A Design for Networked Flash

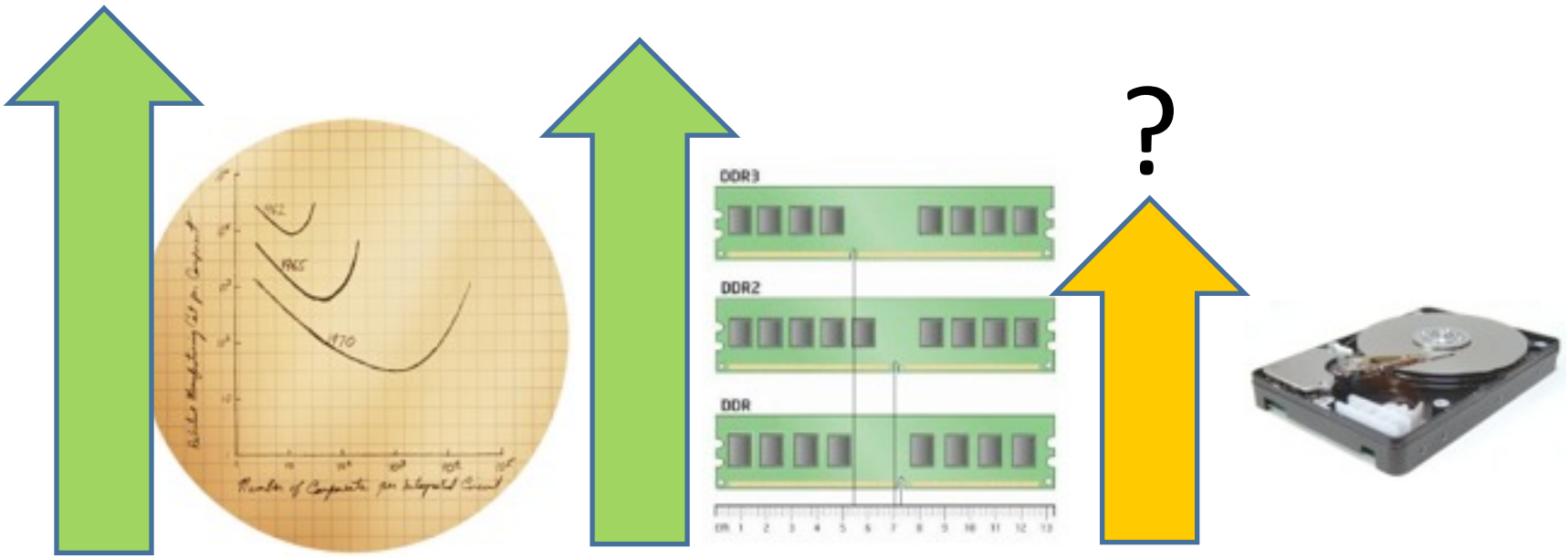
(Clusters **O**f **R**aw **F**lash **U**nits)

Mahesh Balakrishnan, John Davis,
Dahlia Malkhi, Vijayan Prabhakaran,
Michael Wei*, Ted Wobber

Microsoft Research Silicon Valley

* Graduate student at UCSD

The I/O Story



Processors

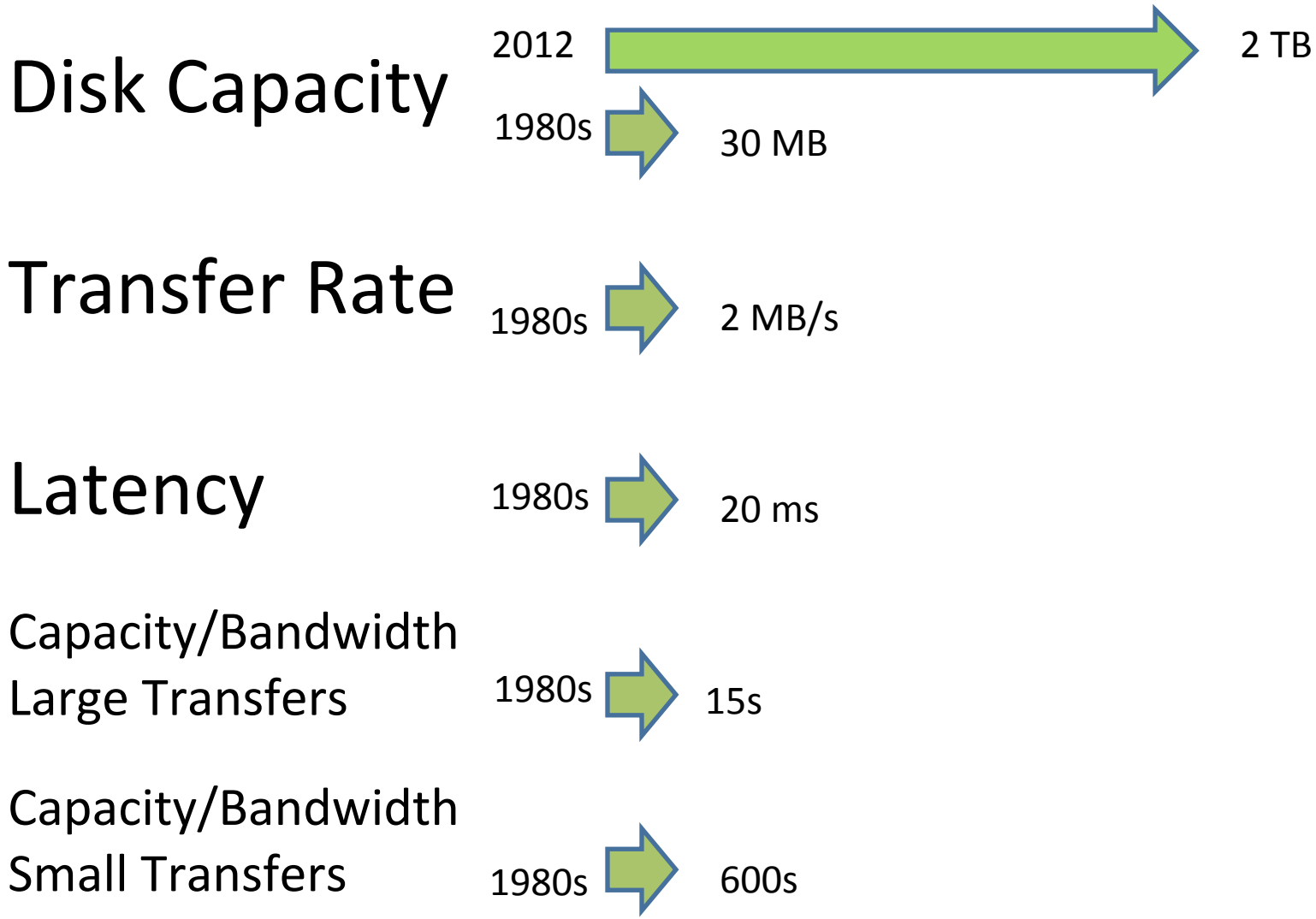
Main Memory

Storage

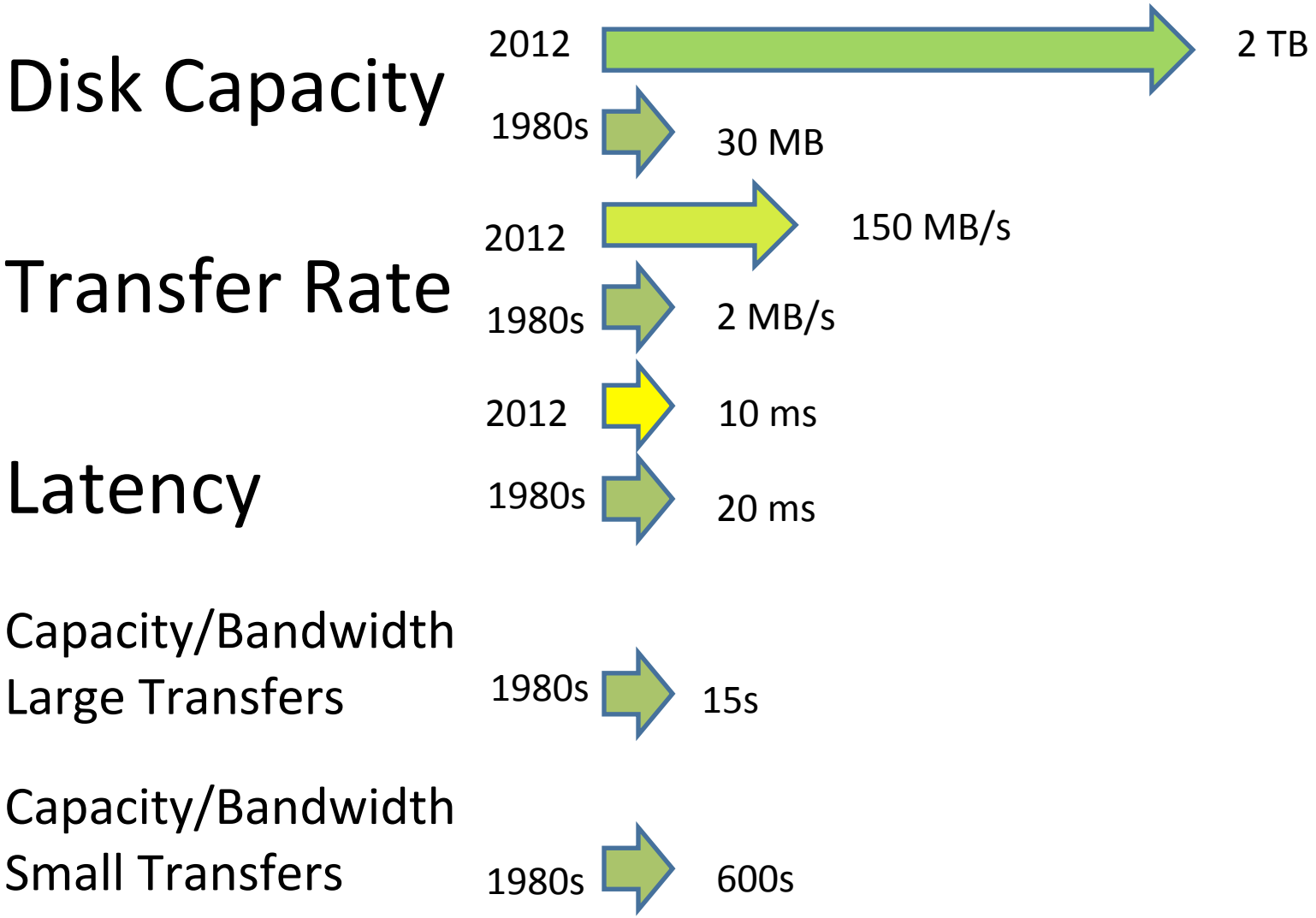
The I/O Story

Disk Capacity	1980s	30 MB
Transfer Rate	1980s	2 MB/s
Latency	1980s	20 ms
Capacity/Bandwidth Large Transfers	1980s	15s
Capacity/Bandwidth Small Transfers	1980s	600s

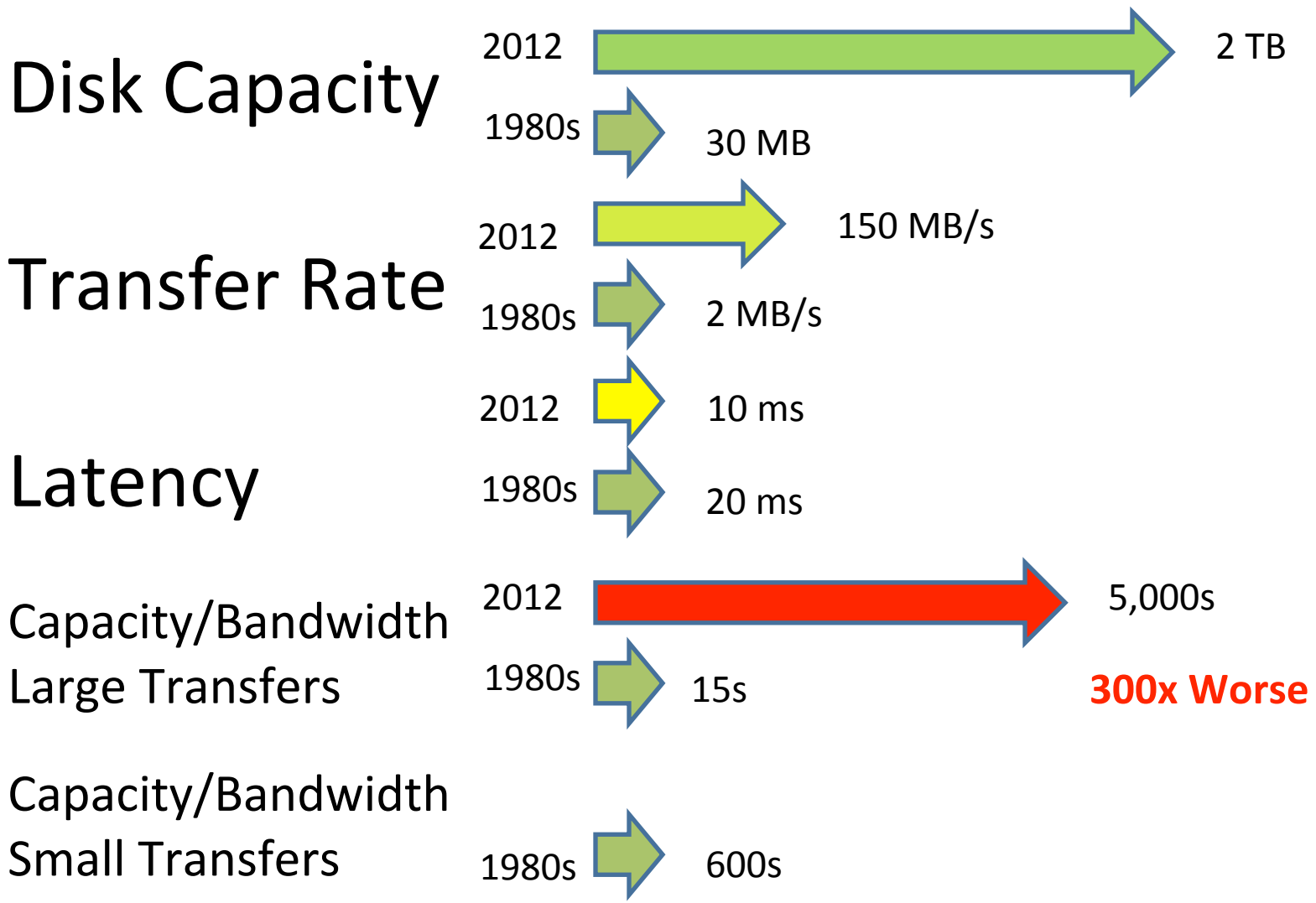
The I/O Story


Disk Capacity
2012
2 TB
1980s
30 MB
Transfer Rate
1980s
2 MB/s
Latency
1980s
20 ms
Capacity/Bandwidth
Large Transfers
1980s
15s
Capacity/Bandwidth
Small Transfers
1980s
600s

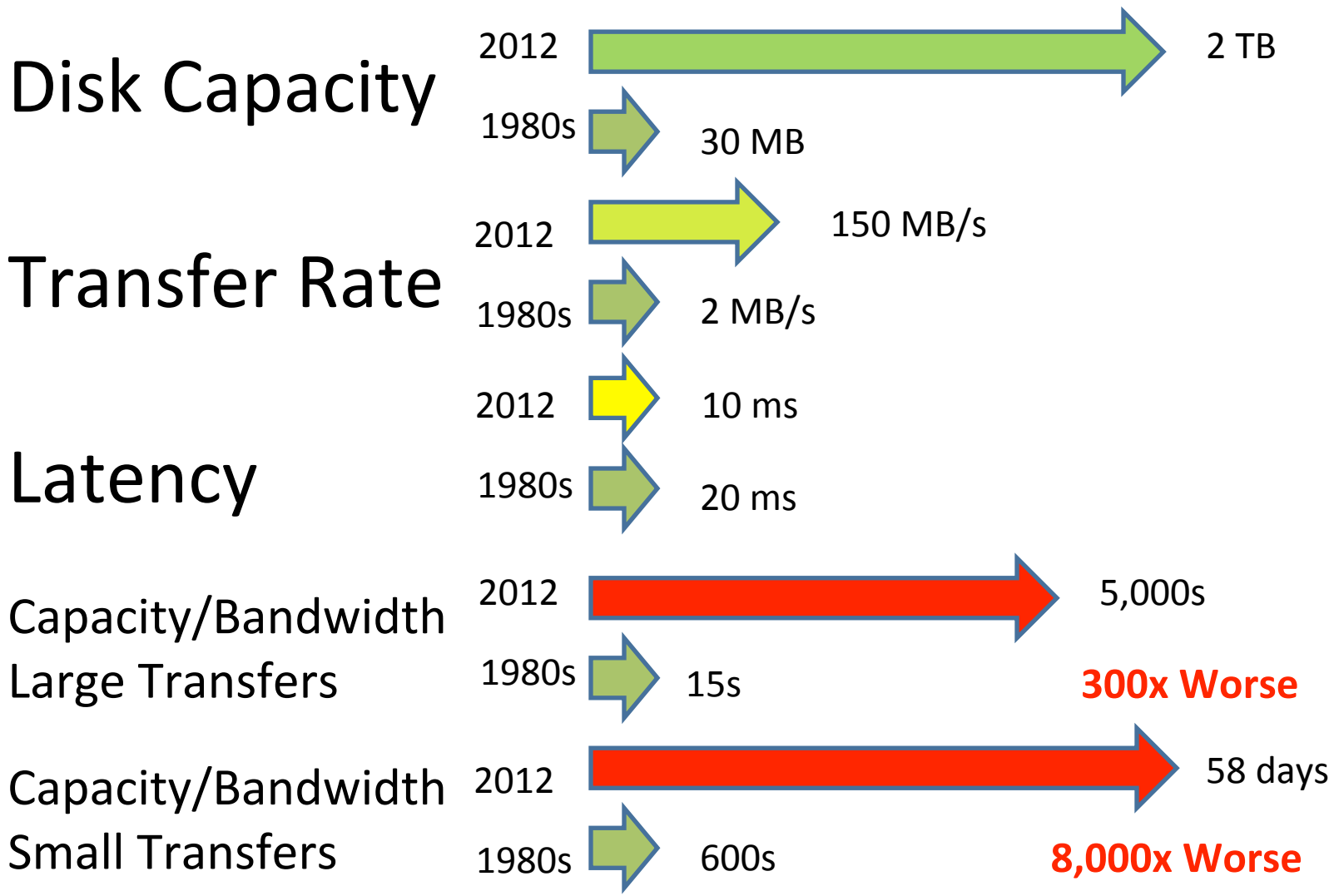
The I/O Story


Disk Capacity
2012
2 TB
1980s
30 MB
Transfer Rate
2012
150 MB/s
1980s
2 MB/s
Latency
2012
10 ms
1980s
20 ms
Capacity/Bandwidth
Large Transfers
1980s
15s
Capacity/Bandwidth
Small Transfers
1980s
600s

The I/O Story

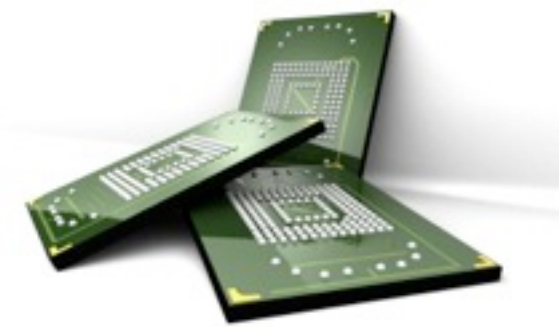

Disk Capacity
2012
2 TB
1980s
30 MB
Transfer Rate
2012
150 MB/s
1980s
2 MB/s
Latency
2012
10 ms
1980s
20 ms
Capacity/Bandwidth
Large Transfers
2012
5,000s
1980s
15s
300x Worse
Capacity/Bandwidth
Small Transfers
1980s
600s

The I/O Story


Disk Capacity
2012
2 TB
1980s
30 MB
Transfer Rate
2012
150 MB/s
1980s
2 MB/s
Latency
2012
10 ms
1980s
20 ms
Capacity/Bandwidth
Large Transfers
2012
5,000s
1980s
15s
300x Worse
Capacity/Bandwidth
Small Transfers
2012
58 days
1980s
600s
8,000x Worse

The I/O Story

512 Gb
~50 MB/s
1,280 s @ 4k

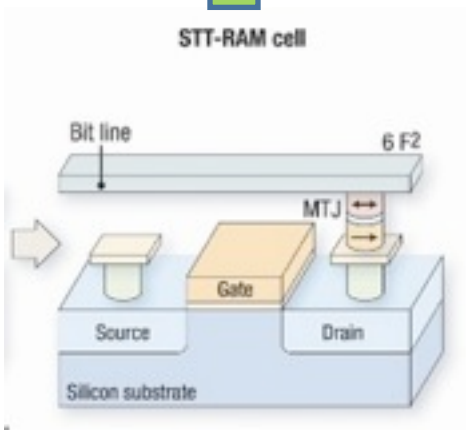


- NAND Flash

128 Mb
~50 MB/s
0.32 s @ 1 byte

- Phase Change

4 Mb
~200 MB/s
2.5ms @ 1 byte



- STT-RAM

~320GB

$7,000

$21/GB

500GB-10TB

$10,000+

$20/GB

2 TB

$88,000

$44/GB

PCIe 3.0 x16: 16 GB/s

iSCSI: 10 Gb/s

SAS : 12 Gb/s

~320GB

$7,000

$21/GB

500GB-10TB

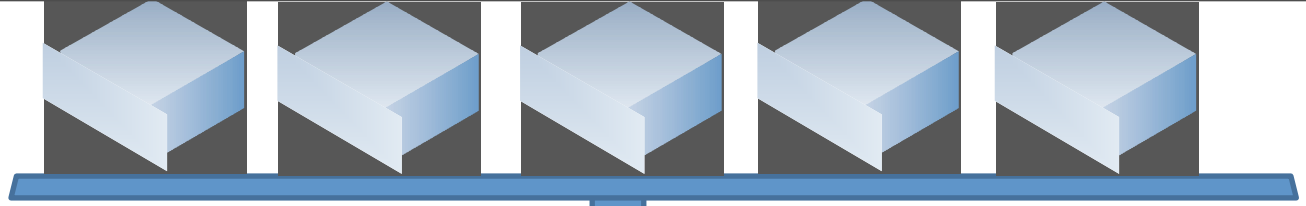
$10,000+

$20/GB

2 TB

$88,000

$44/GB



Ethernet

PCIe 3.0 x16: 16 GB/s

iSCSI: 10 Gb/s

SAS : 12 Gb/s

PCI EXPRESS

iSCSI

~320GB

$7,000

$21/GB

500GB-10TB

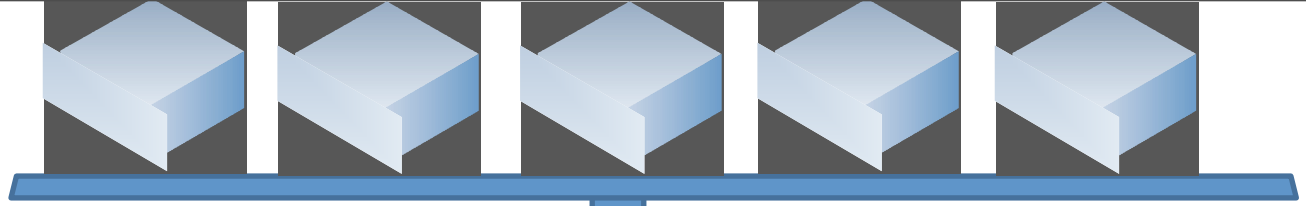
$10,000+

$20/GB

2 TB

$88,000

$44/GB



Ethernet

- Bottleneck
- Single Point of Failure
- Difficult to Scale
- Power-Inefficient
- How fast does it need to be?

