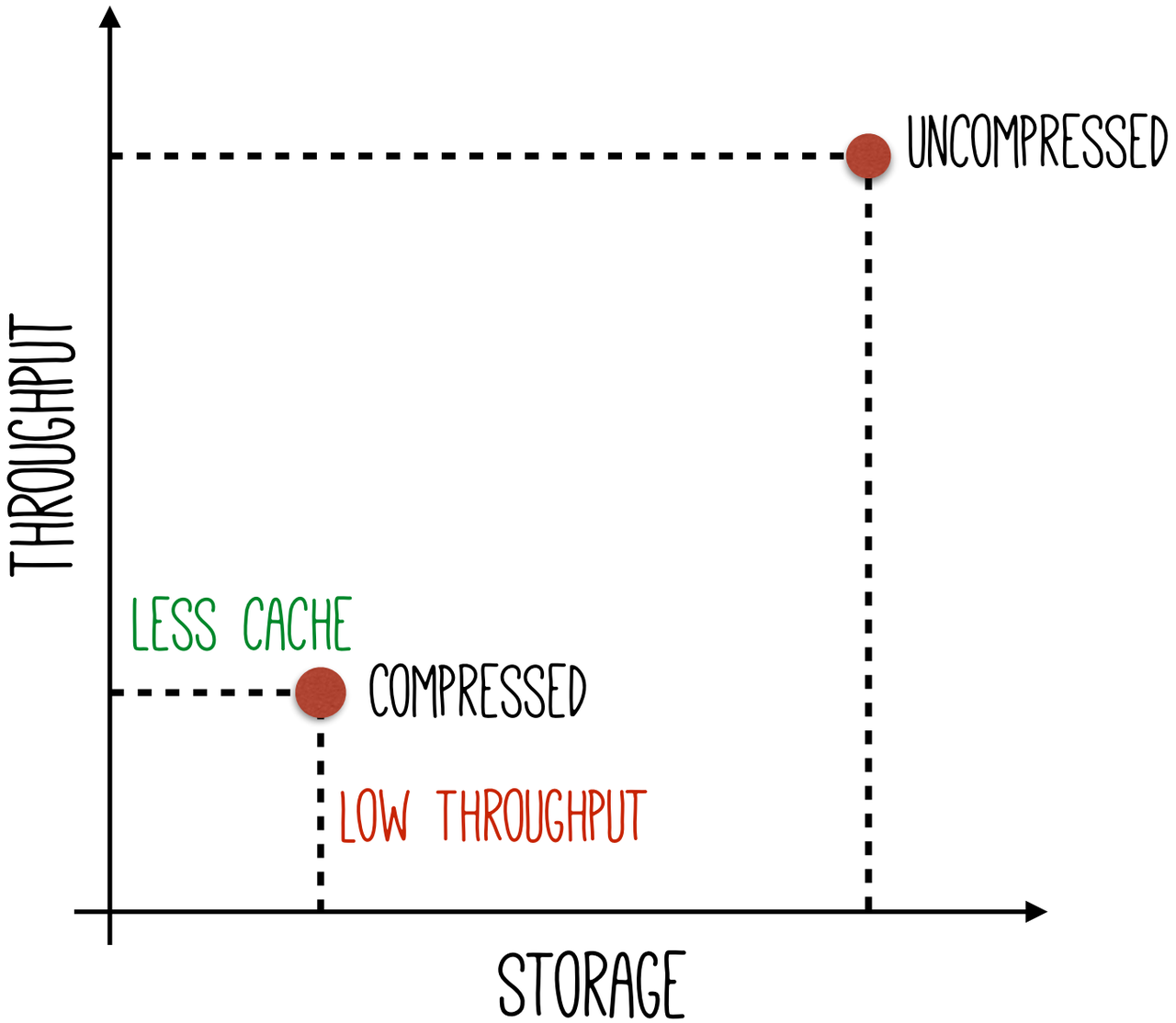
Existing Data Stores



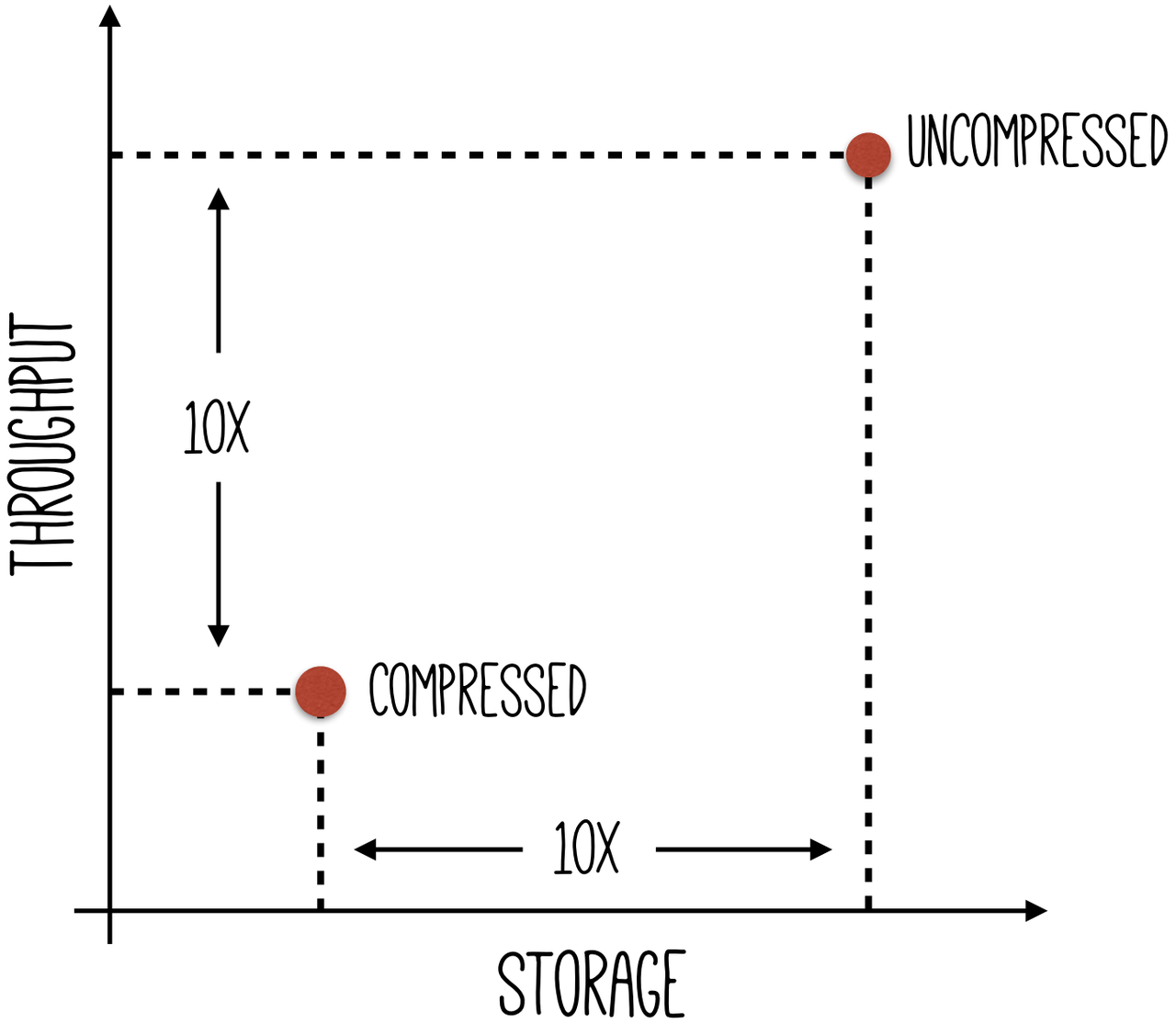
Existing Data Stores



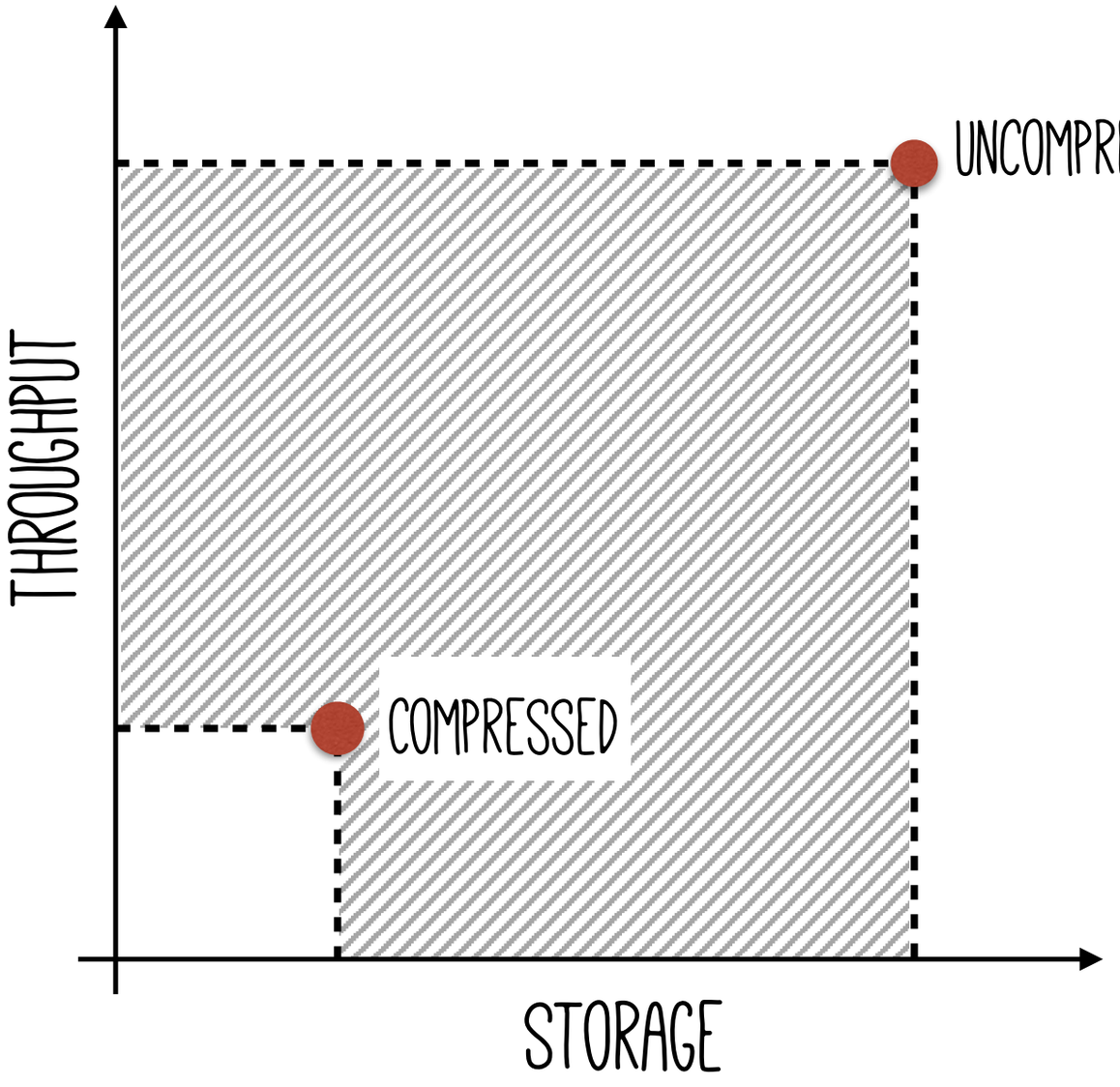
Existing Data Stores



Existing Data Stores

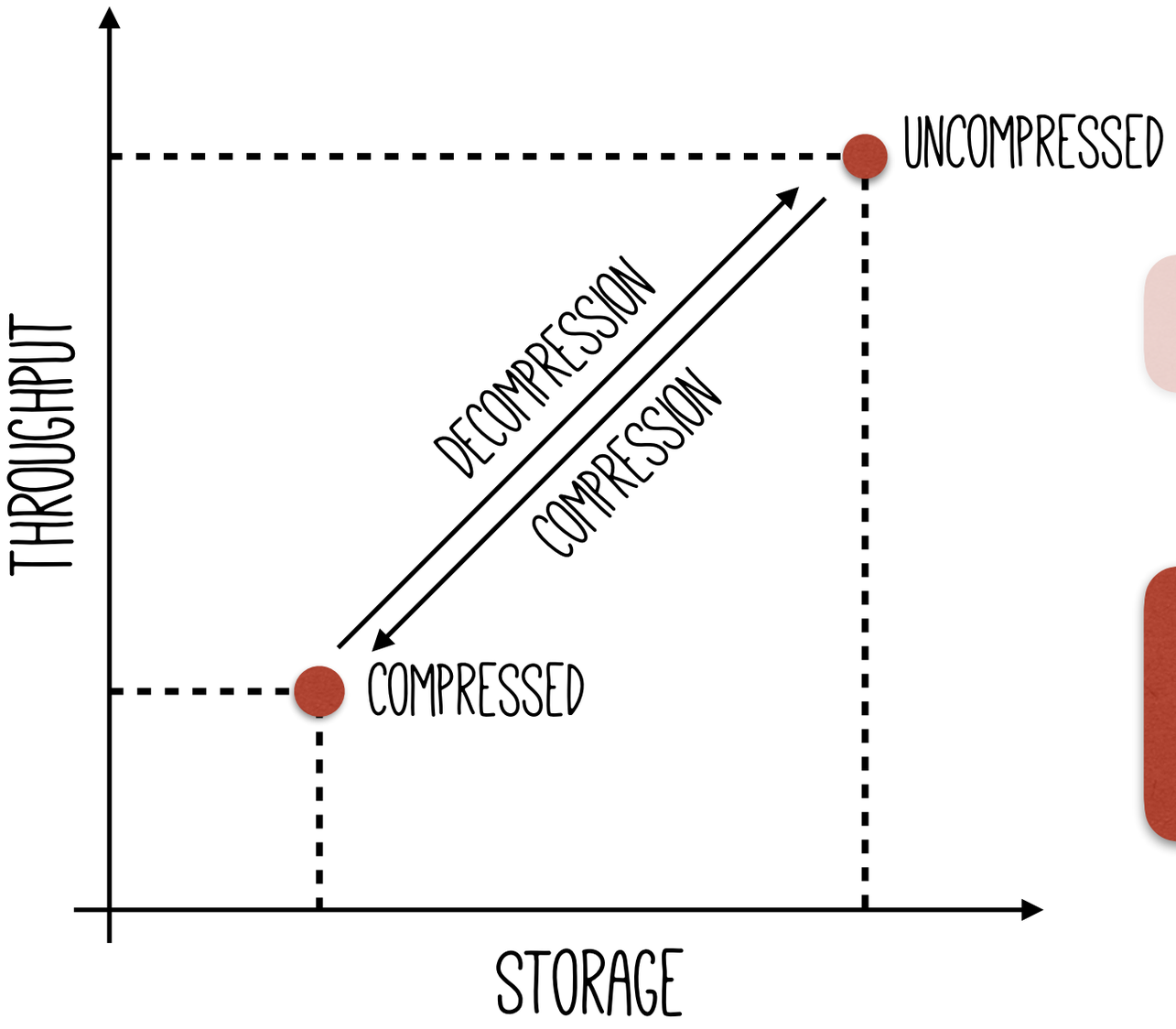


Existing Data Stores



Unachievable points

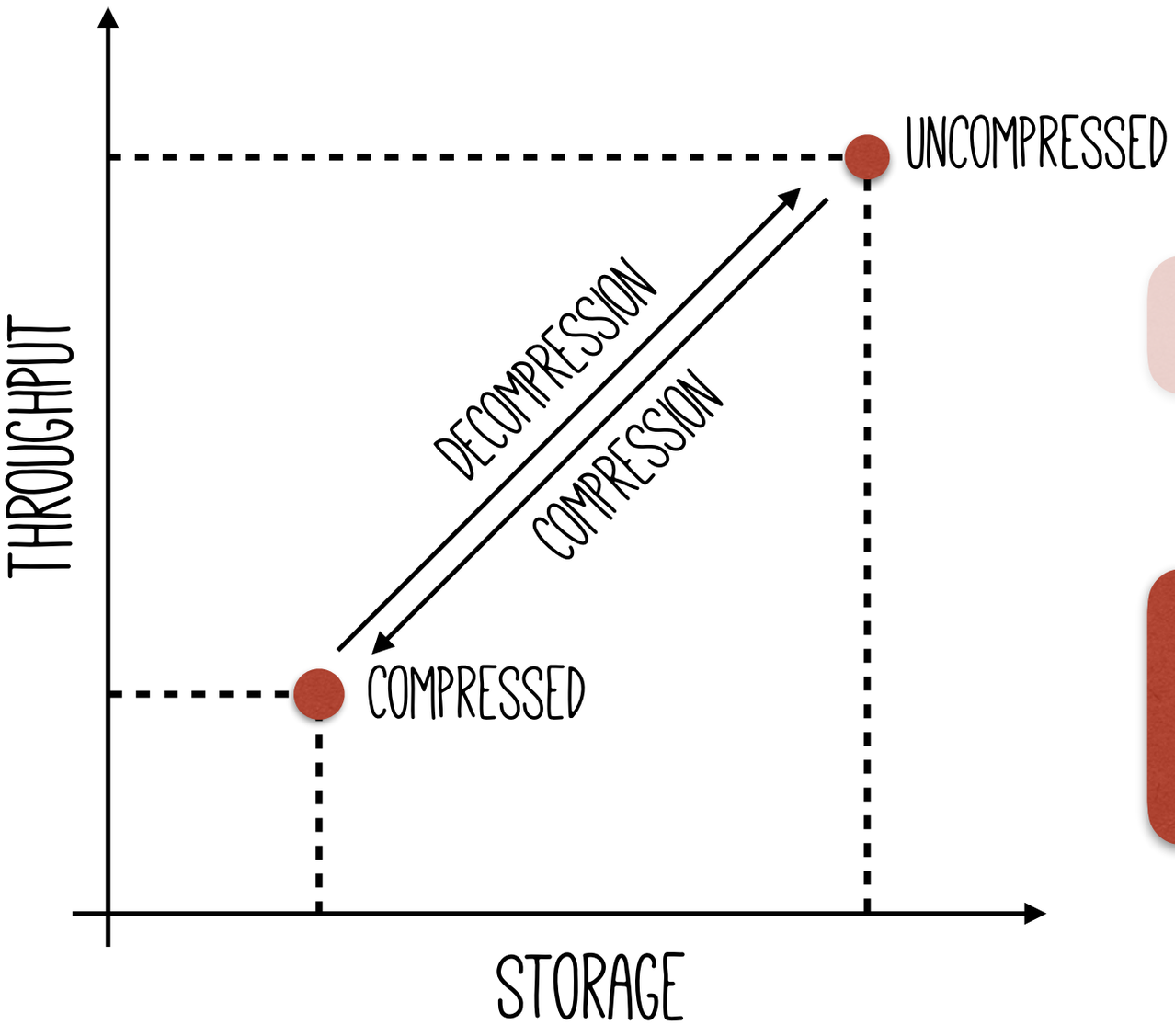
Existing Data Stores



Unachievable points

Switching between the two
incurs high latency & CPU

Existing Data Stores



Unachievable points

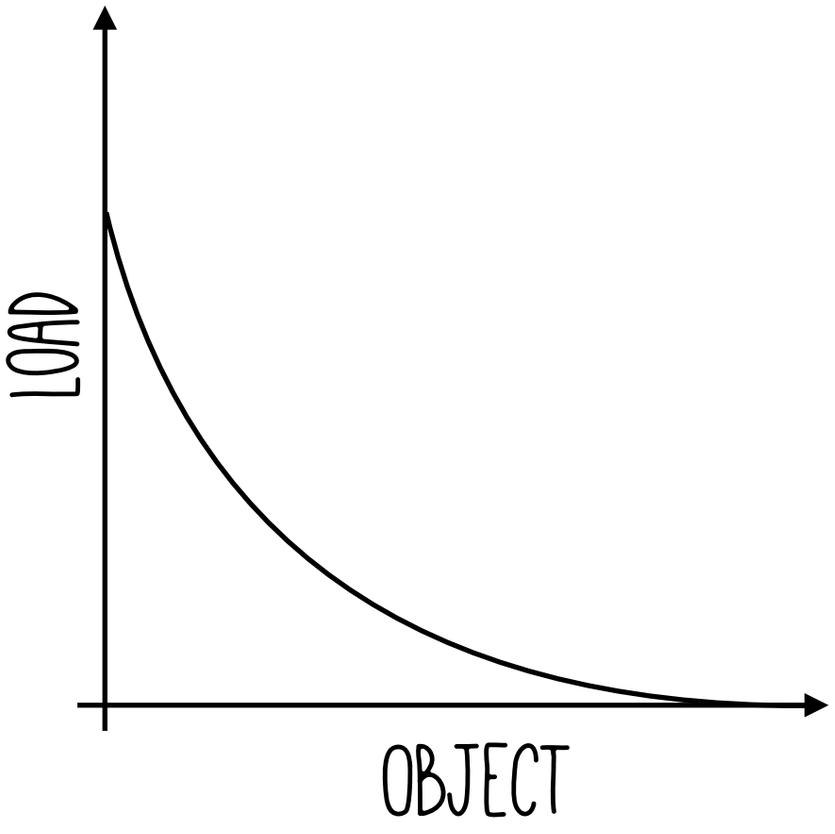
Switching between the two incurs high latency & CPU