PCIe 3.0 x16: 16 GB/s

iSCSI: 10 Gb/s

SAS : 12 Gb/s

~320GB

$7,000

$21/GB

500GB-10TB

$10,000+

$20/GB

2 TB

$88,000

$44/GB

Outline

- The I/O Story
- **CORFU Overview**
- Hardware Platform
- Conclusion

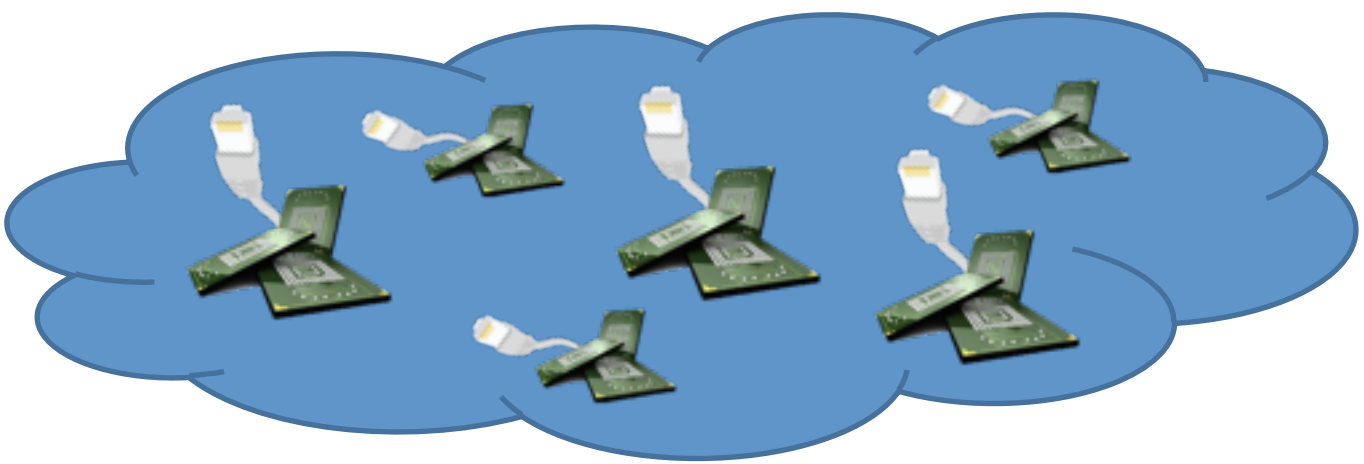
flash in the data center

How can we leverage flash in distributed systems?

Can flash clusters eliminate the trade-off between consistency and performance?

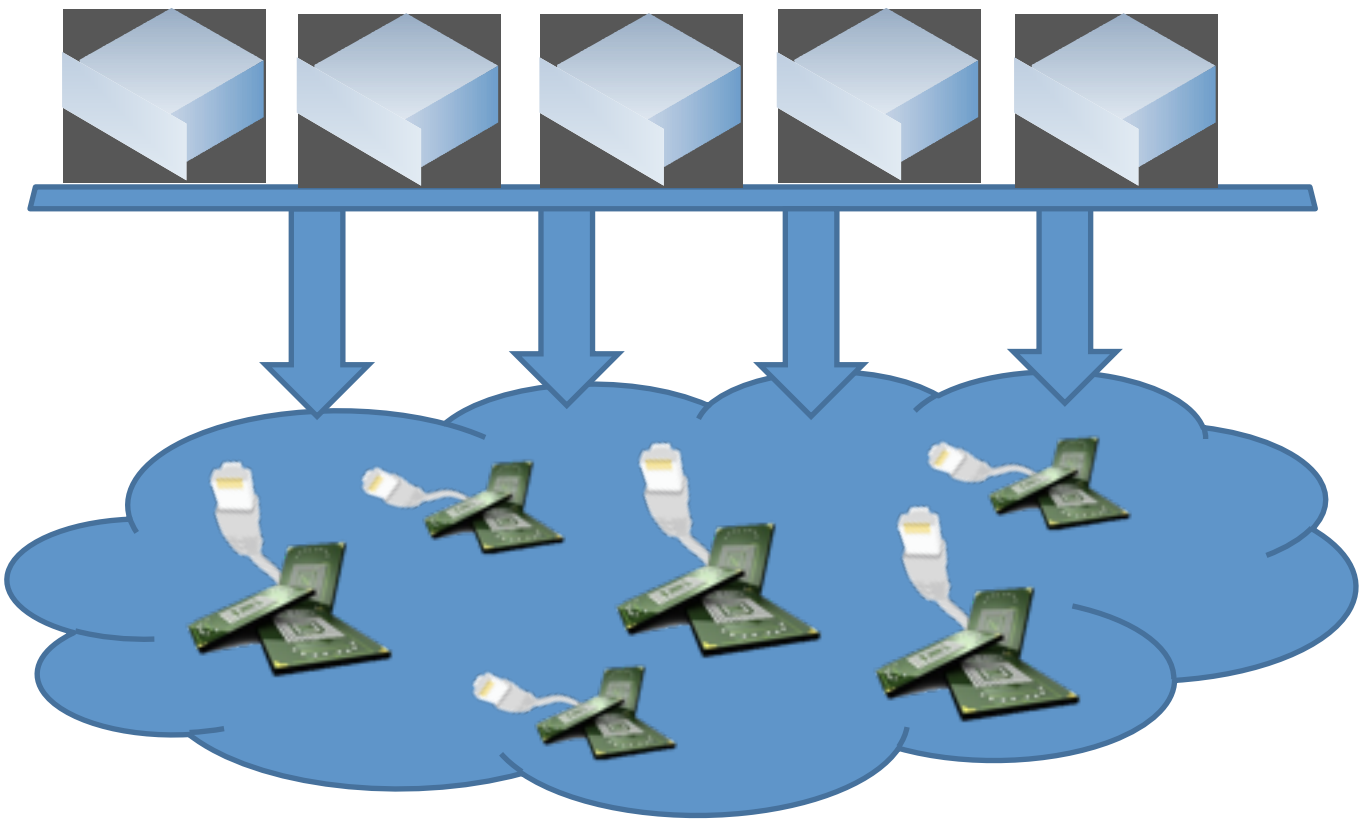
What new abstractions are required to manage and access flash clusters?

The CORFU Architecture



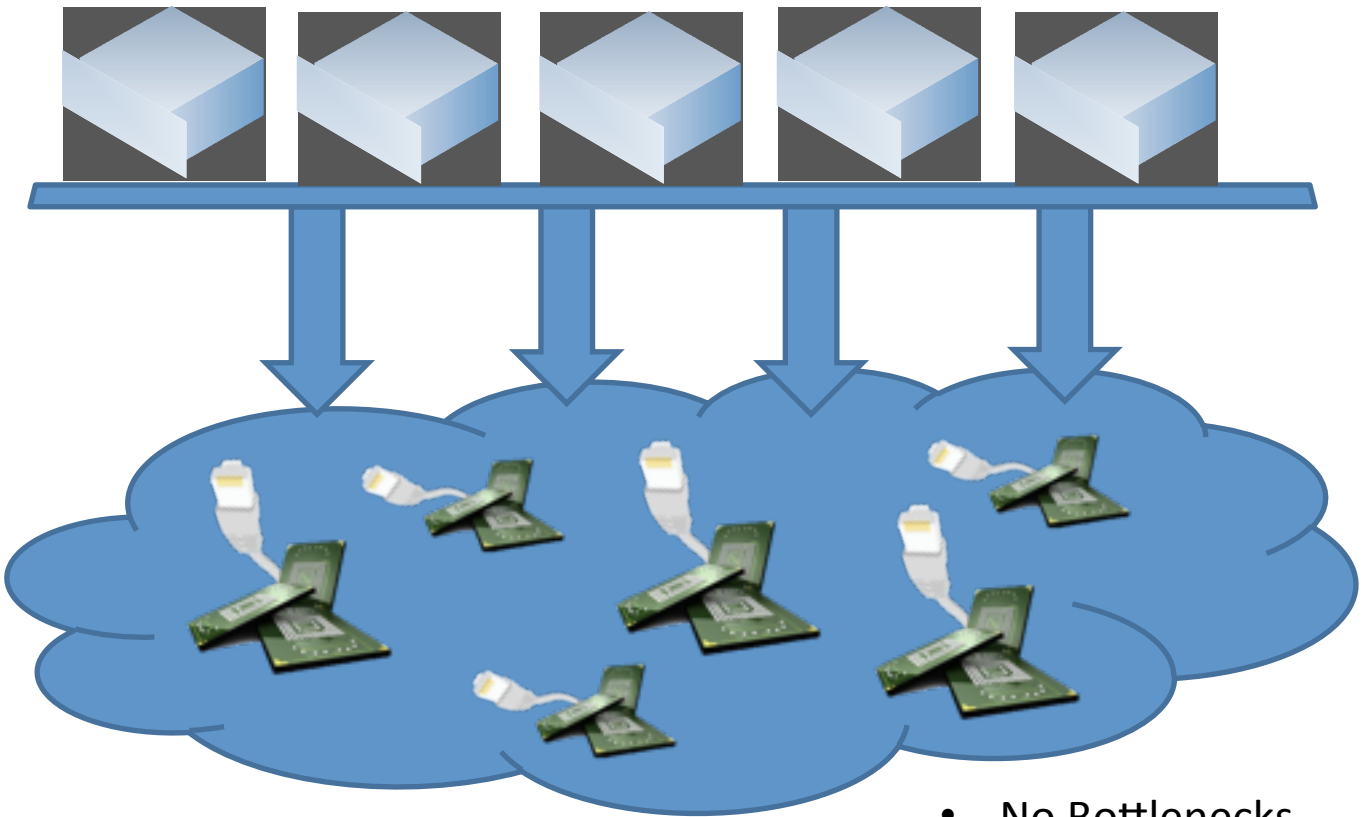
Cluster of raw flash units

The CORFU Architecture



Cluster of raw flash units

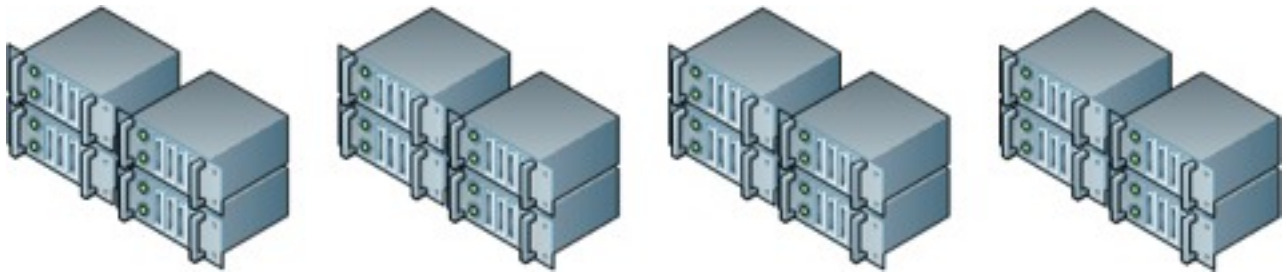
The CORFU Architecture



Cluster of raw flash units

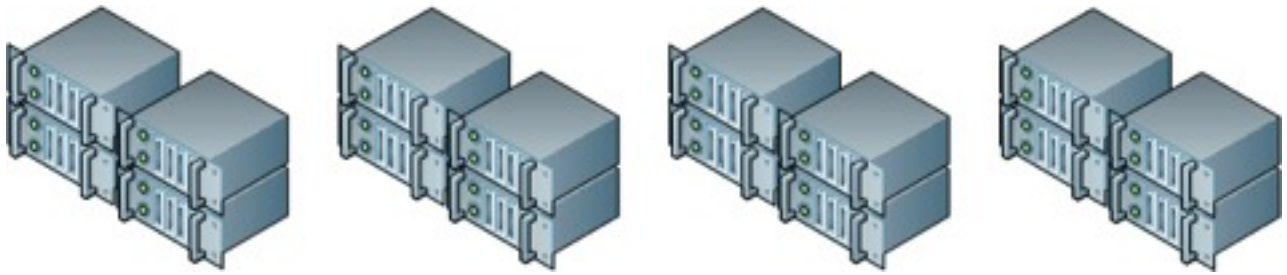
- No Bottlenecks
- Fault Tolerant
- Highly Scalable
- Low Power (10W /unit)
- Cheap (@ Cost of Flash)

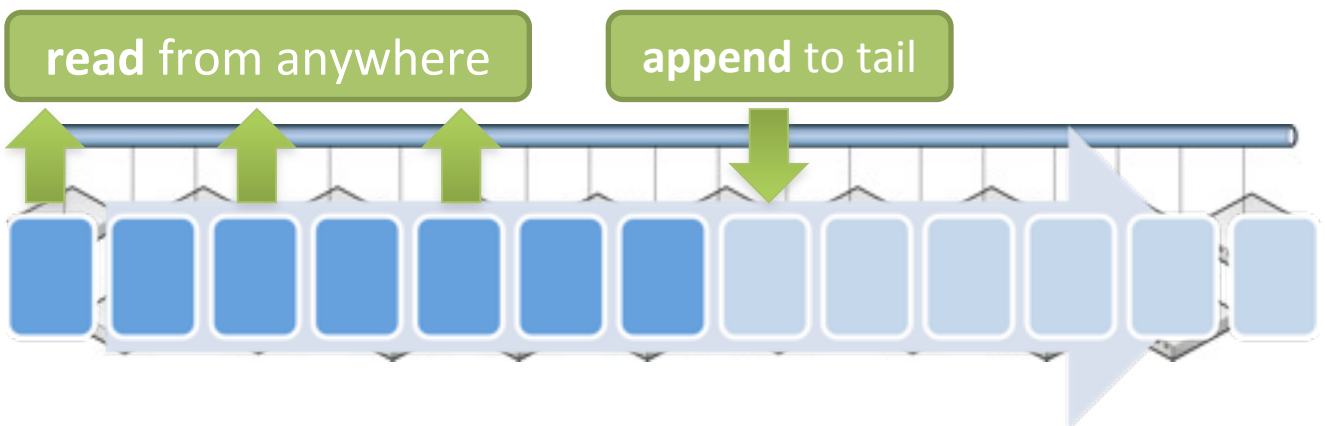
what is CORFU?



flash
cluster

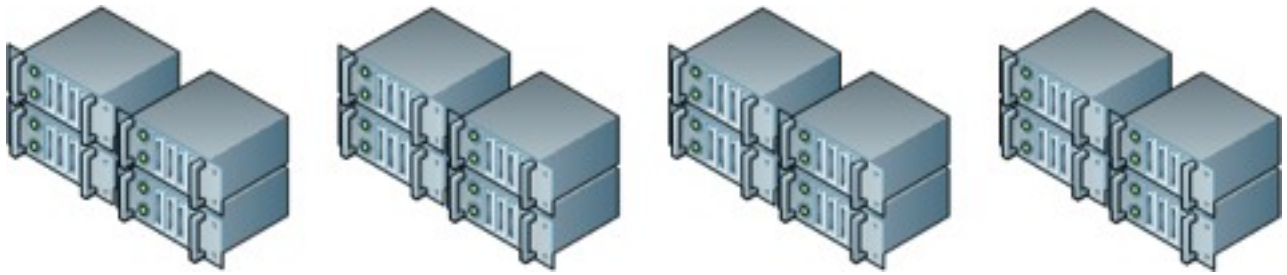
what is CORFU?



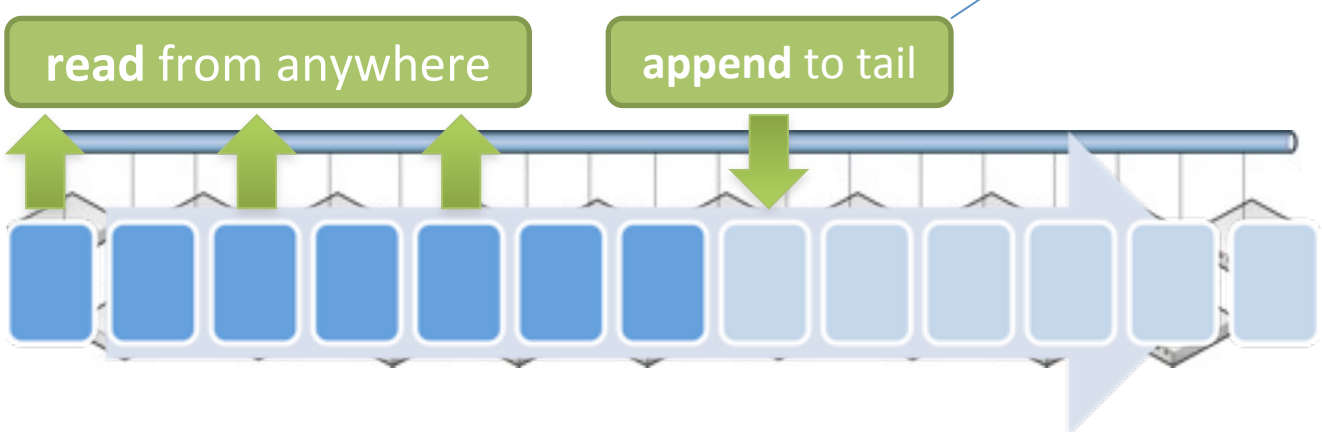


flash cluster

what is CORFU?

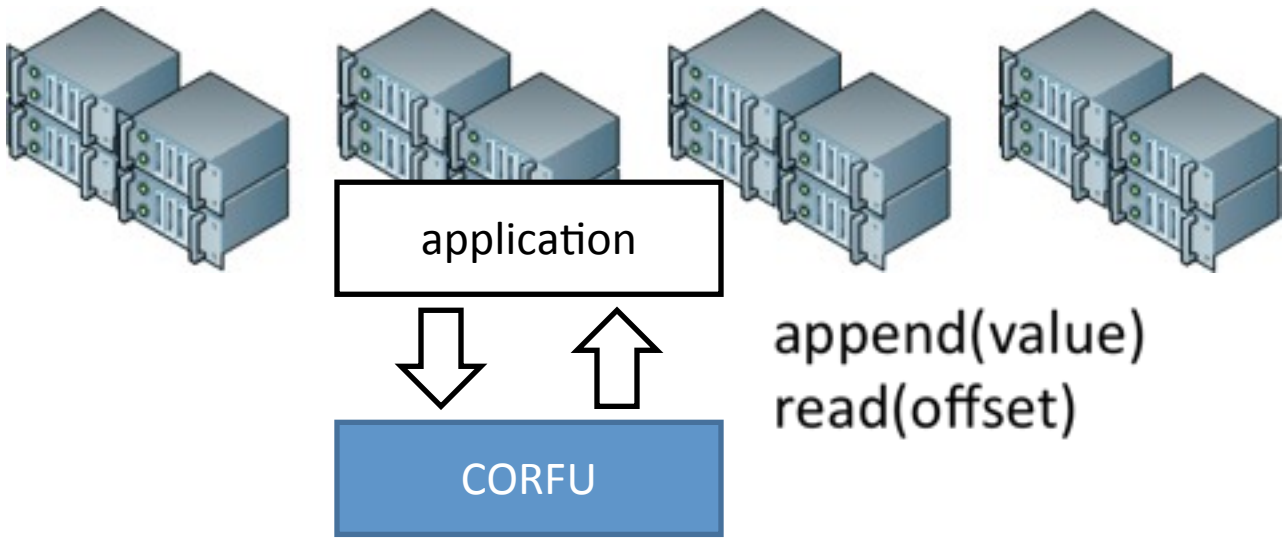


Replicated, fault-tolerant append

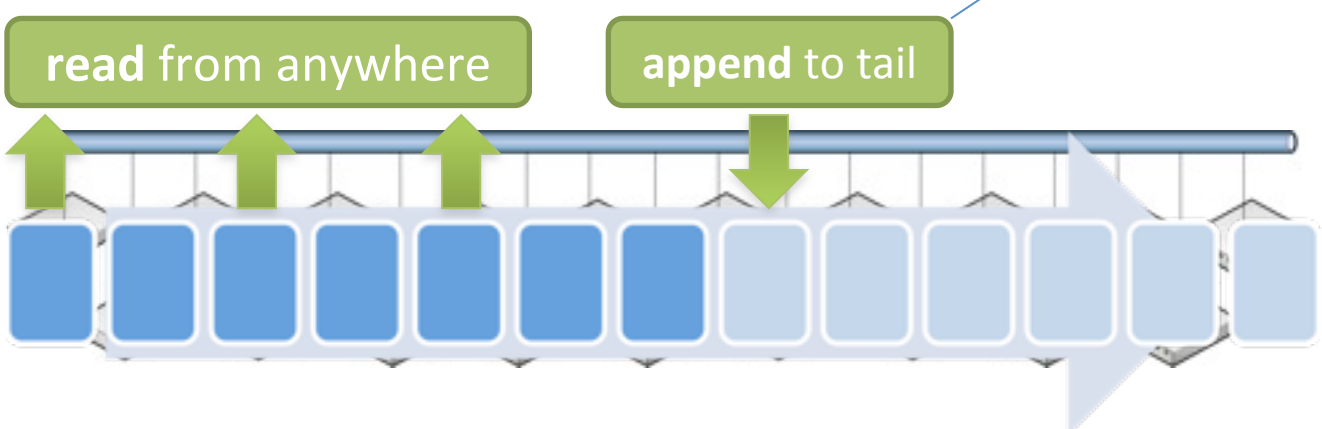


flash cluster

what is CORFU?

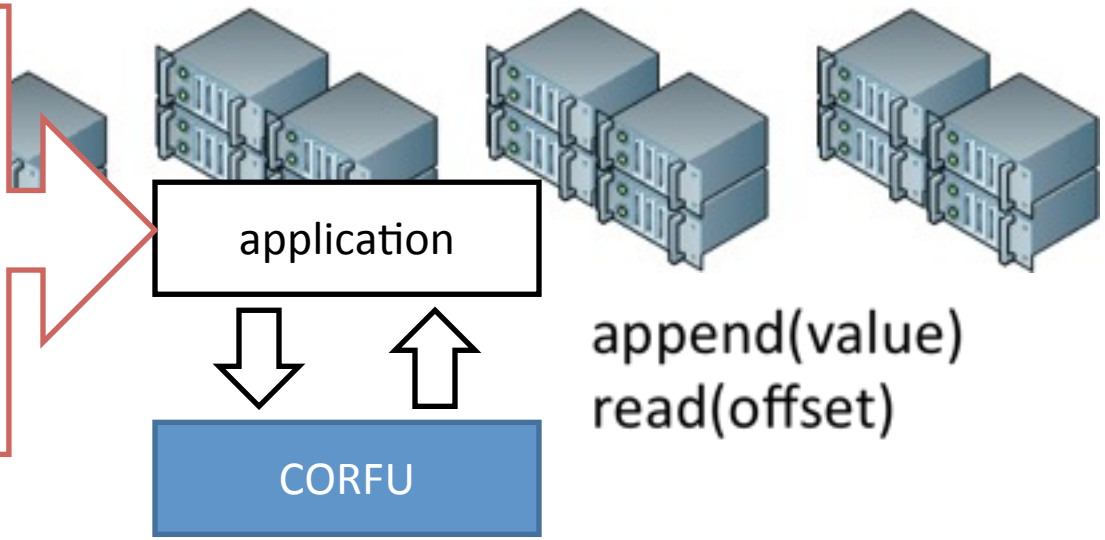


Replicated, fault-tolerant append

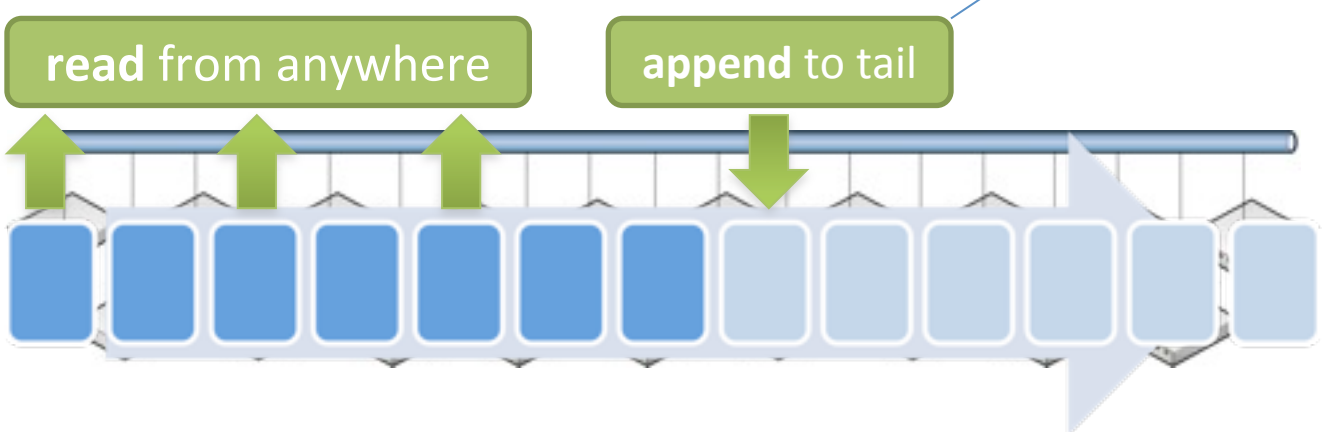


what is CORFU?

infrastructure applications:
key-value stores
databases
SMR
filesystems
virtual disks



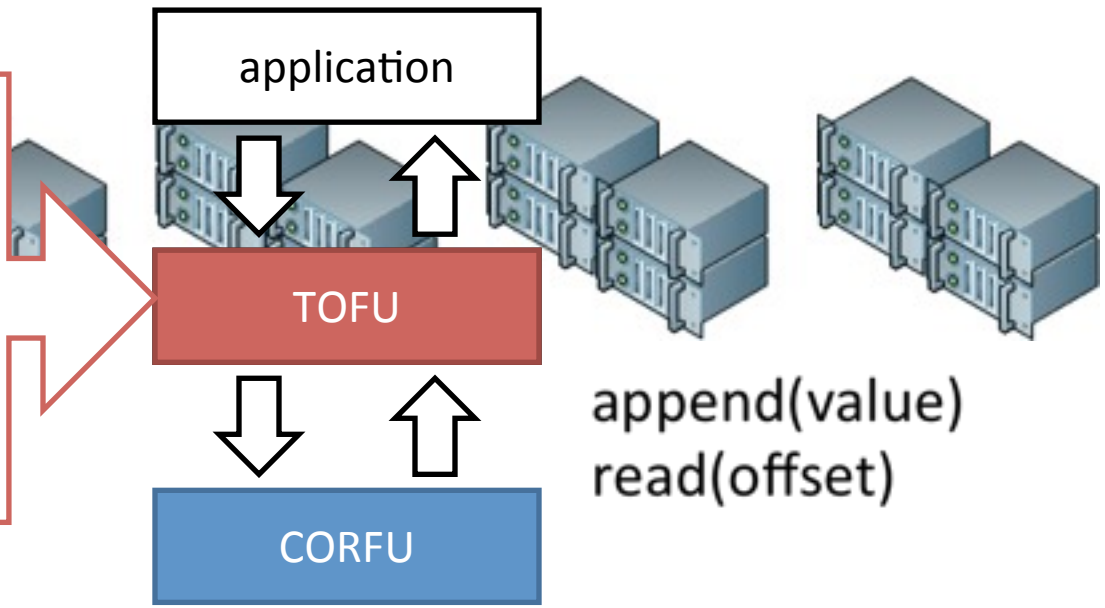
Replicated, fault-tolerant append



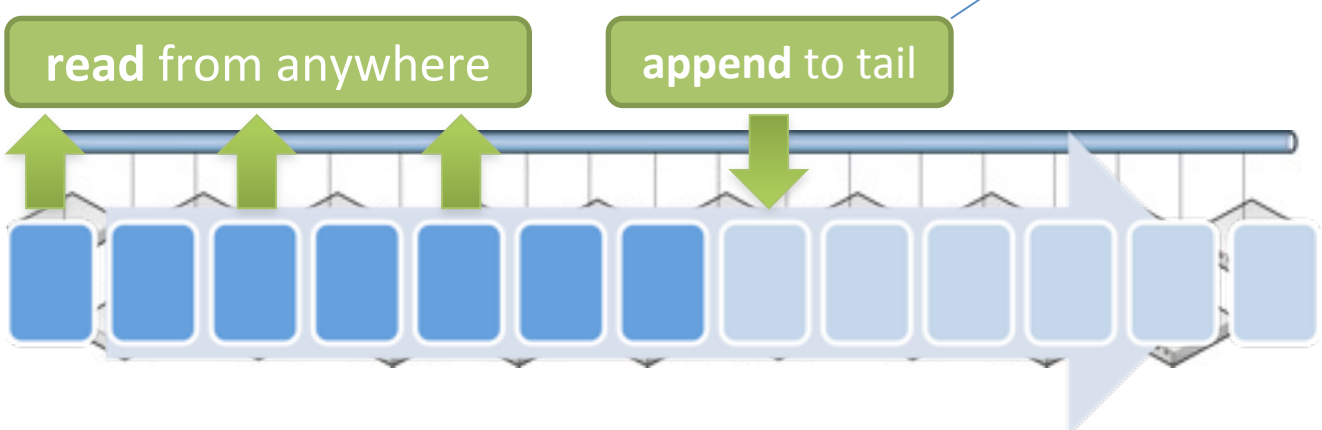
flash cluster

what is CORFU?

infrastructure applications:
key-value stores
databases
SMR
filesystems
virtual disks



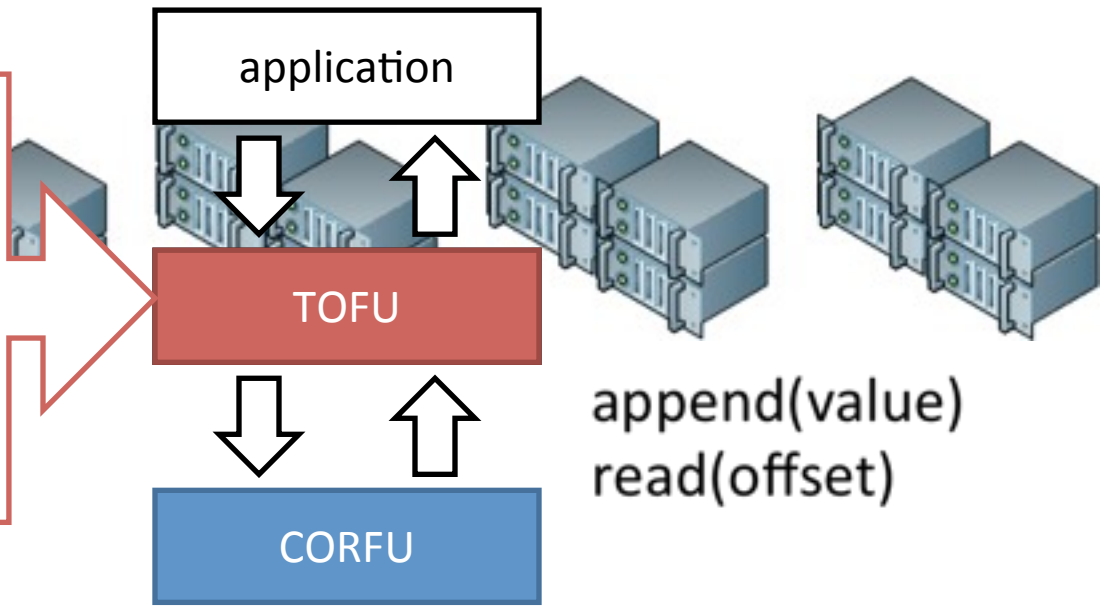
Replicated, fault-tolerant append



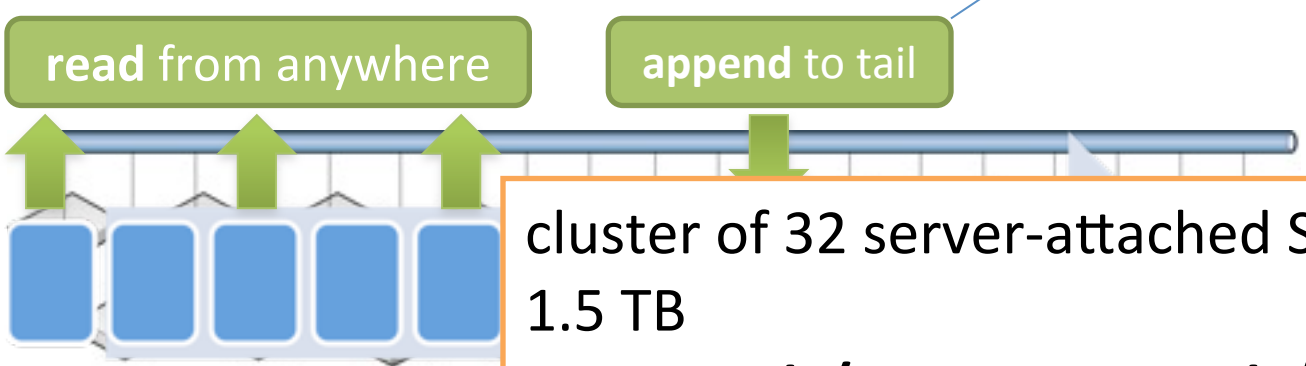
flash cluster

what is CORFU?

infrastructure applications:
key-value stores
databases
SMR
filesystems
virtual disks



Replicated, fault-tolerant append

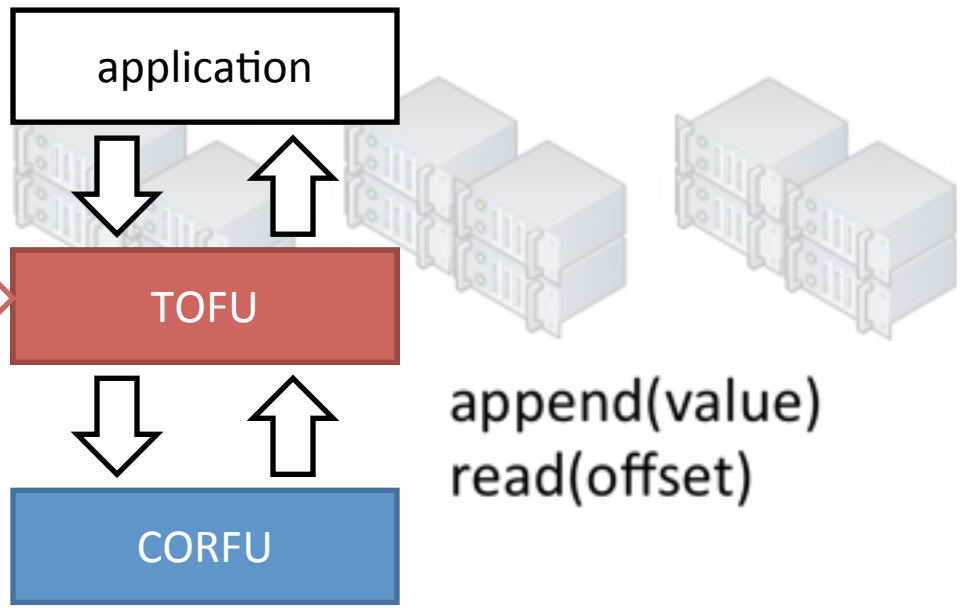


flash cluster

cluster of 32 server-attached SSDs, 1.5 TB
0.5M reads/s, 0.2M appends/s (4KB)

what is CORFU?

infrastructure applications:
key-value stores
databases
SMR
filesystems
virtual disks



Replicated, fault-tolerant append

read from anywhere

append to tail

network-attached flash
1 Gbps, **15W, 75 µs reads, 200 µs writes** (4KB)

cluster of 32 server-attached SSDs, 1.5 TB
0.5M reads/s, 0.2M appends/s (4KB)

the case for CORFU

applications append/read data

why a shared log interface?

1. easy to build strongly consistent (transactional) applications
2. effective way to pool flash:
    1. SSD uses logging to avoid write-in-place
    2. random reads are fast
    3. GC is feasible

CORFU is a distributed SSD with a shared log interface

getting 10 Gbps random-IO from 1TB flash farm:

	Configuration	Unit cost	Unit power consumption	Summary
competition	ten SATA SSDs in 1Gbps server	$200/SSD + $2K/ server	150W	$22K 1500W fault tolerance ~~incremental scalability~~
	individual PCI-e controller in 10 Gbps server	$20K/Enterprise controller + $10K/server	500W	$30K 500W ~~fault tolerance~~ ~~incremental scalability~~
CORFU	ten 1Gbps CORFU units (no servers)	$50/raw flash + $200/custom-made controller	10W	$2.5K 100W fault tolerance incremental scalability

the CORFU design

application

CORFU library

CORFU API:
read(*O*)
append(*V*)
trim(*O*)

read from anywhere

append to tail

filled 1 | filled 2 | filled 3 | filled 4 | filled 5 | filled 6 | filled 7 | filled 8 | filled 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16

4KB log entry

the CORFU design

application

CORFU library

CORFU API:
read(*O*)
append(*V*)
trim(*O*)

read from anywhere

append to tail

filled 1 | filled 2 | filled 3 | filled 4 | filled 5 | filled 6 | filled 7 | filled 8 | filled 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16

4KB log entry

each logical entry is mapped to a replica set of physical flash pages

the CORFU design

mapping resides **at the client**

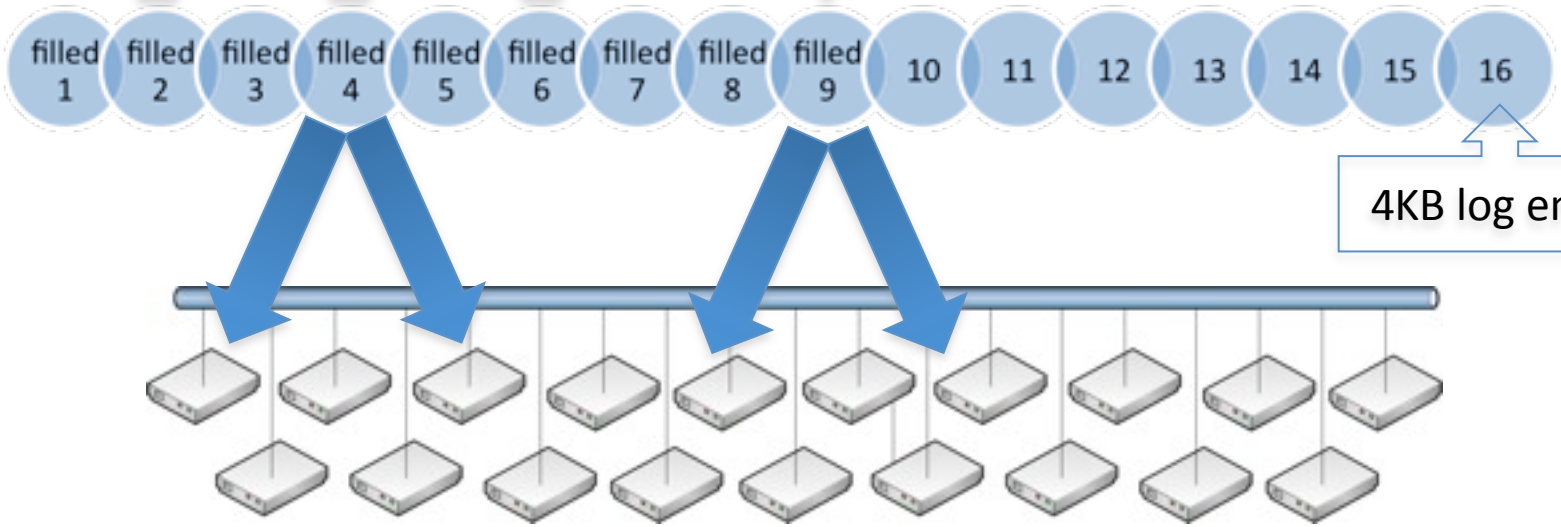
application

CORFU library

CORFU API:
read(*O*)
append(*V*)
trim(*O*)

read from anywhere

append to tail



each logical entry is mapped to a replica set of physical flash pages

the CORFU design

mapping resides **at the client**

application

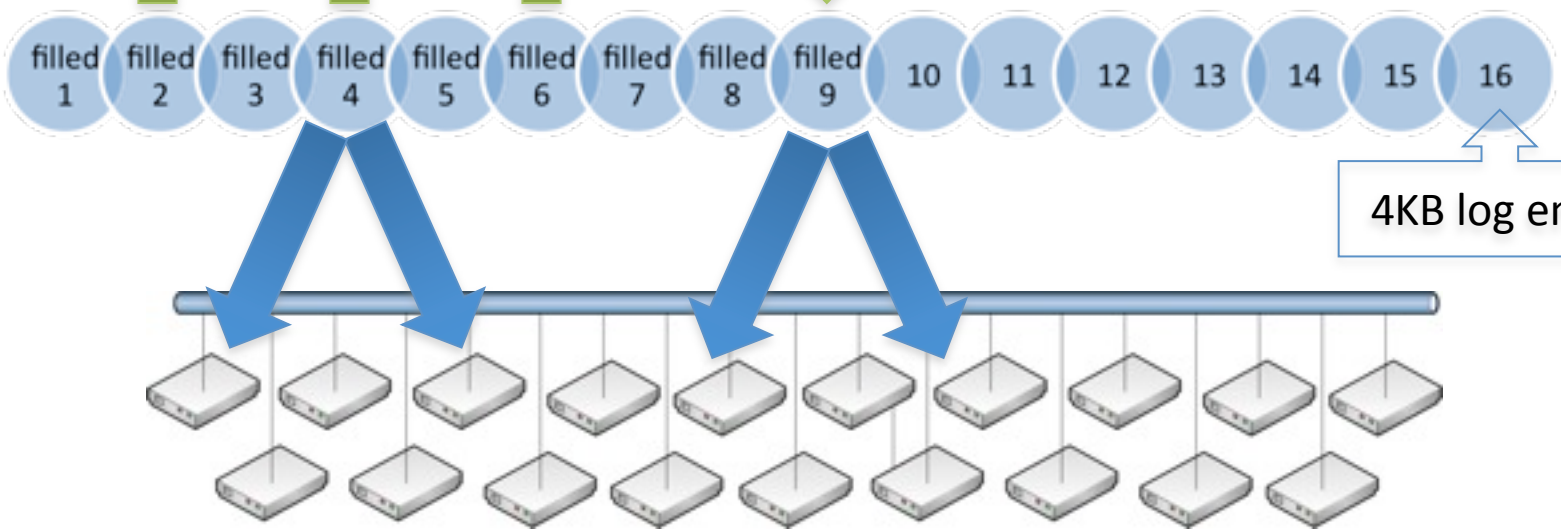
CORFU library

CORFU API:
read(*O*)
append(*V*)
trim(*O*)

read from anywhere

append to tail

sequencer



each logical entry is mapped to a replica set of physical flash pages

the CORFU design

mapping resides **at the client**

application

CORFU library

CORFU API:

CORFU append throughput: # of 64-bit tokens issued per second

read from anywhere

append to tail

sequencer

filled 1 | filled 2 | filled 3 | filled 4 | filled 5 | filled 6 | filled 7 | filled 8 | filled 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16