Leads to degraded performance when underlying workload or infrastructure changes

A Motivating Example

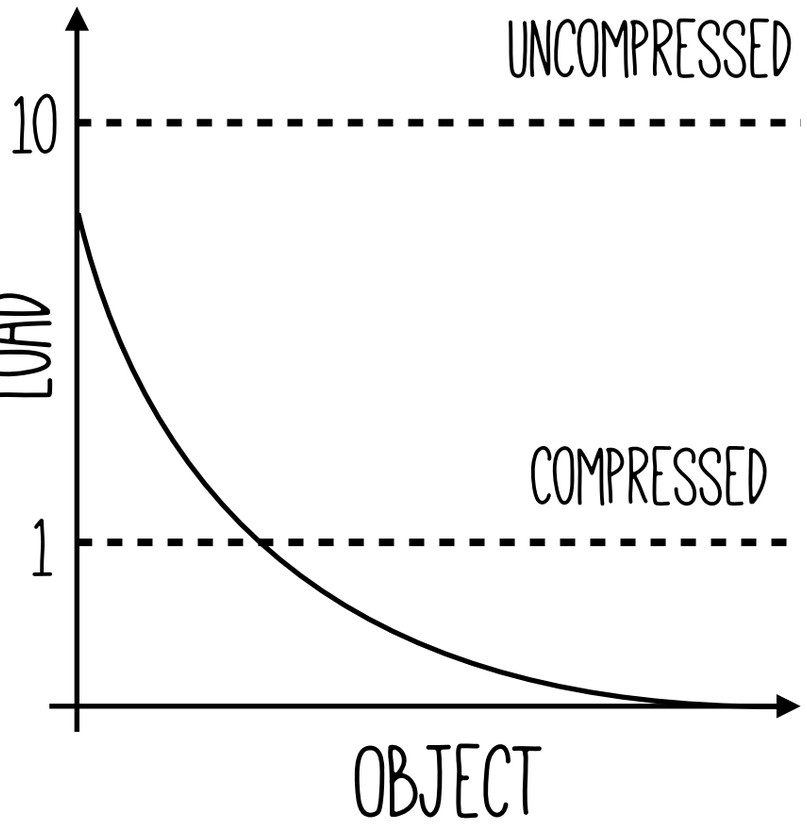
A Motivating Example

Load across items heavily skewed



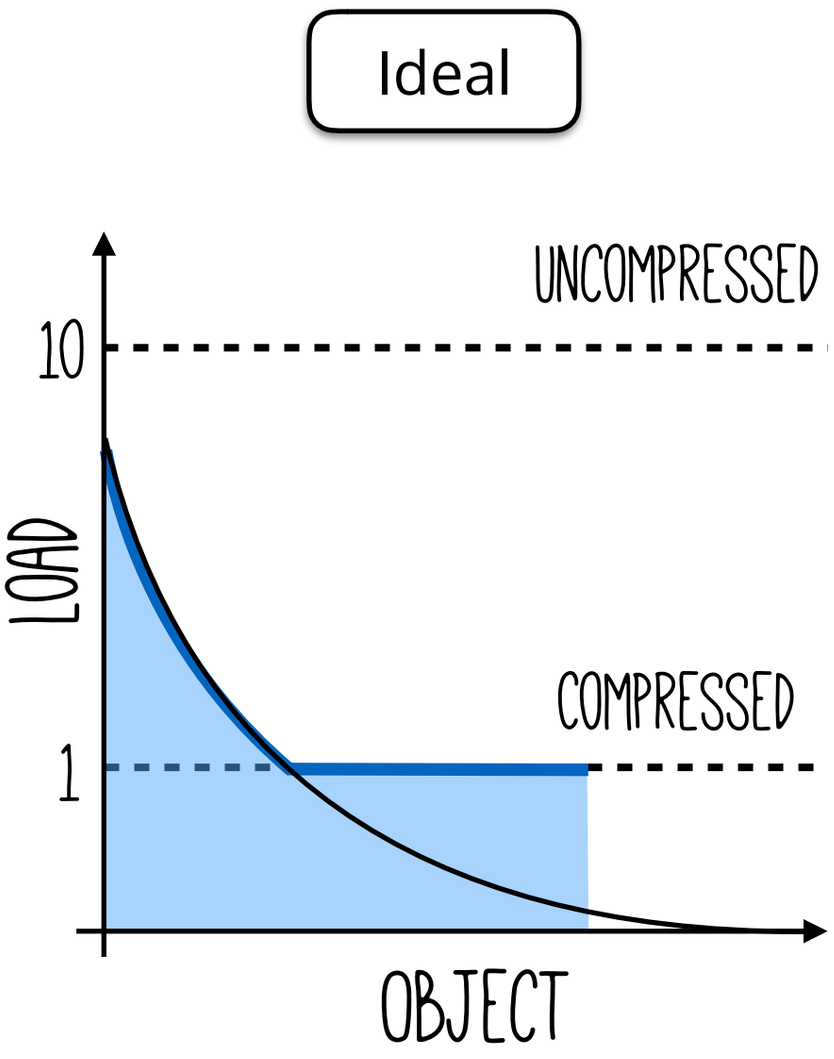
A Motivating Example

Load across items heavily skewed



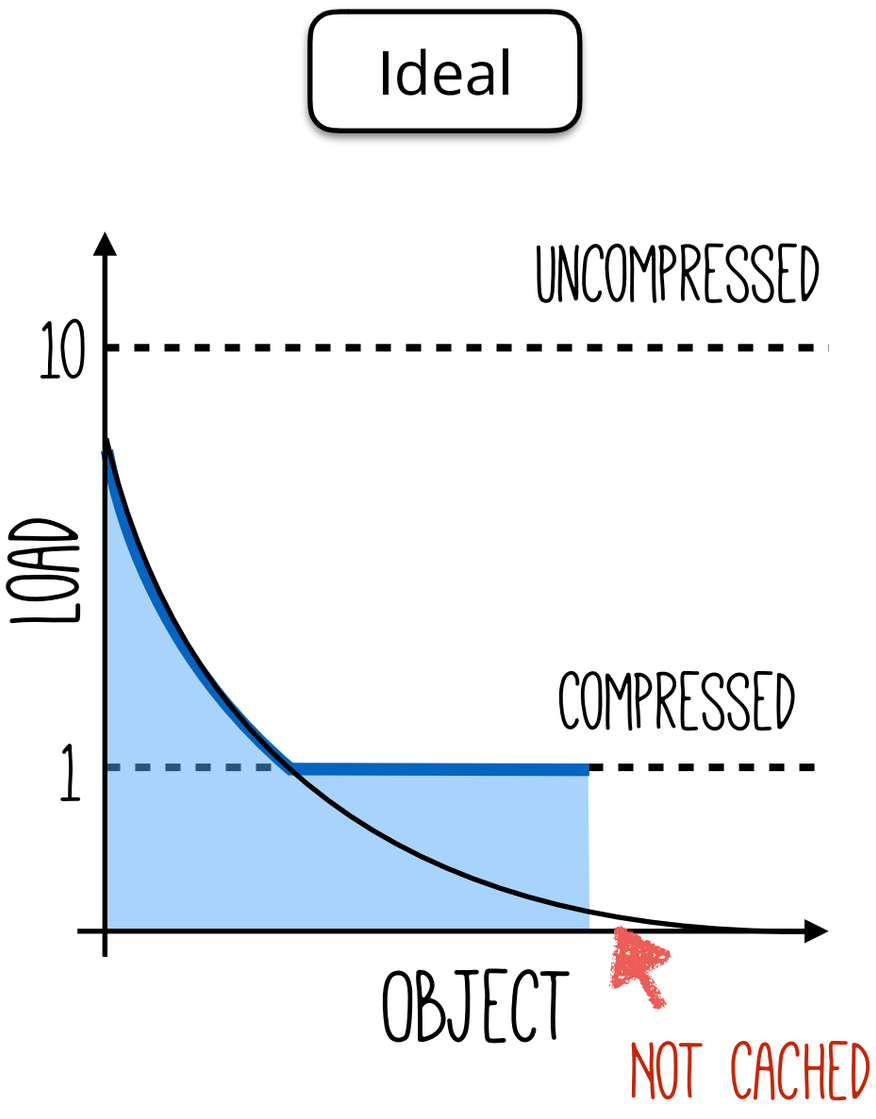
A Motivating Example

Load across items heavily skewed



A Motivating Example

Load across items heavily skewed

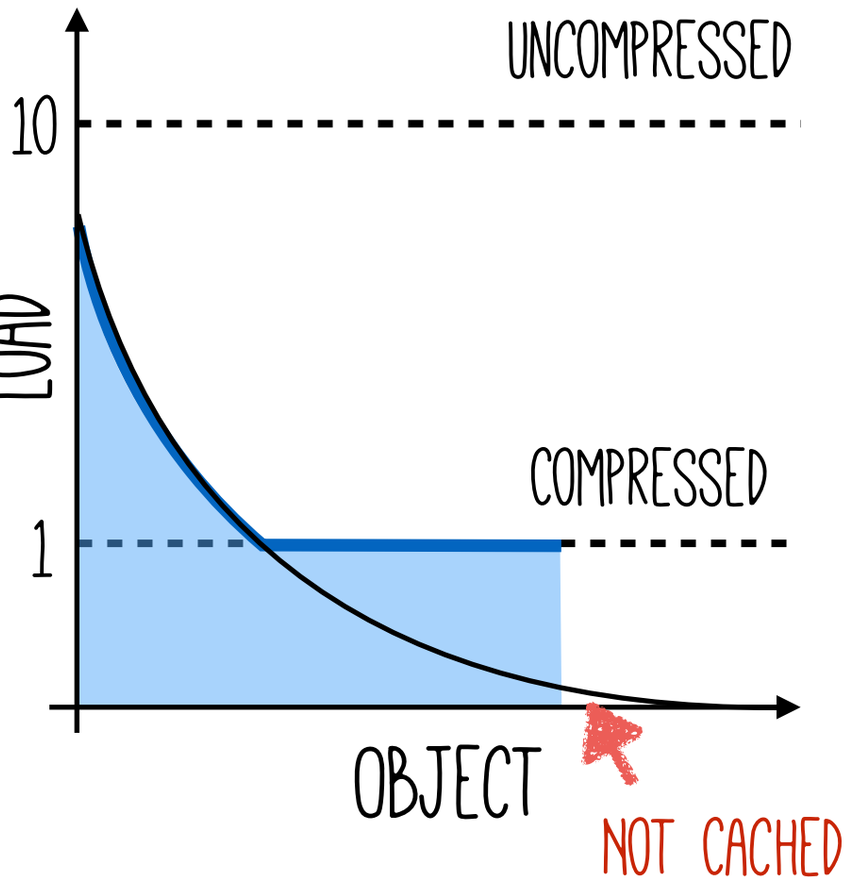


A Motivating Example

Load across items heavily skewed

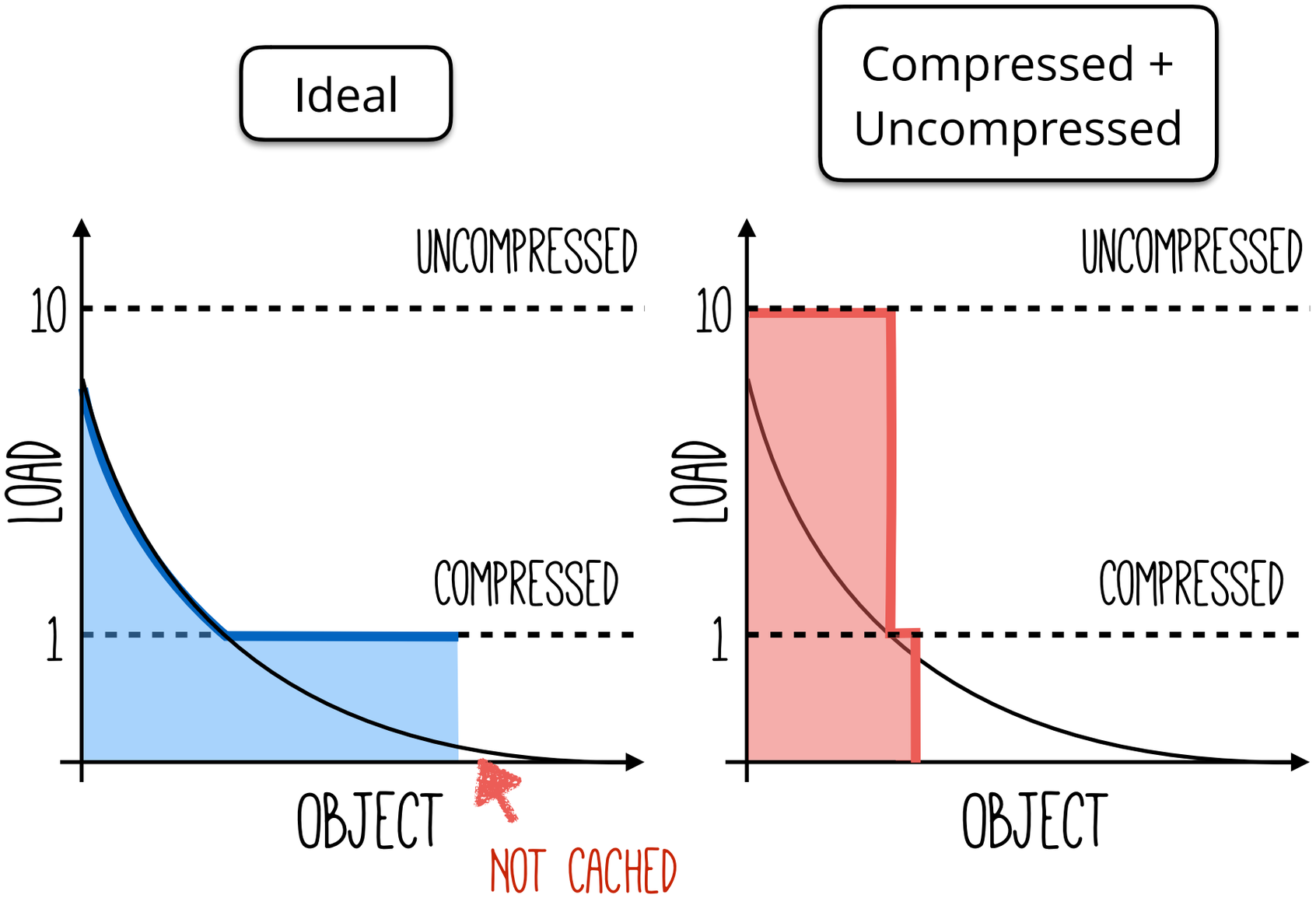
Ideal

Compressed + Uncompressed



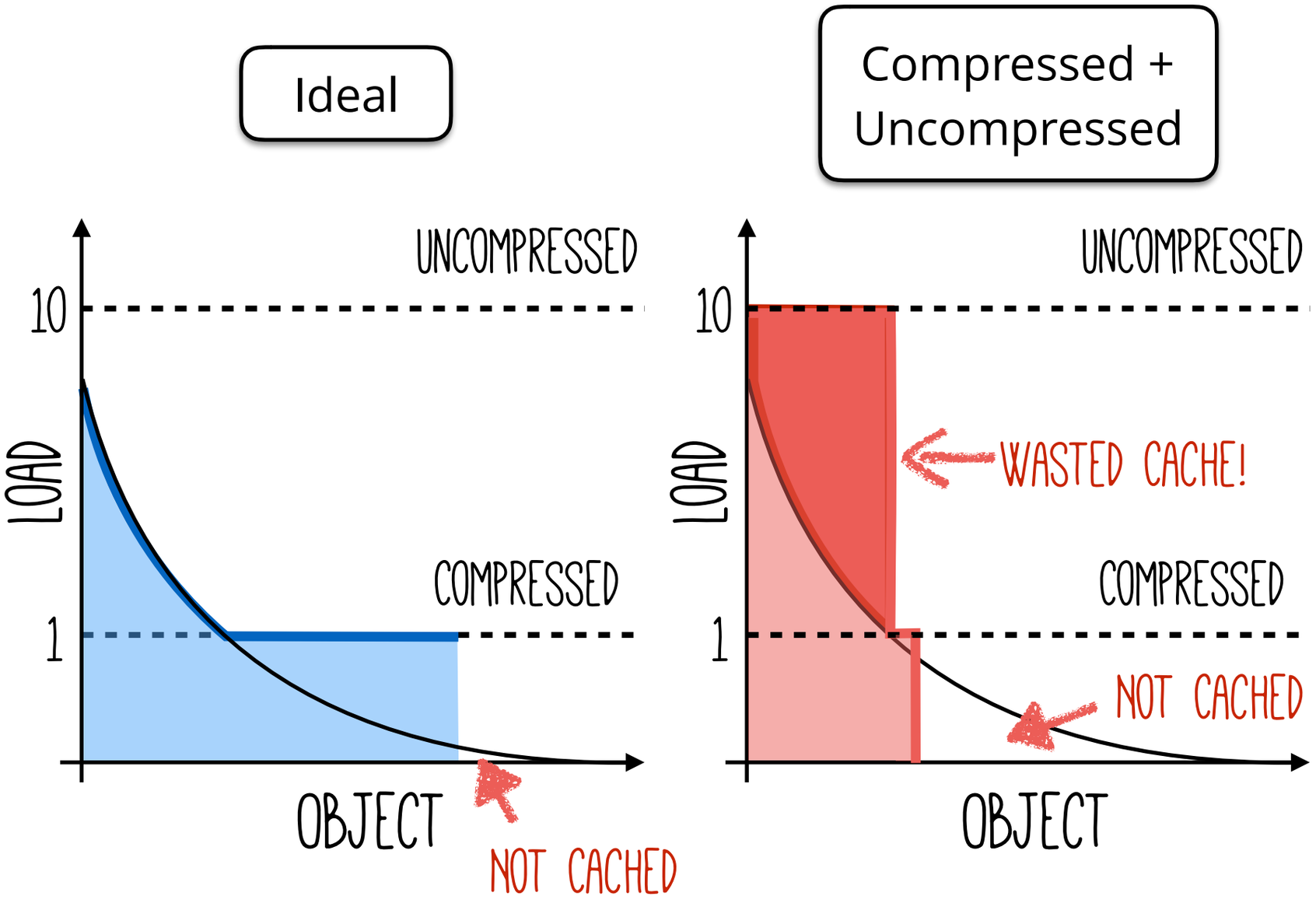
A Motivating Example

Load across items heavily skewed



A Motivating Example

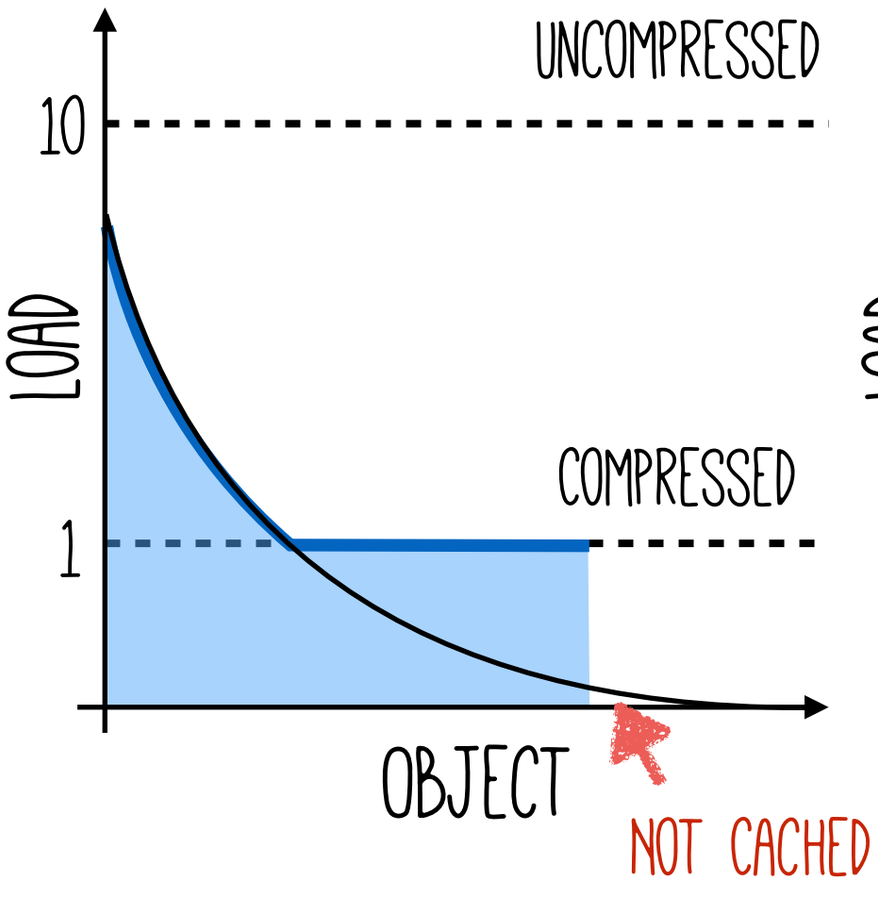
Load across items heavily skewed



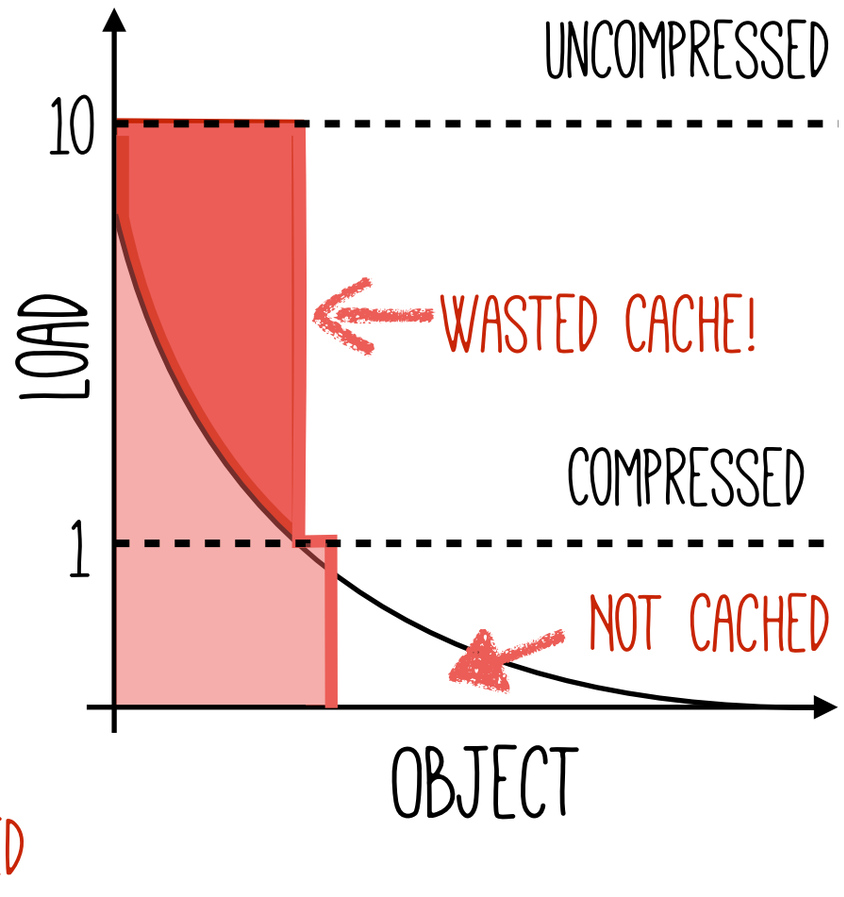
A Motivating Example

Load across items heavily skewed

Ideal



Compressed + Uncompressed

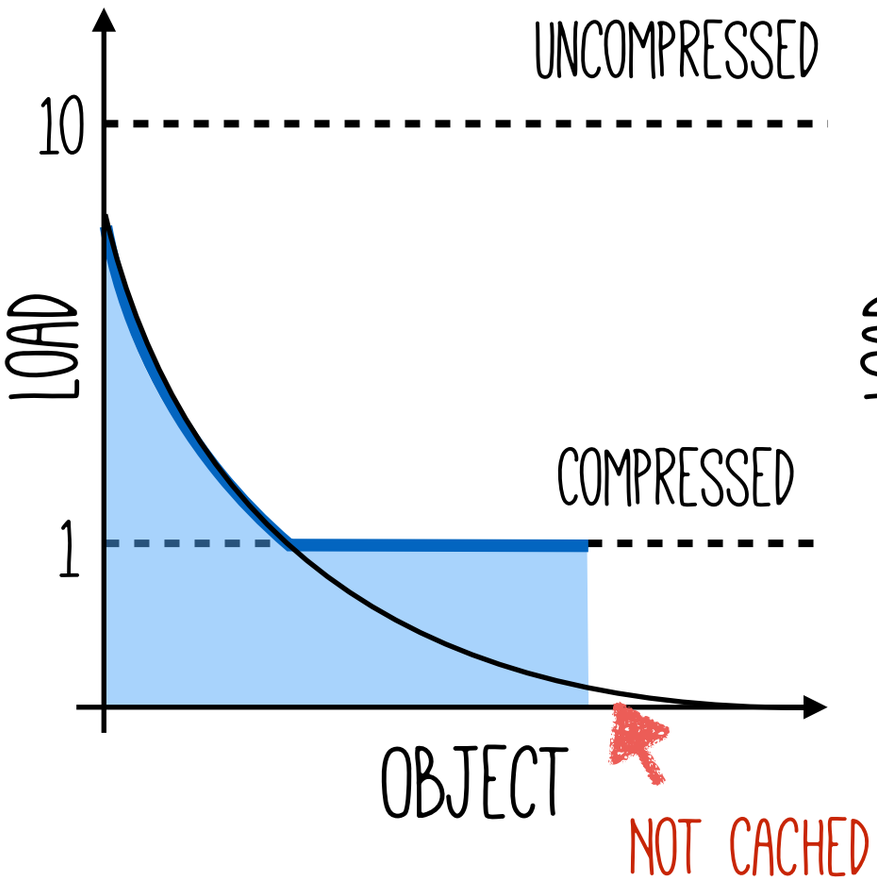


Selective Replication: #Replicas α Load

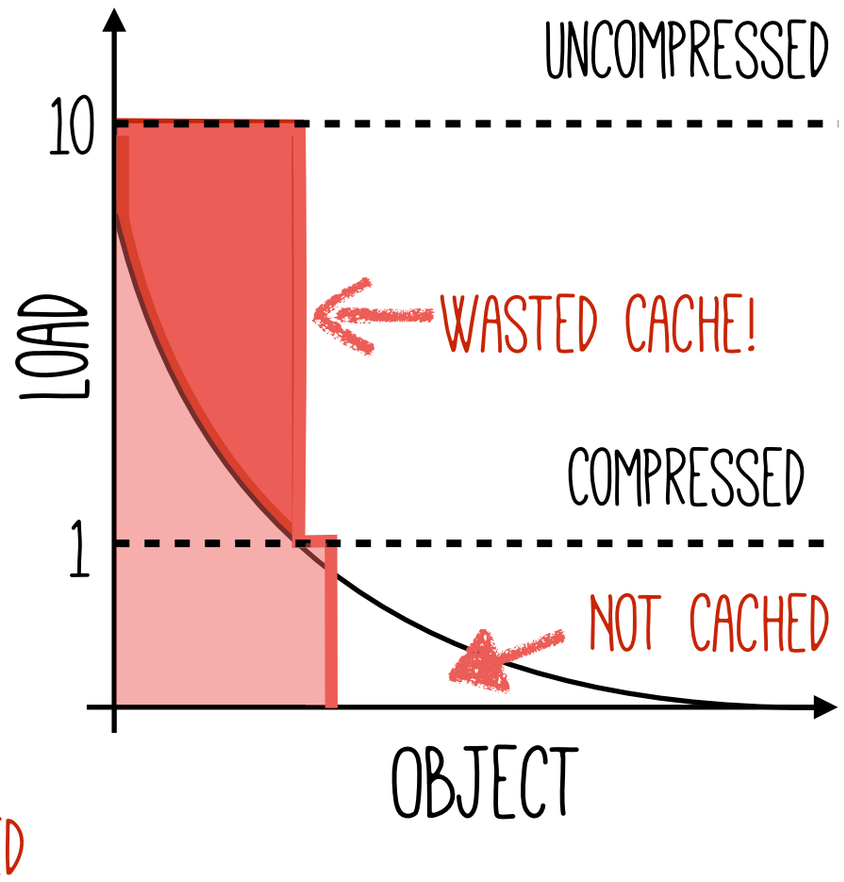
A Motivating Example

Load across items heavily skewed

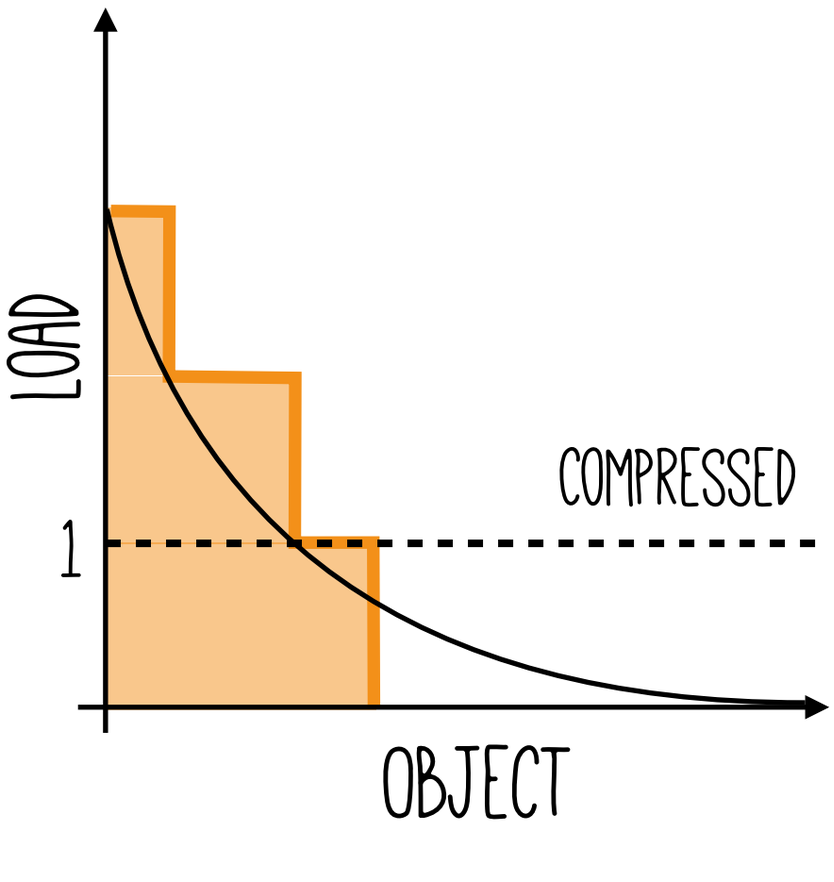
Ideal



Compressed + Uncompressed



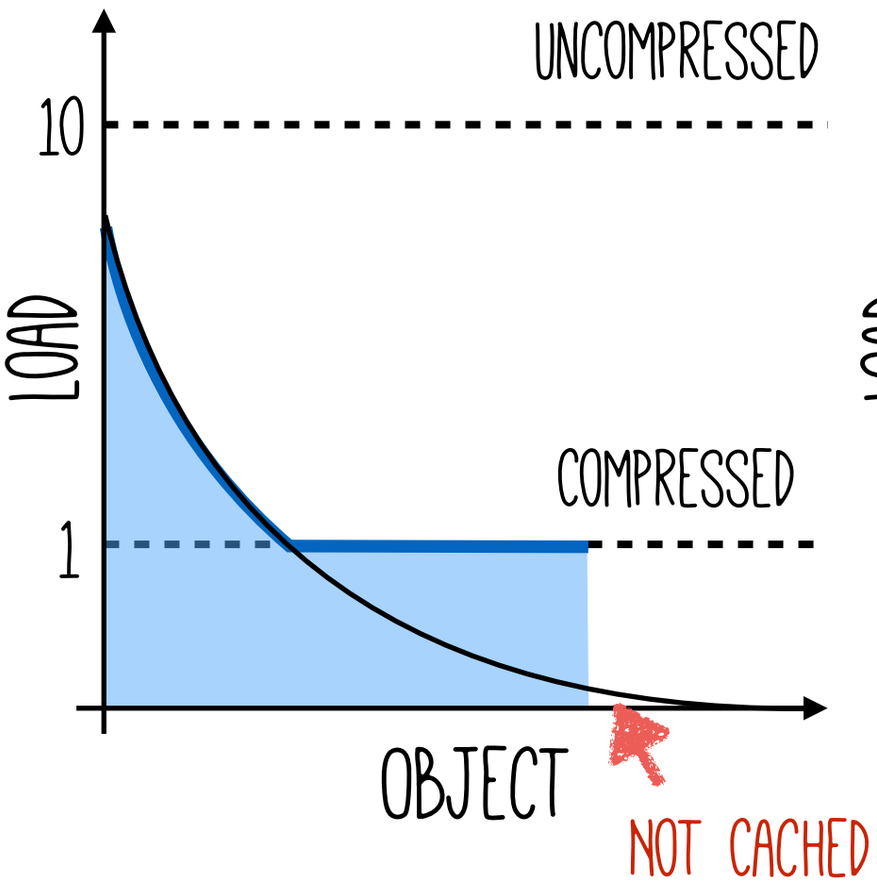
Selective Replication: #Replicas α Load



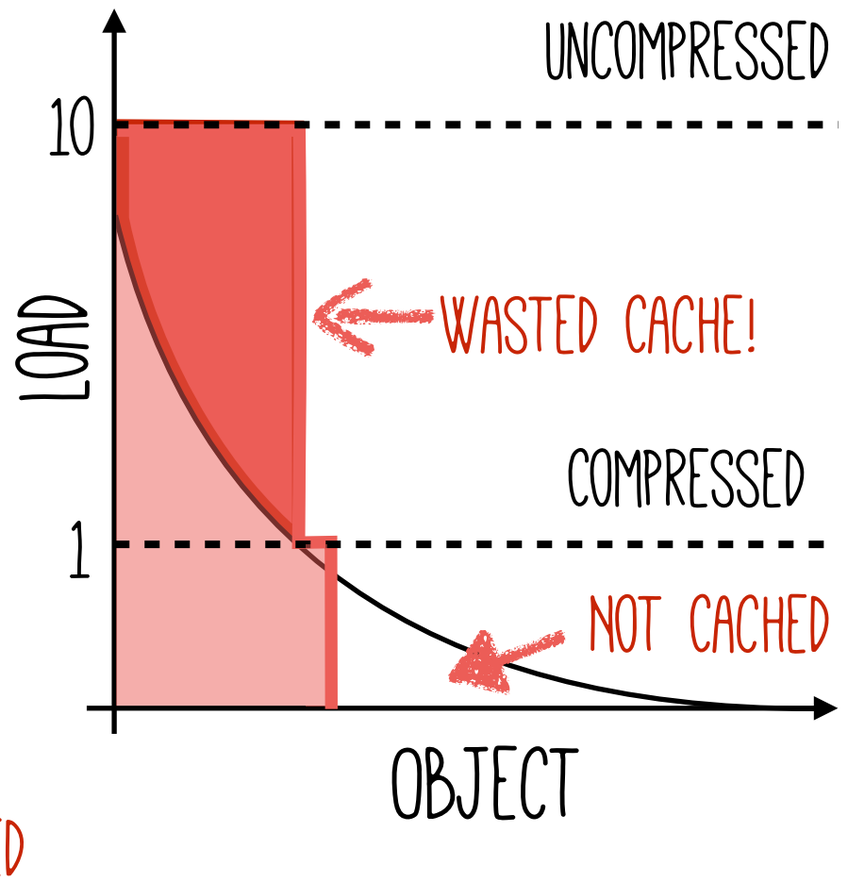