4KB log entry

each logical entry is mapped to a replica set of physical flash pages

Striping

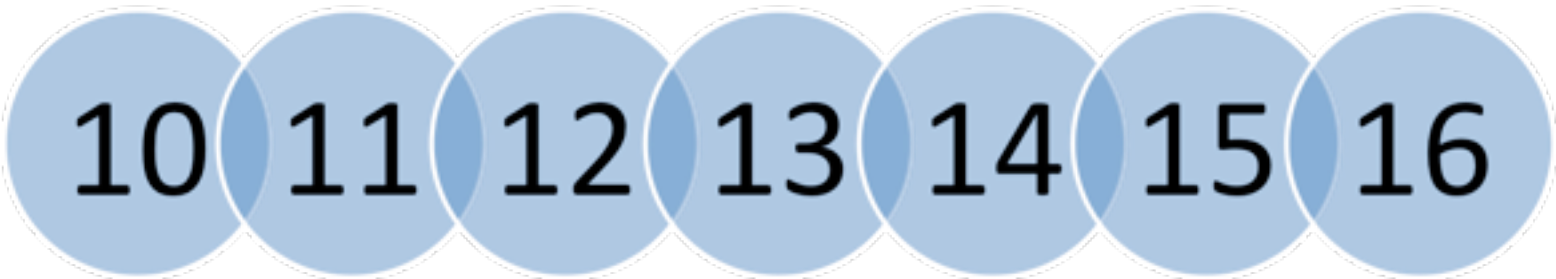
application

CORFU library

Striping

application

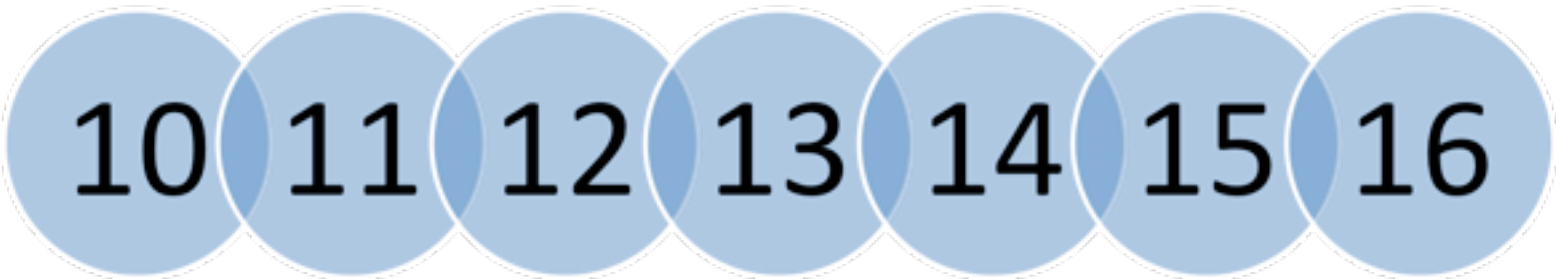
CORFU library



Striping

application

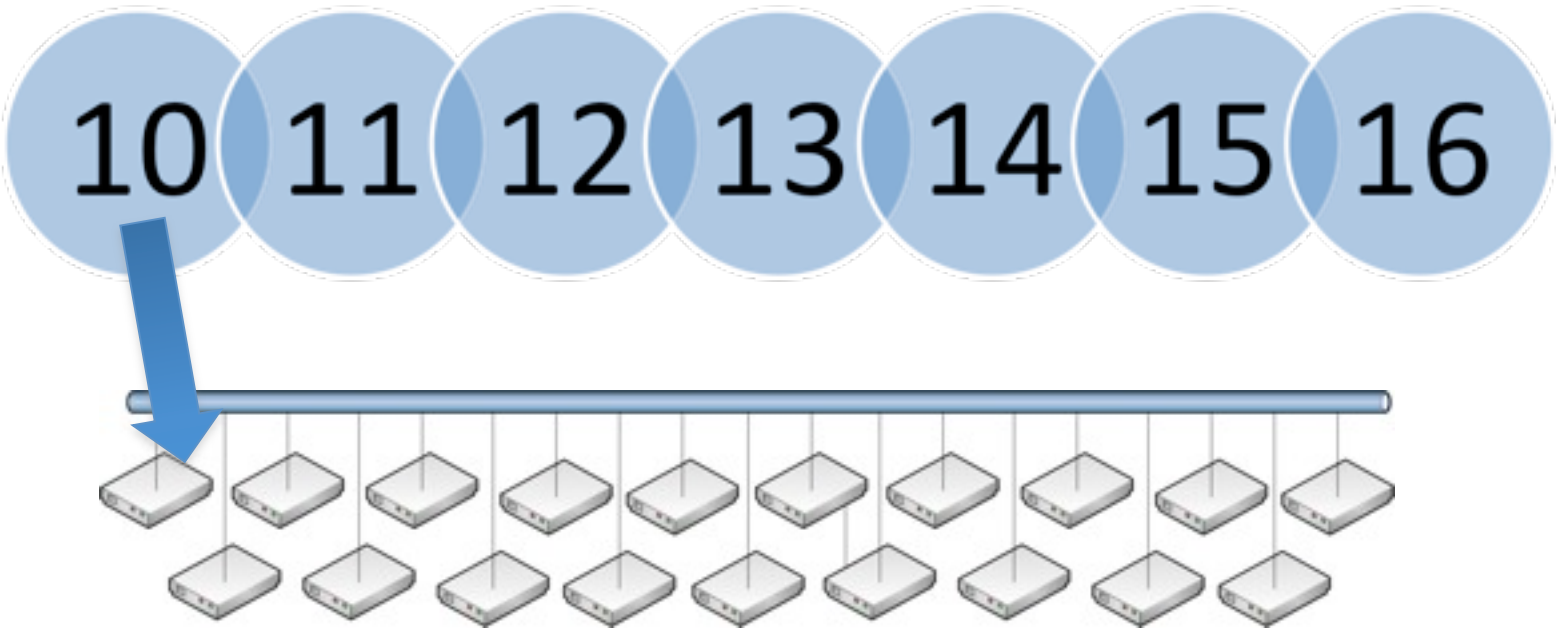
CORFU library



Striping

application

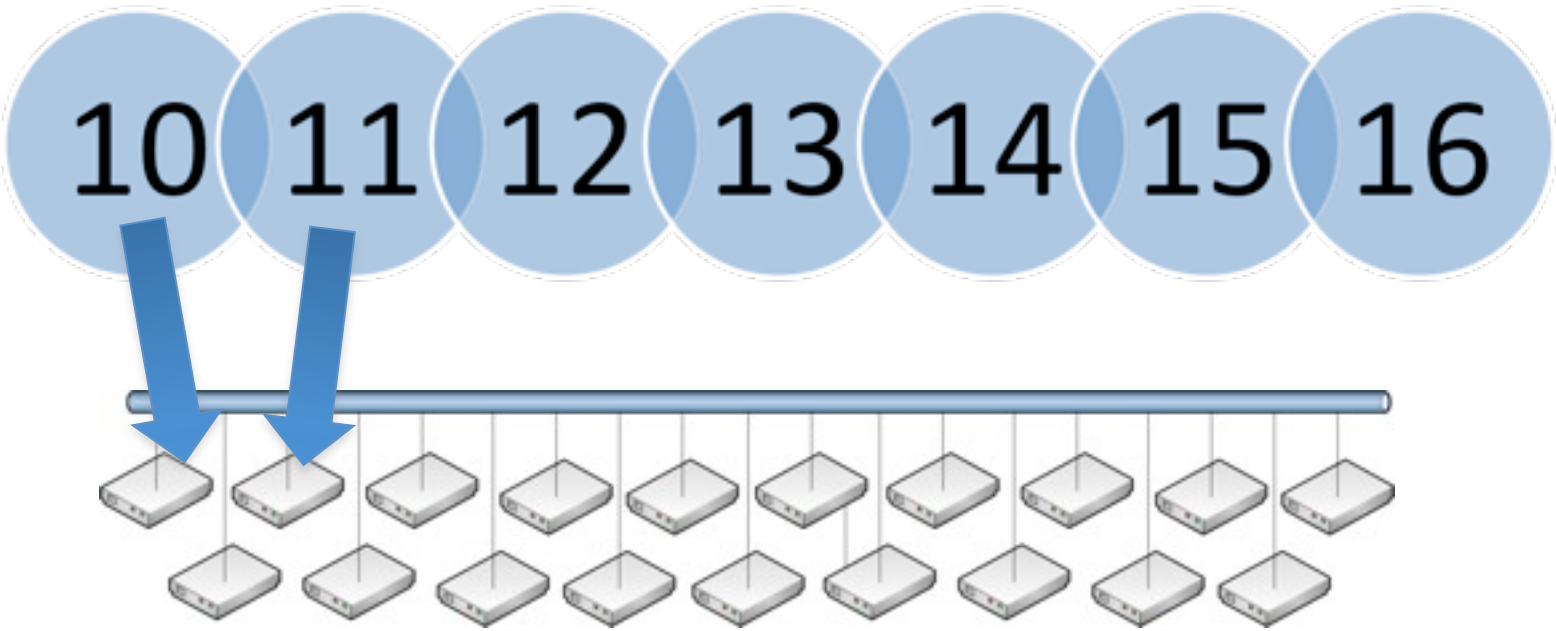
CORFU library



Striping

application

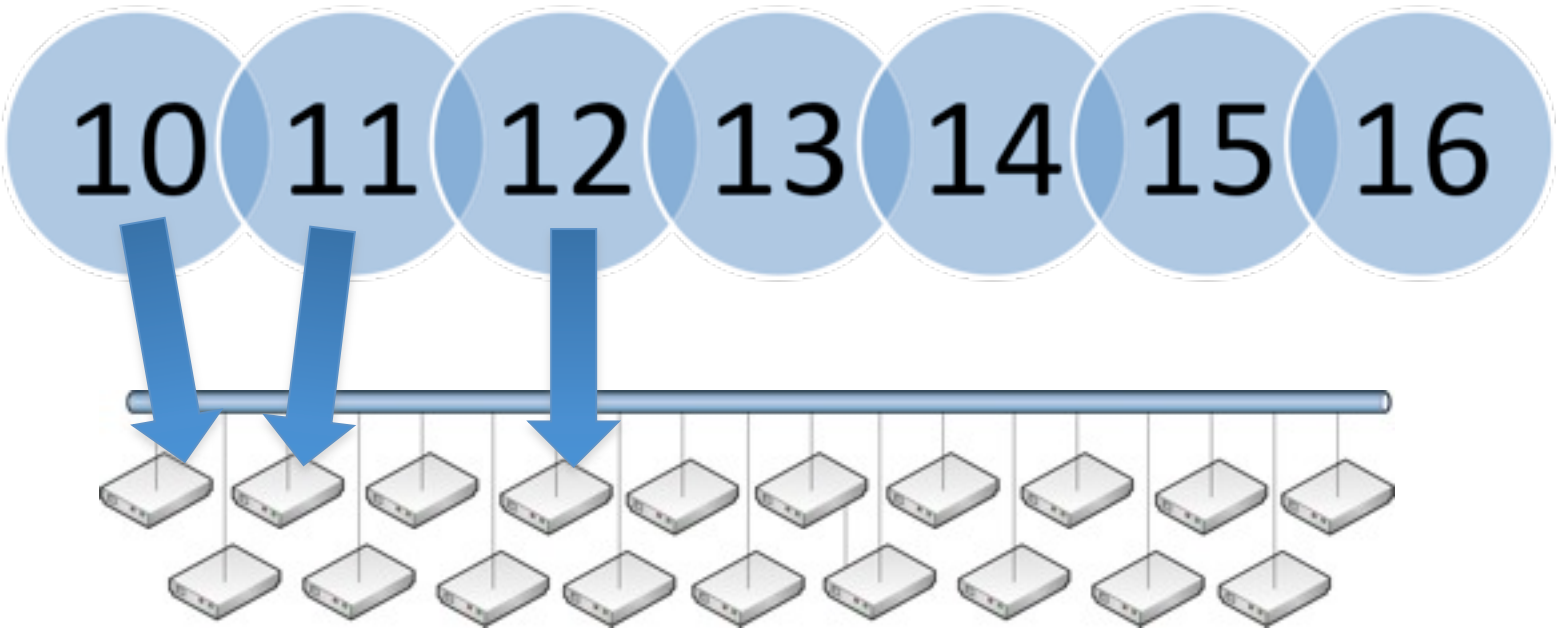
CORFU library



Striping

application

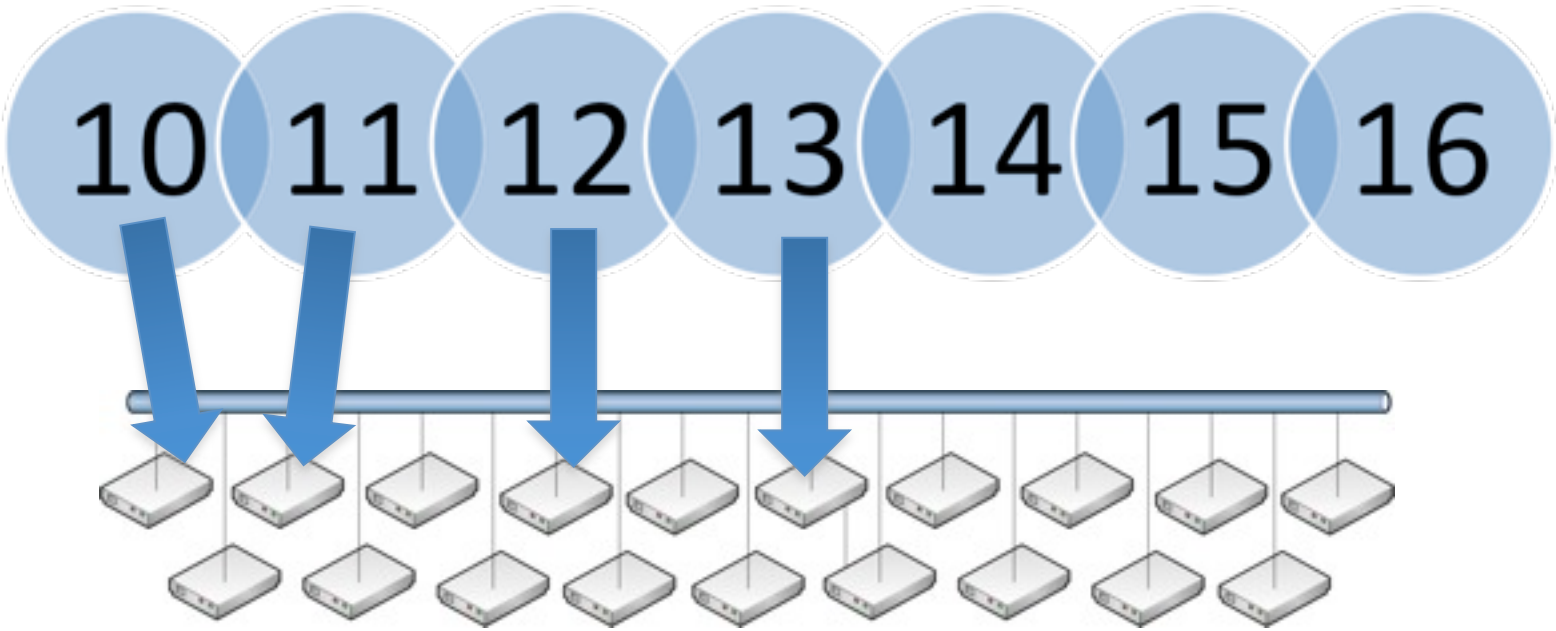
CORFU library



Striping

application

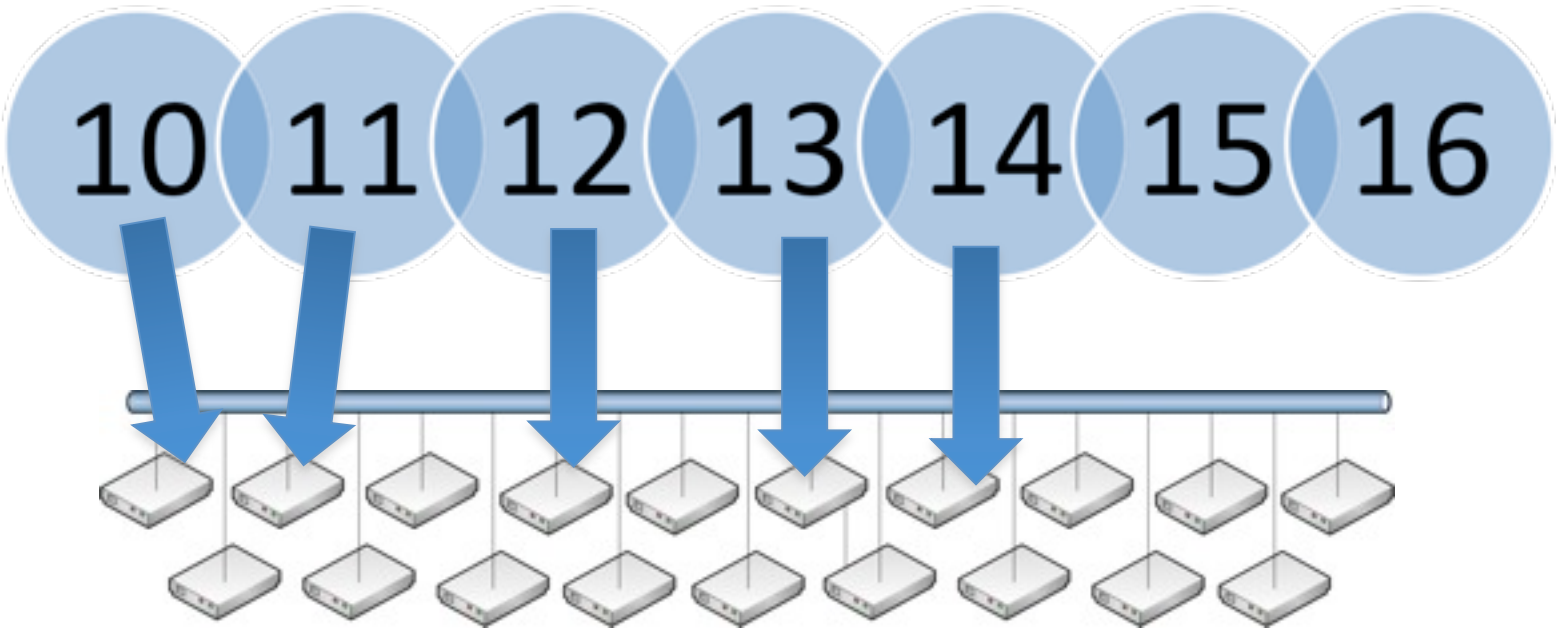
CORFU library



Striping

application

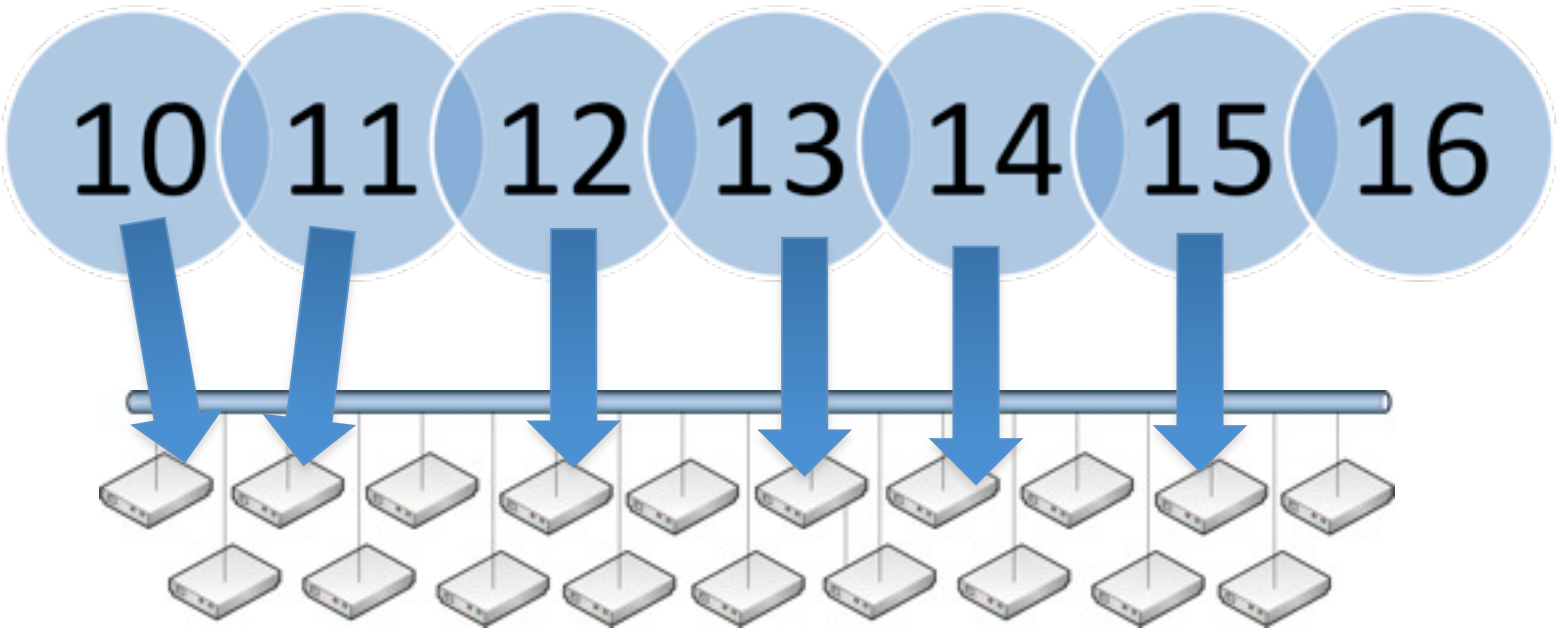
CORFU library



Striping

application

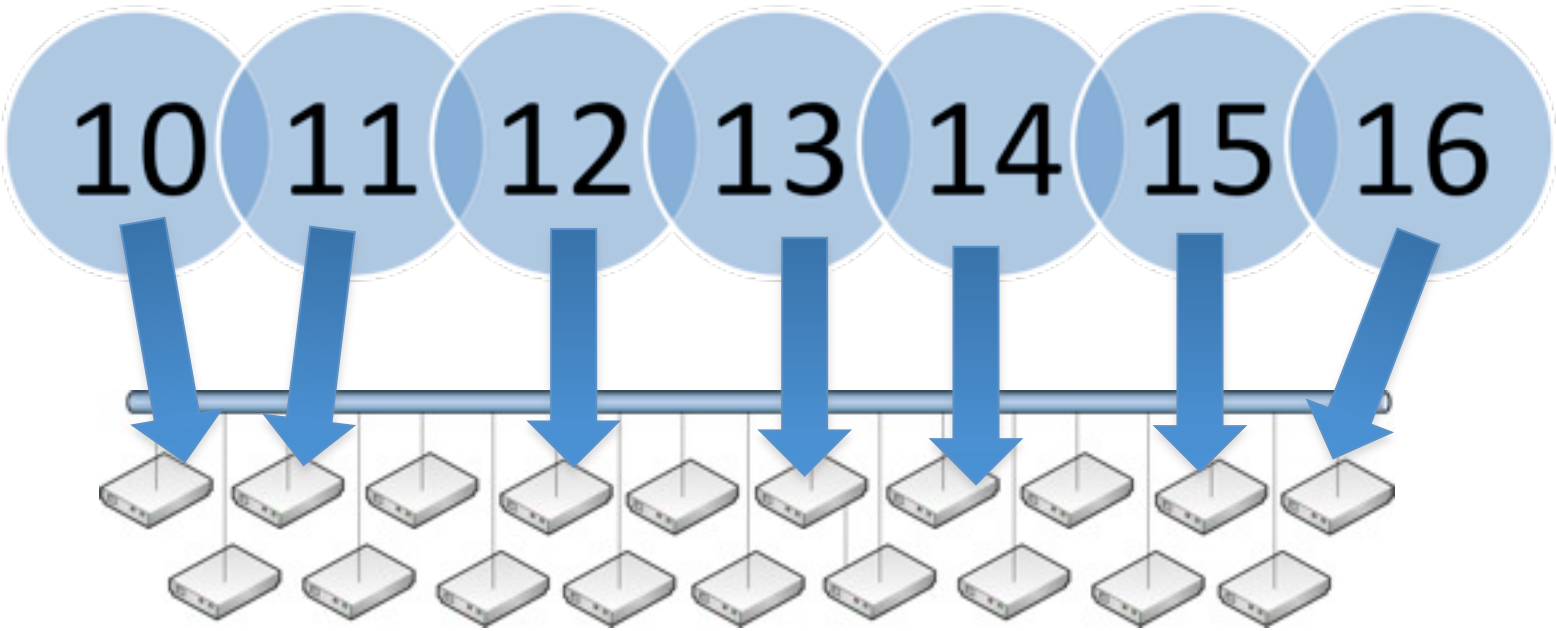
CORFU library



Striping

application

CORFU library



Chaining (Replication)