A Motivating Example

Load across items heavily skewed

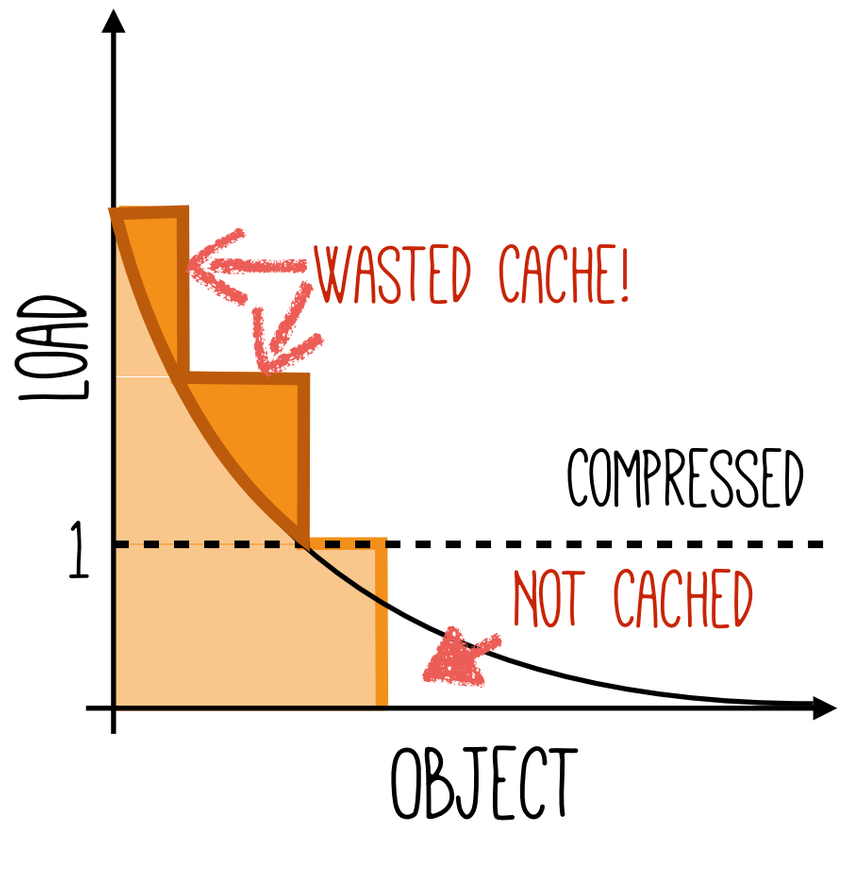
Ideal



Compressed + Uncompressed



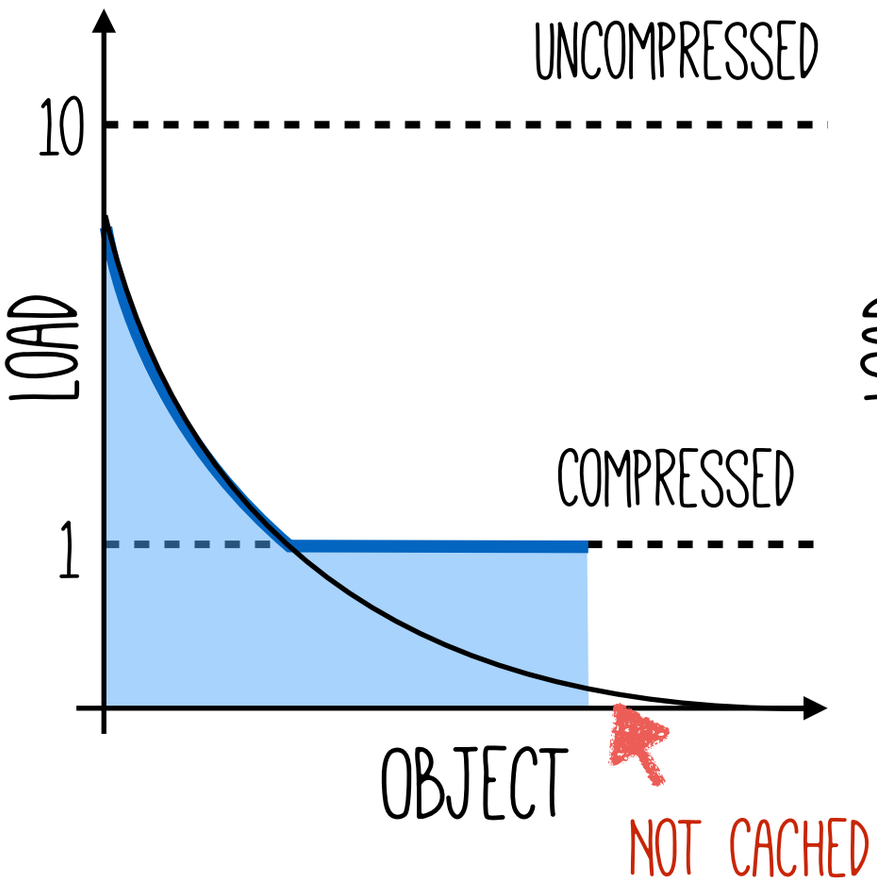
Selective Replication: #Replicas α Load



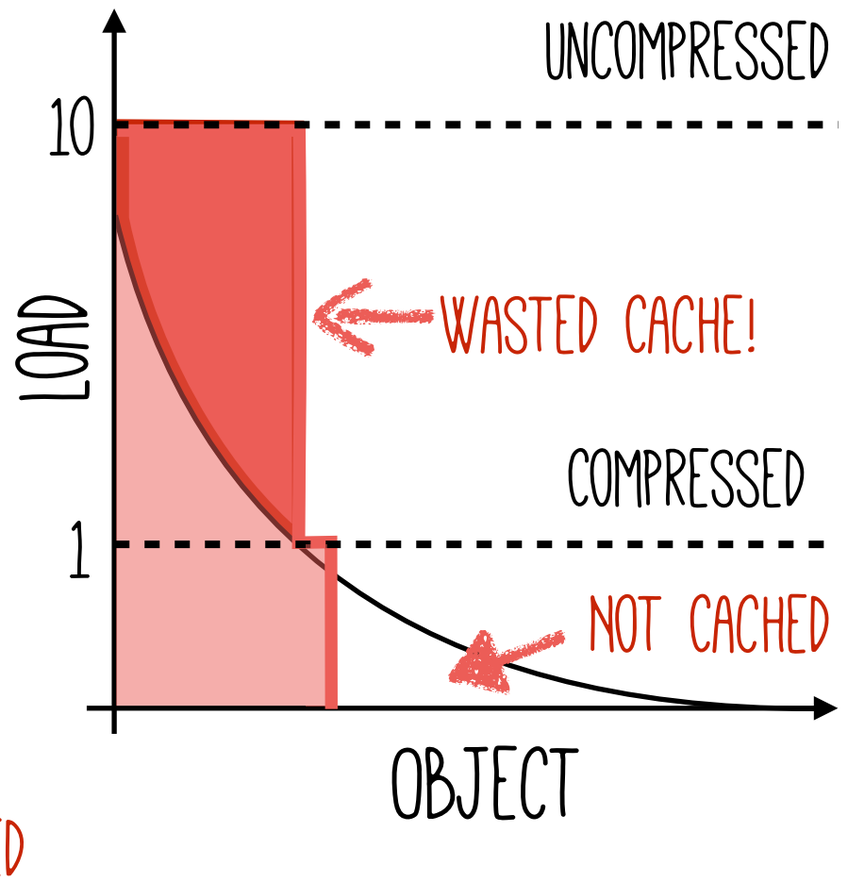
A Motivating Example

Load across items heavily skewed

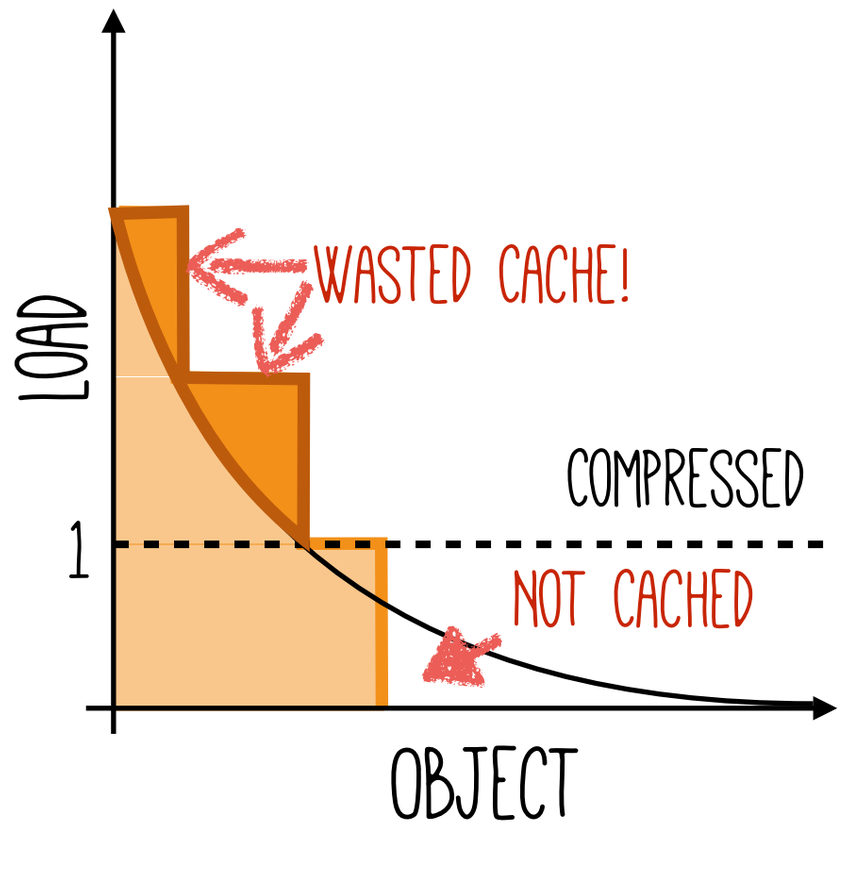
Ideal



Compressed + Uncompressed



Selective Replication: #Replicas α Load

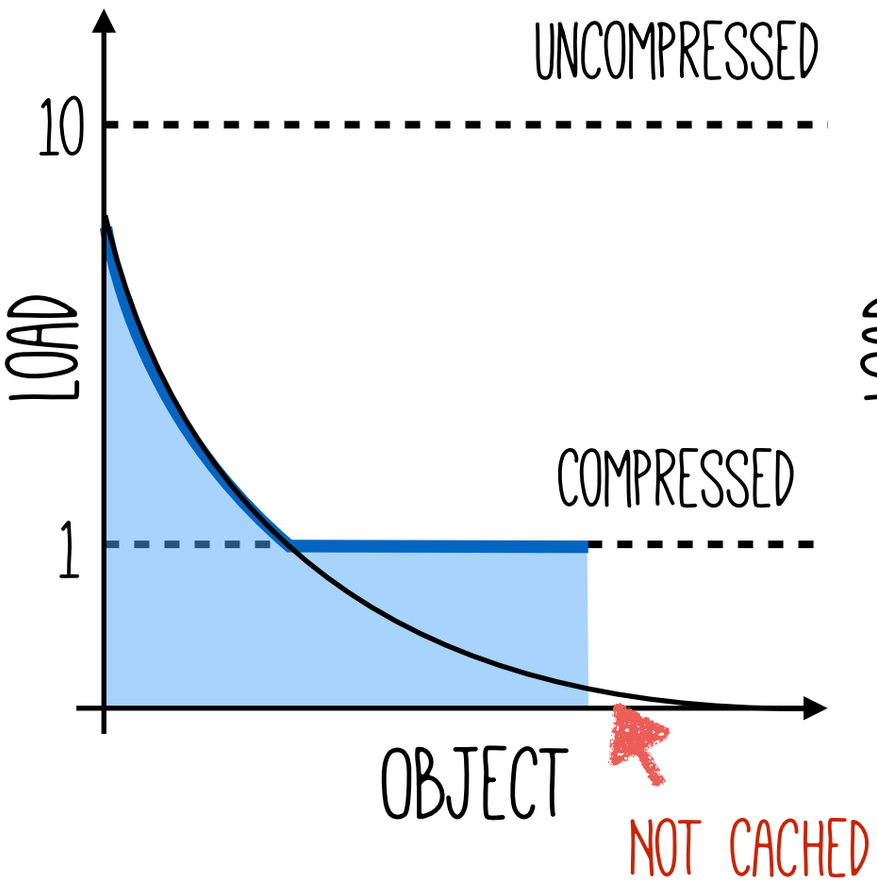


Load changes over time

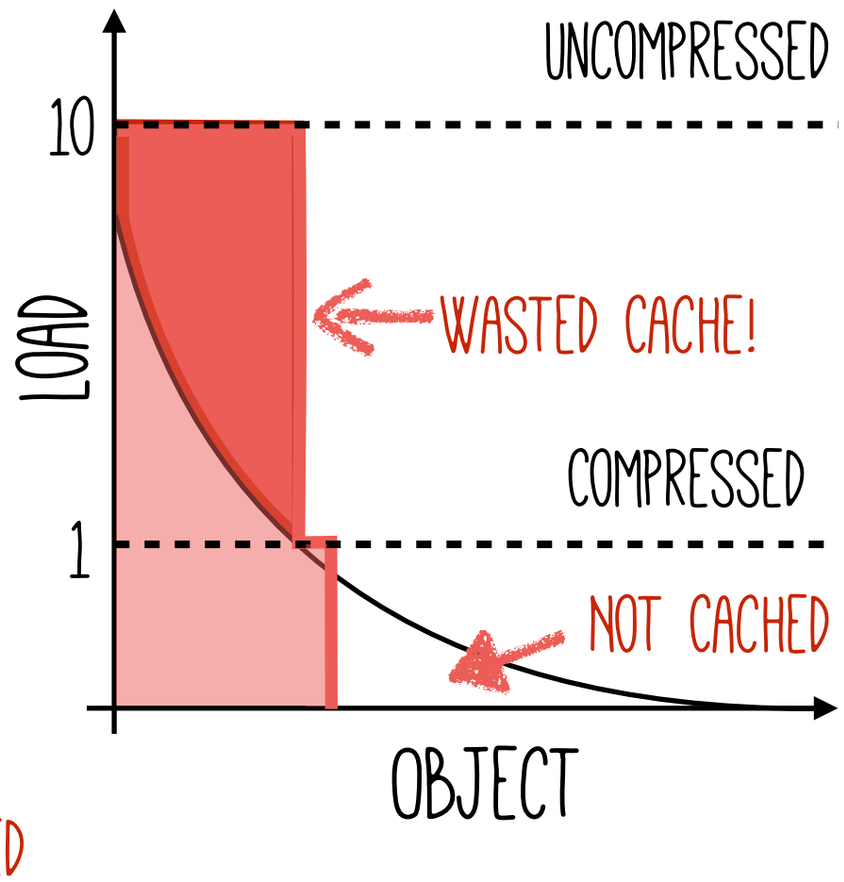
A Motivating Example

Load across items heavily skewed

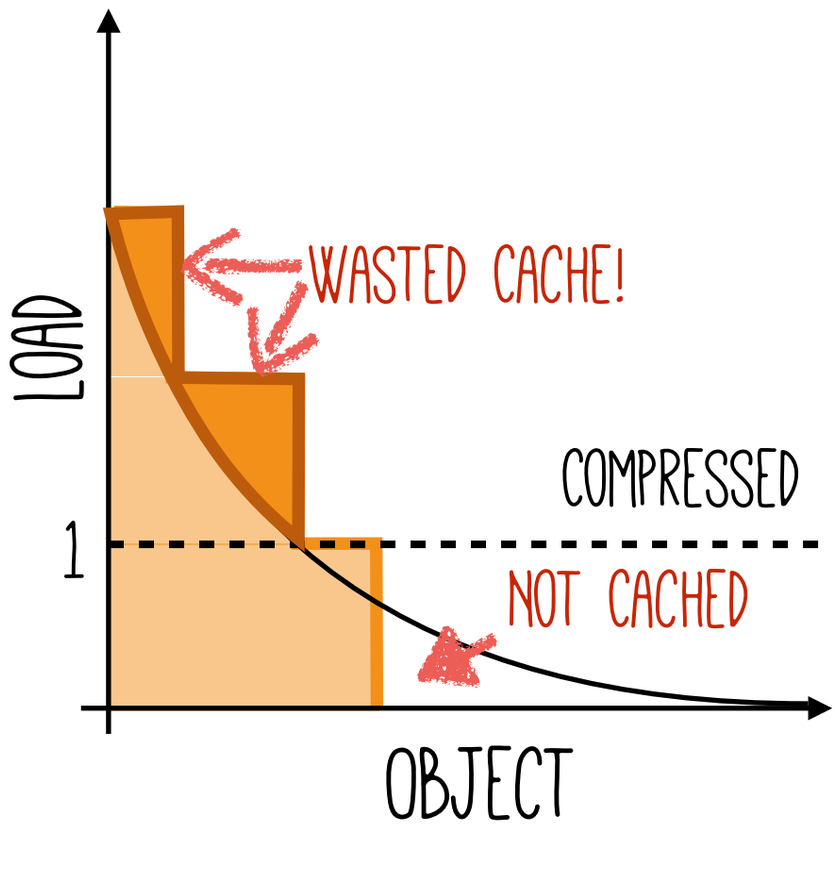
Ideal



Compressed + Uncompressed

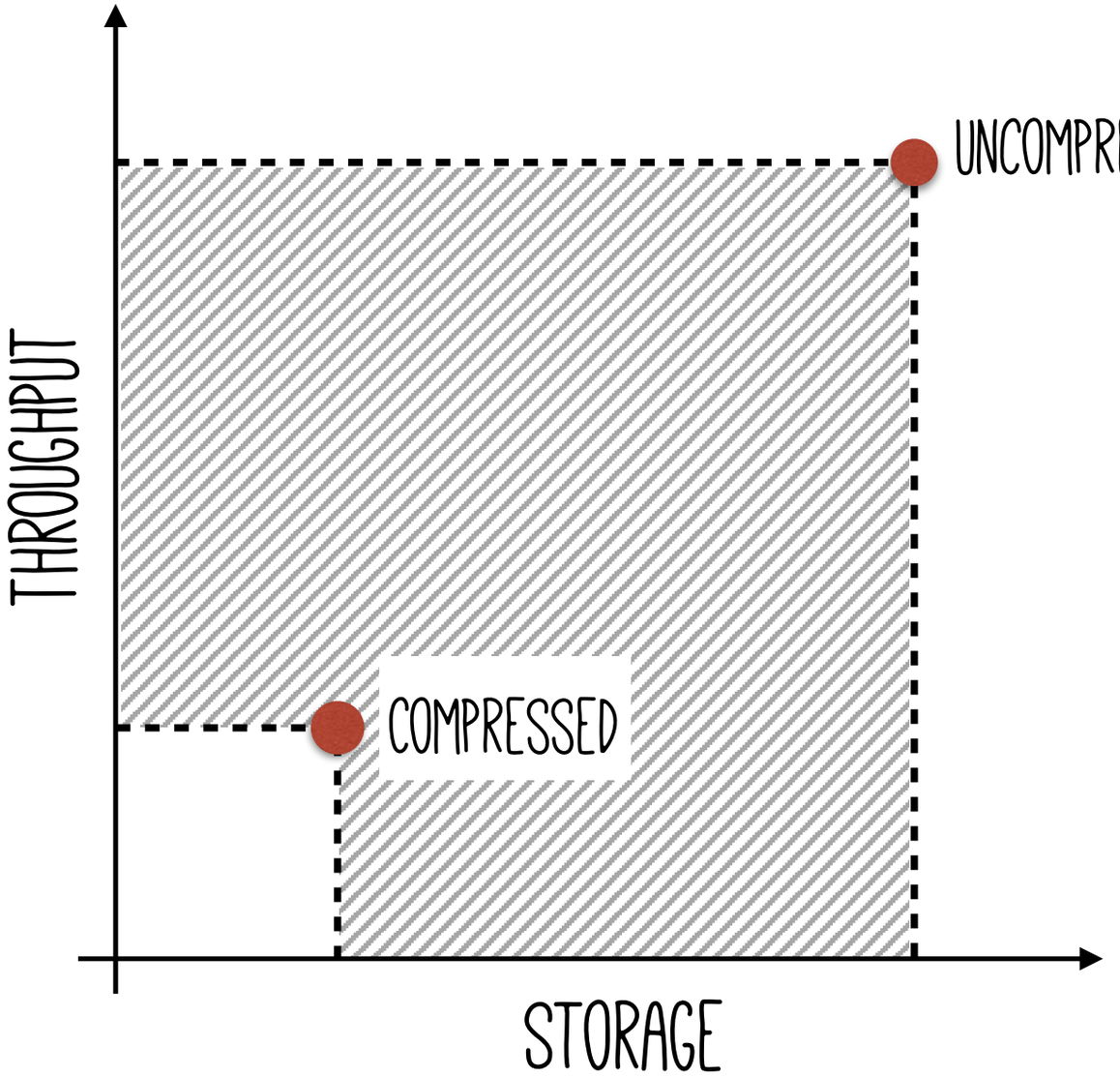


Selective Replication: #Replicas α Load

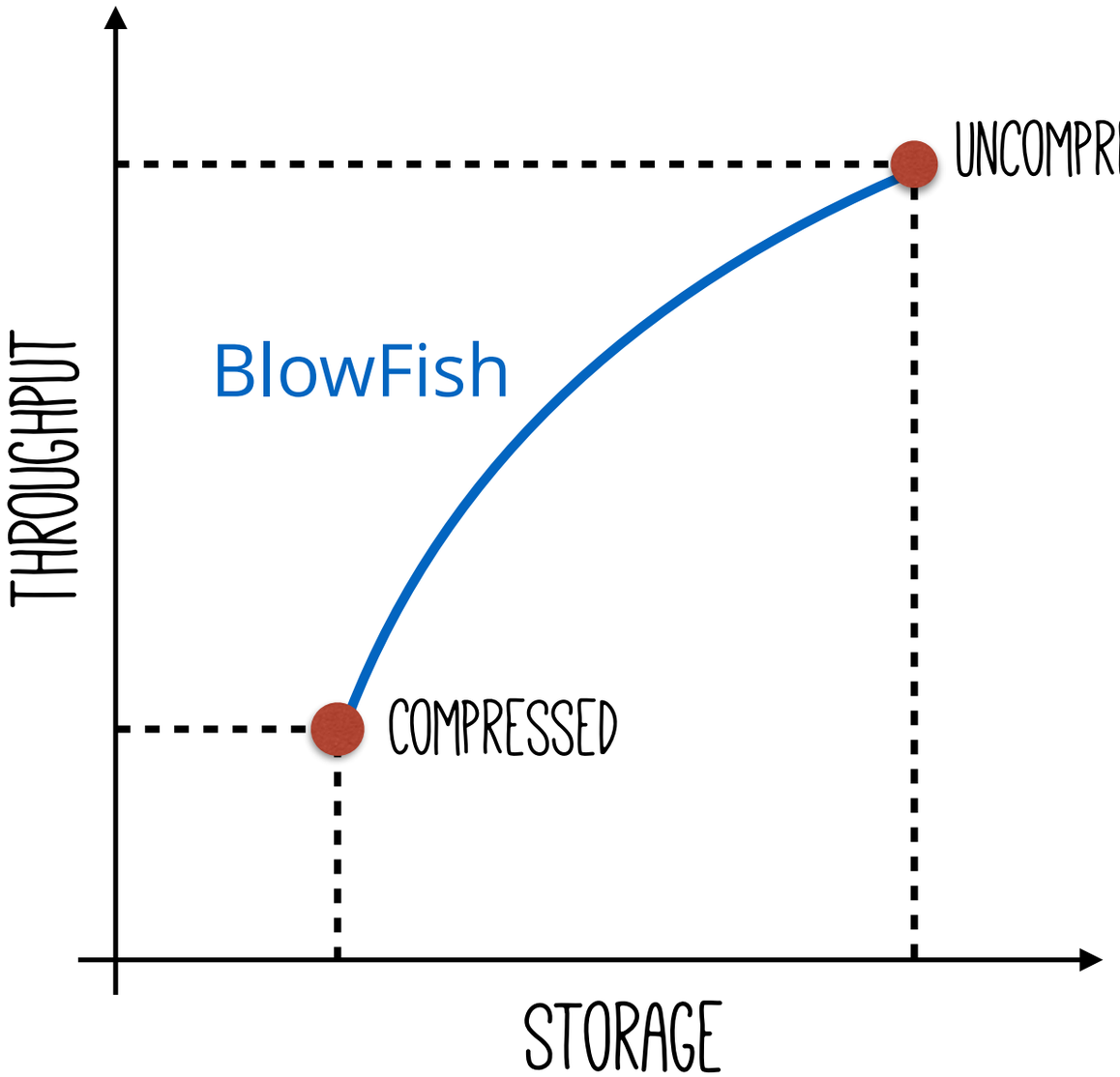


Load changes over time → Degraded performance

BlowFish

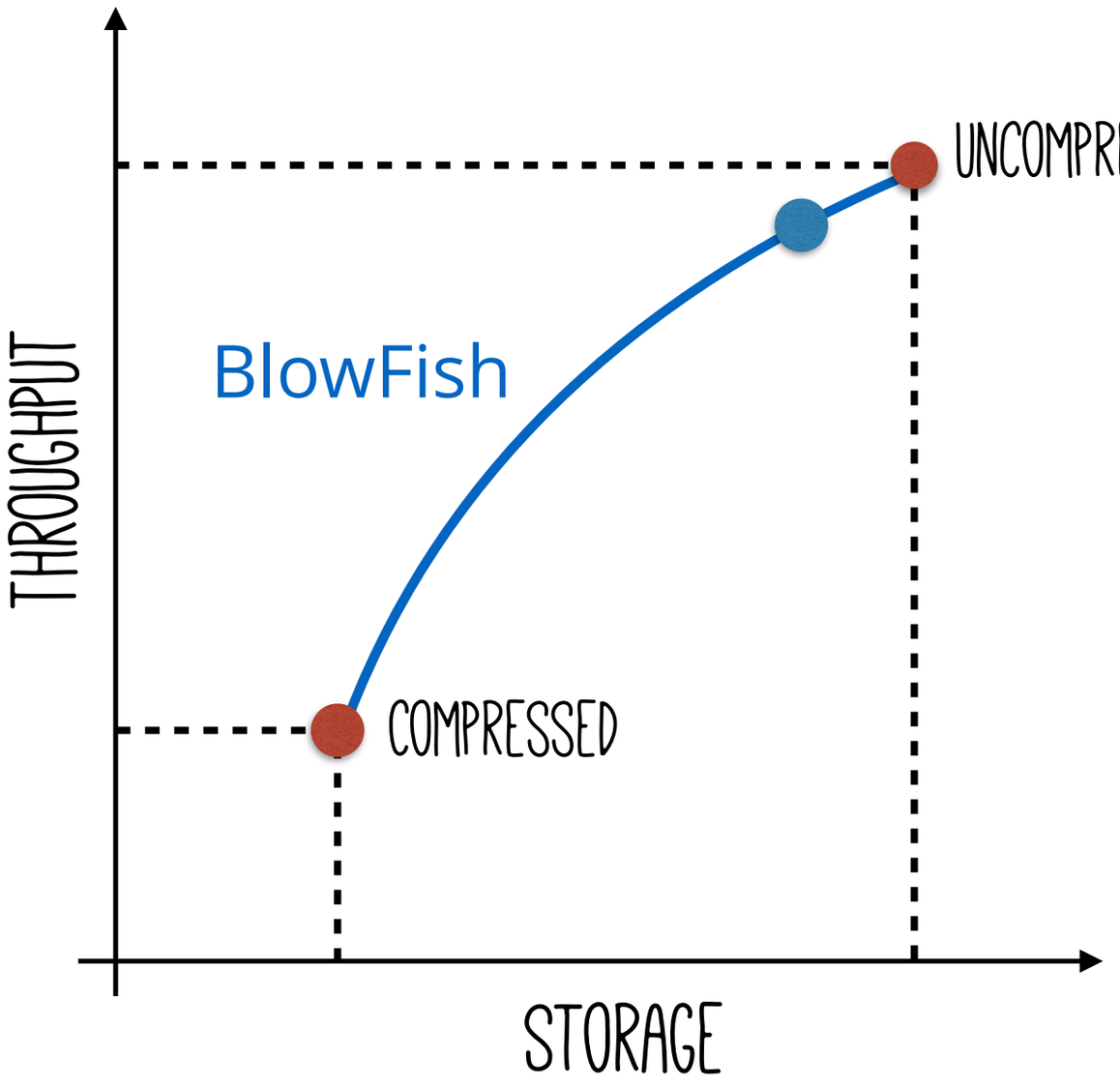


BlowFish



Smooth Tradeoff Curve

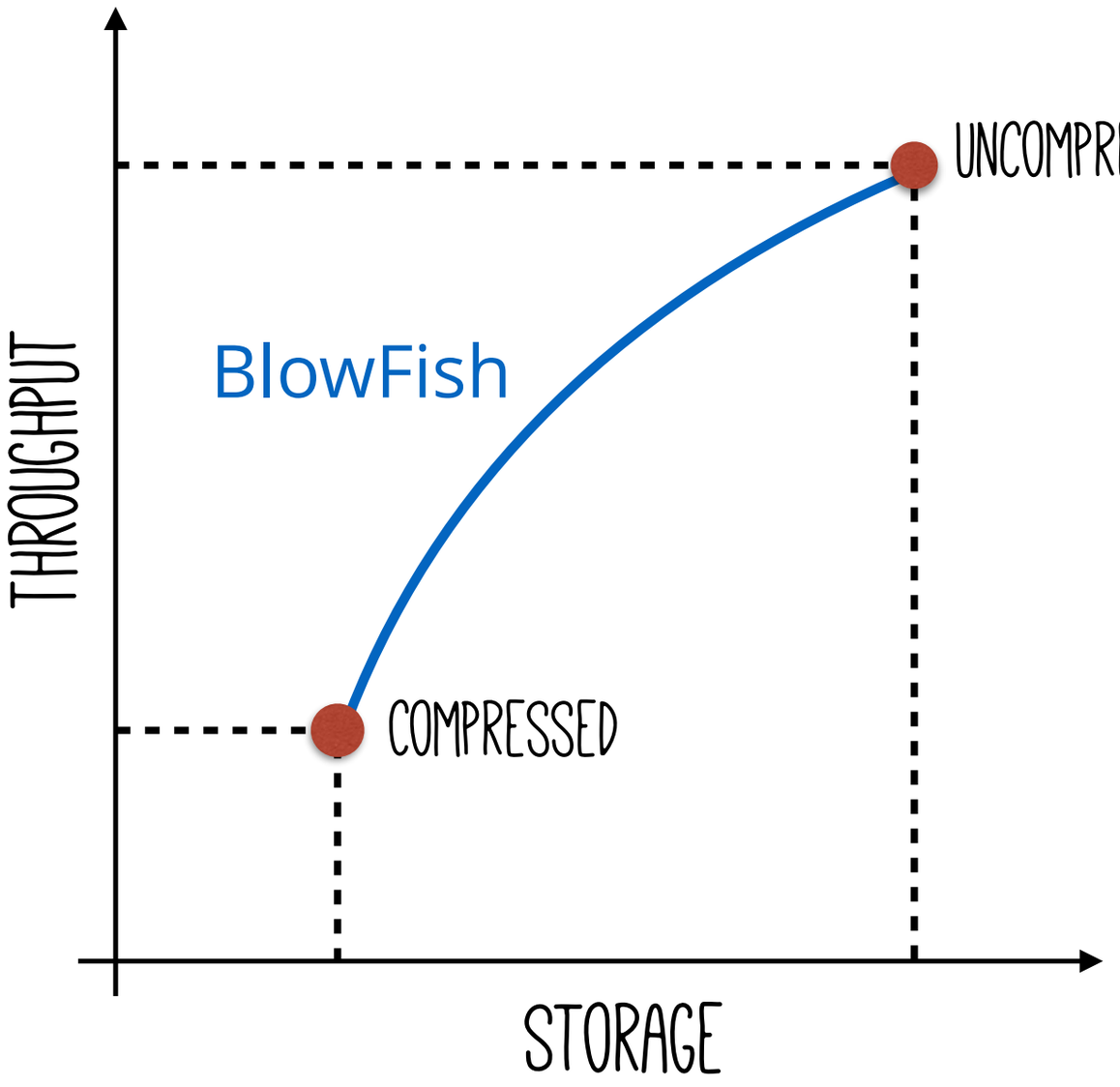
BlowFish



Smooth Tradeoff Curve

Dynamic Navigation

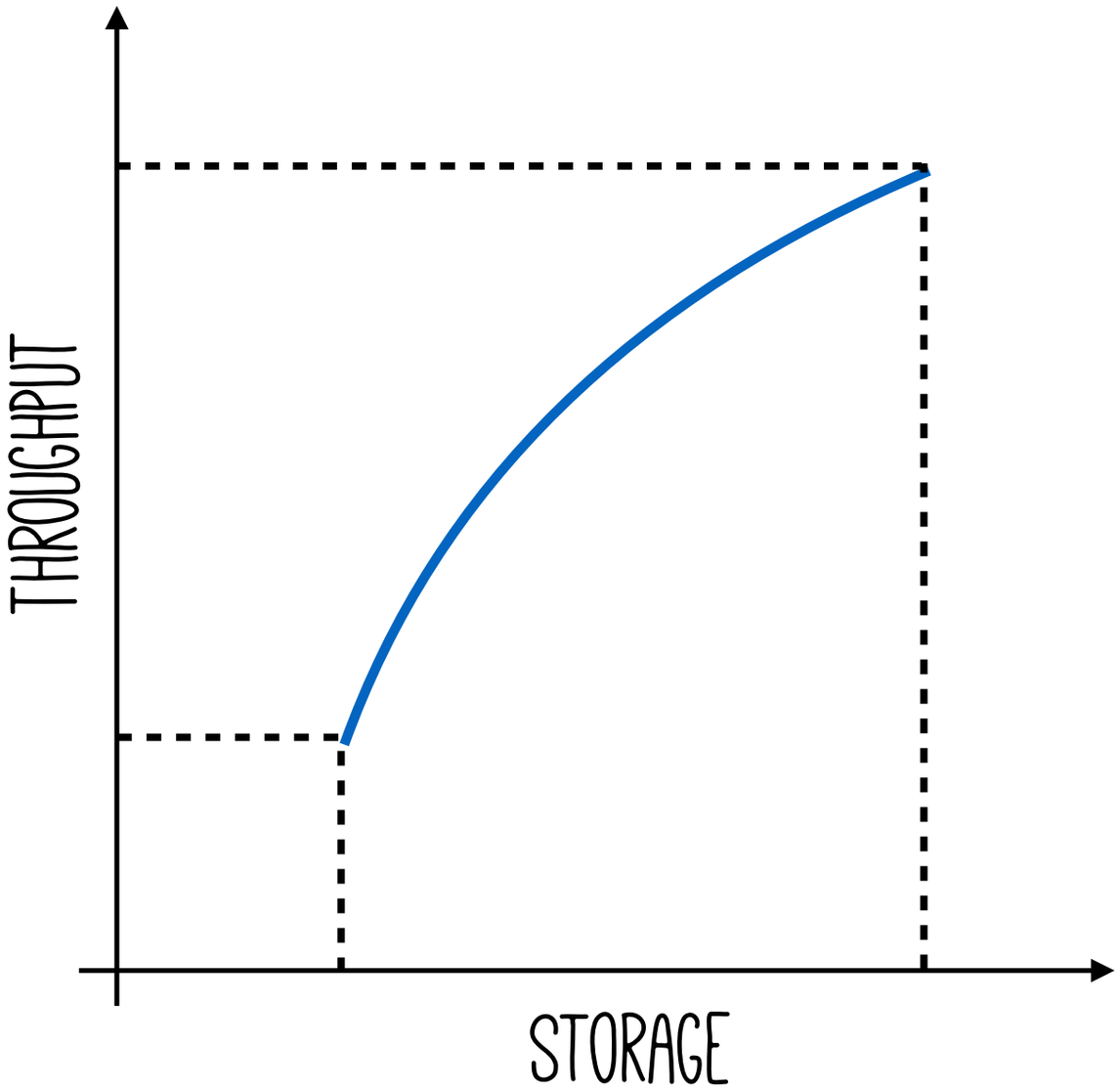
BlowFish

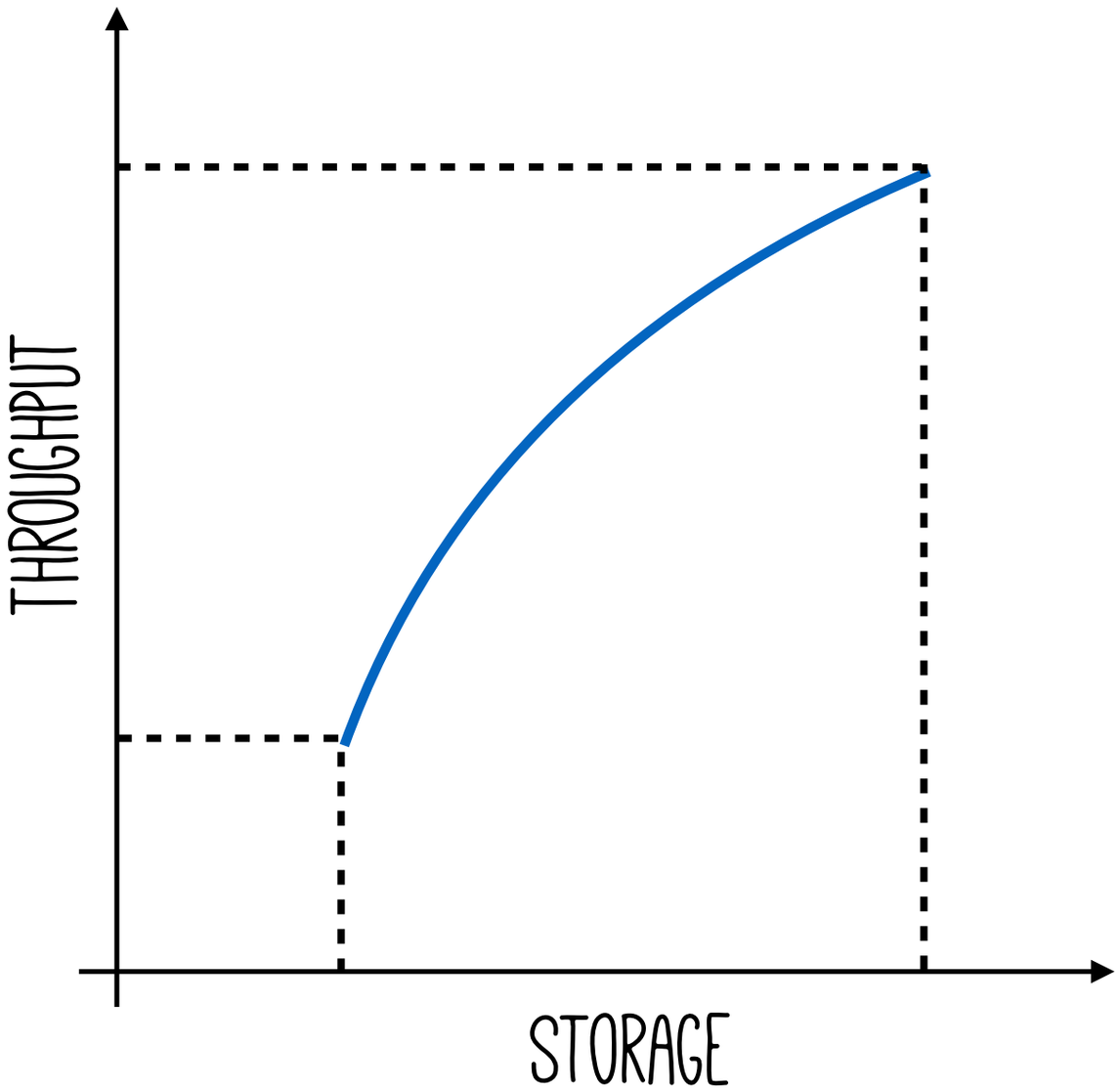


Smooth Tradeoff Curve

Dynamic Navigation

Applications in Several Classical Systems Problems


THROUGHPUT
STORAGE



Storage-Performance Tradeoff

Background

Background

Builds on Succinct [NSDI'15]