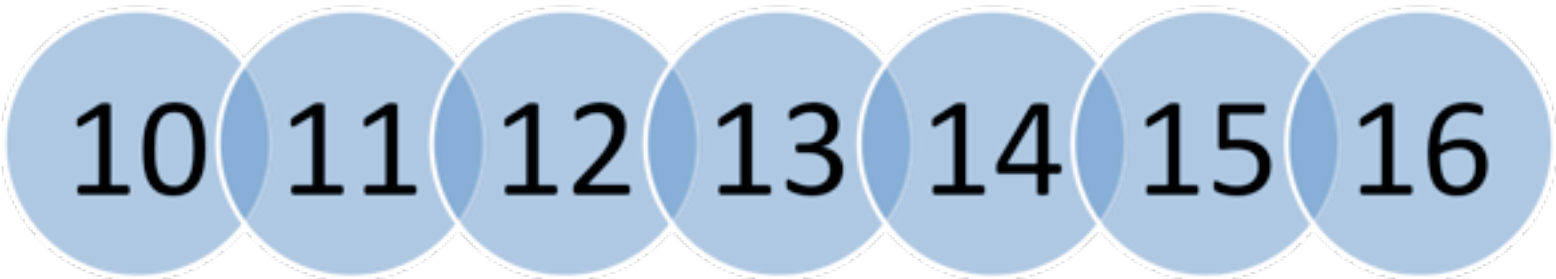
application

CORFU library

Chaining (Replication)

application

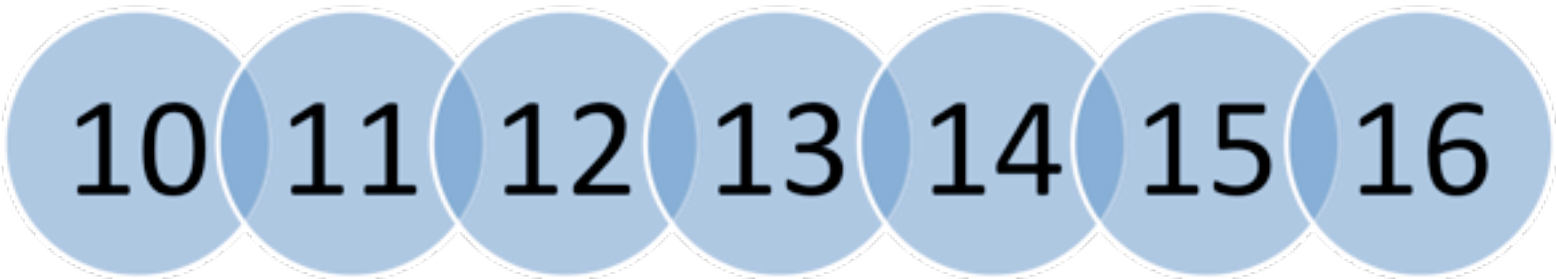
CORFU library



Chaining (Replication)

application

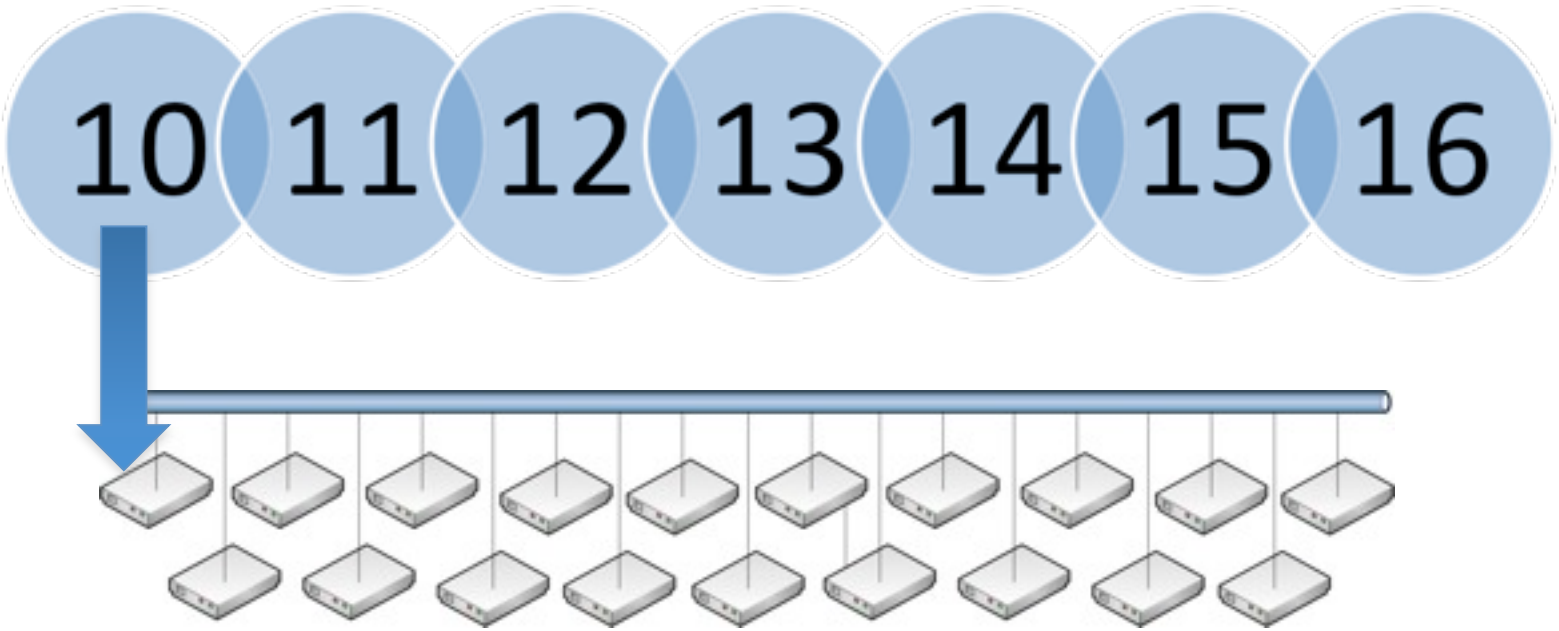
CORFU library



Chaining (Replication)

application

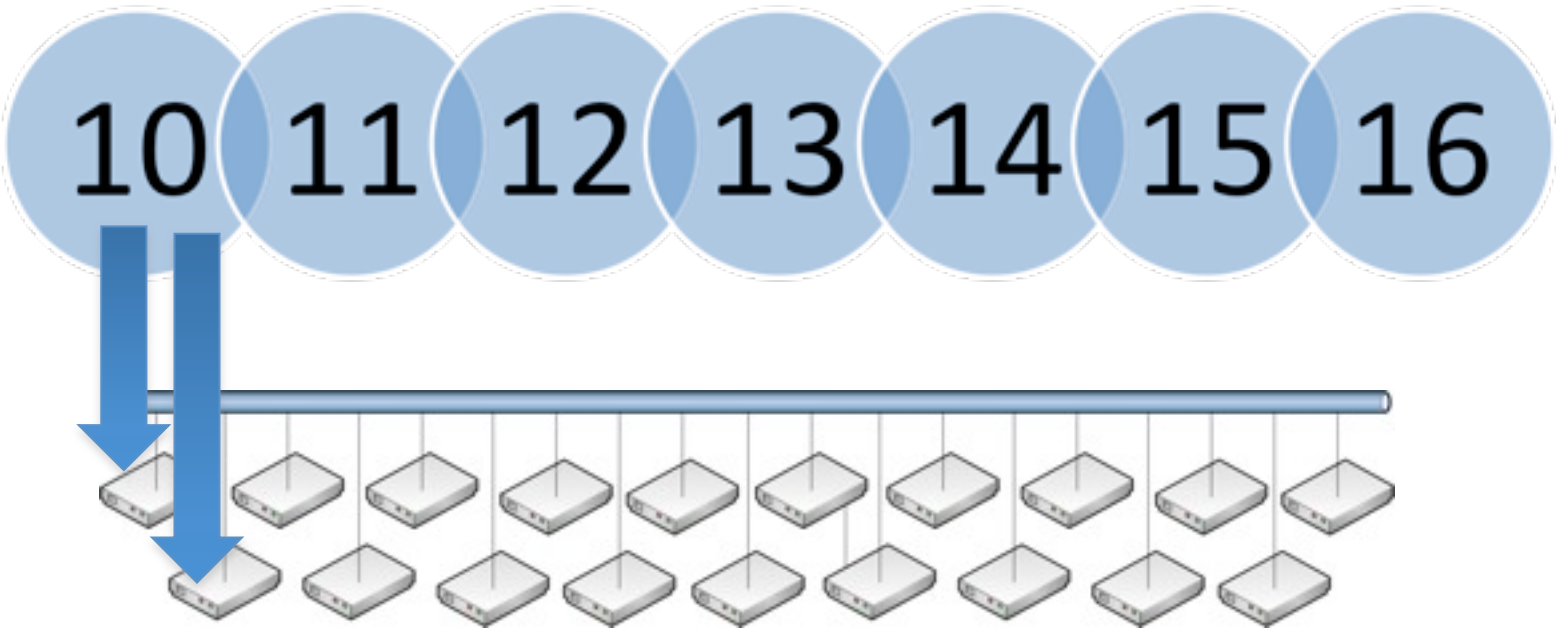
CORFU library



Chaining (Replication)

application

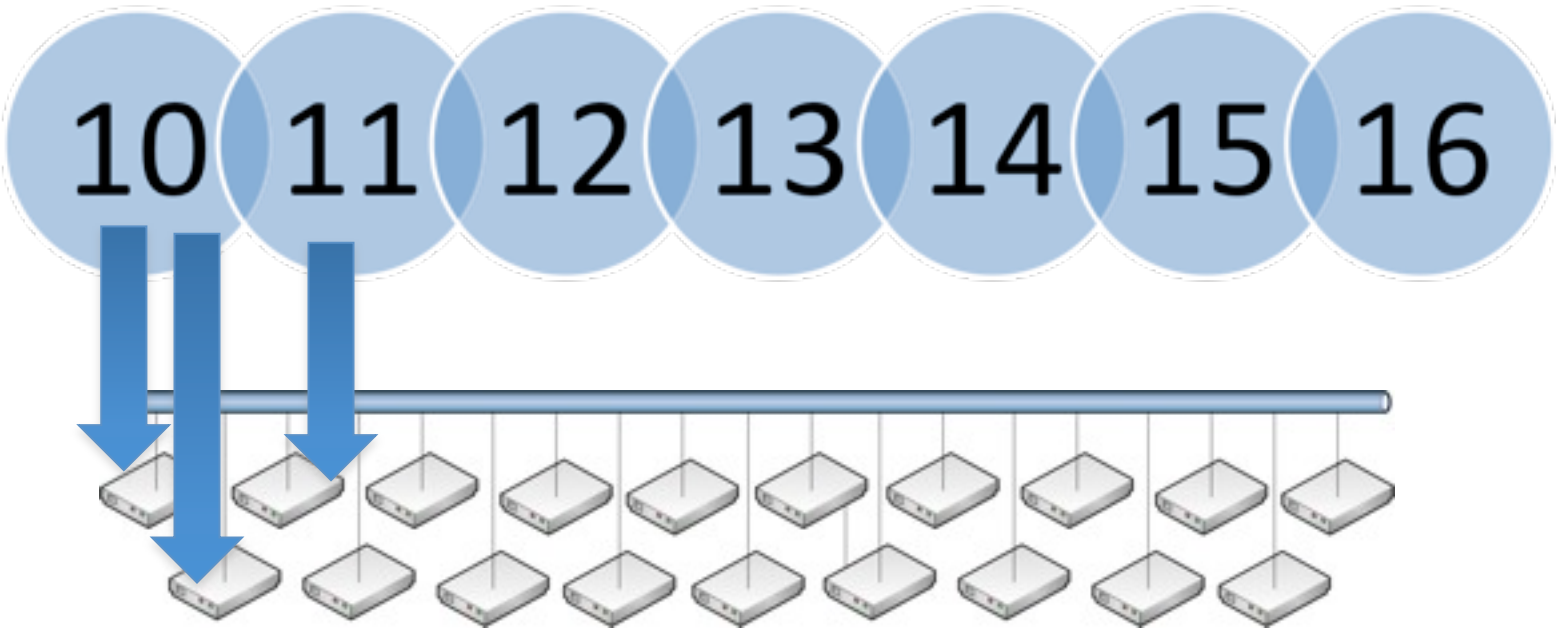
CORFU library



Chaining (Replication)

application

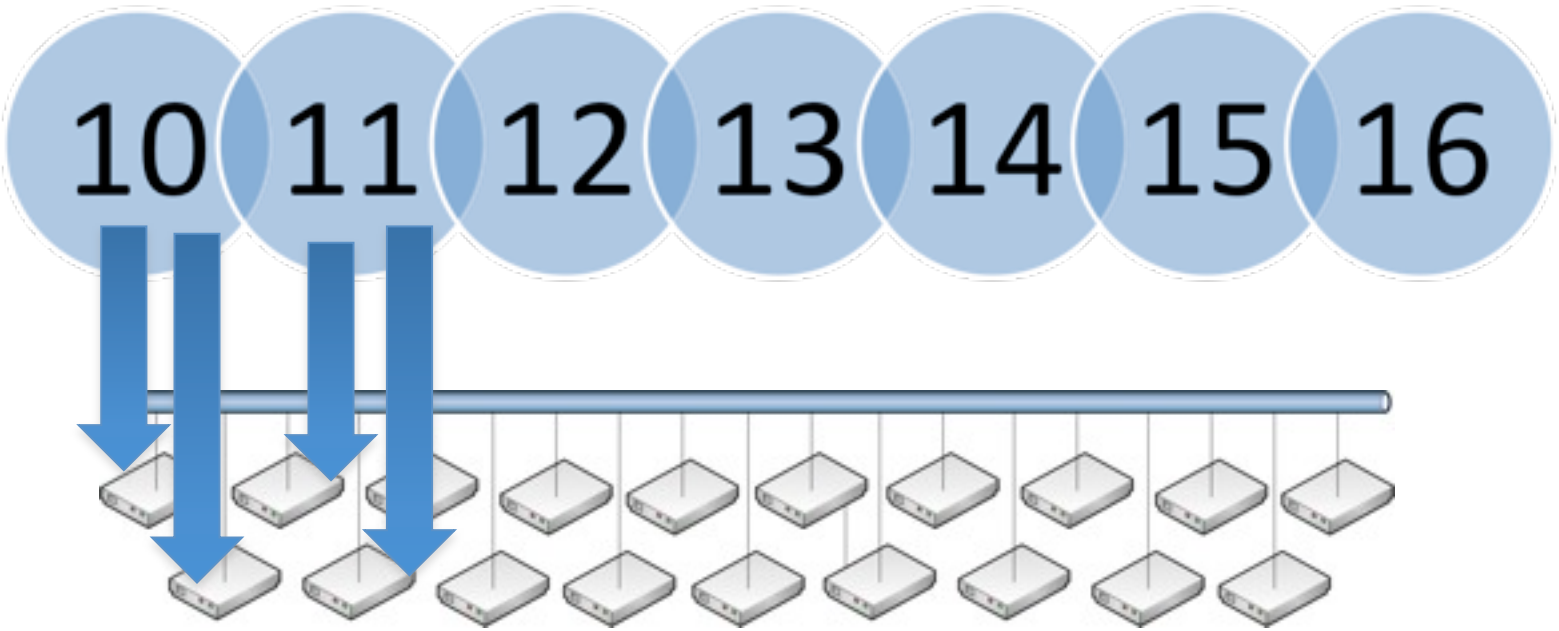
CORFU library



Chaining (Replication)

application

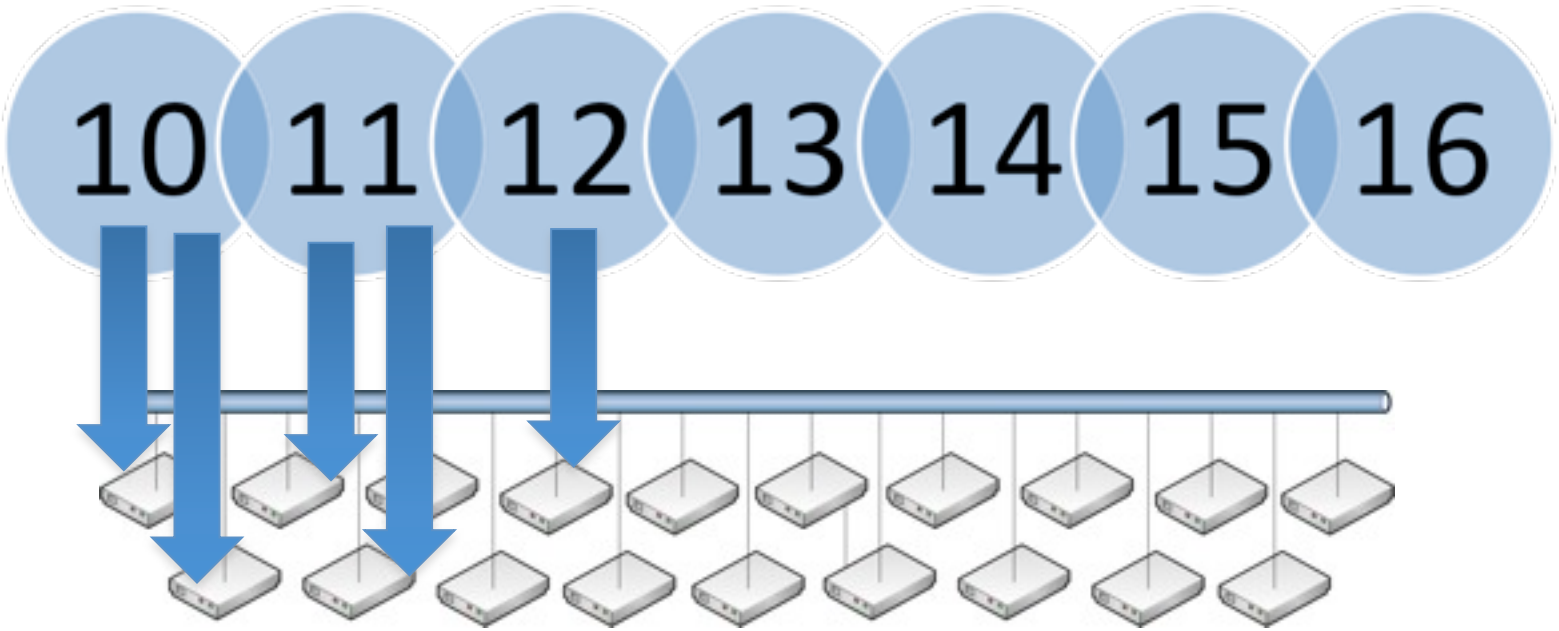
CORFU library



Chaining (Replication)

application

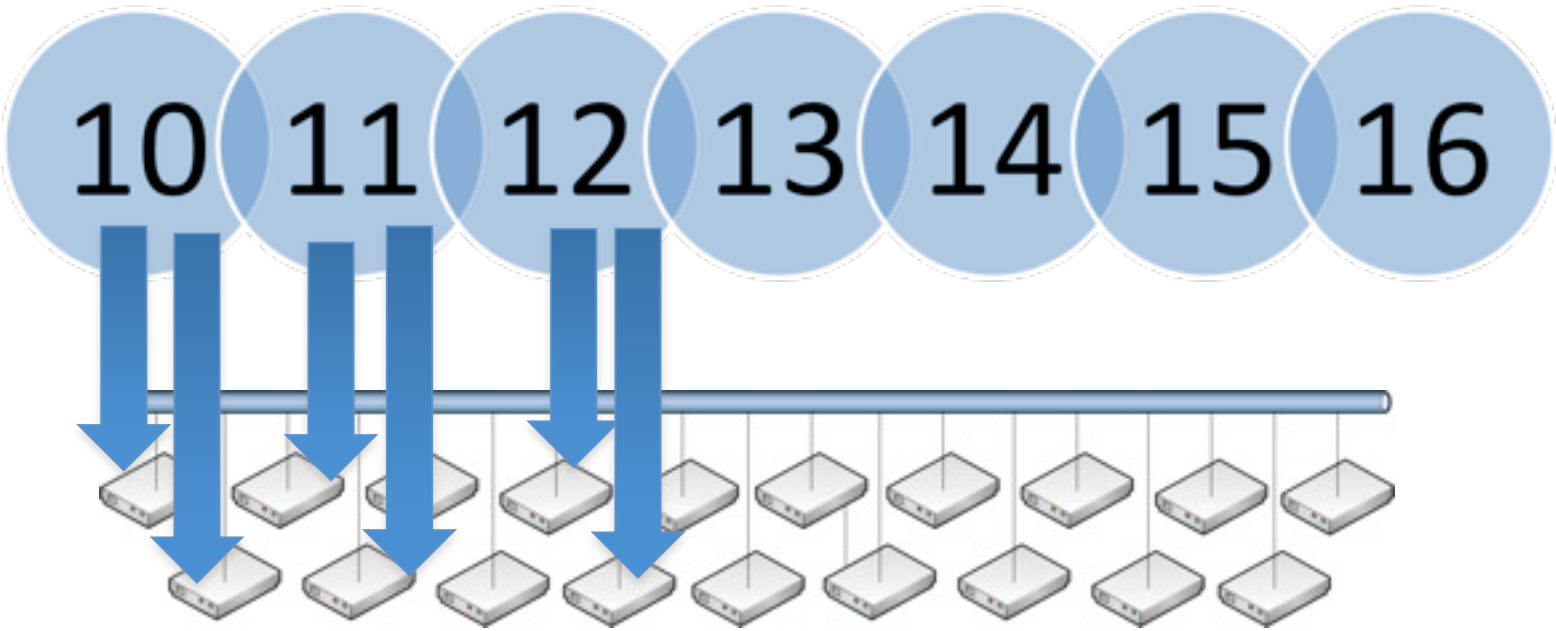
CORFU library



Chaining (Replication)