Background

Builds on Succinct [NSDI'15]

Succinct stores:

Background

Builds on Succinct [NSDI'15]

Succinct stores:

Background

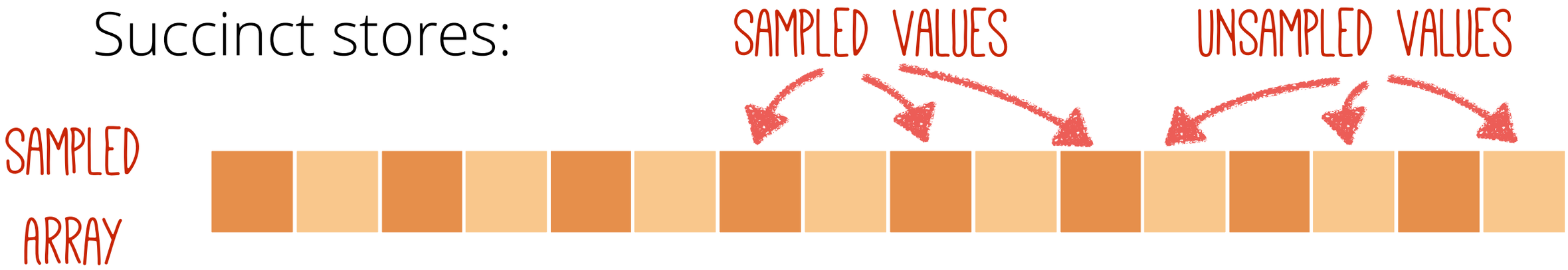
Builds on Succinct [NSDI'15]

Succinct stores:

Background

Builds on Succinct [NSDI'15]

Succinct stores:

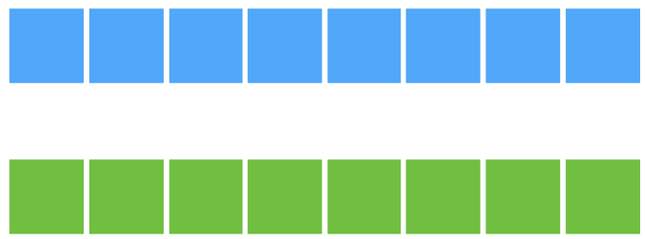


Background

Builds on Succinct [NSDI'15]

Succinct stores:

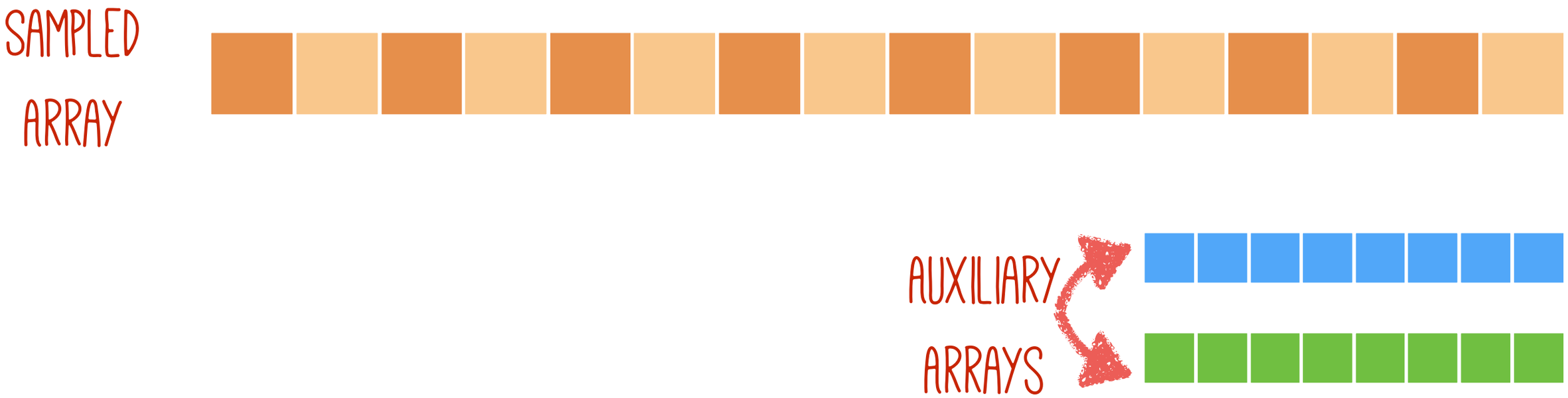
SAMPLED ARRAY



Background

Builds on Succinct [NSDI'15]

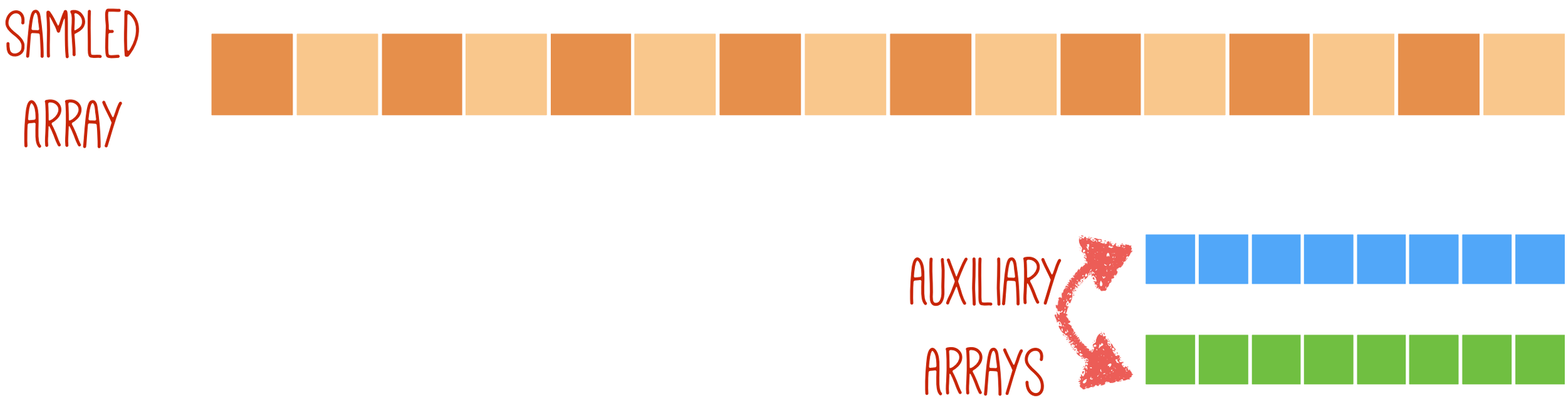
Succinct stores:



Background

Builds on Succinct [NSDI'15]

Succinct stores:

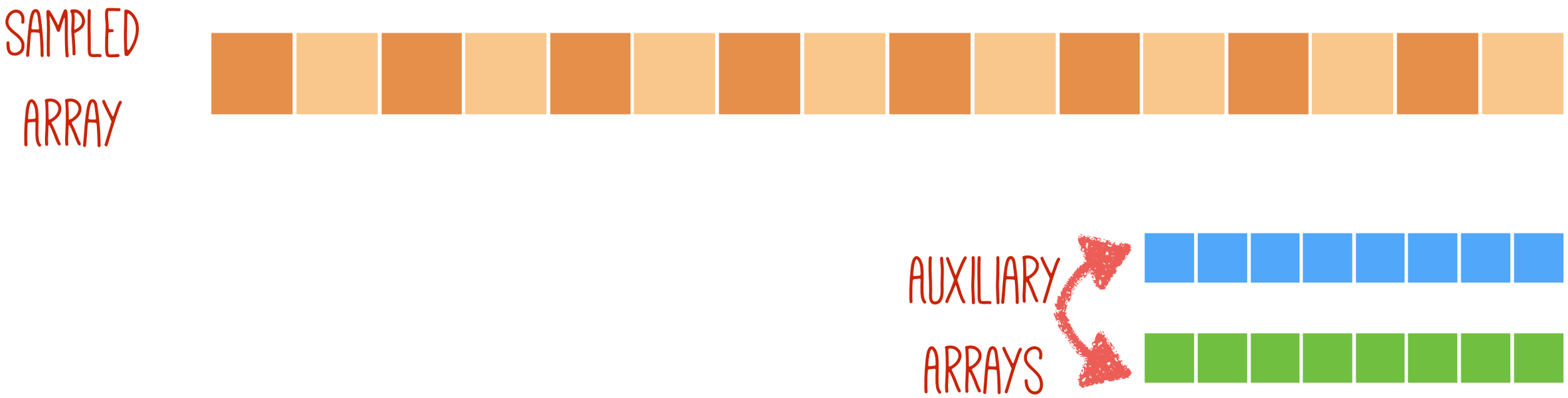


- Small

Background

Builds on Succinct [NSDI'15]

Succinct stores:

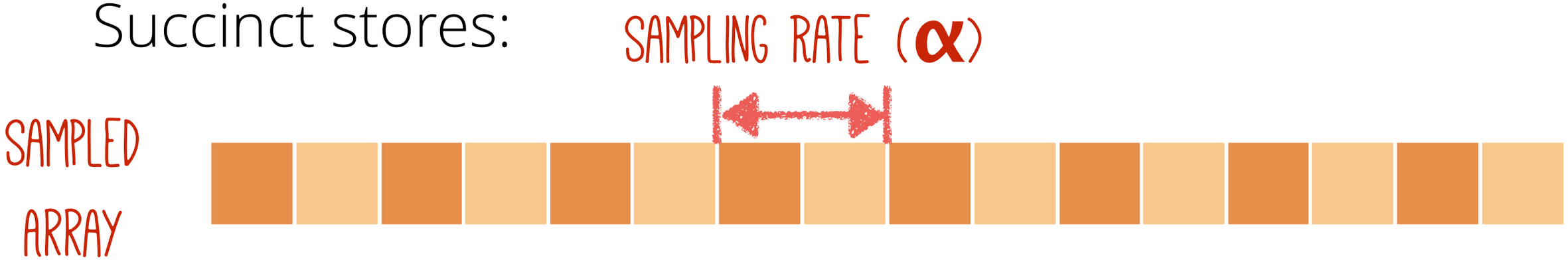


- Small
- Compute unsampled values on the fly

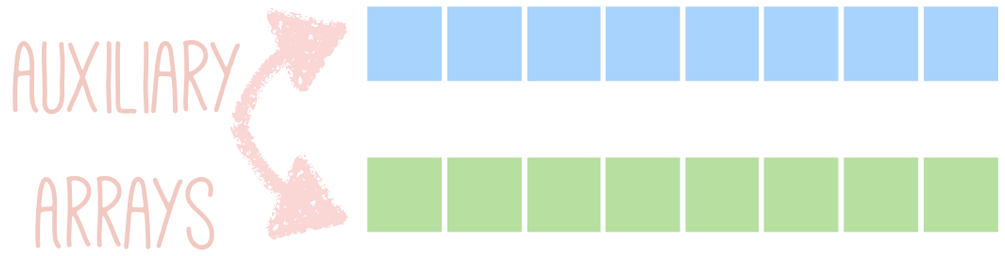
Background

Builds on Succinct [NSDI'15]

Succinct stores:



Sampling Rate proxy for
Storage & Performance

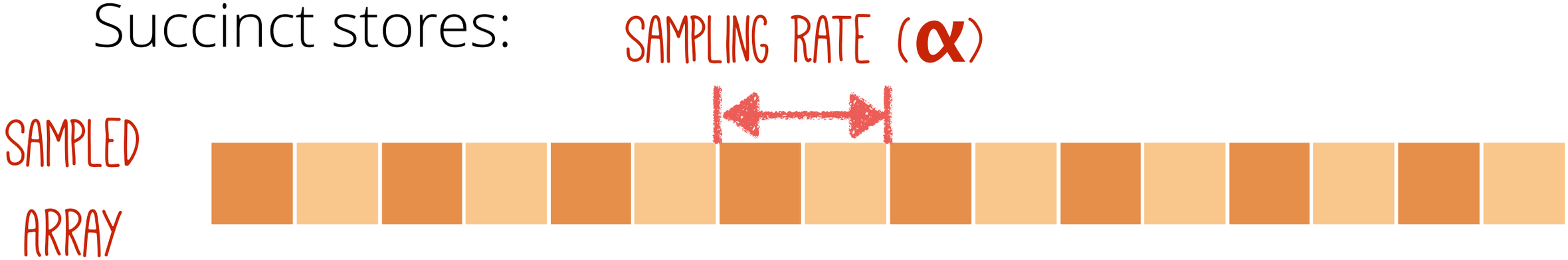


- Small
- Compute unsampled values on the fly

Background

Builds on Succinct [NSDI'15]

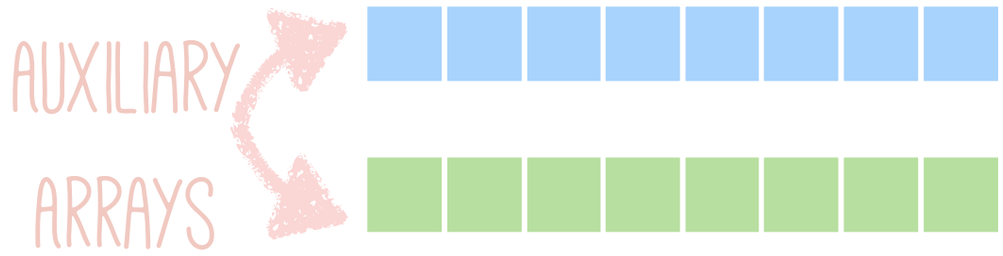
Succinct stores:



Sampling Rate proxy for
Storage & Performance

Storage ≈ OriginalSize/α

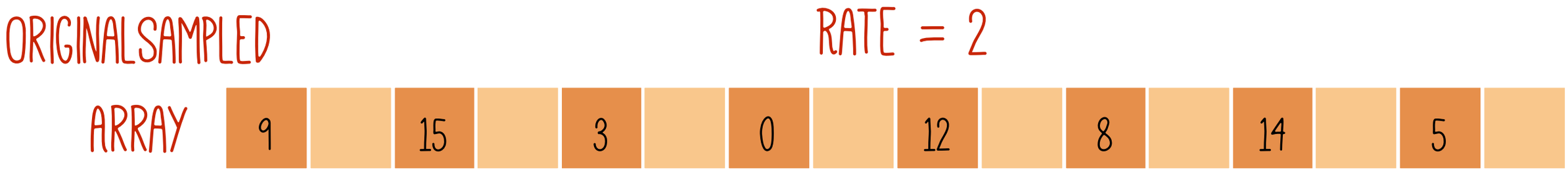
Latency ≈ α



- Small
- Compute unsampled values on the fly

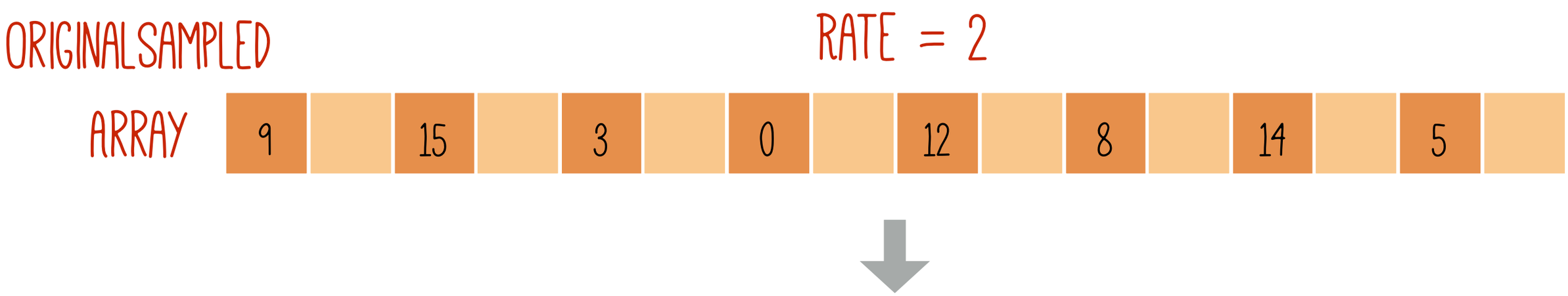
Layered Sampled Array

Inspired by multi-layered video encoding techniques



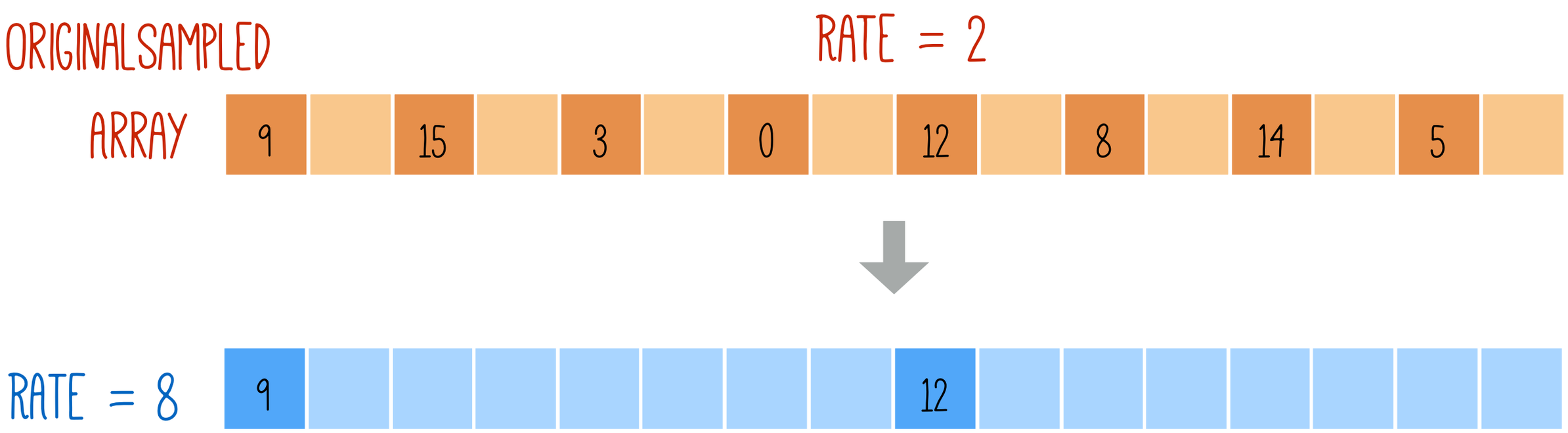
Layered Sampled Array

Inspired by multi-layered video encoding techniques



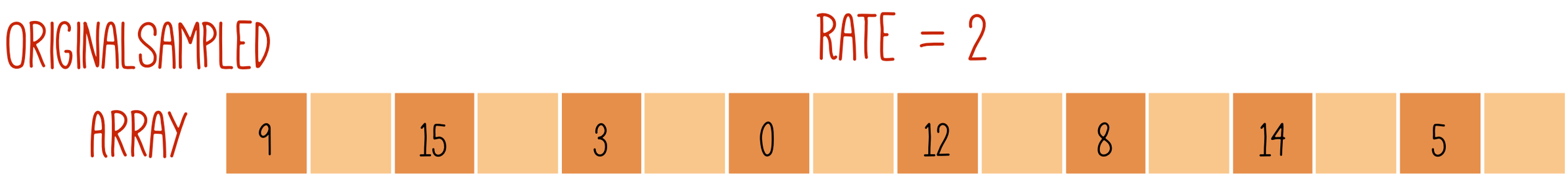
Layered Sampled Array

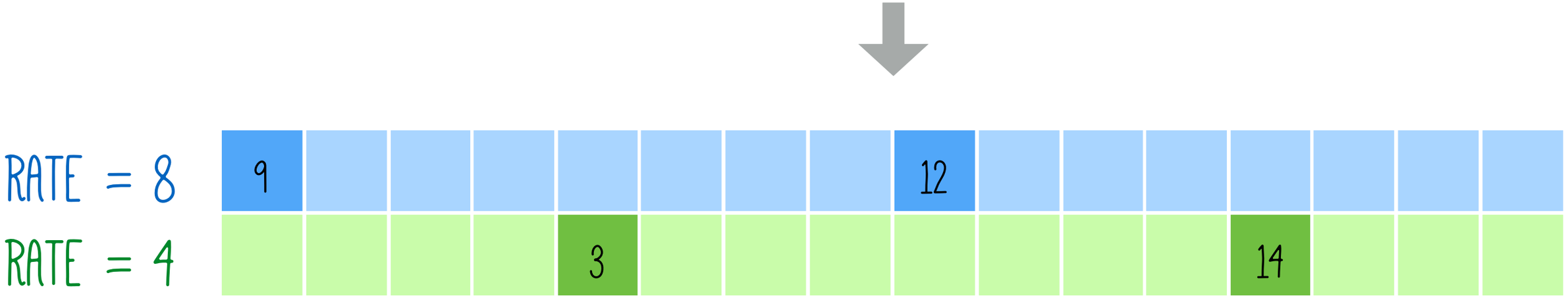
Inspired by multi-layered video encoding techniques



Layered Sampled Array

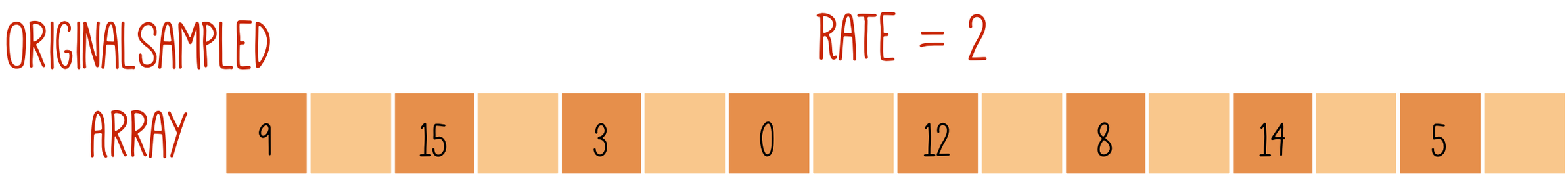
Inspired by multi-layered video encoding techniques

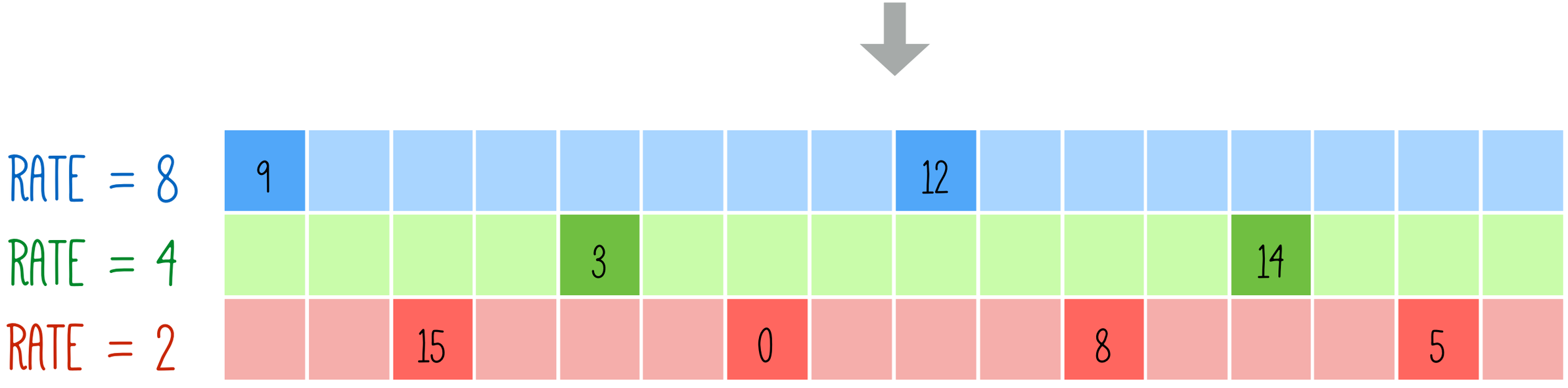




Layered Sampled Array

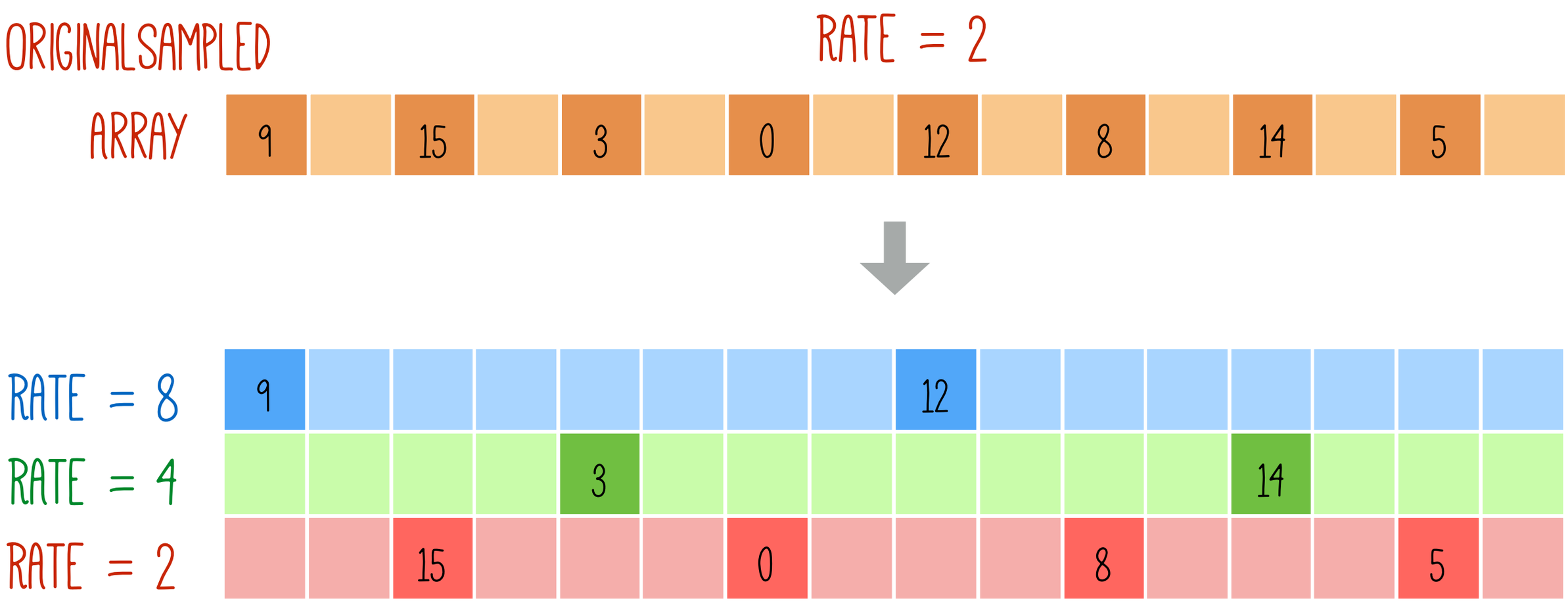
Inspired by multi-layered video encoding techniques





Layered Sampled Array

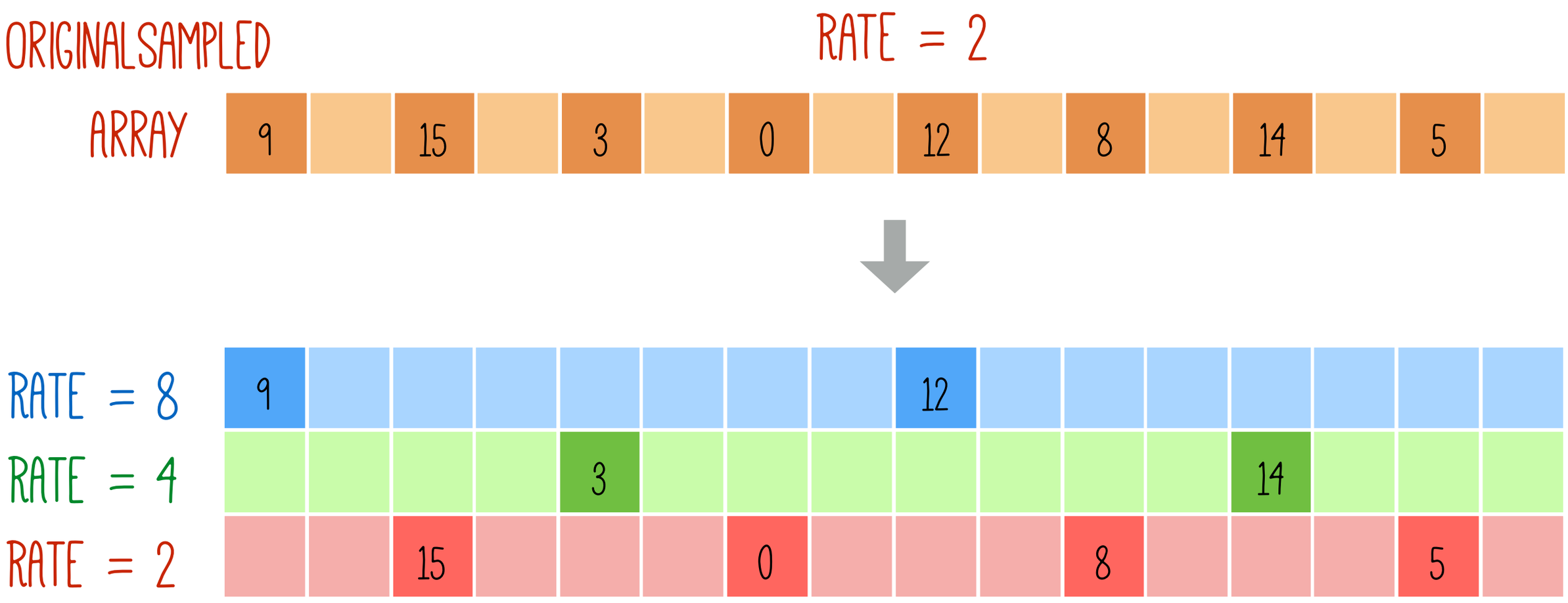
Inspired by multi-layered video encoding techniques



Different combination of layers

Layered Sampled Array

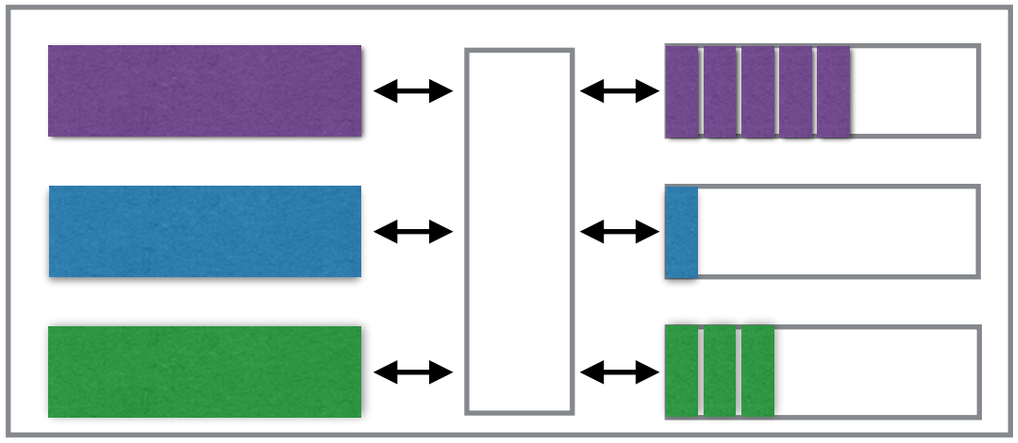
Inspired by multi-layered video encoding techniques



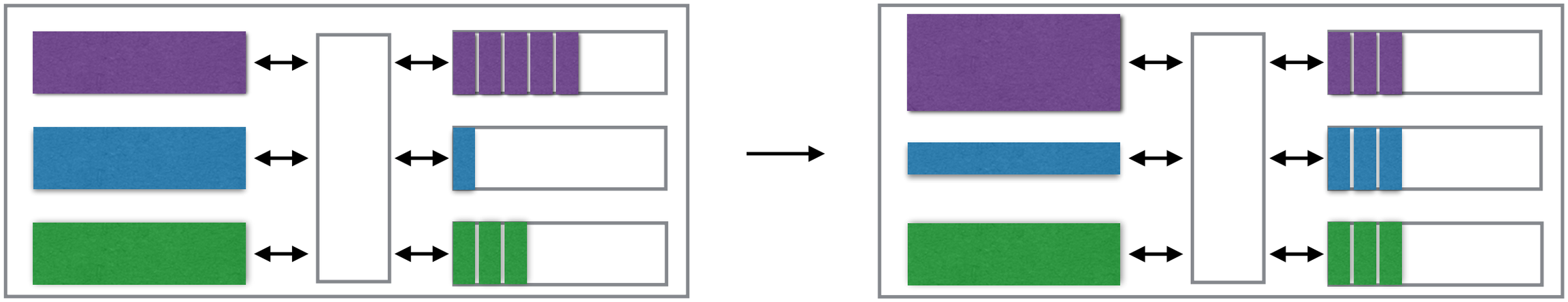
Different combination of layers → **Different points** on tradeoff curve

Technical **Details**

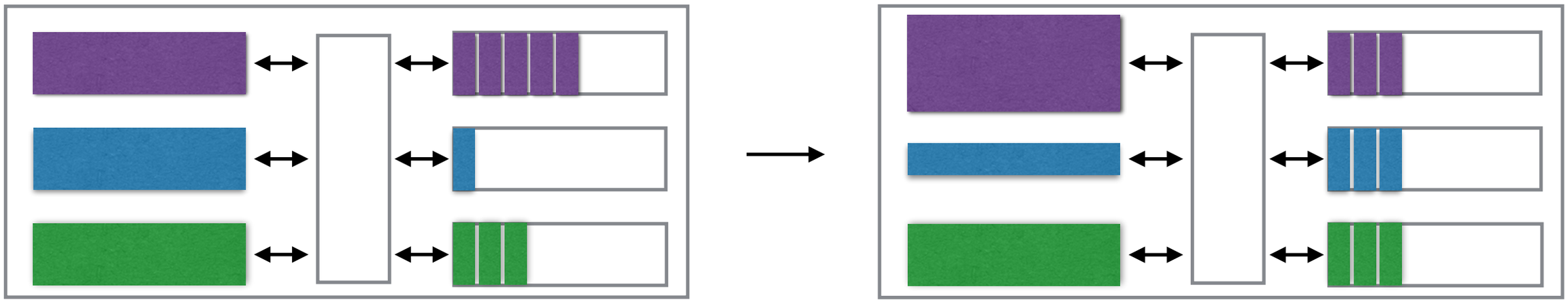
Technical **Details**



Technical **Details**

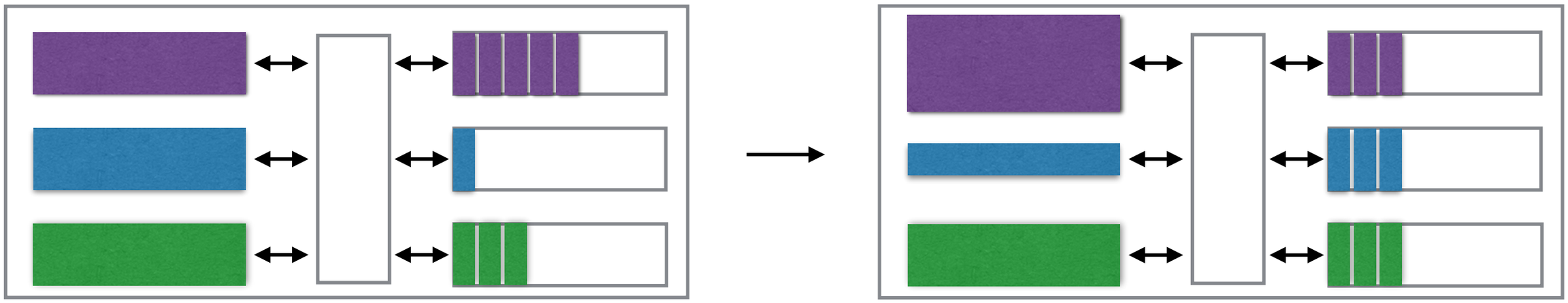


Technical **Details**



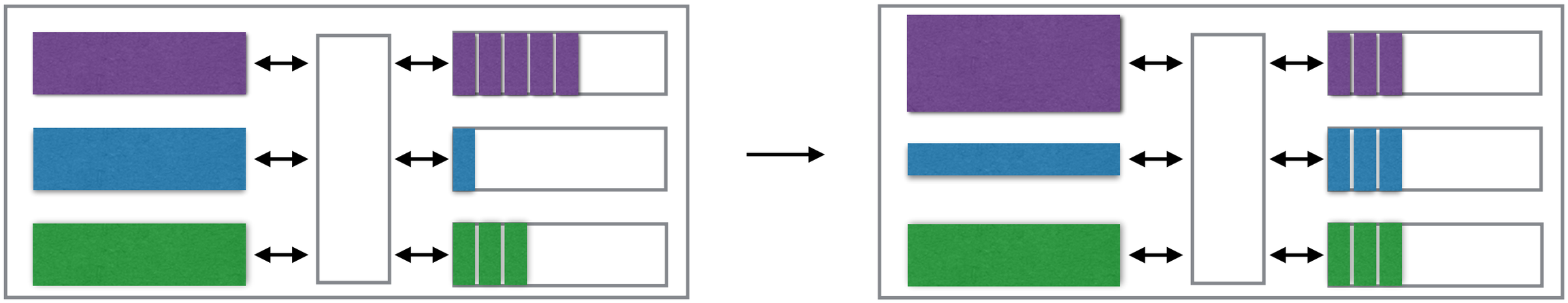
- How should partitions share cache on a server?

Technical **Details**



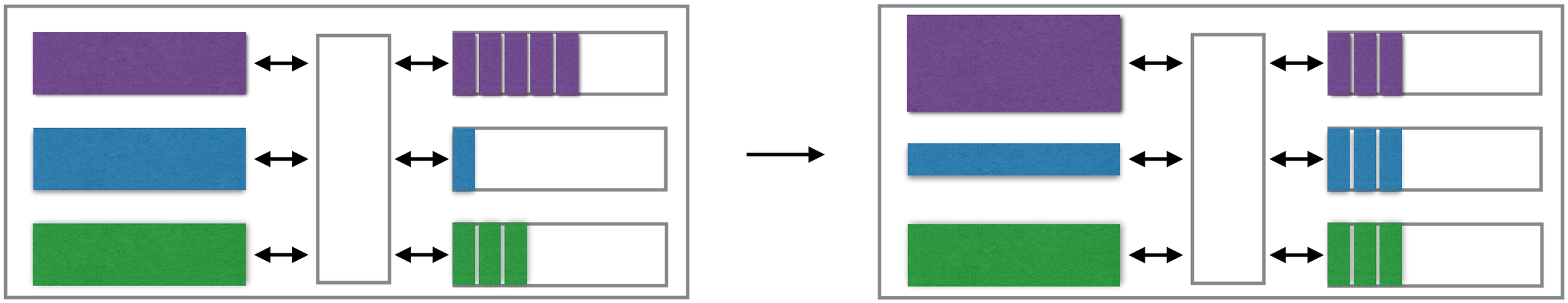
- How should partitions share cache on a server?
- How should partitions share cache across servers?

Technical **Details**



- How should partitions share cache on a server?
- How should partitions share cache across servers?
- How should requests be scheduled across replicas?

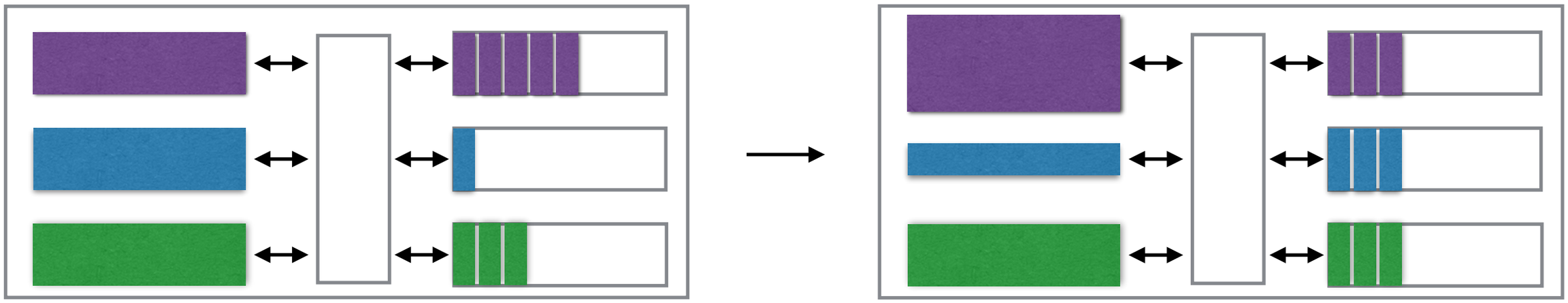
Technical **Details**



- How should partitions share cache on a server?
- How should partitions share cache across servers?
- How should requests be scheduled across replicas?

Unified Solution: Back-pressure style scheduling

Technical **Details**

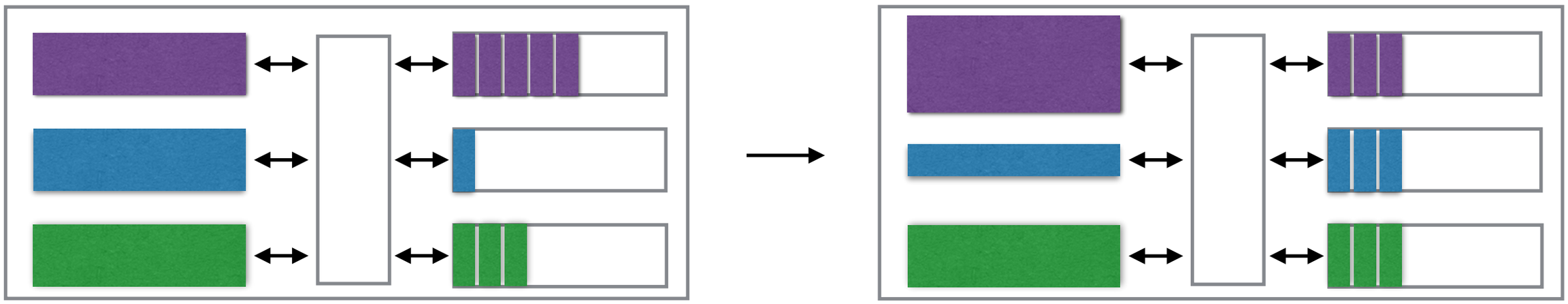


- How should partitions share cache on a server?
- How should partitions share cache across servers?
- How should requests be scheduled across replicas?

Unified Solution: Back-pressure style scheduling

Cache proportional to load,

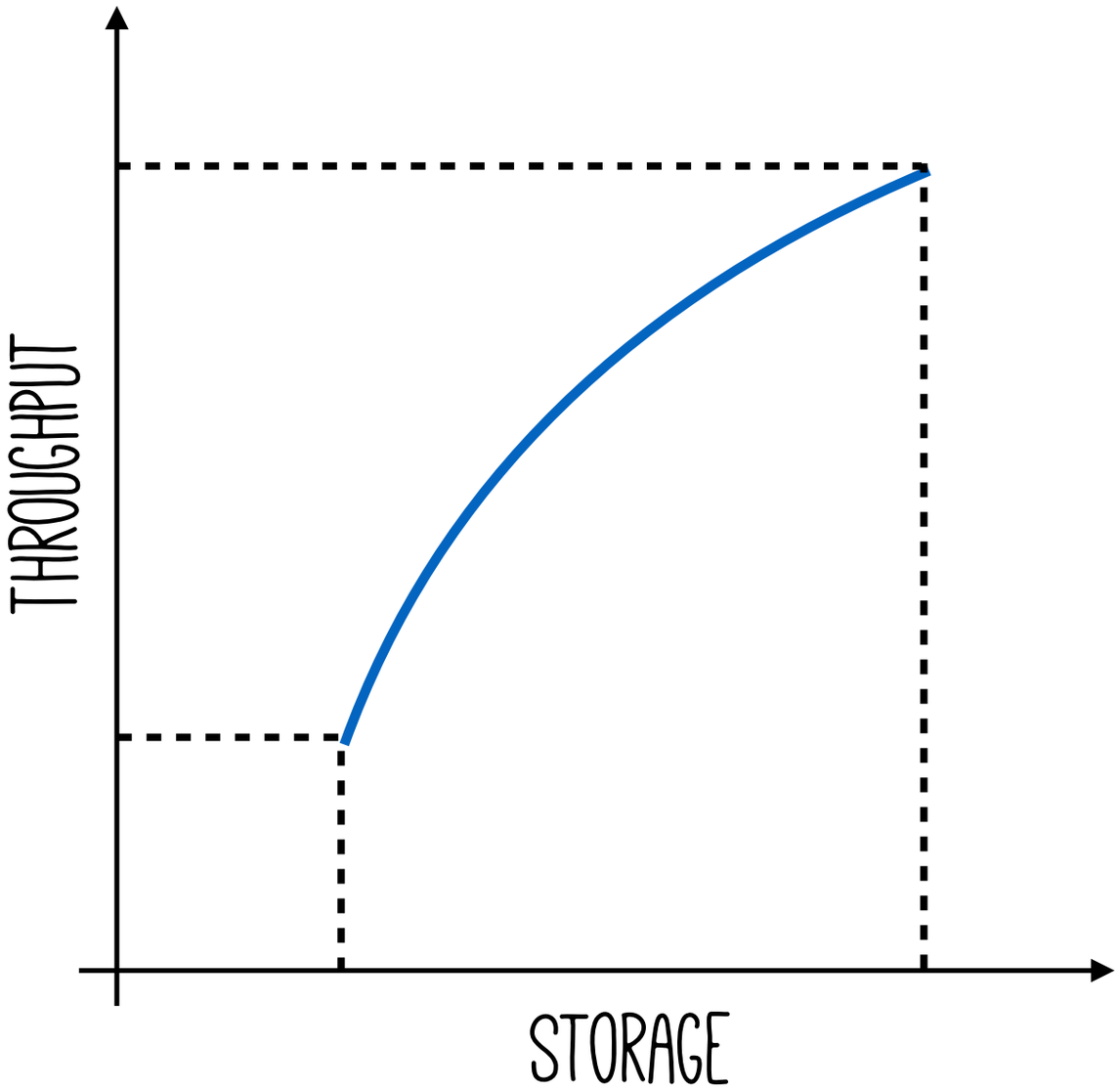
Technical **Details**

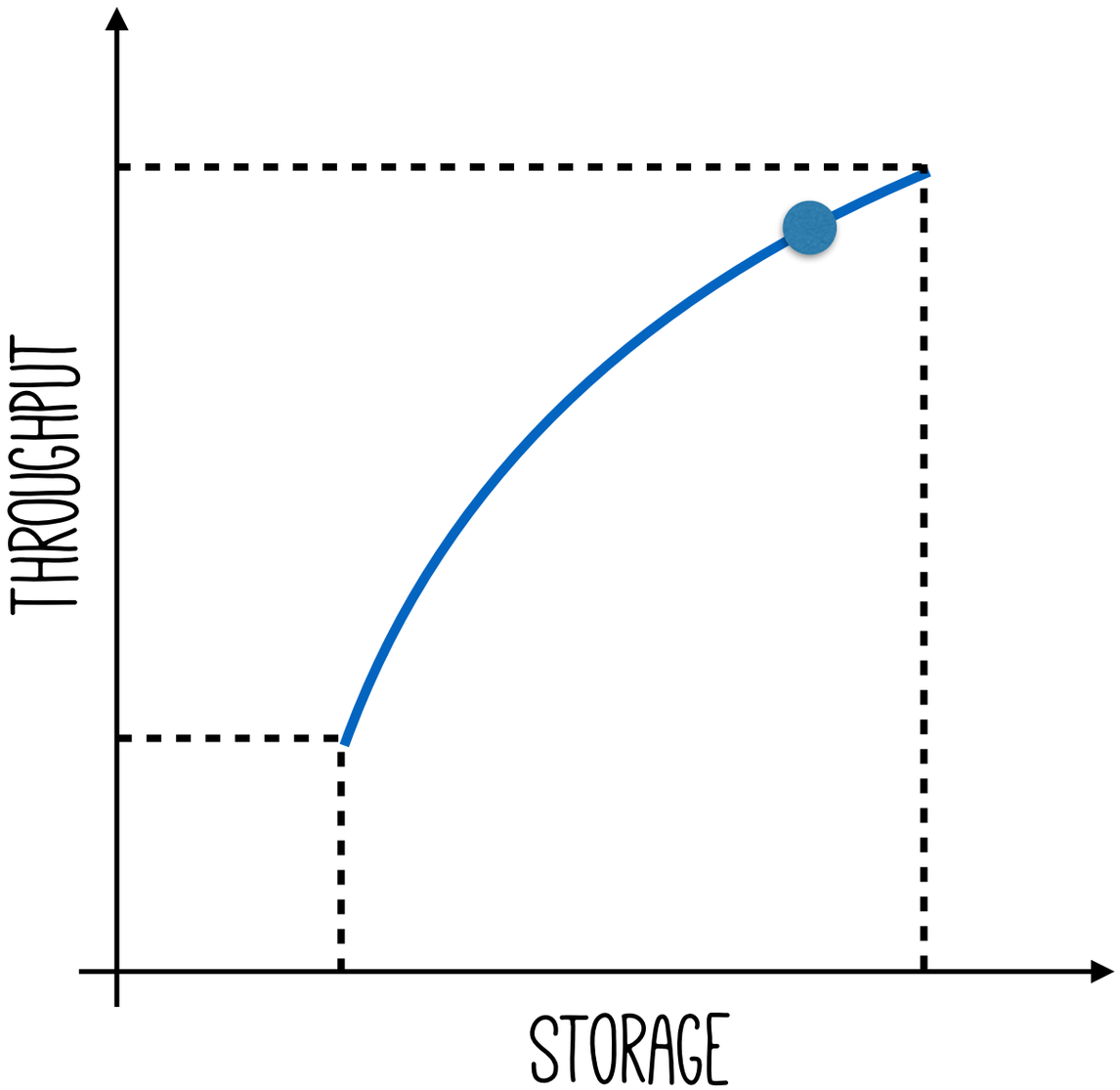


- How should partitions share cache on a server?
- How should partitions share cache across servers?
- How should requests be scheduled across replicas?

Unified Solution: Back-pressure style scheduling

Cache proportional to load, without explicit coordination


THROUGHPUT
STORAGE

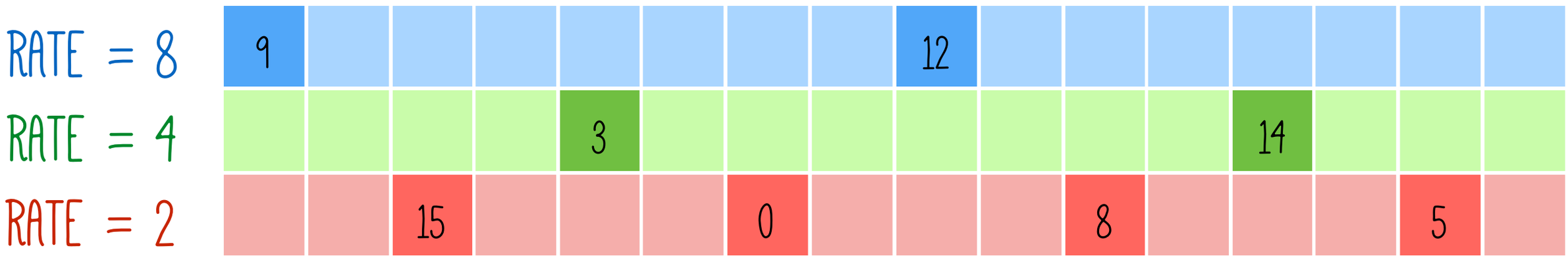


Dynamic Navigation

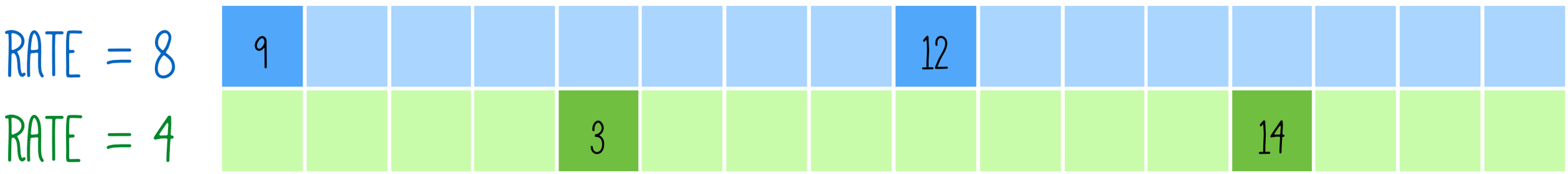
of tradeoff curve

Layer Additions & Deletions

Layer Additions & Deletions

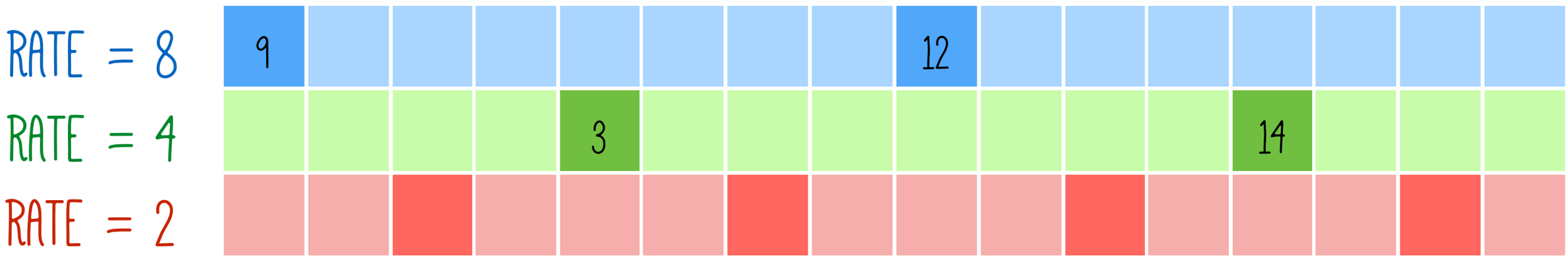


Layer Additions & Deletions



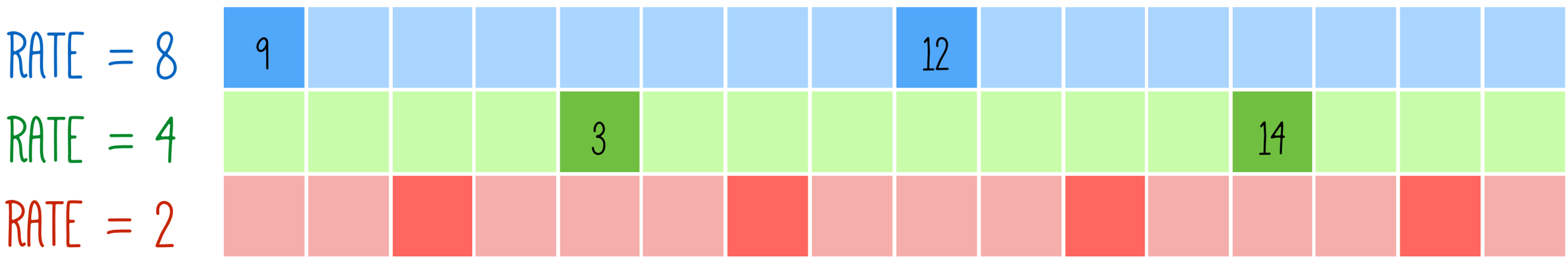
Layer Deletions: simple

Layer Additions & Deletions



Layer Addition:

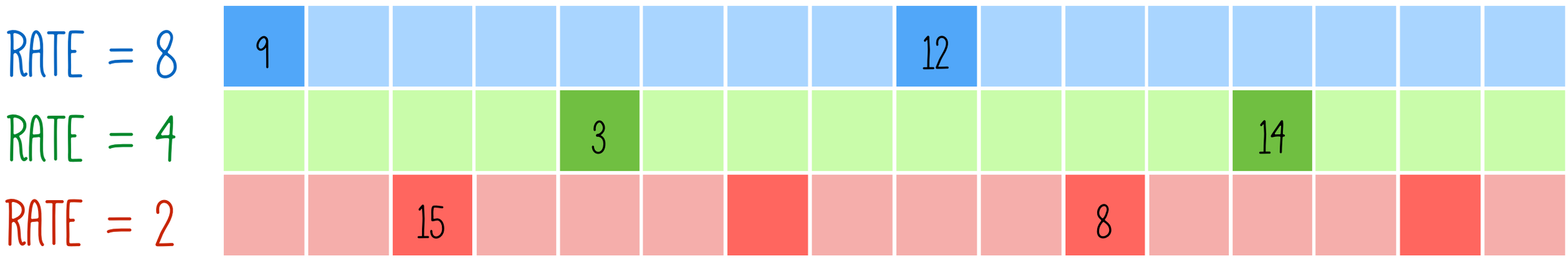
Layer Additions & Deletions



Layer Addition:

Unsampled values already computed during query execution

Layer Additions & Deletions



Layer Addition:

Unsampled values already computed during query execution

Layers in LSA populated *opportunistically*!!