application

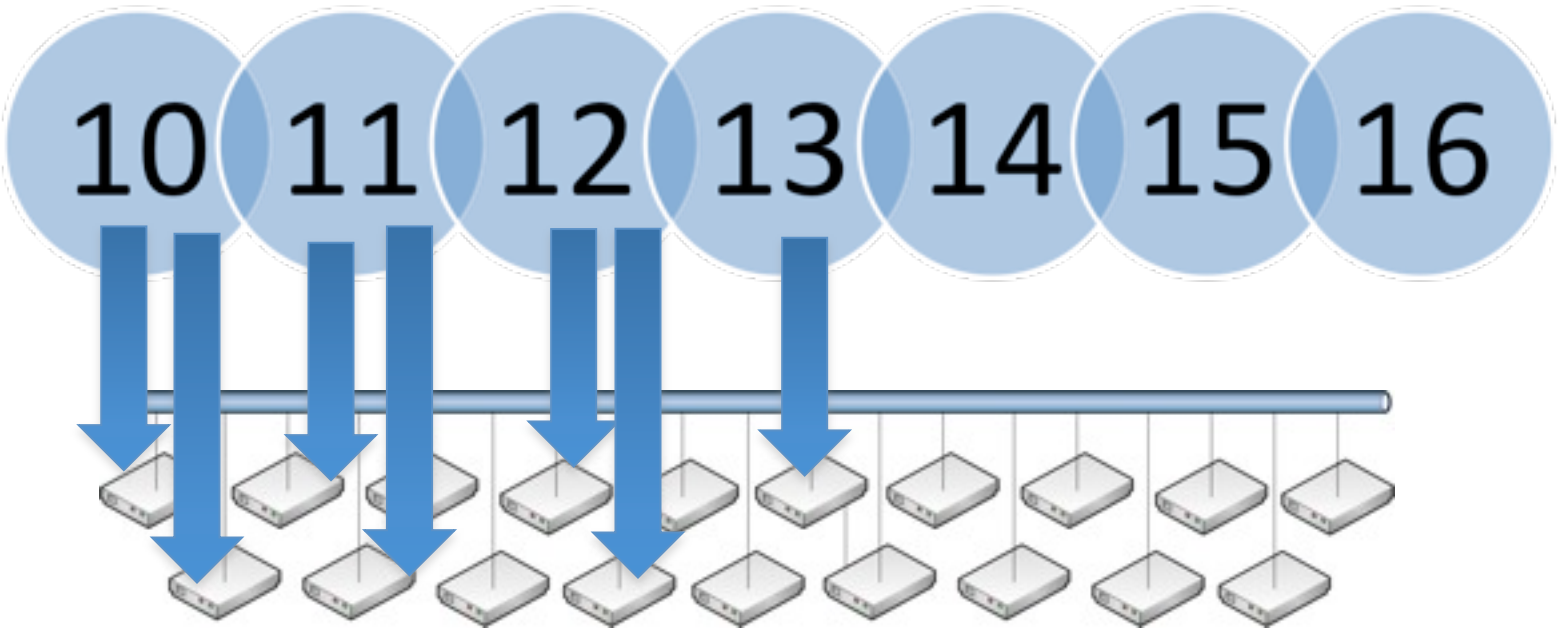
CORFU library



Chaining (Replication)

application

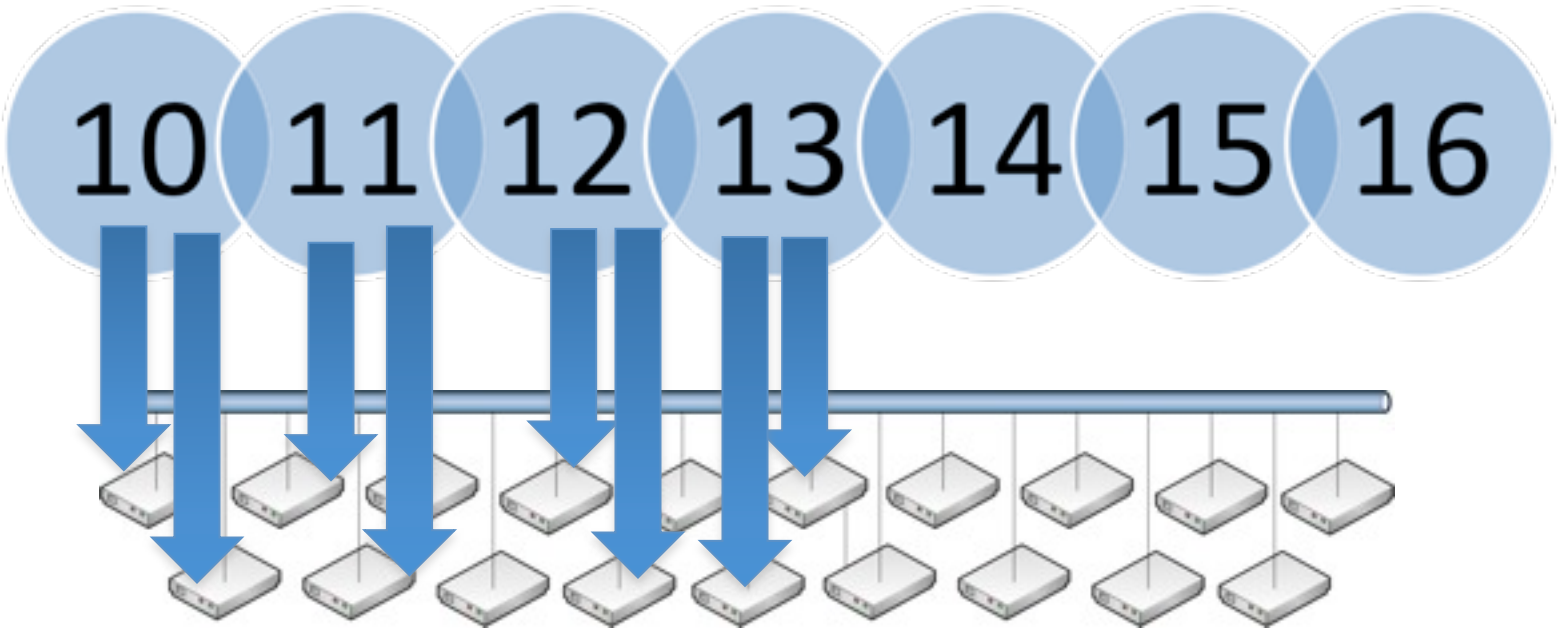
CORFU library



Chaining (Replication)

application

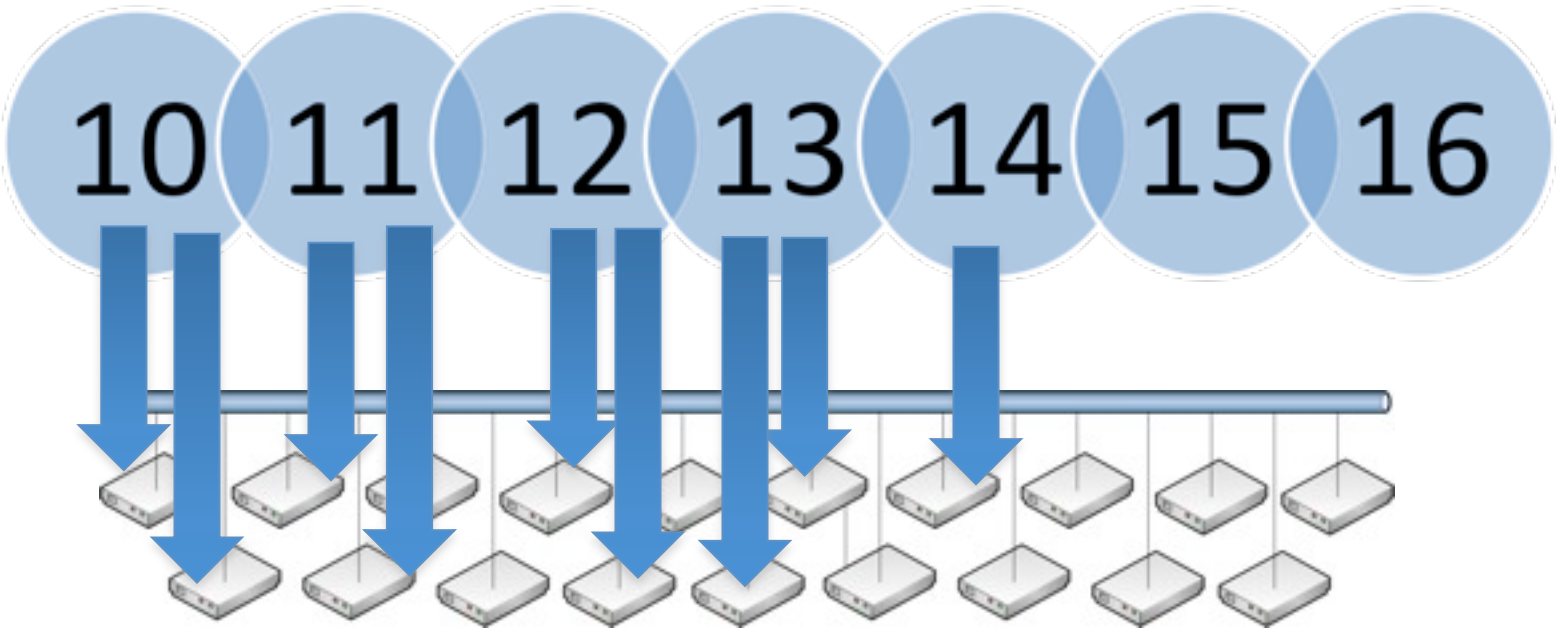
CORFU library



Chaining (Replication)

application

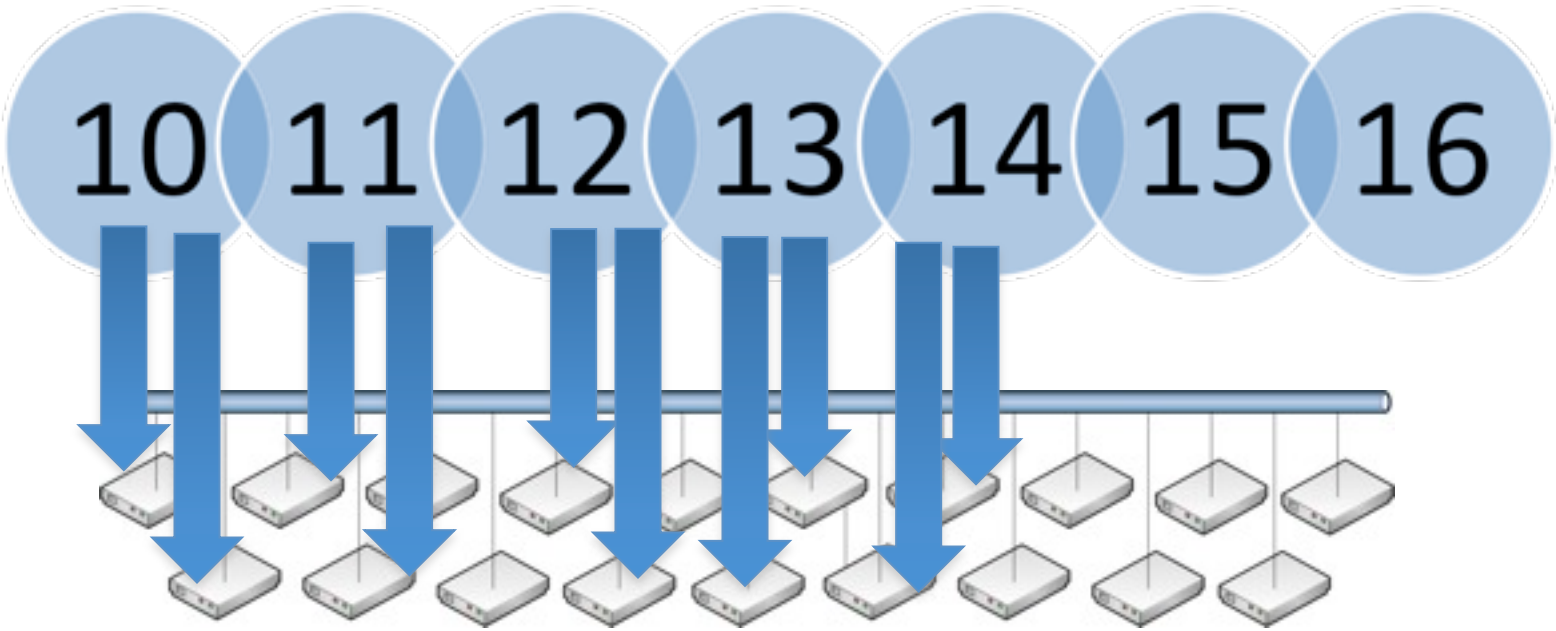
CORFU library



Chaining (Replication)

application

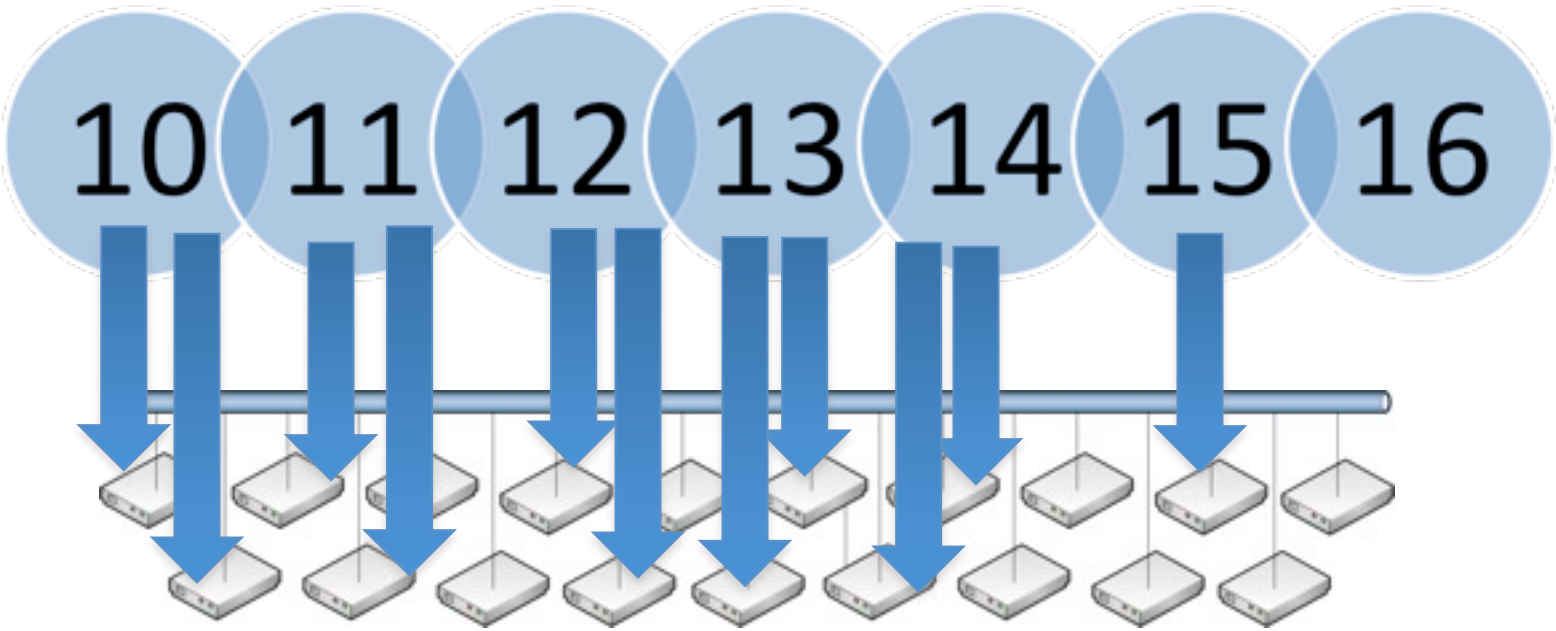
CORFU library



Chaining (Replication)

application

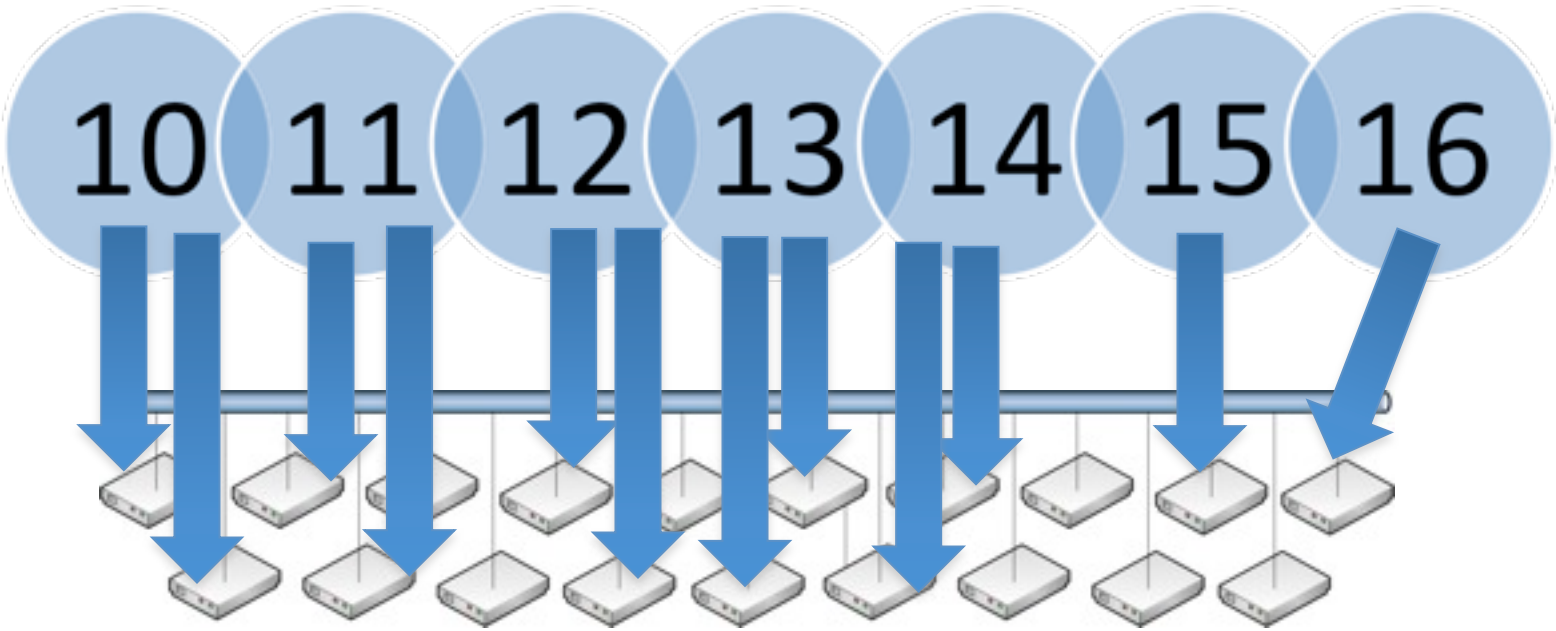
CORFU library



Chaining (Replication)

application

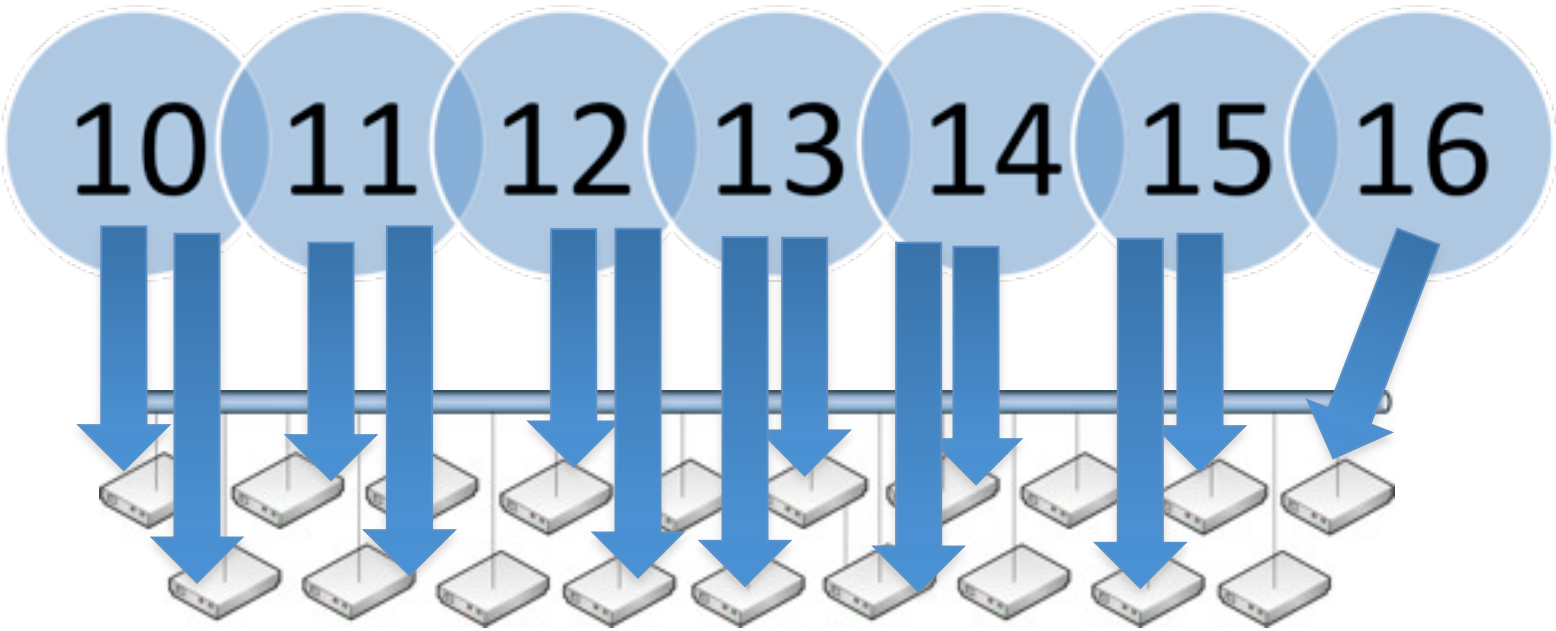
CORFU library



Chaining (Replication)

application

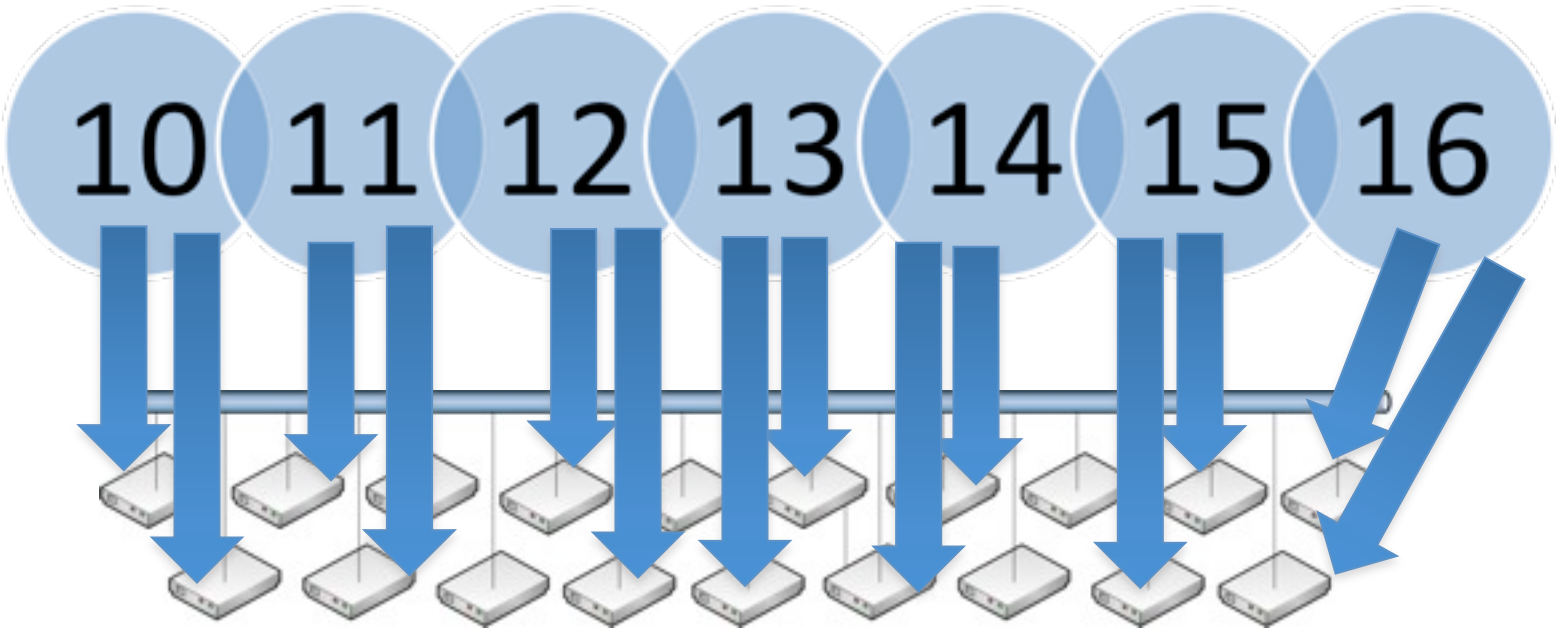
CORFU library



Chaining (Replication)