Assumptions & Limitations

Assumptions & Limitations

- Functionality close to state-of-the-art NoSQL stores

Assumptions & Limitations

- Functionality close to state-of-the-art NoSQL stores

`get()` `put()` `delete()` `search()` `regex()`

Assumptions & Limitations

- Functionality close to state-of-the-art NoSQL stores

`get()` `put()` `delete()` `search()` `regex()`

- Queries do not touch all servers

Assumptions & Limitations

- Functionality close to state-of-the-art NoSQL stores

`get()` `put()` `delete()` `search()` `regex()`

- Queries do not touch all servers

Many systems employ sharding schemes to avoid touching all servers, e.g., [Schism, VLDB'10]

Assumptions & Limitations

- Functionality close to state-of-the-art NoSQL stores

`get()` `put()` `delete()` `search()` `regex()`

- Queries do not touch all servers

  Many systems employ sharding schemes to avoid touching all servers, e.g., [Schism, VLDB'10]

- System is not network-bottlenecked

Assumptions & Limitations

- Functionality close to state-of-the-art NoSQL stores

`get()` `put()` `delete()` `search()` `regex()`

- Queries do not touch all servers

  Many systems employ sharding schemes to avoid touching all servers, e.g., [Schism, VLDB'10]

- System is not network-bottlenecked

  [MICA, NSDI'14] → True for most data stores today

Applications

Applications

Look at classical systems problems through a new “lens”

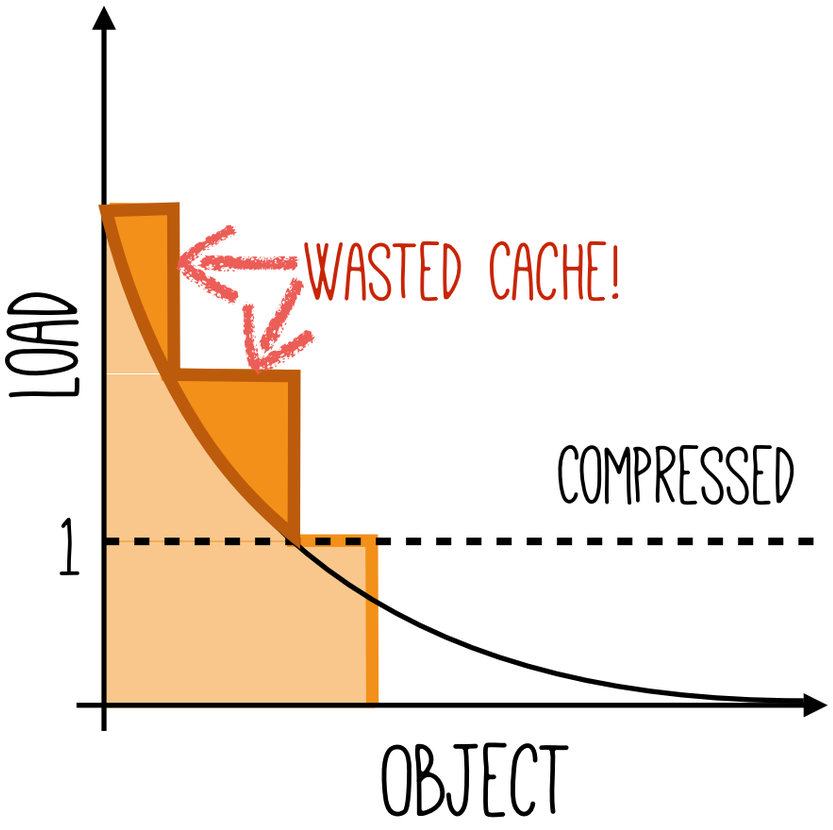
Spatial Skew

Spatial Skew

Load distribution across partitions is heavily skewed

Spatial Skew

Load distribution across partitions is heavily skewed

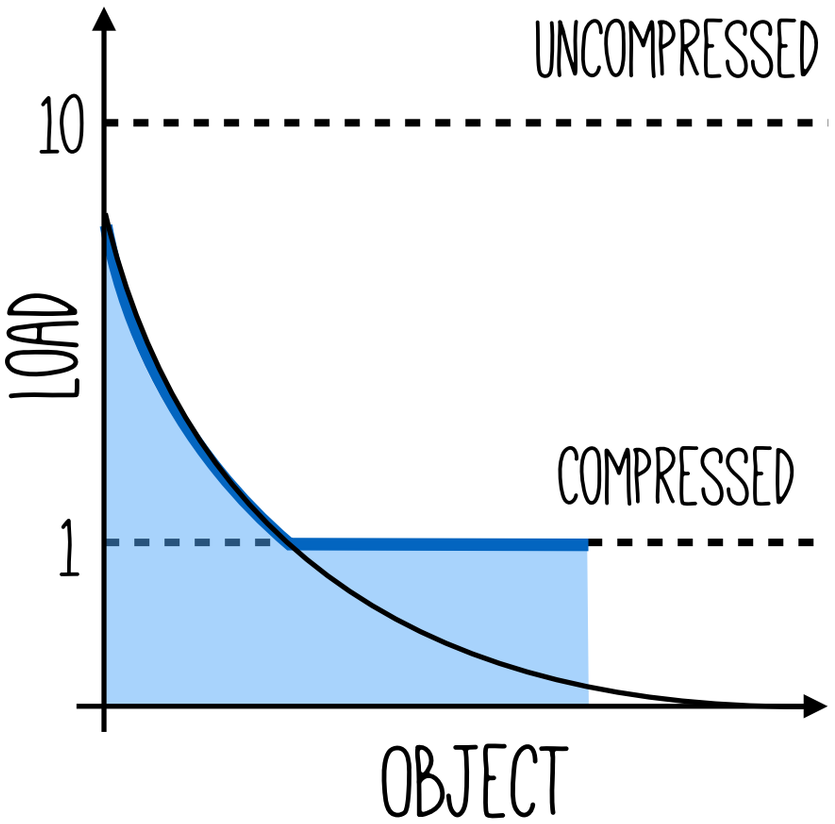


Selective Replication

#Replicas **α** Load

Spatial Skew

Load distribution across partitions is heavily skewed



Selective Replication

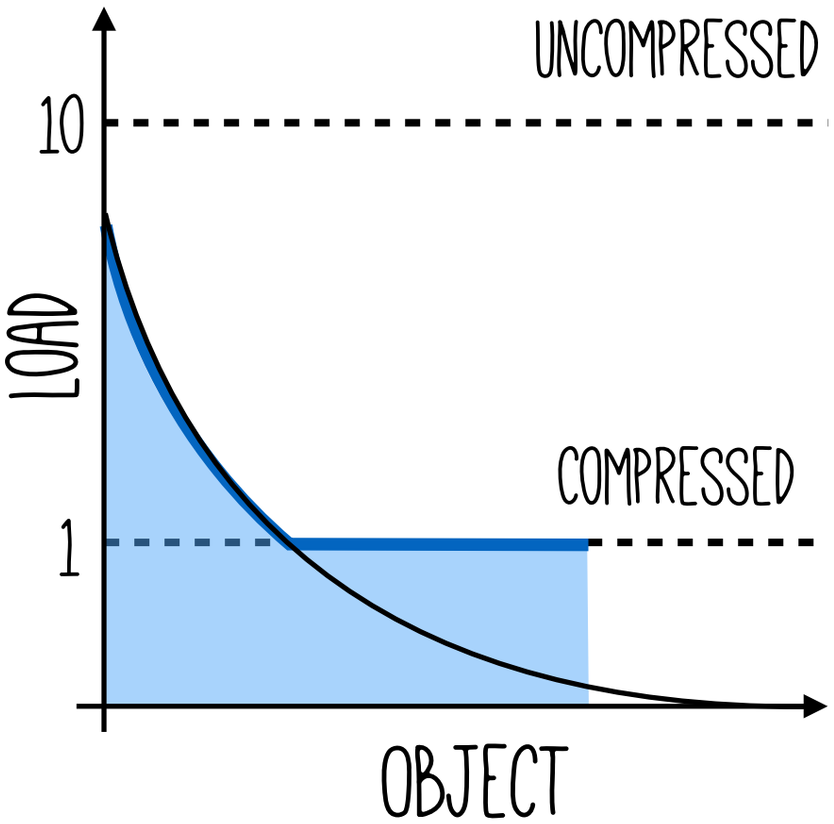
#Replicas **α** Load

BlowFish

Fractionally change storage just enough to meet load

Spatial Skew

Load distribution across partitions is heavily skewed



Selective Replication

#Replicas **α** Load

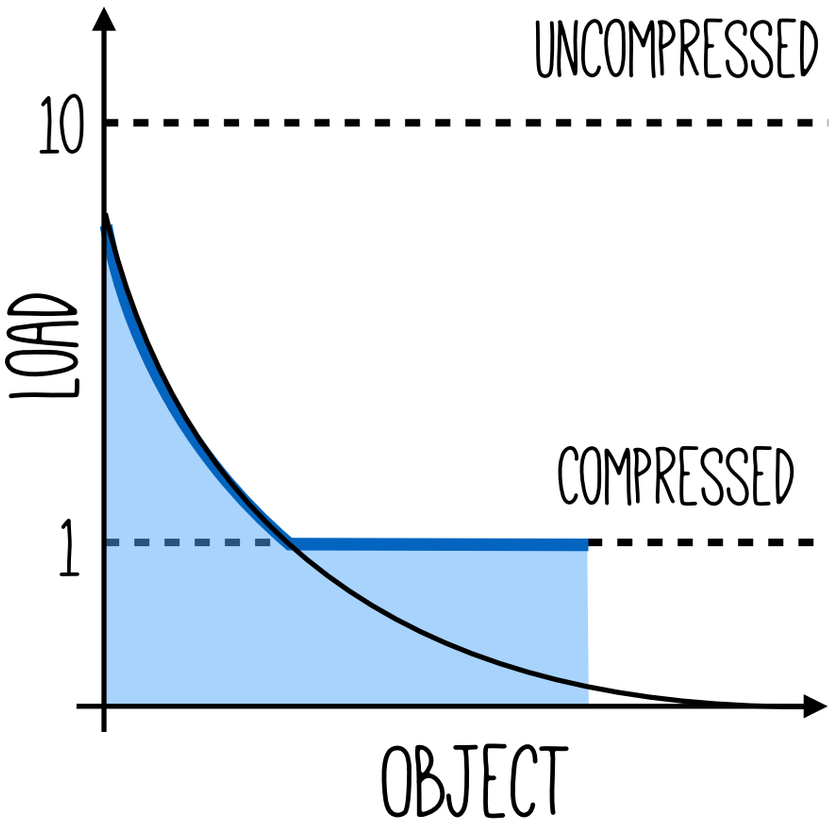
BlowFish

Fractionally change storage just enough to meet load

1.5x higher throughput than Selective Replication,

Spatial Skew

Load distribution across partitions is heavily skewed



Selective Replication

#Replicas **α** Load

BlowFish

Fractionally change storage just enough to meet load

1.5x higher throughput than Selective Replication,

within 10% of optimal

Changes in Spatial Skew

Changes in Spatial Skew

Study on Facebook Warehouse Cluster [HotStorage'13]

Changes in Spatial Skew

Study on Facebook Warehouse Cluster [HotStorage'13]

Transient failures → 90% of failures

Changes in Spatial Skew

Study on Facebook Warehouse Cluster [HotStorage'13]

Transient failures → 90% of failures

Replica creation delayed by 15 mins

Changes in Spatial Skew

Study on Facebook Warehouse Cluster [HotStorage'13]

Transient failures → 90% of failures

Replica creation delayed by 15 mins

Leads to variation in load over time

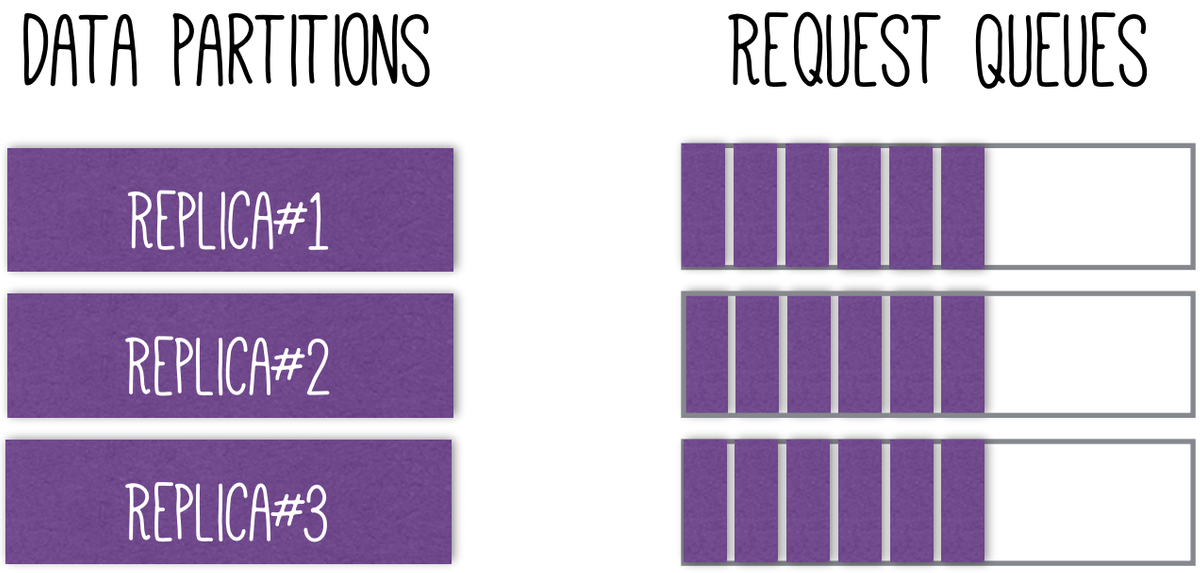
Changes in Spatial Skew

Study on Facebook Warehouse Cluster [HotStorage'13]

Transient failures → 90% of failures

Replica creation delayed by 15 mins

Leads to variation in load over time



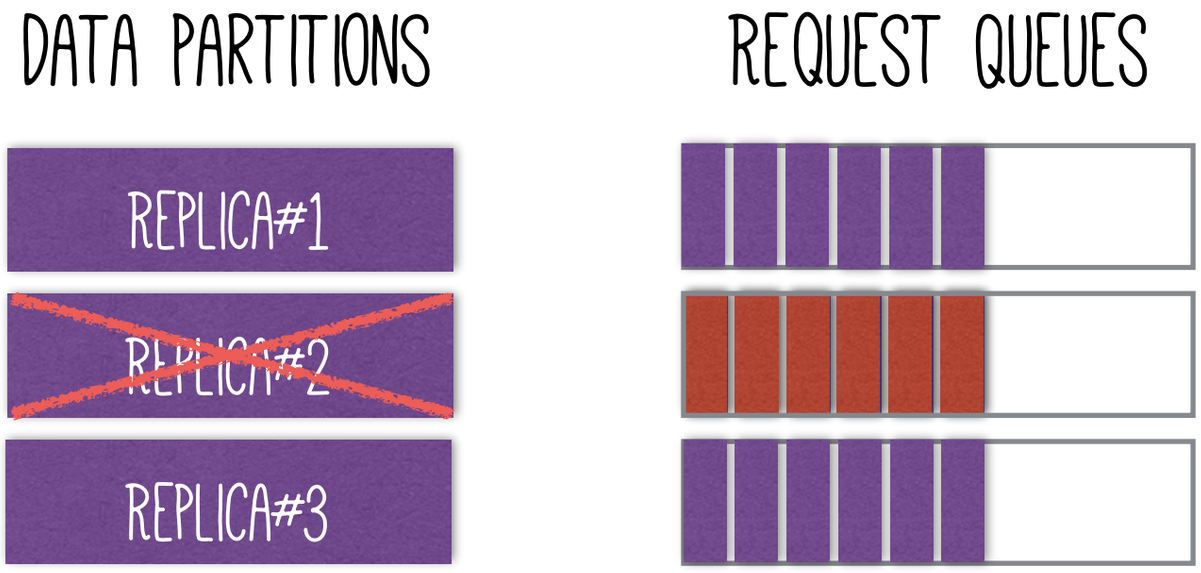
Changes in Spatial Skew

Study on Facebook Warehouse Cluster [HotStorage'13]

Transient failures → 90% of failures

Replica creation delayed by 15 mins

Leads to variation in load over time



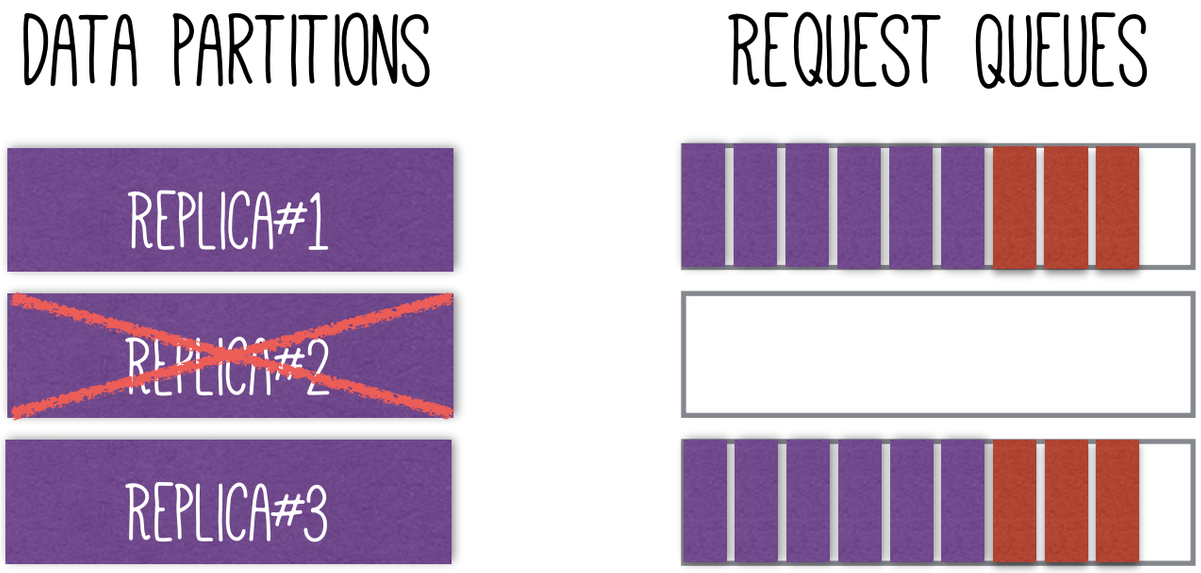
Changes in Spatial Skew

Study on Facebook Warehouse Cluster [HotStorage'13]