application

CORFU library



Holes

application

CORFU library

application

CORFU library

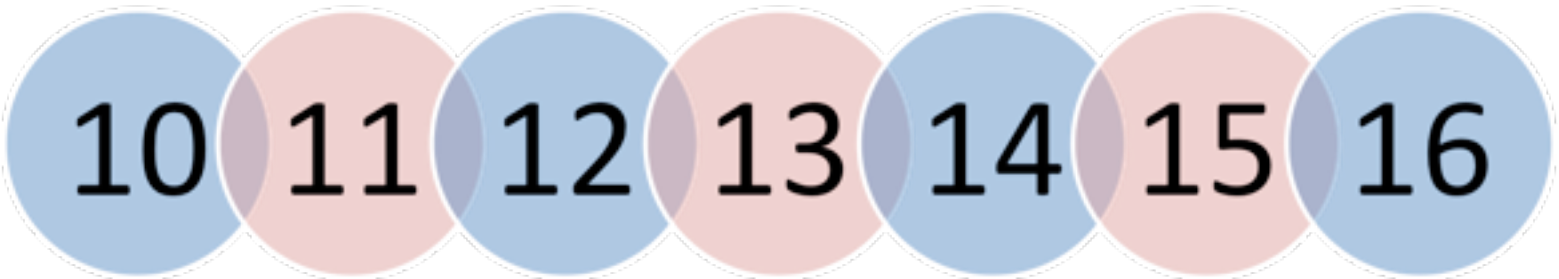
Holes

application

CORFU library

application

CORFU library



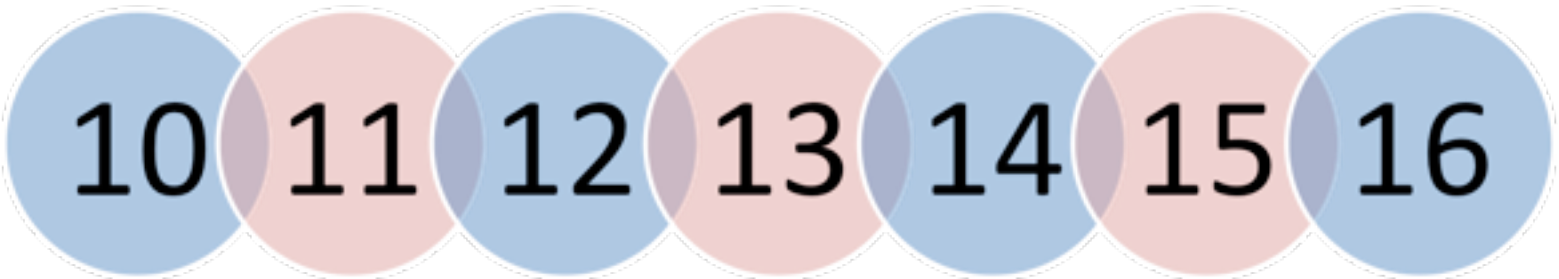
Holes

application

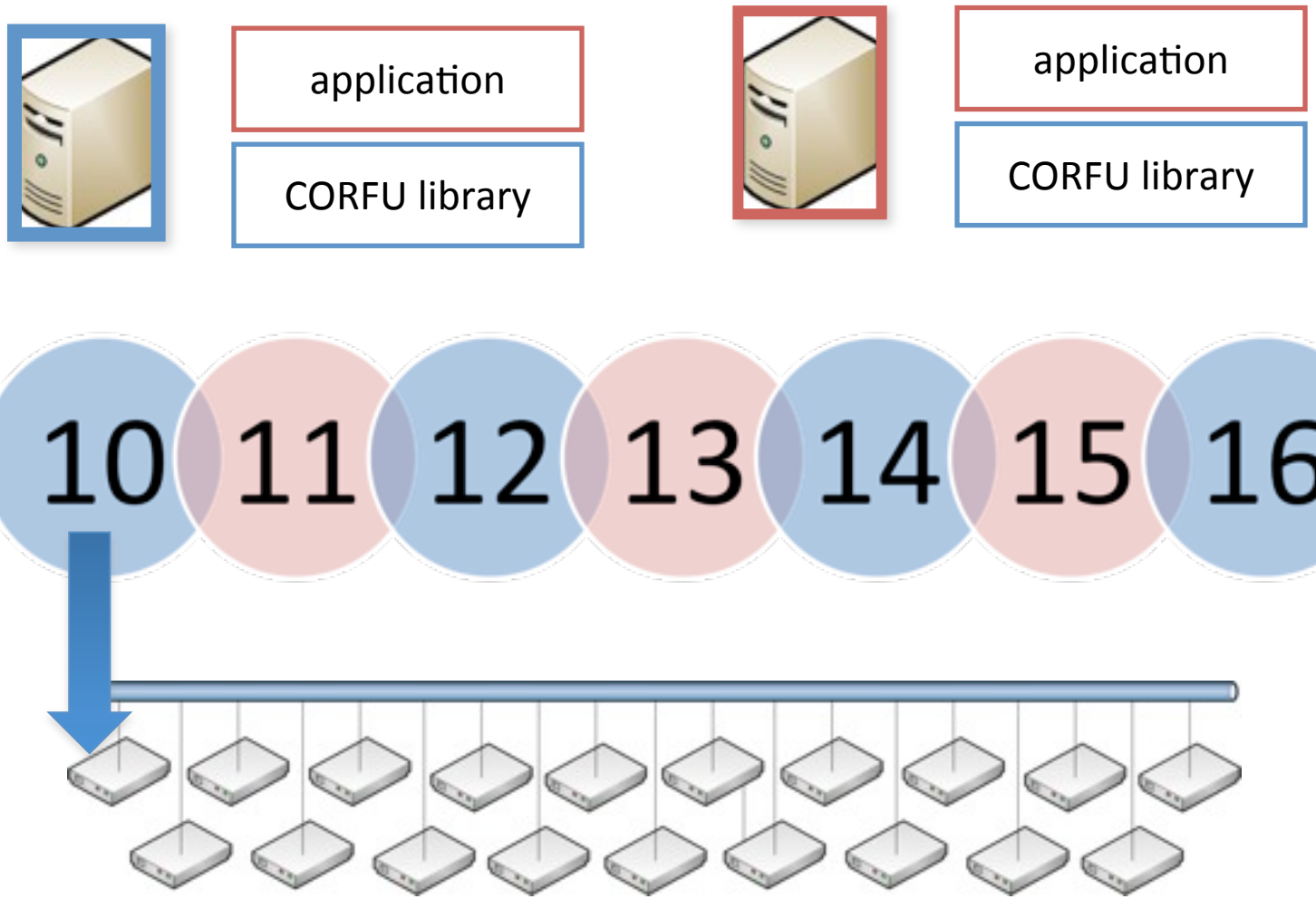
CORFU library

application

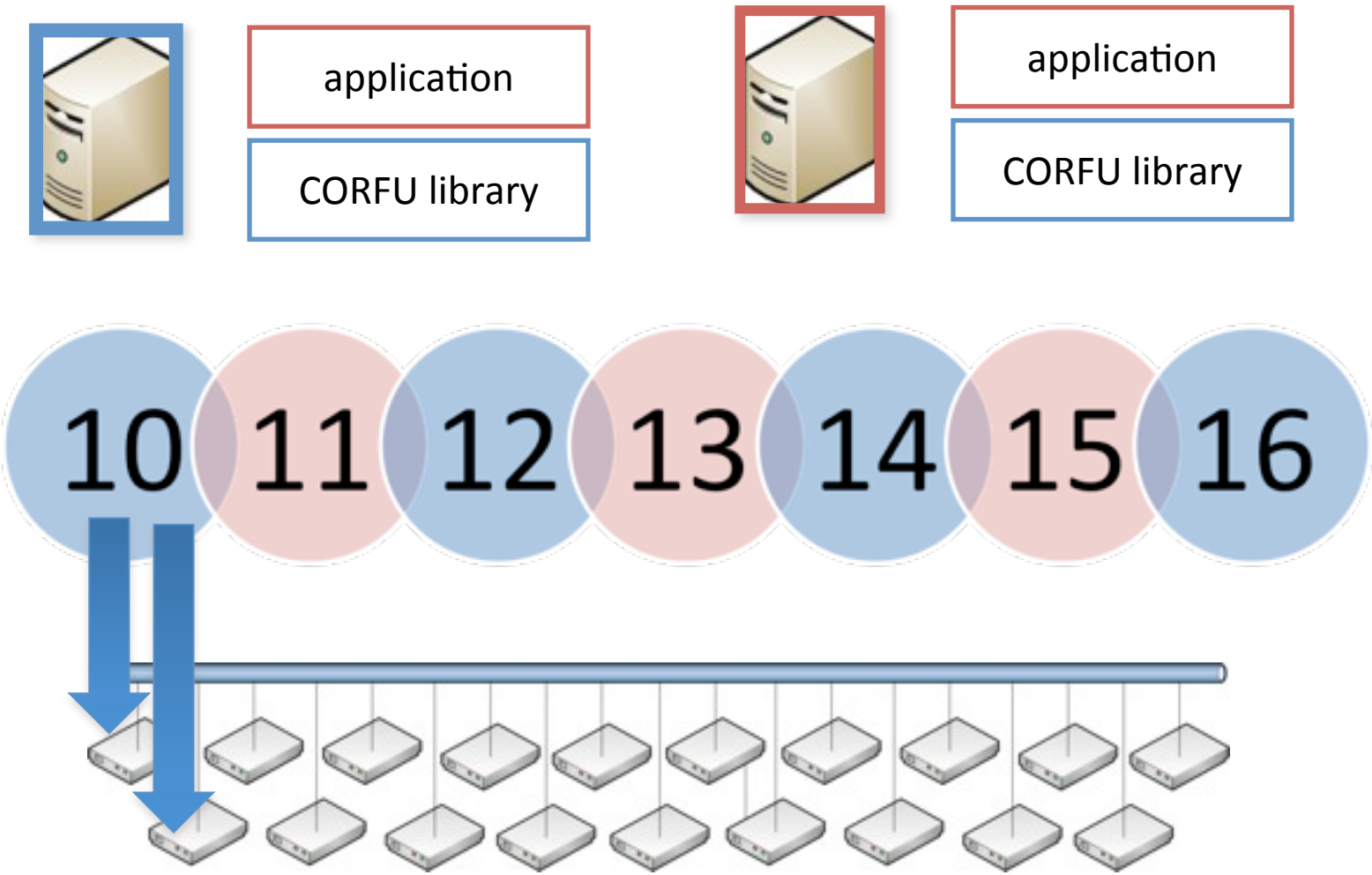
CORFU library



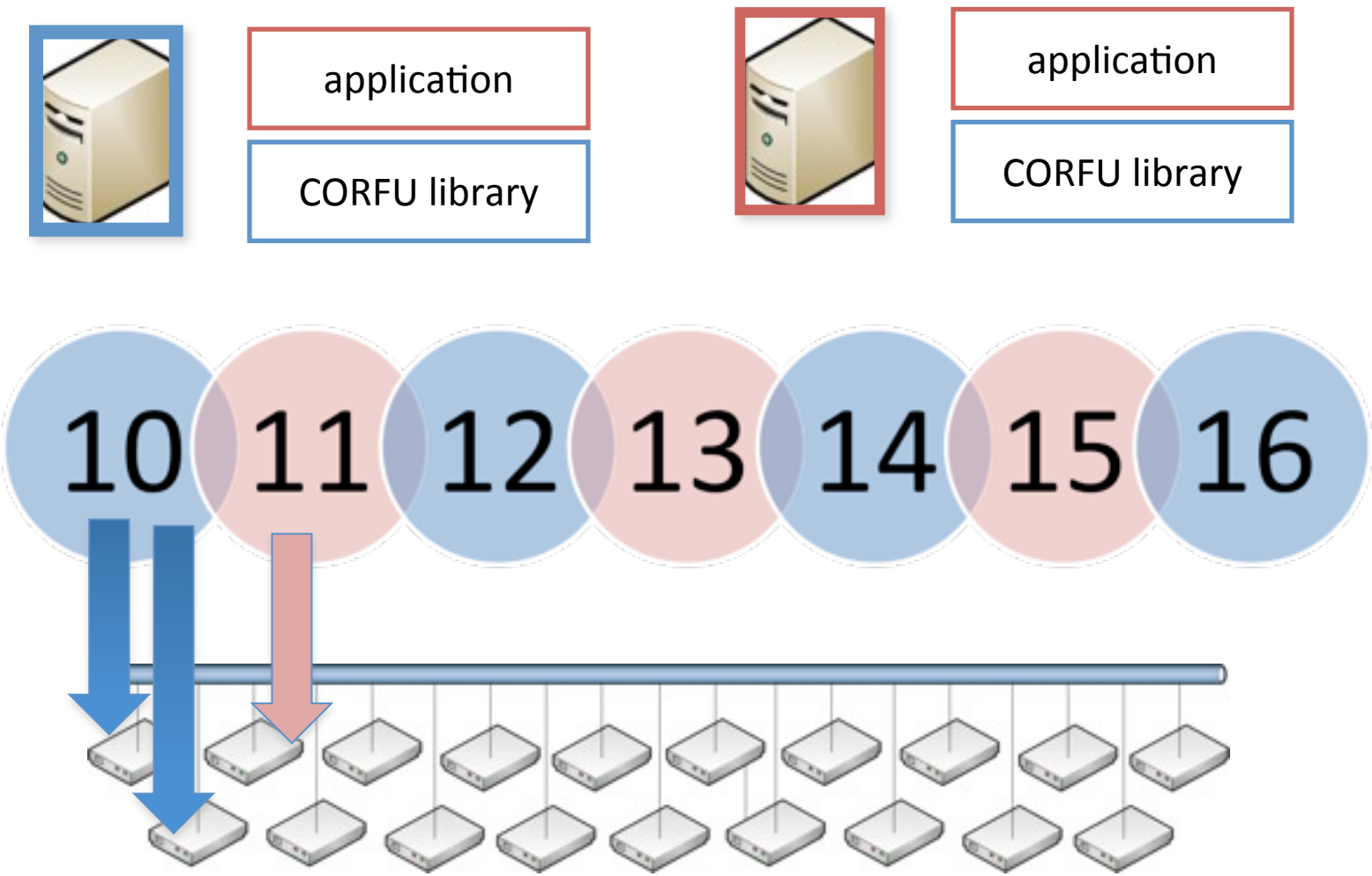
Holes


application
CORFU library
application
CORFU library
10
11
12
13
14
15
16

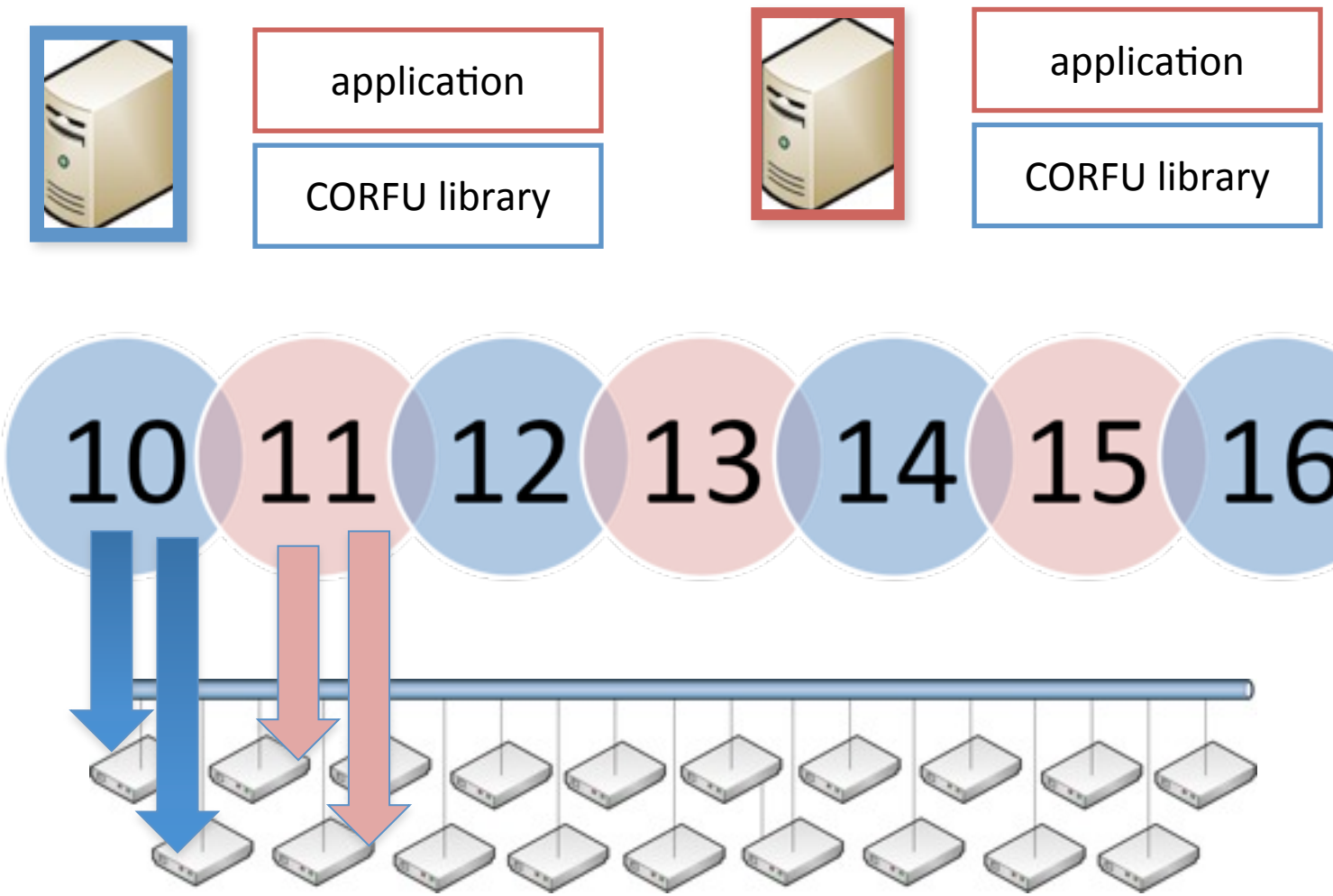
Holes


application
CORFU library
application
CORFU library
10
11
12
13
14
15
16

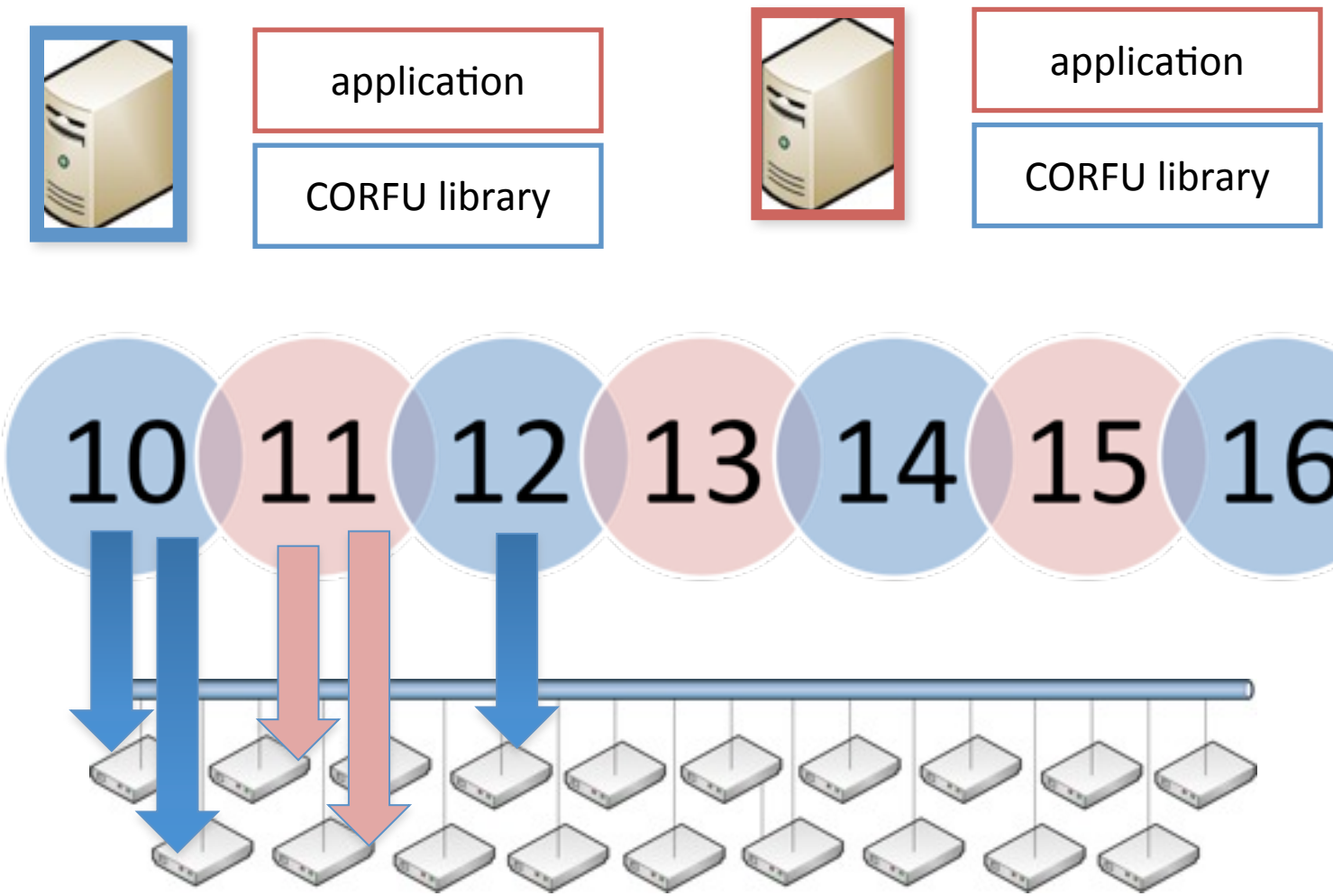
Holes


application
CORFU library
application
CORFU library
10
11
12
13