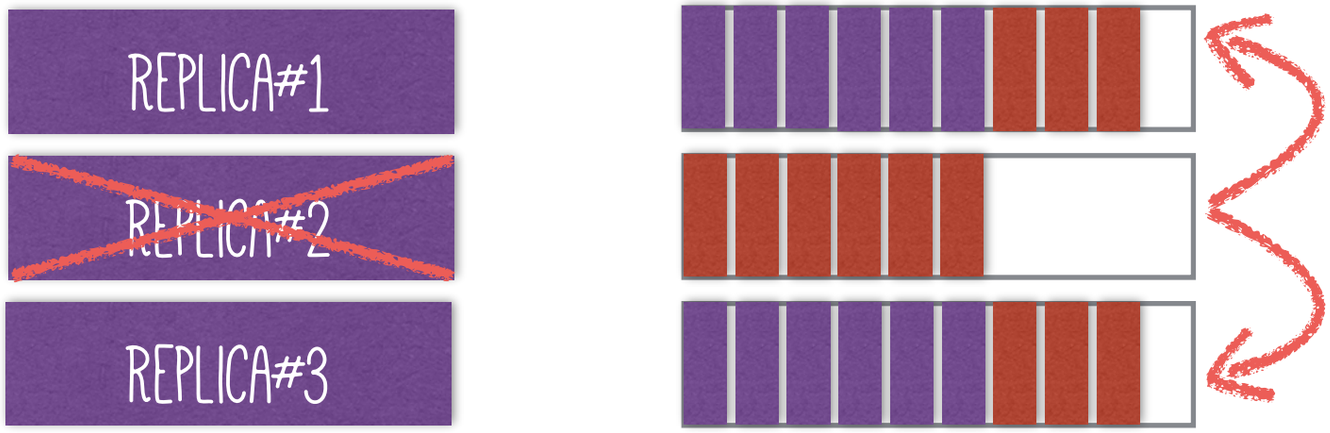
Transient failures → 90% of failures

Replica creation delayed by 15 mins

Leads to variation in load over time

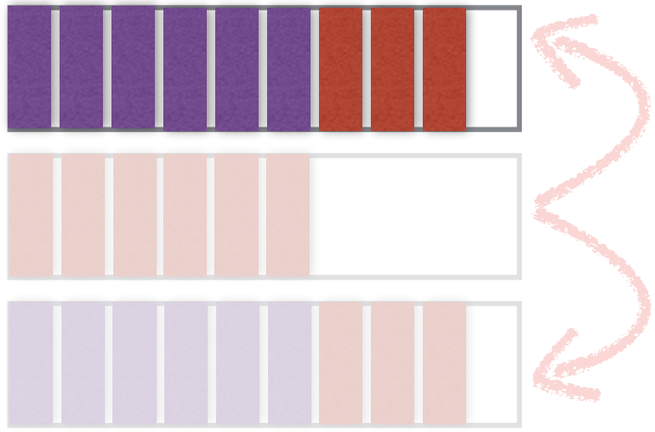


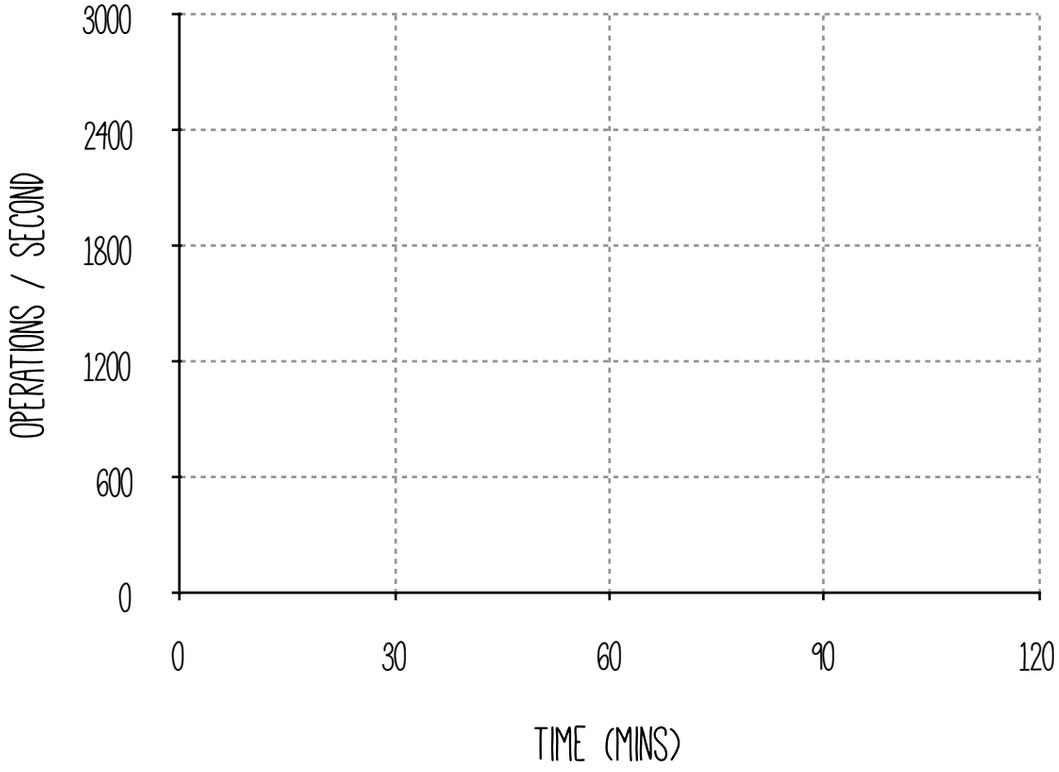
Changes in Spatial Skew



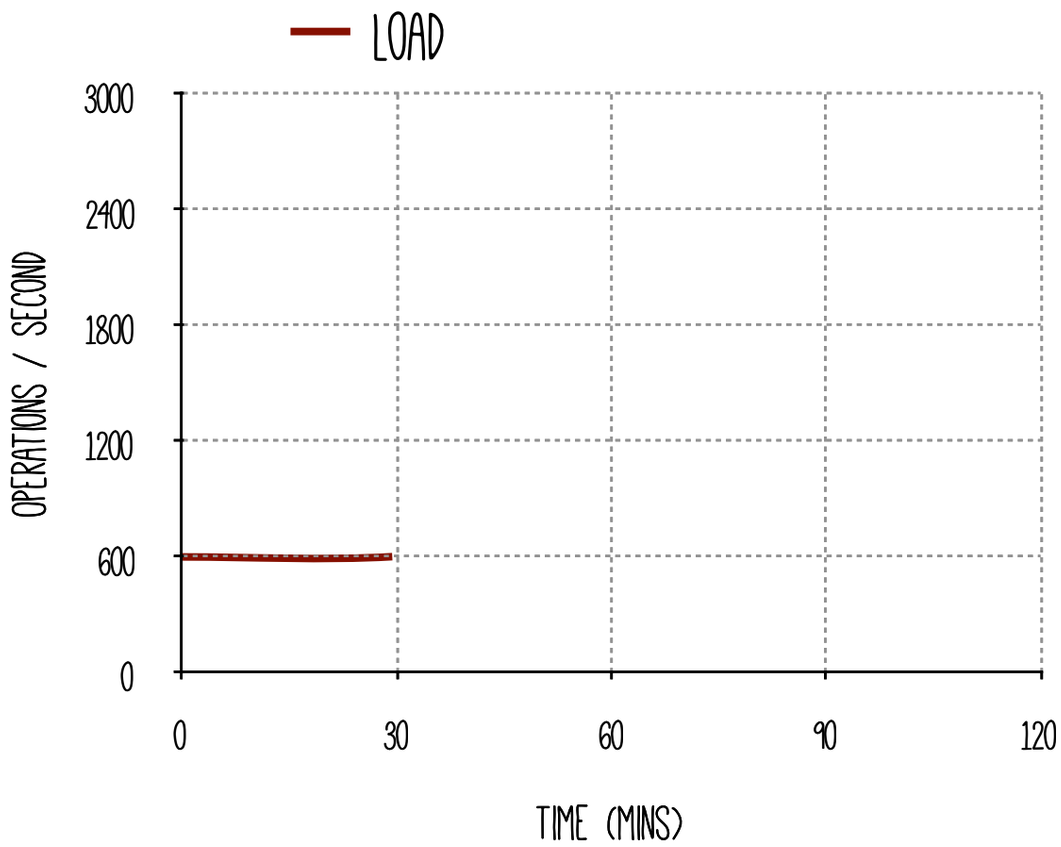
Changes in Spatial Skew

Changes in Spatial Skew

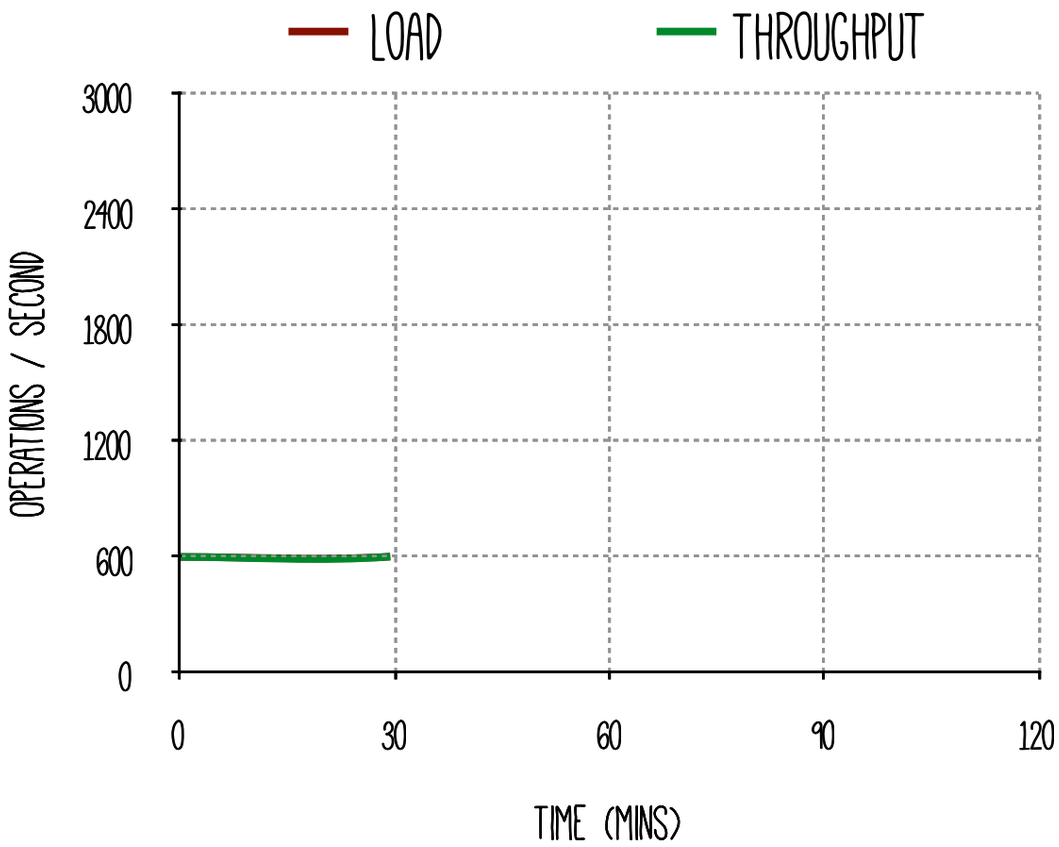




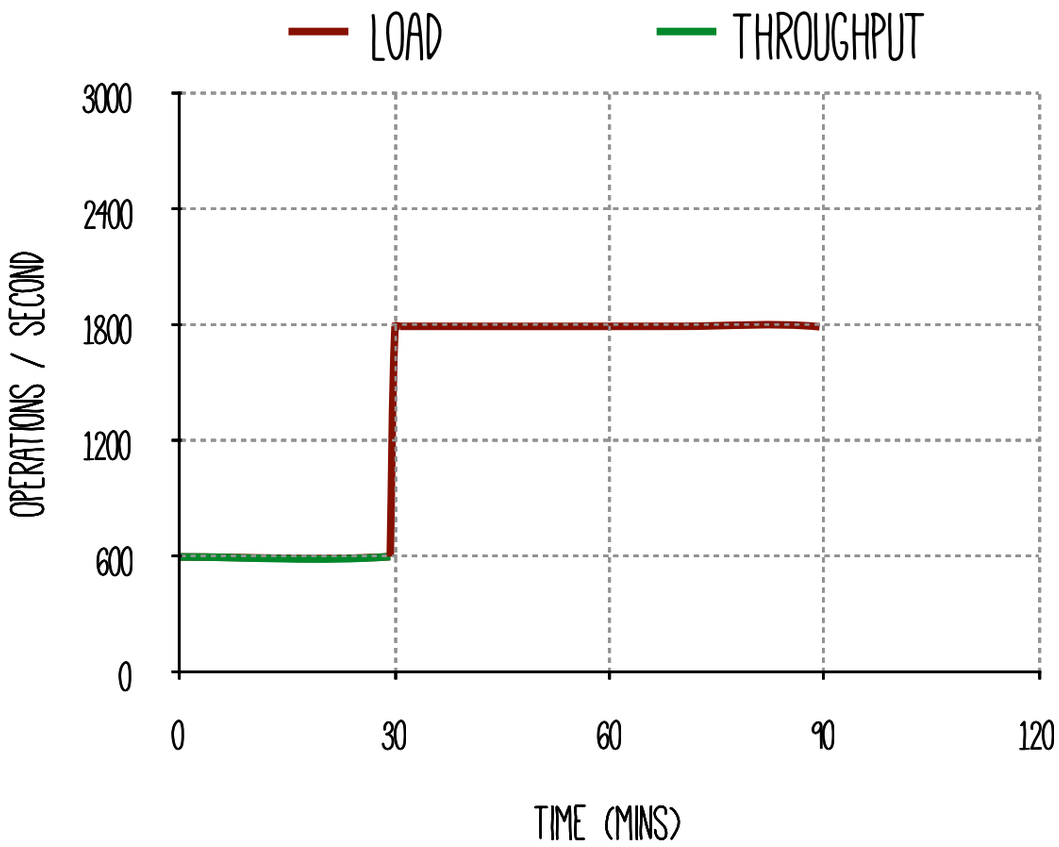
Changes in Spatial Skew



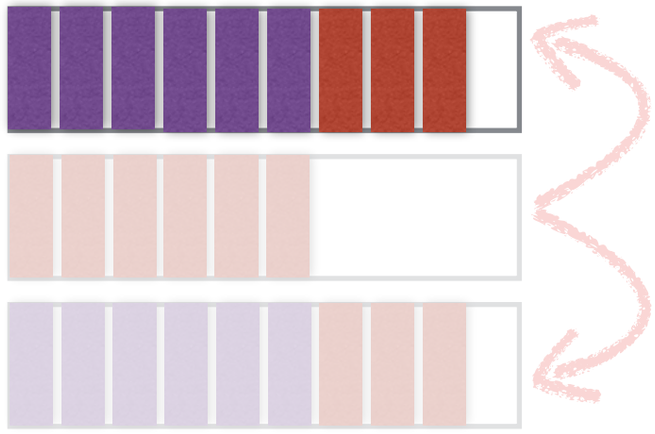
Changes in Spatial Skew

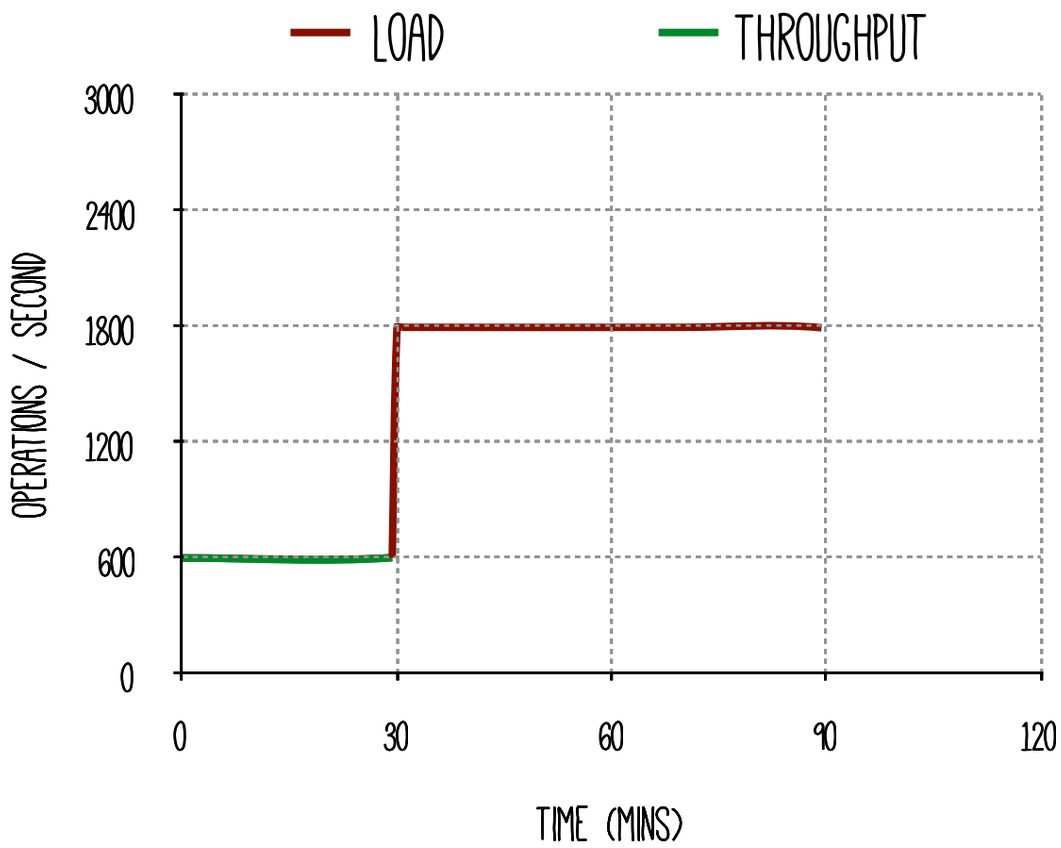


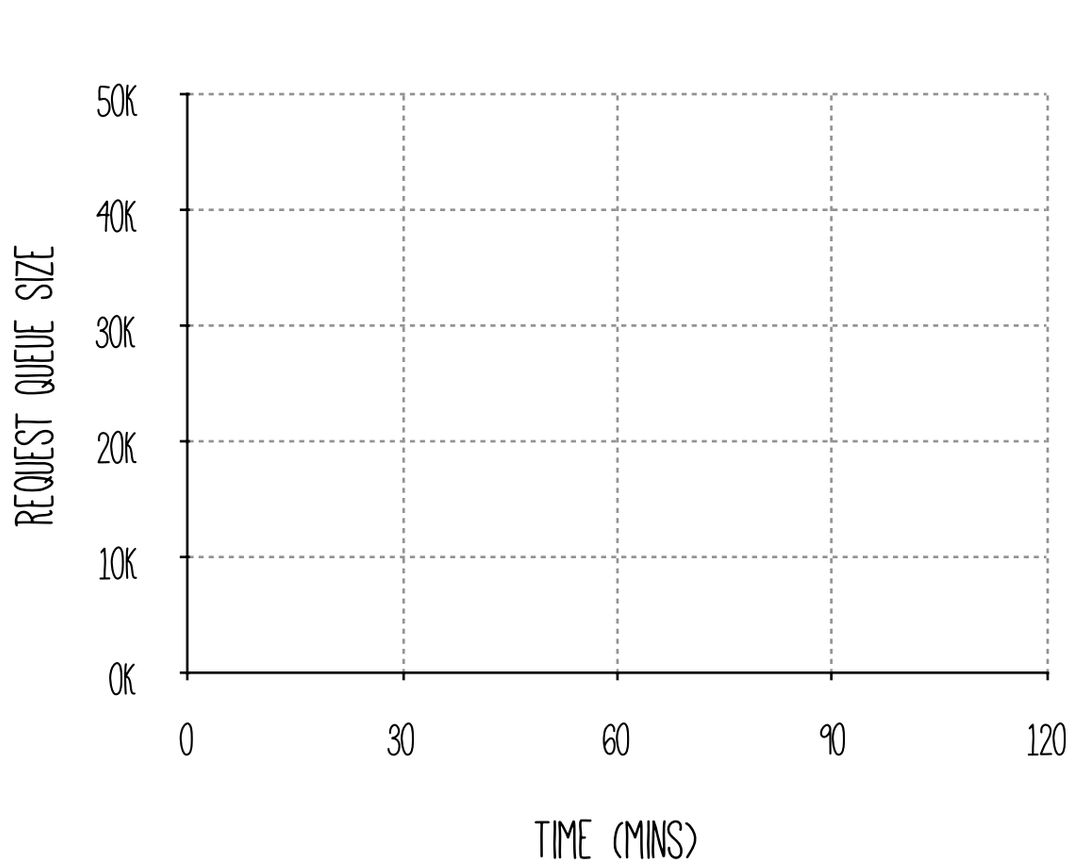
Changes in Spatial Skew



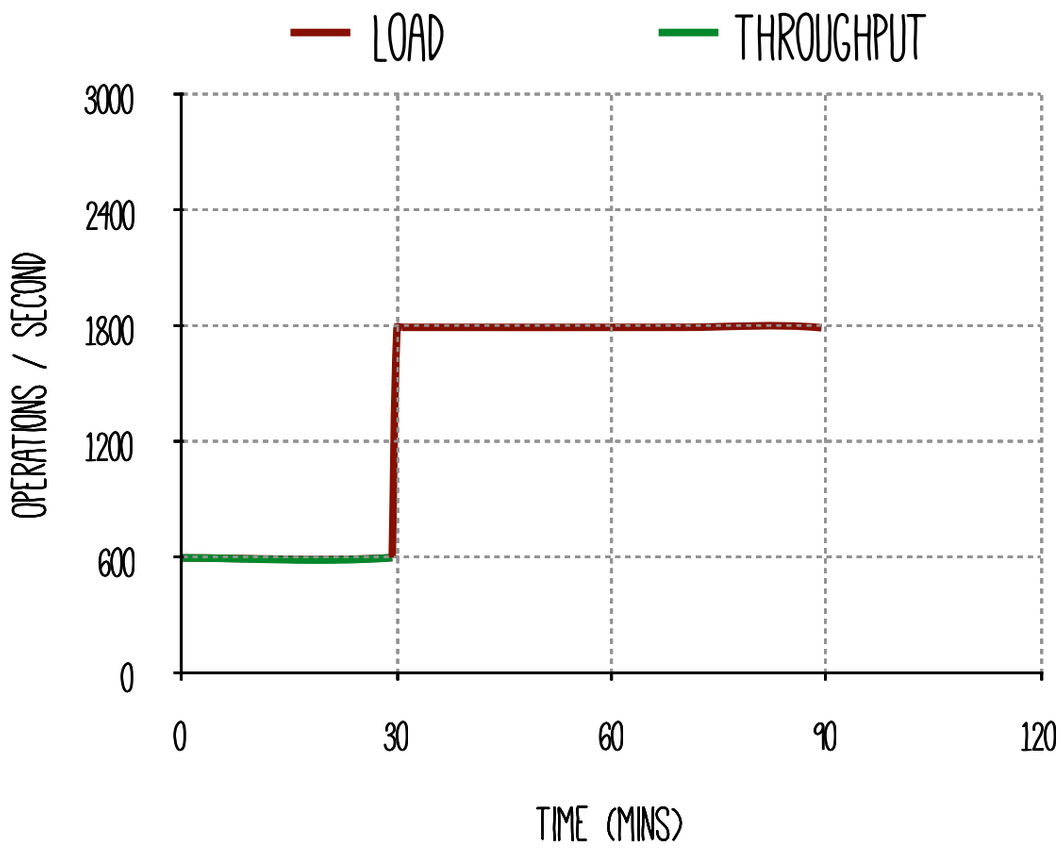
Changes in Spatial Skew

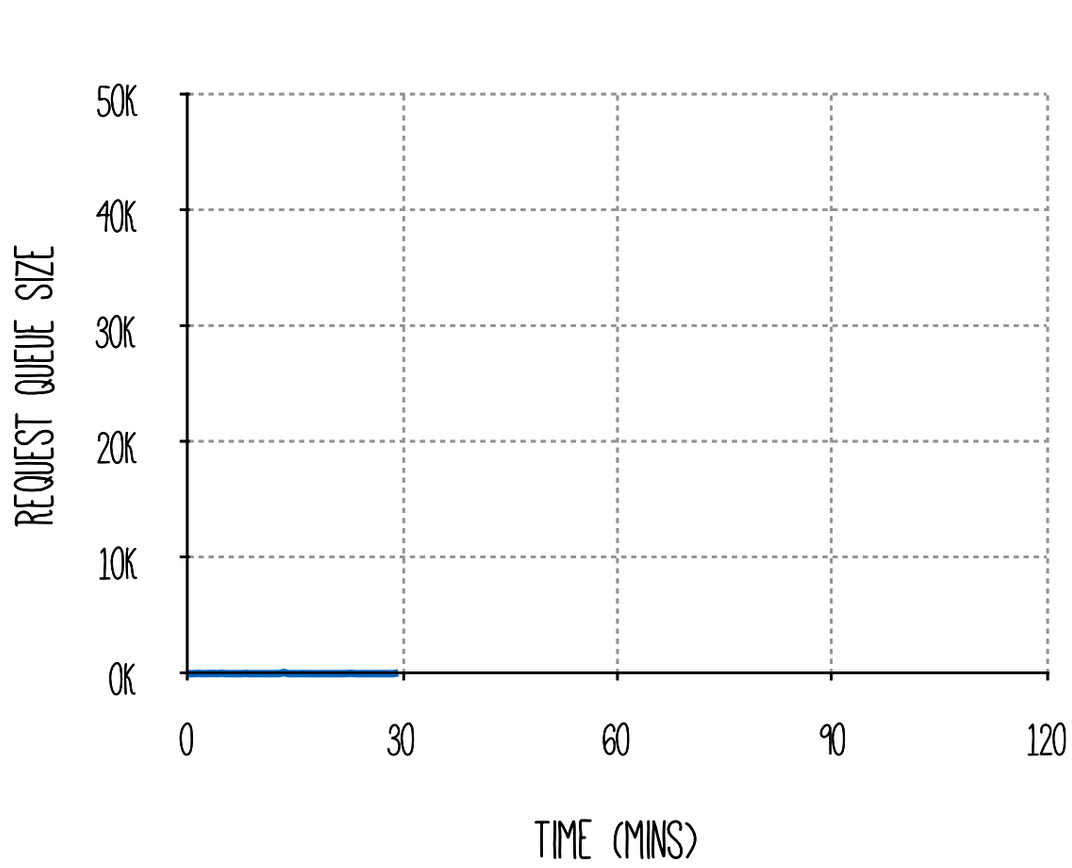




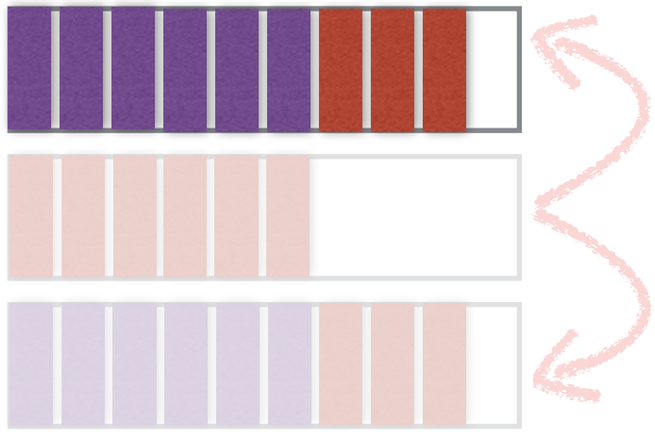


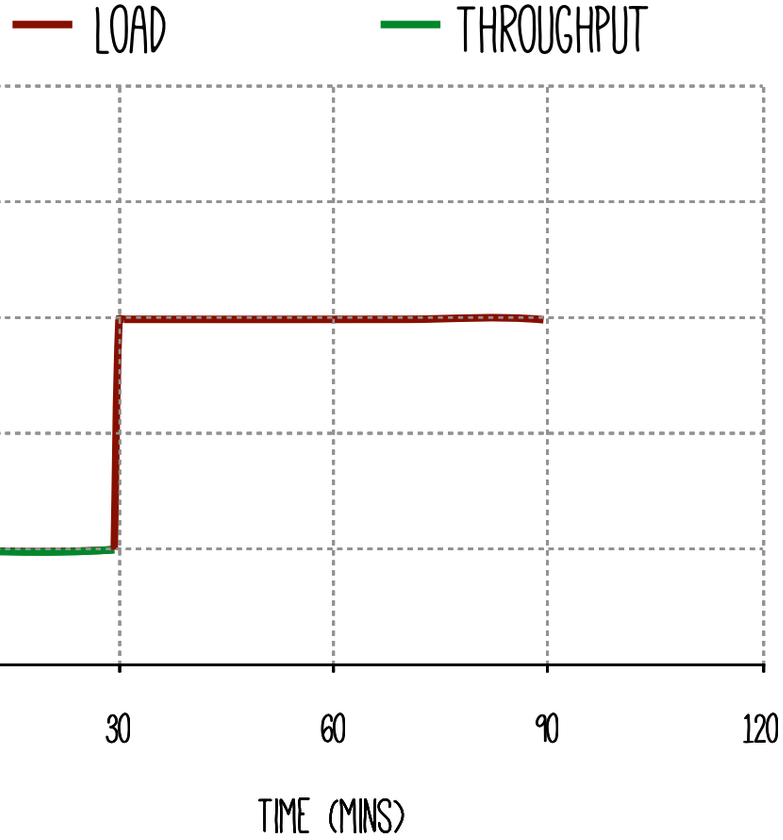
Changes in Spatial Skew

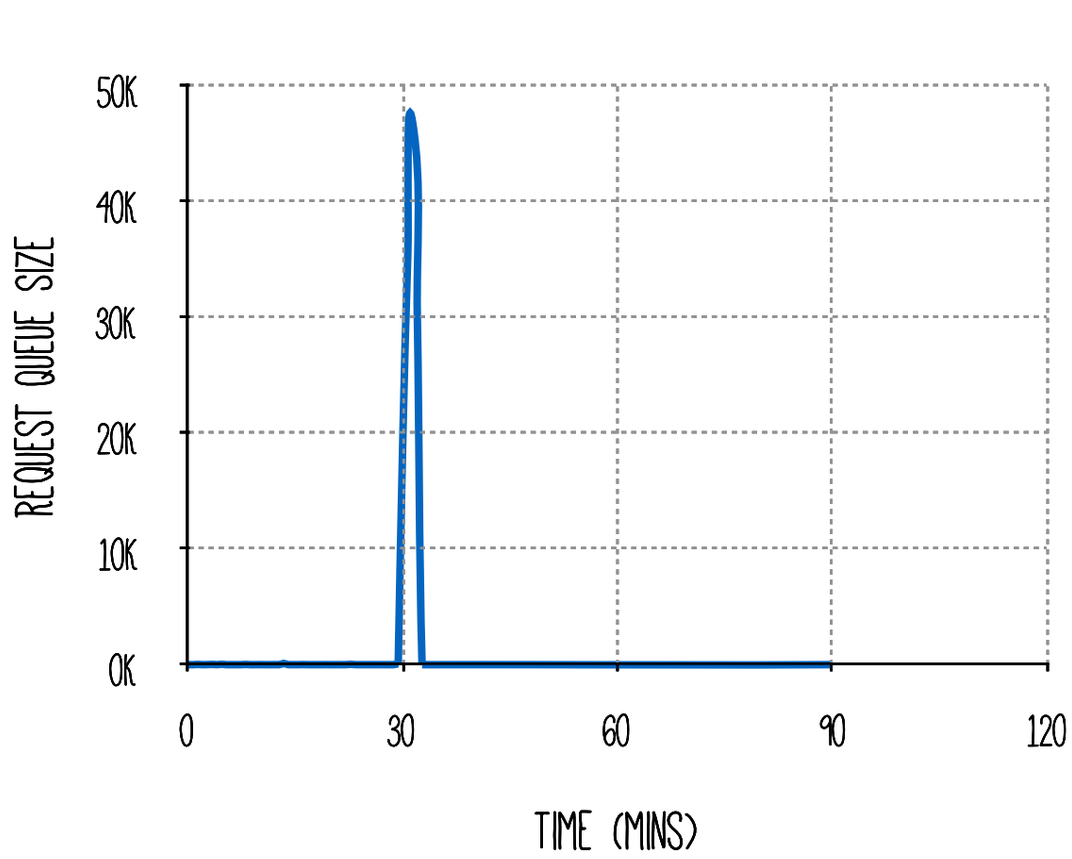




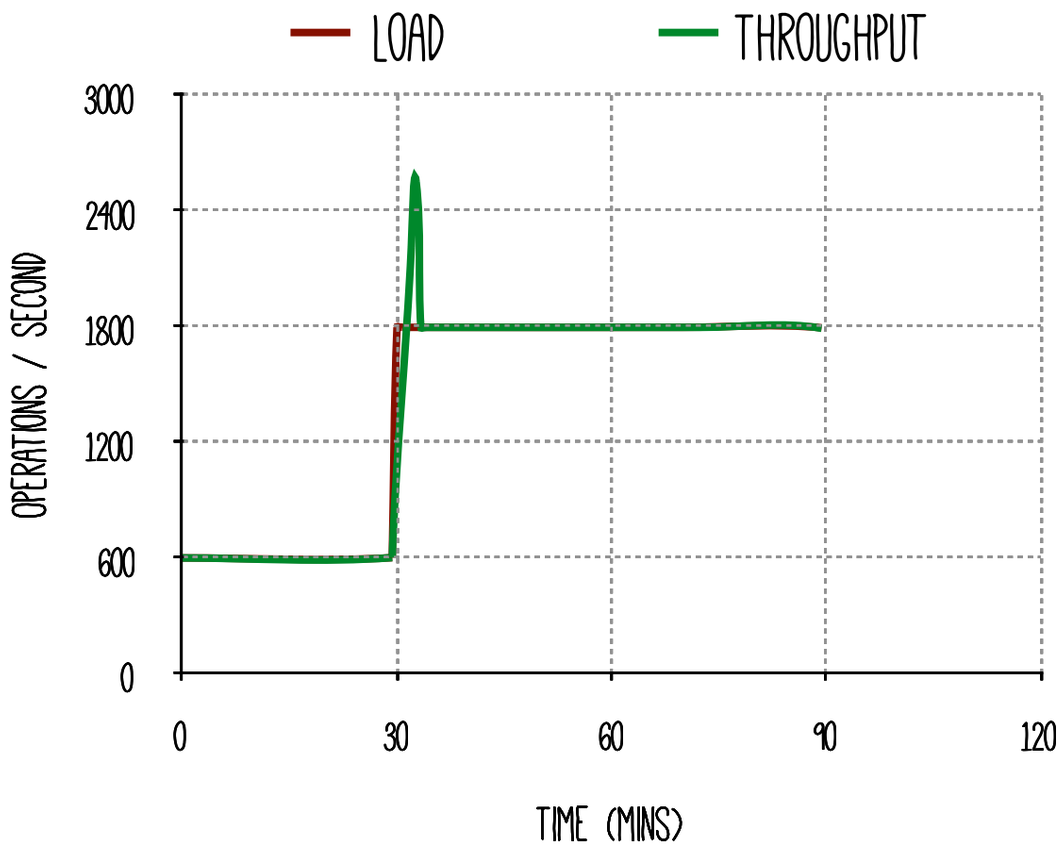
Changes in Spatial Skew

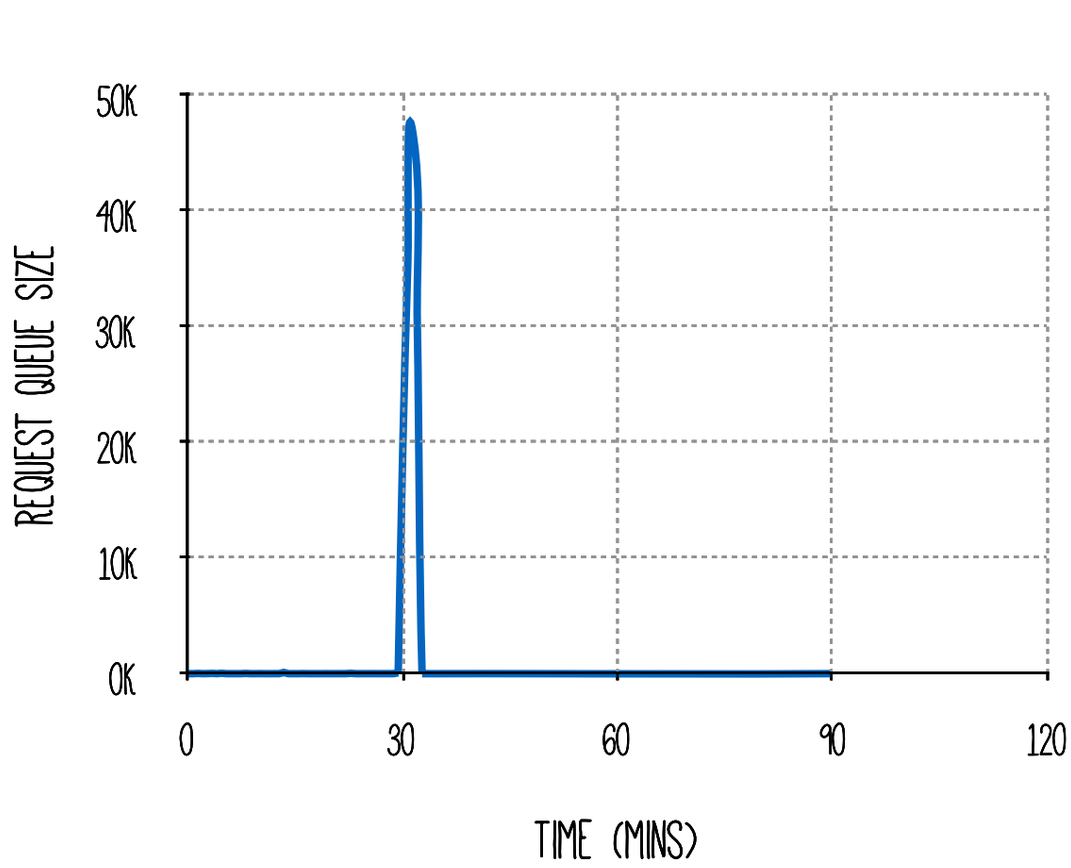




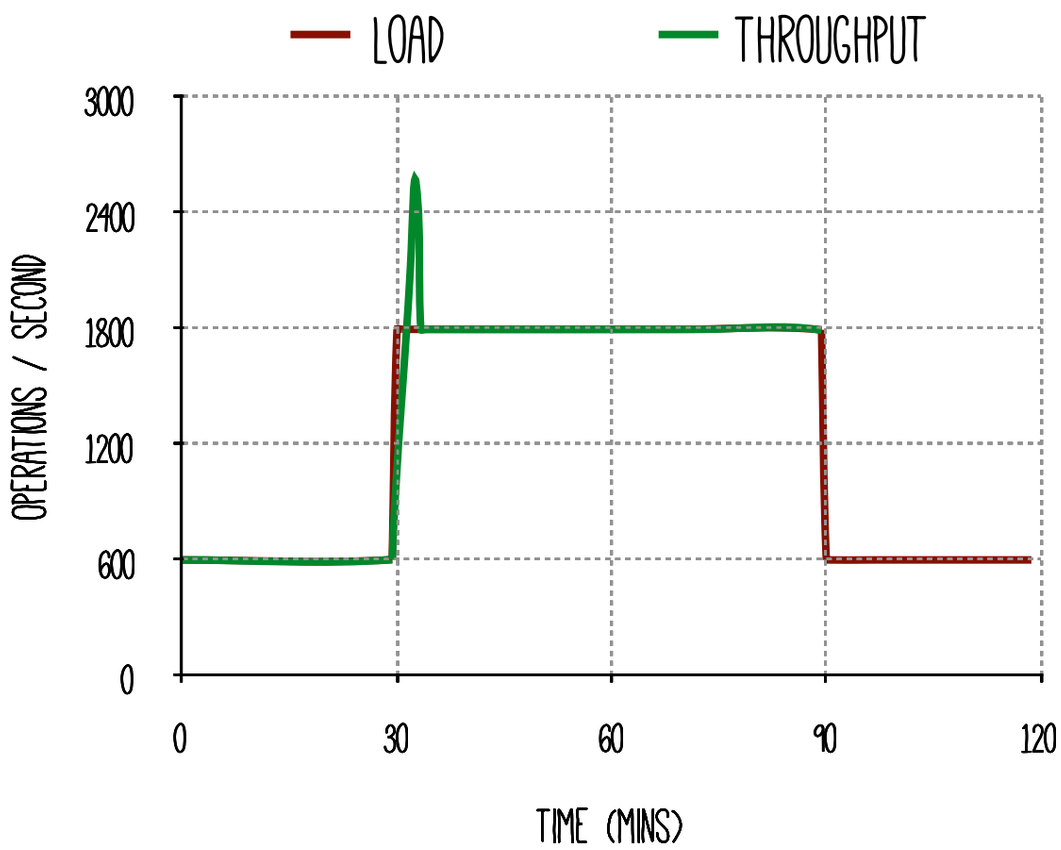


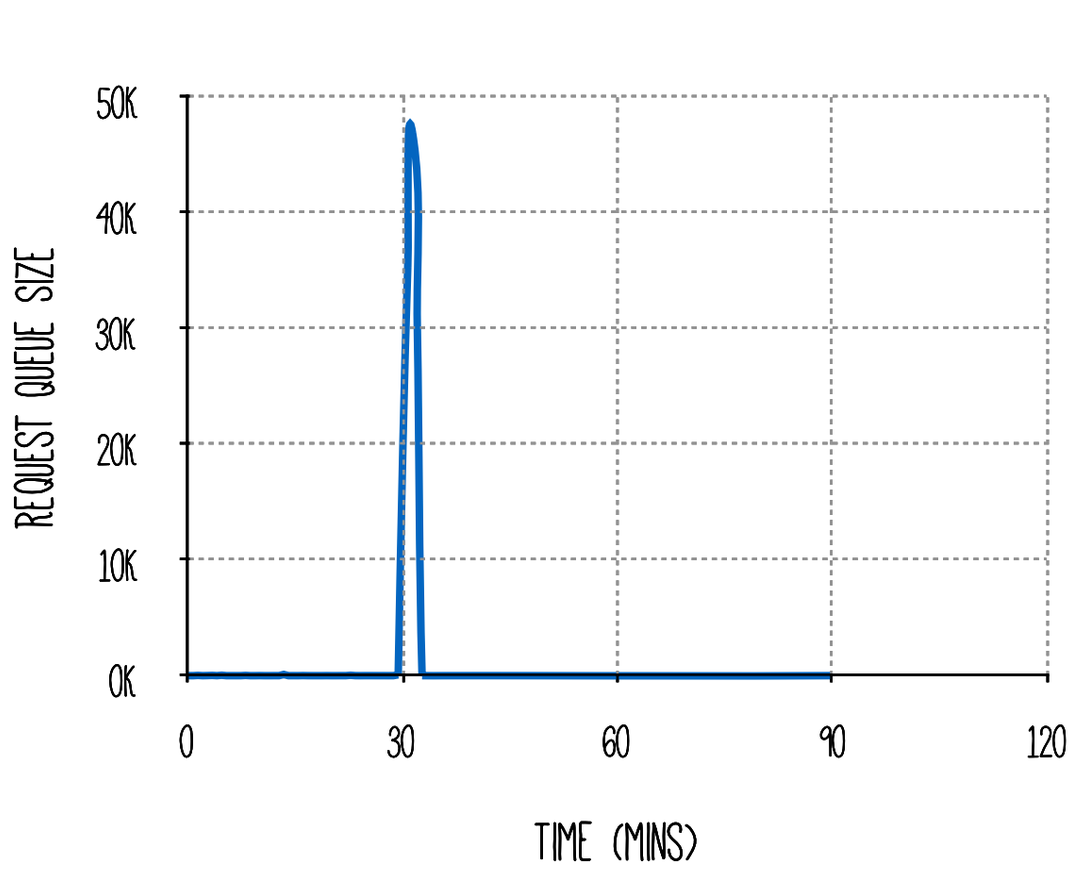
Changes in Spatial Skew



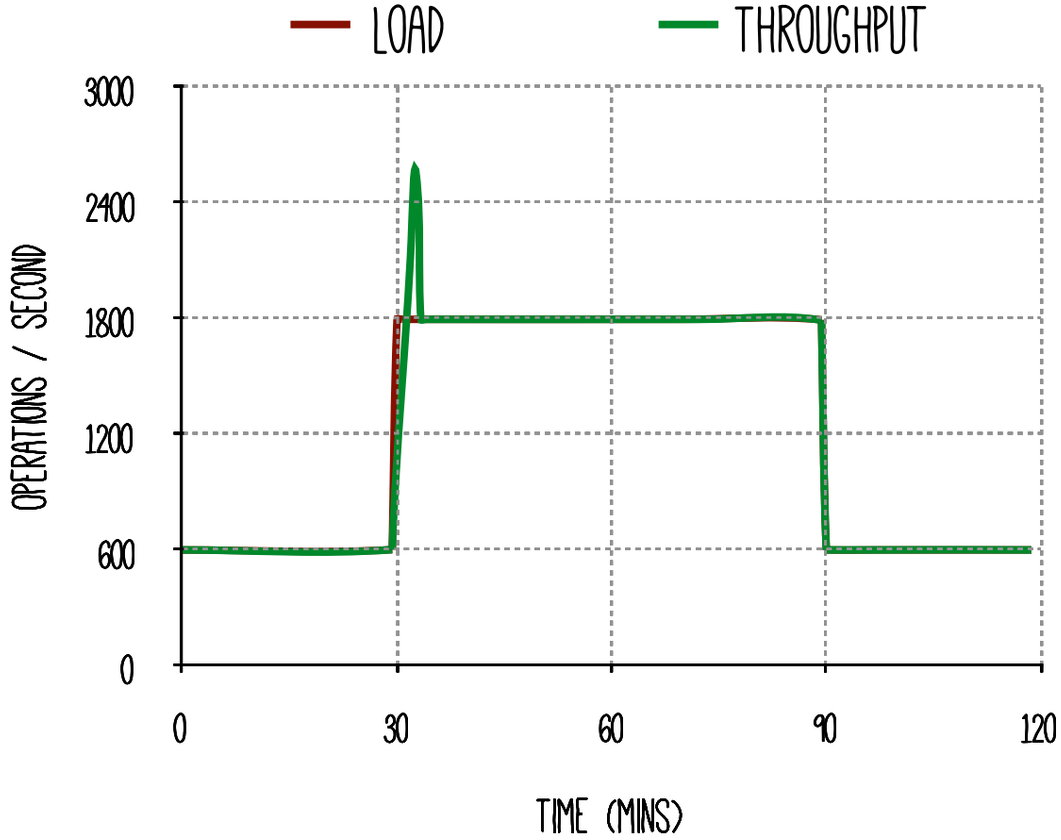


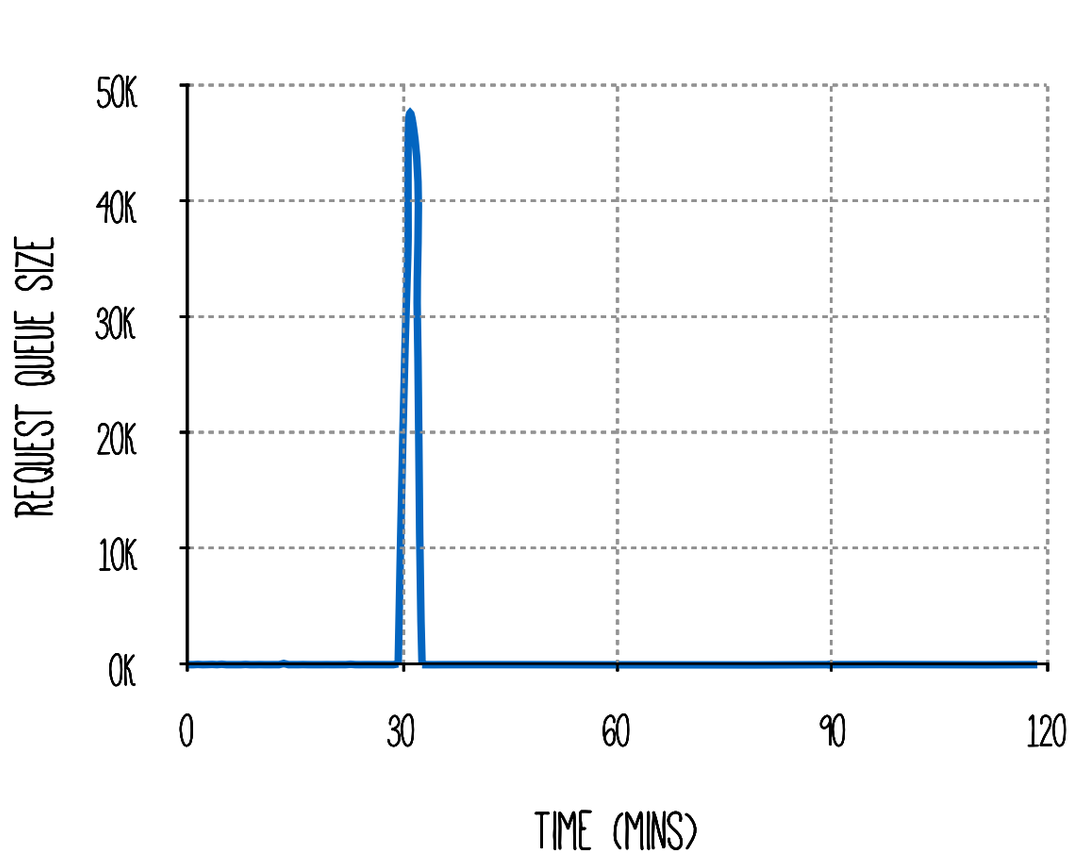
Changes in Spatial Skew



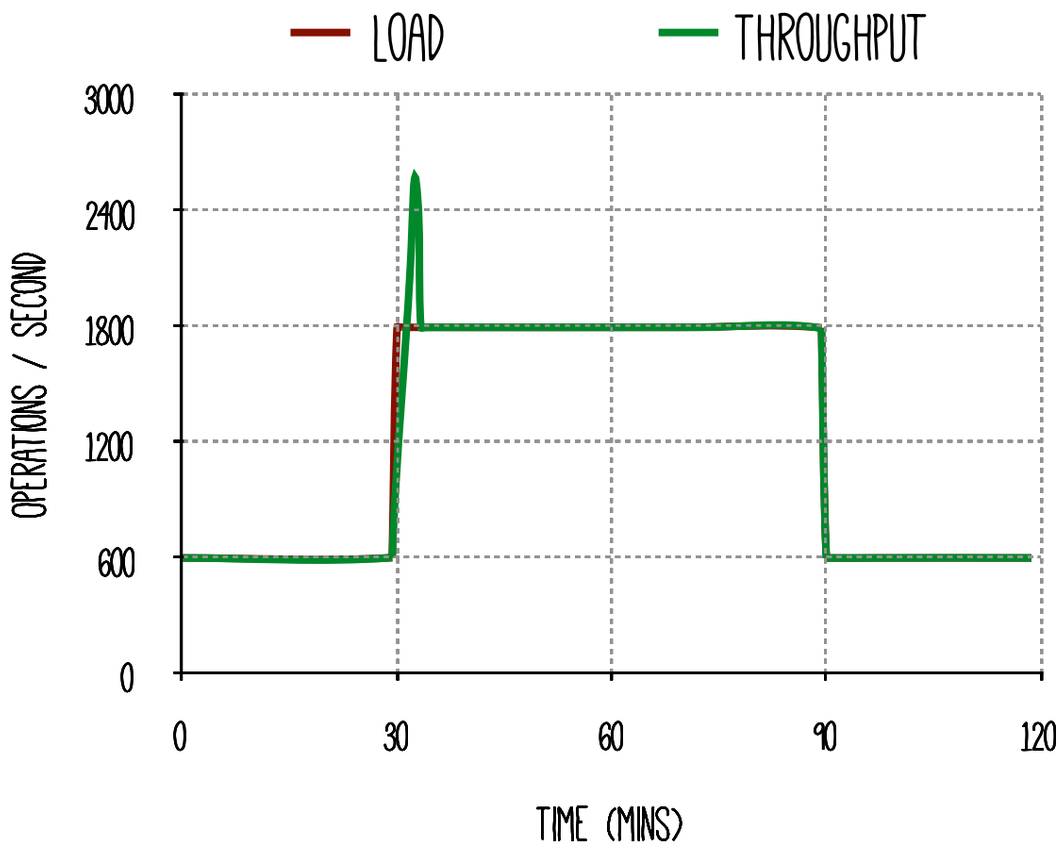


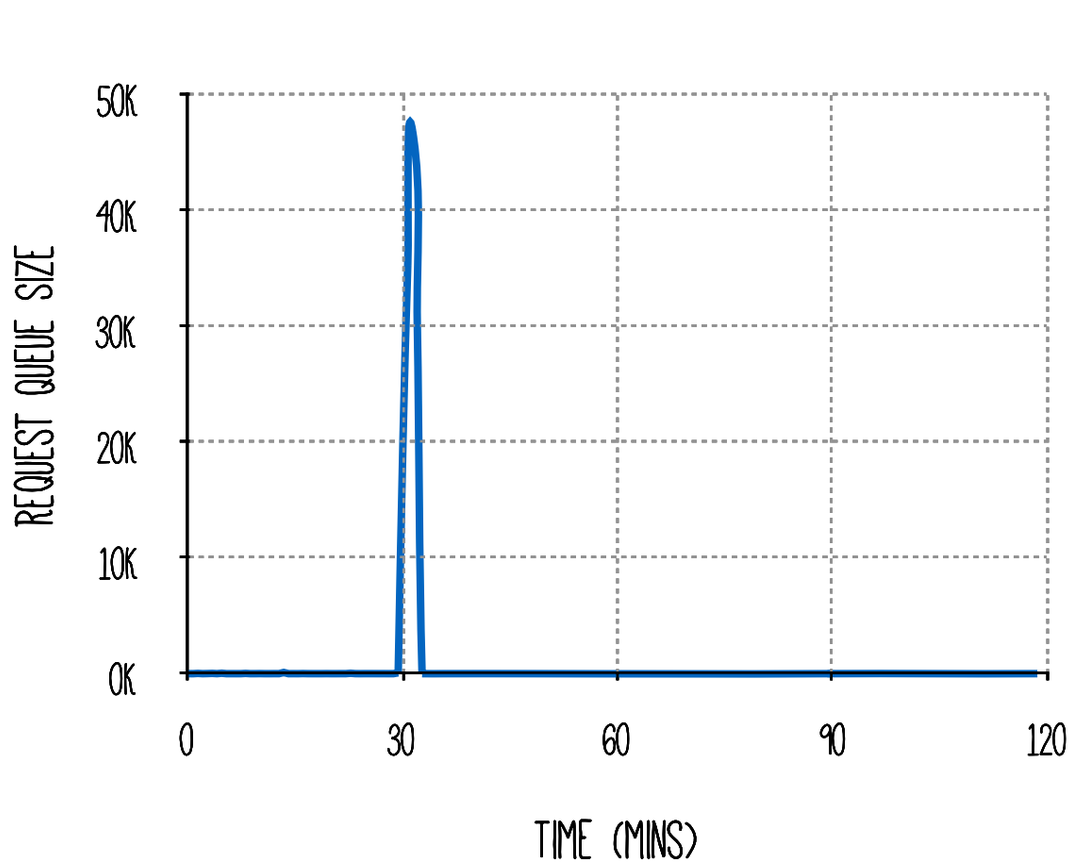
Changes in Spatial Skew





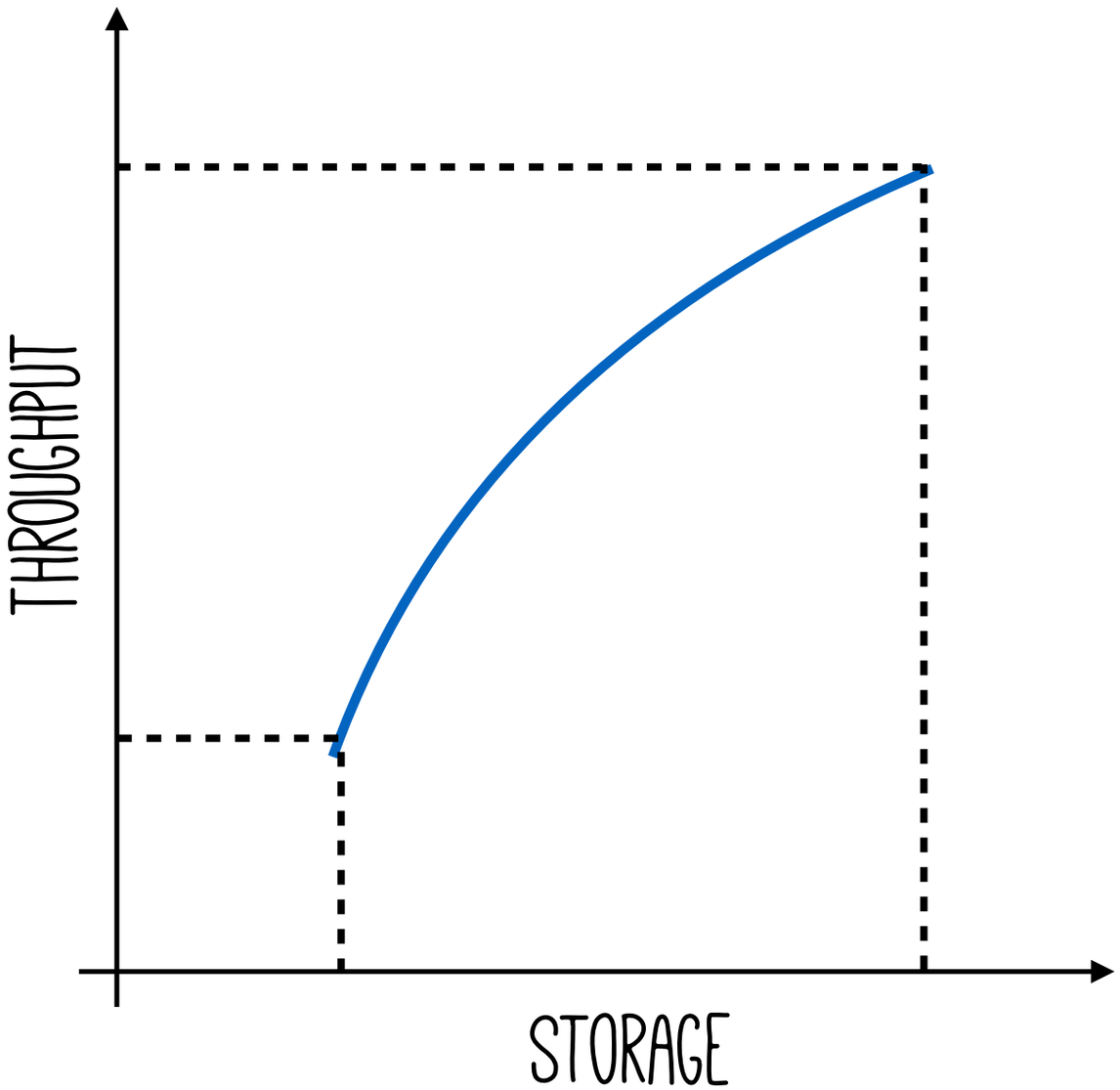
Changes in Spatial Skew



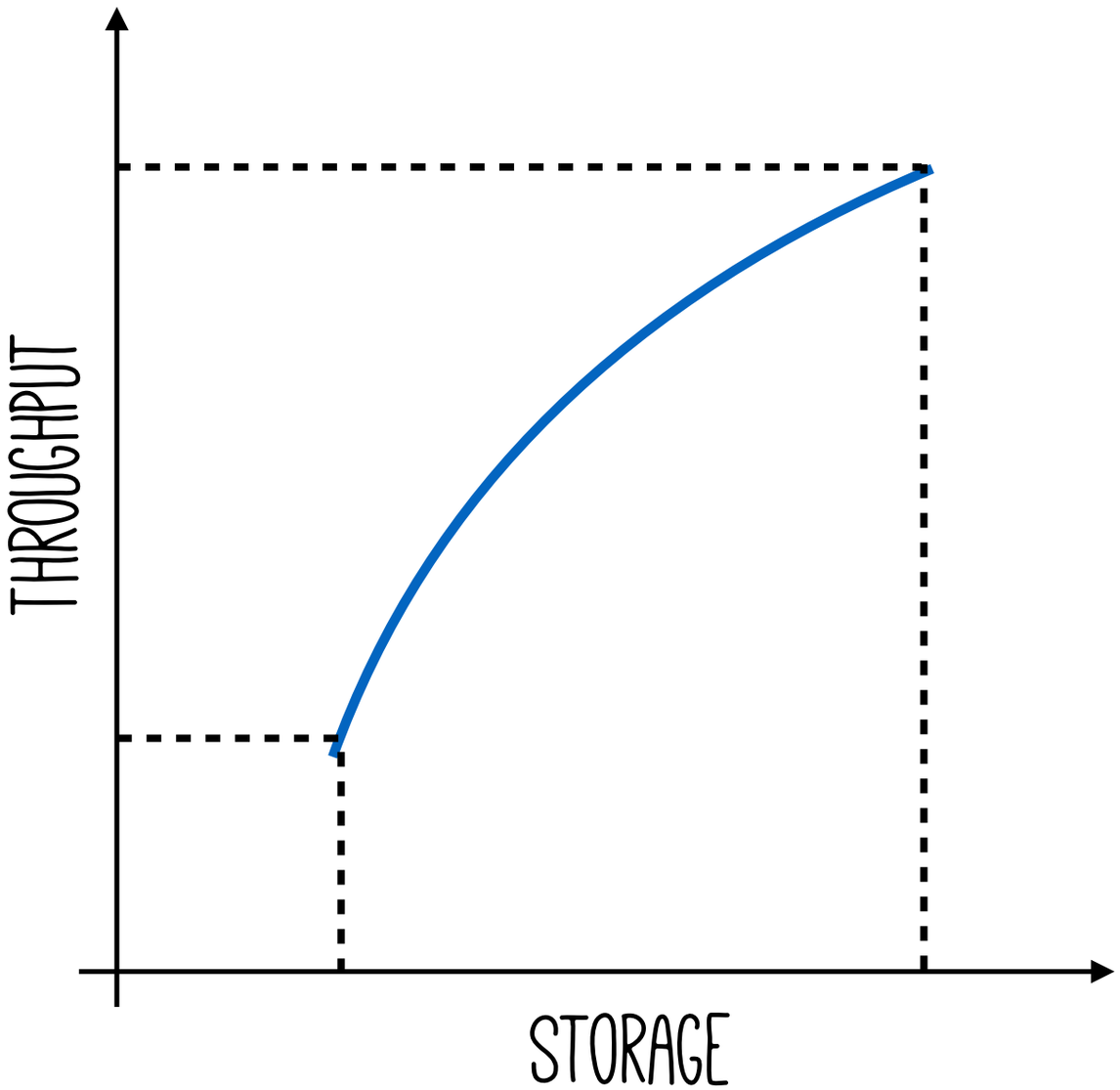


Adapts to 3x higher load in < 5 mins

Summary

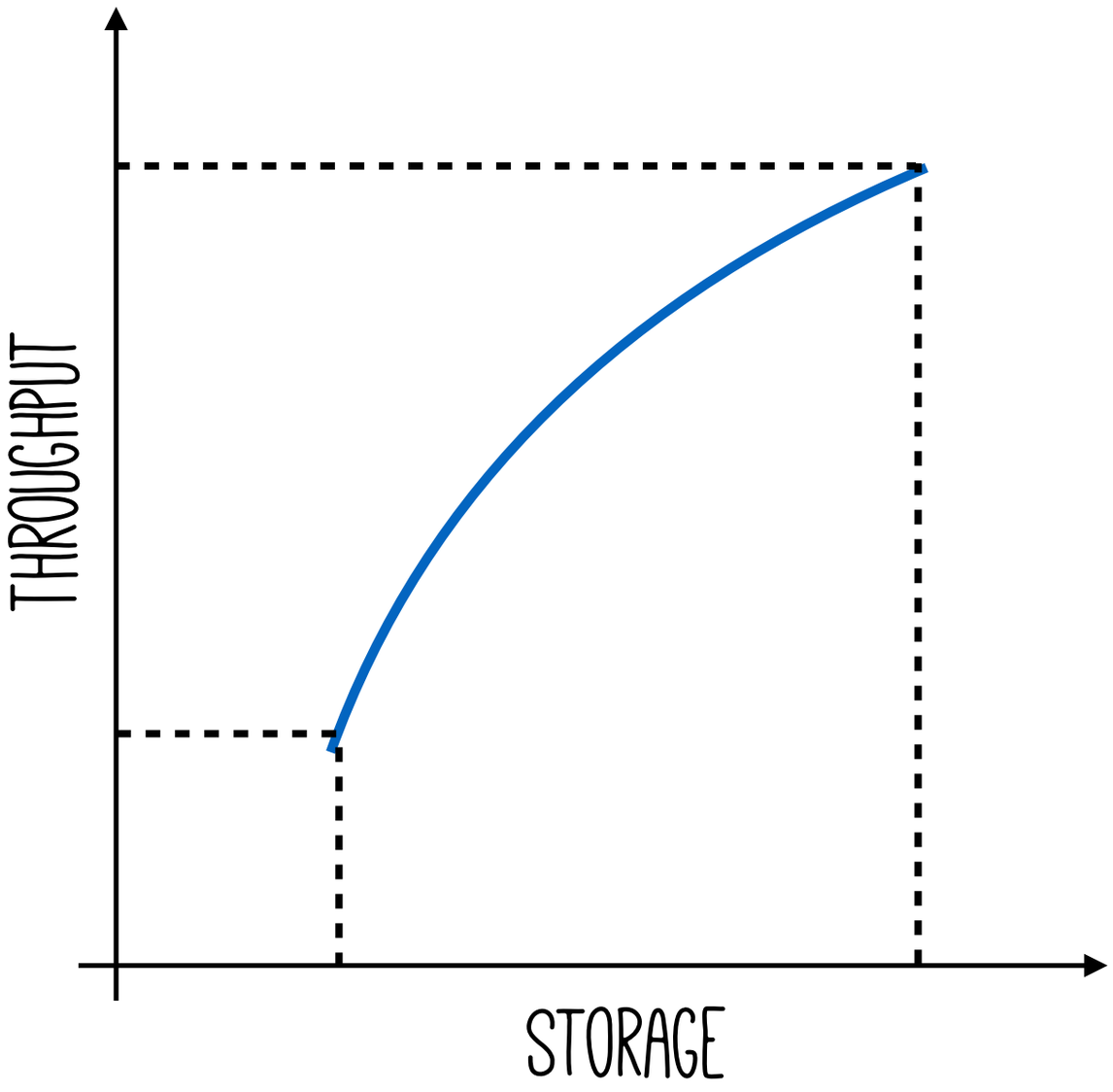


Summary



Smooth Tradeoff Curve

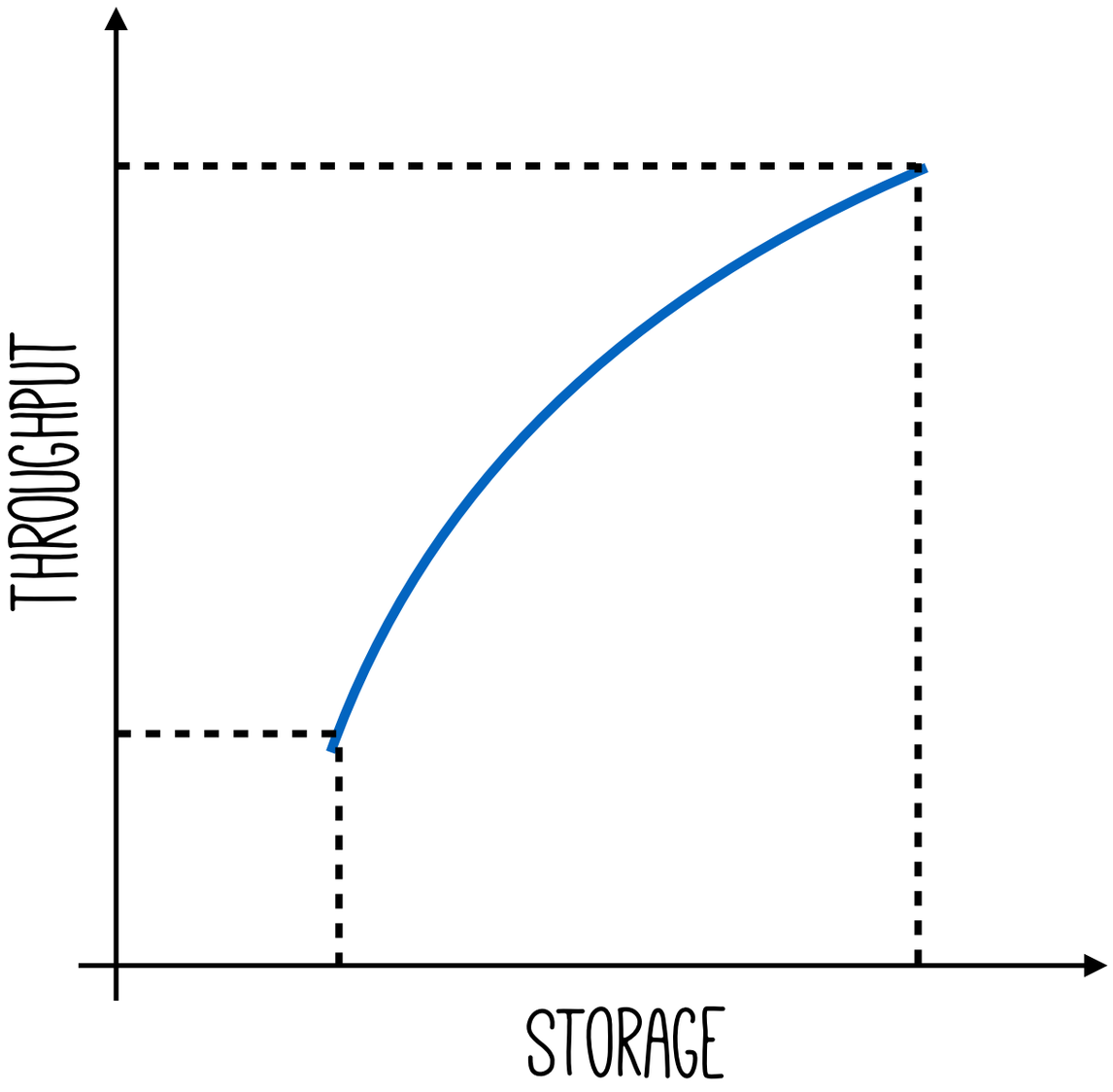
Summary



Smooth Tradeoff Curve

Dynamic Navigation

Summary

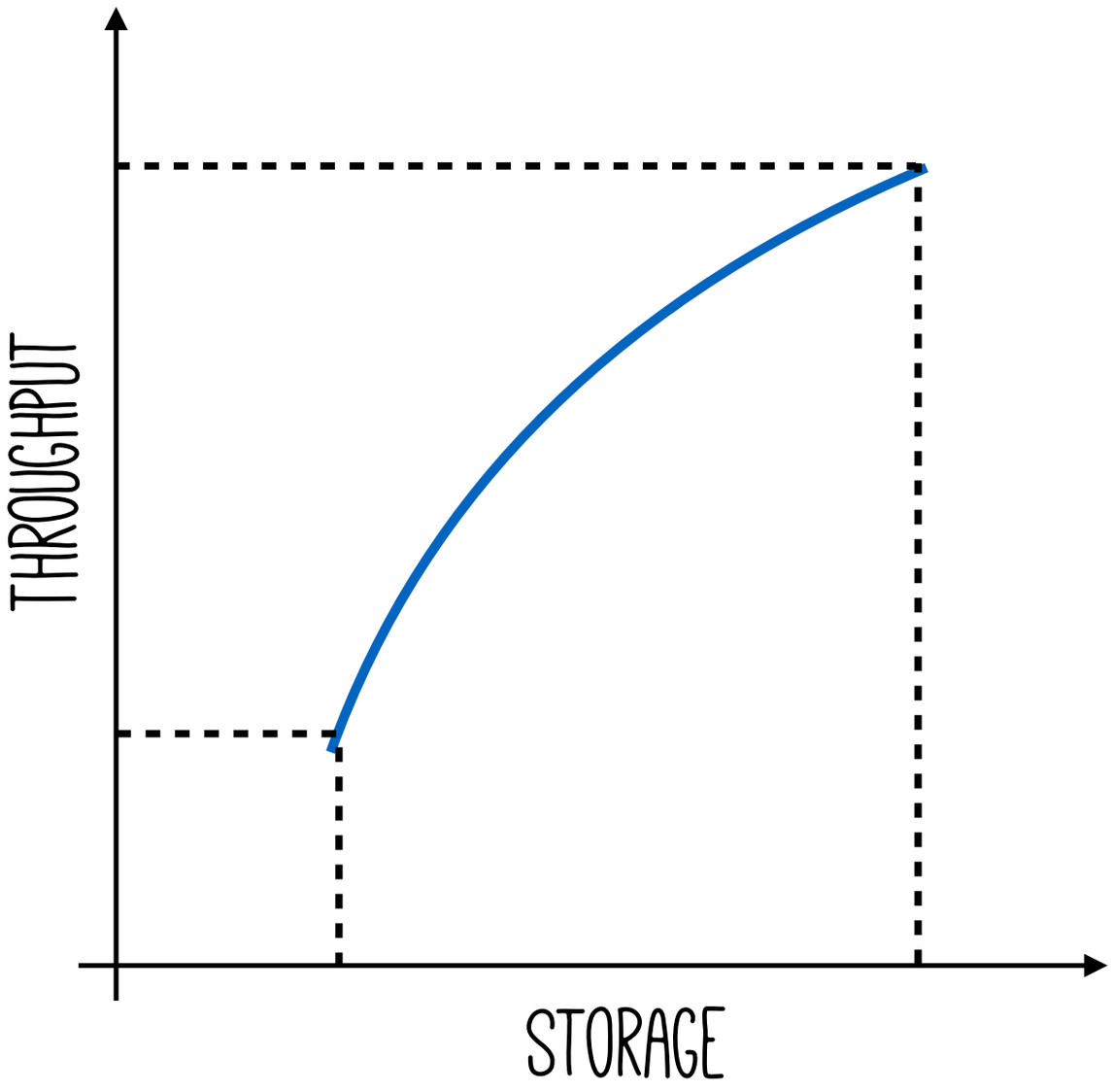


Smooth Tradeoff Curve

Dynamic Navigation

Applications in Several Classical Systems Problems

Summary



Smooth Tradeoff Curve

Dynamic Navigation

Applications in Several Classical Systems Problems

Thank You! Questions?

Backup Slides