14
15
16

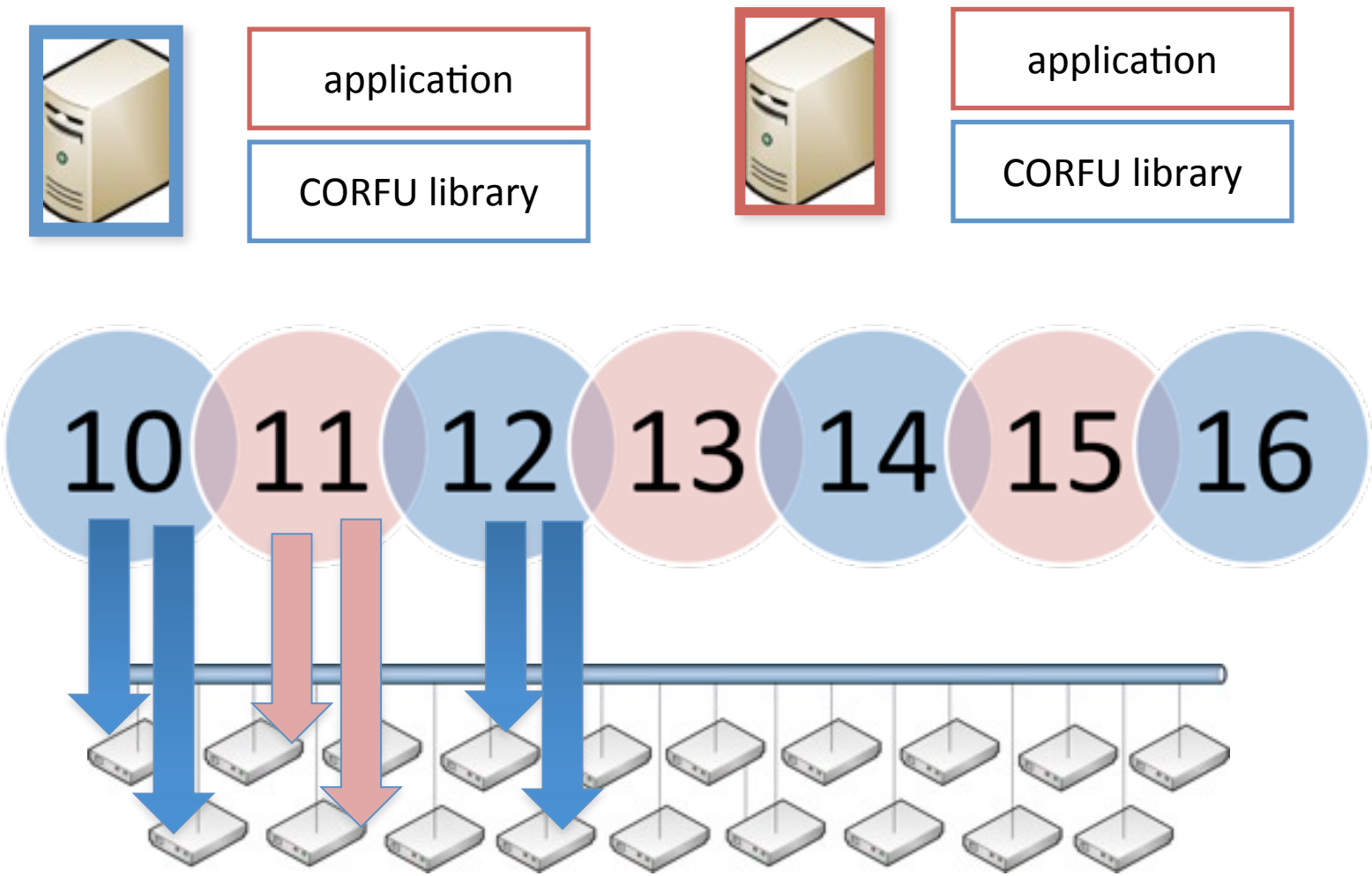
Holes


application
CORFU library
application
CORFU library
10
11
12
13
14
15
16

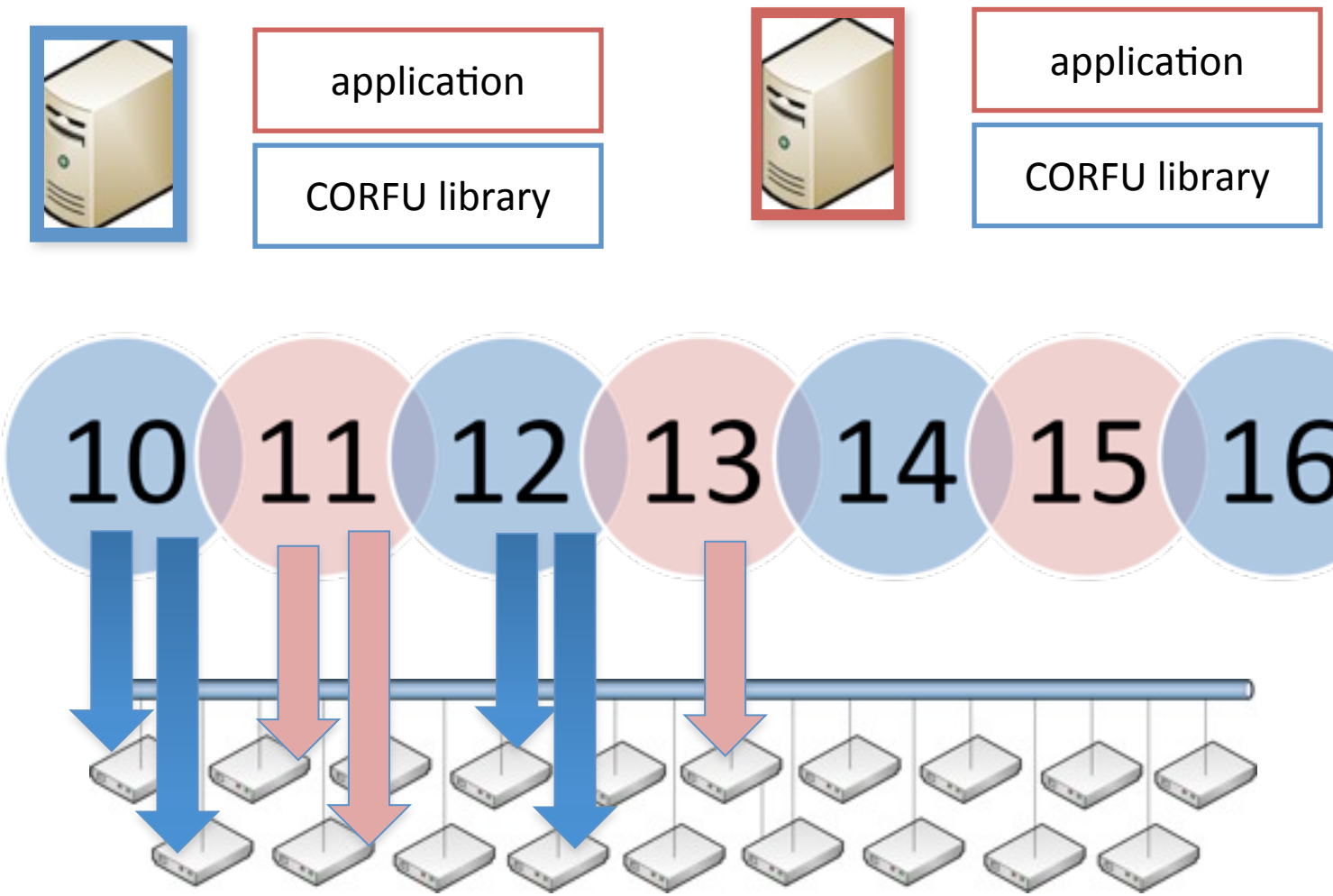
Holes


application
CORFU library
application
CORFU library
10
11
12
13
14
15
16

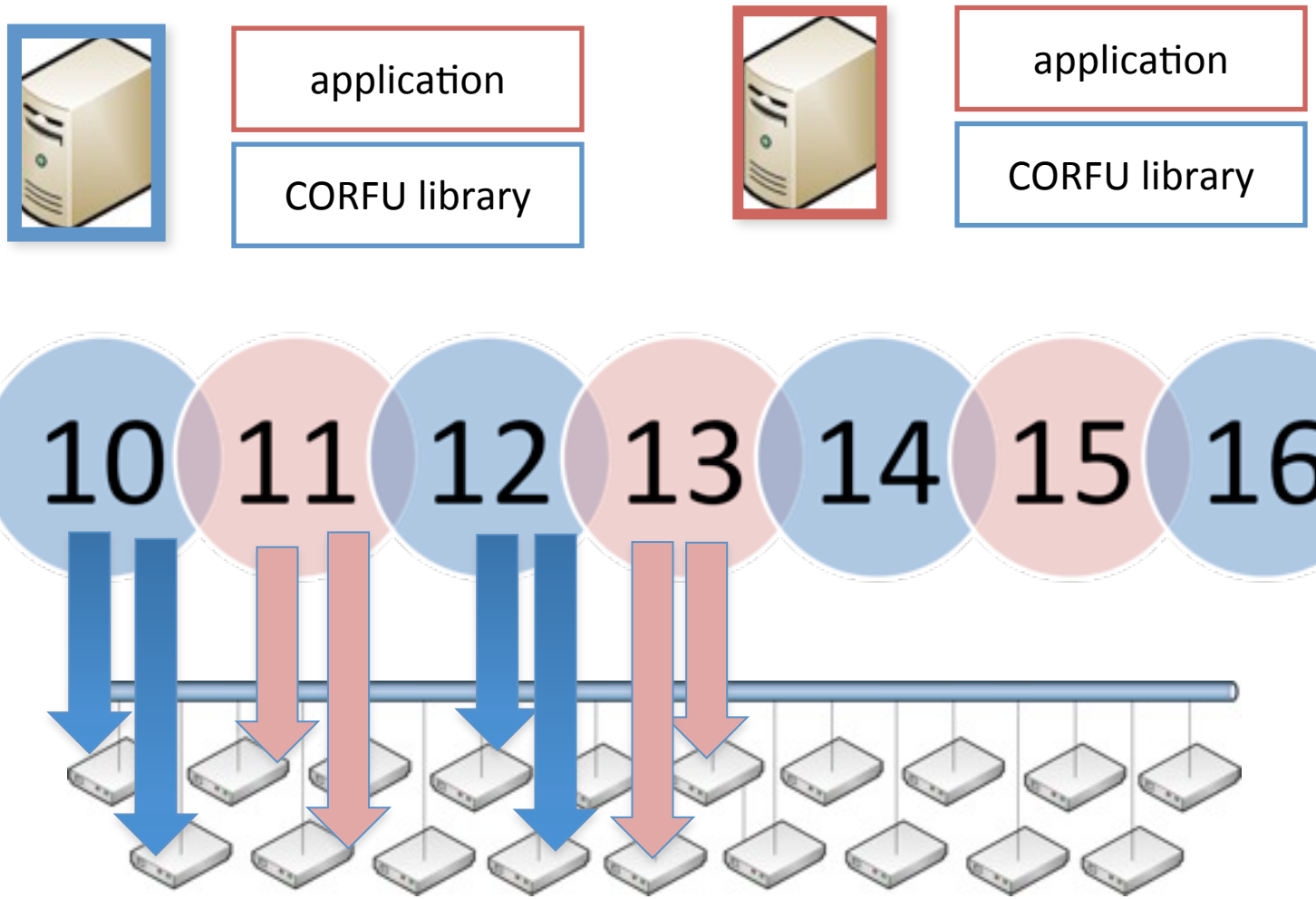
Holes


application
CORFU library
application
CORFU library
10
11
12
13
14
15
16

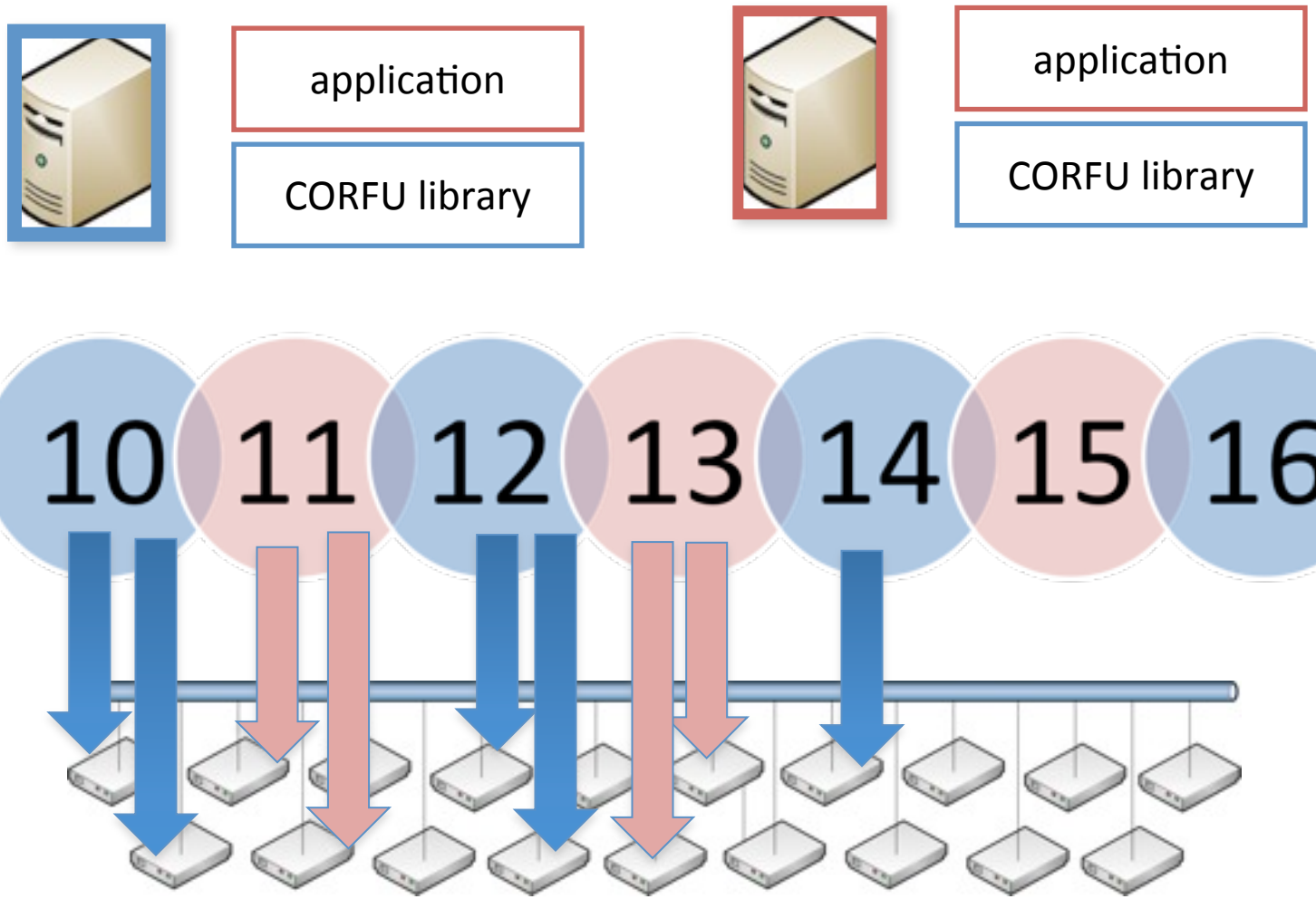
Holes


application
CORFU library
application
CORFU library
10
11
12
13
14
15
16

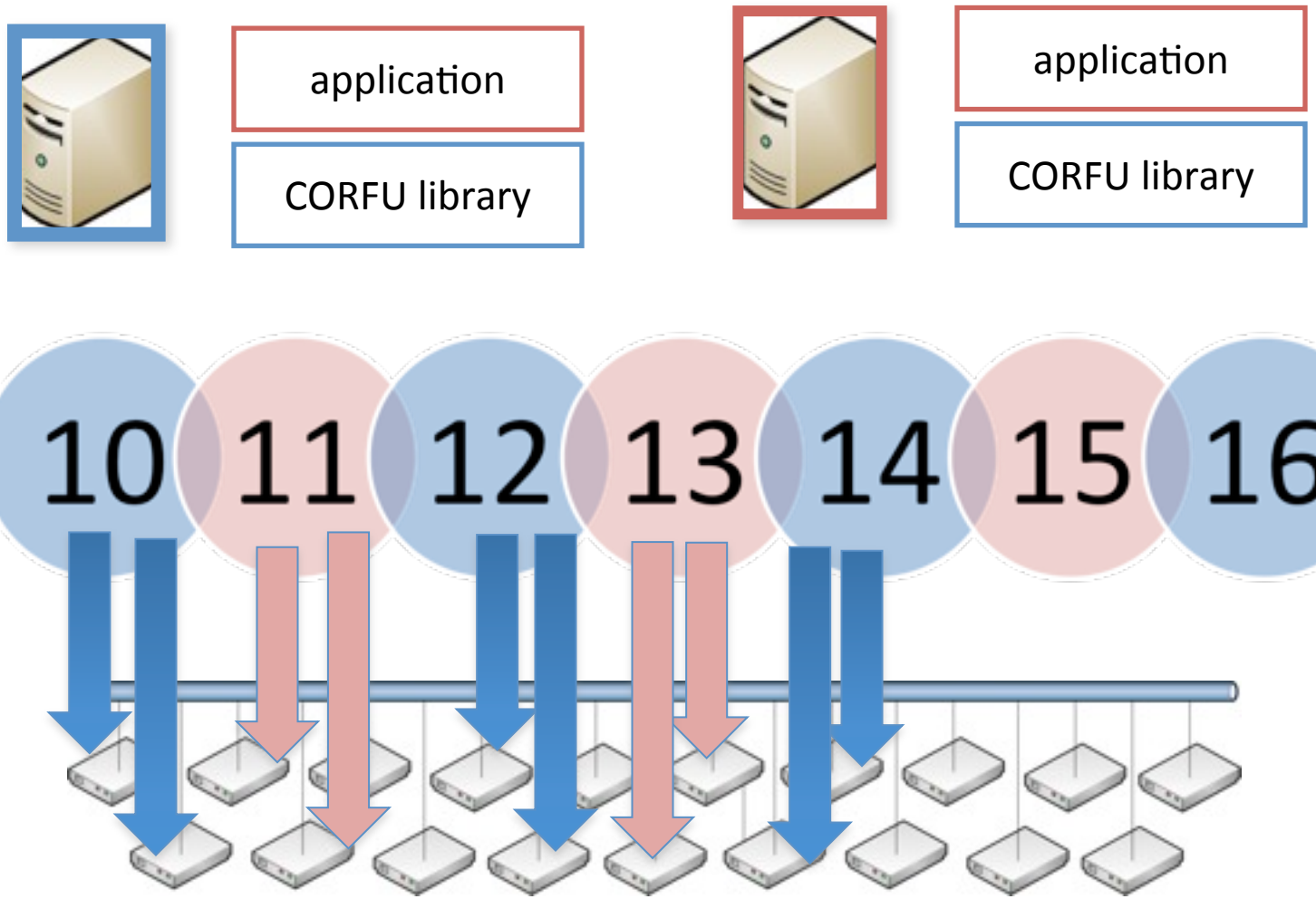
Holes


application
CORFU library
application
CORFU library
10
11
12
13
14
15
16

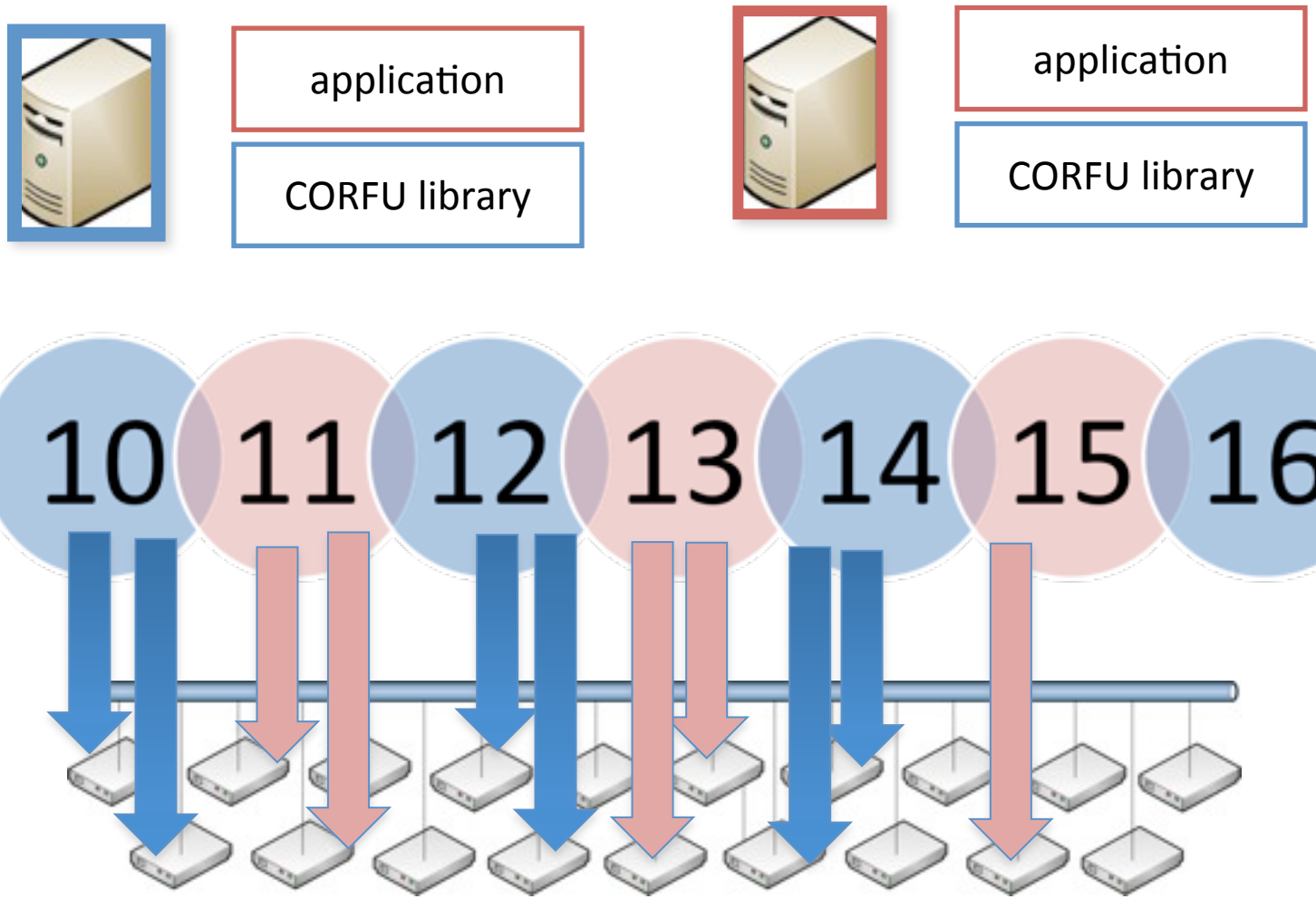
Holes


application
CORFU library
application
CORFU library
10
11
12
13
14
15
16

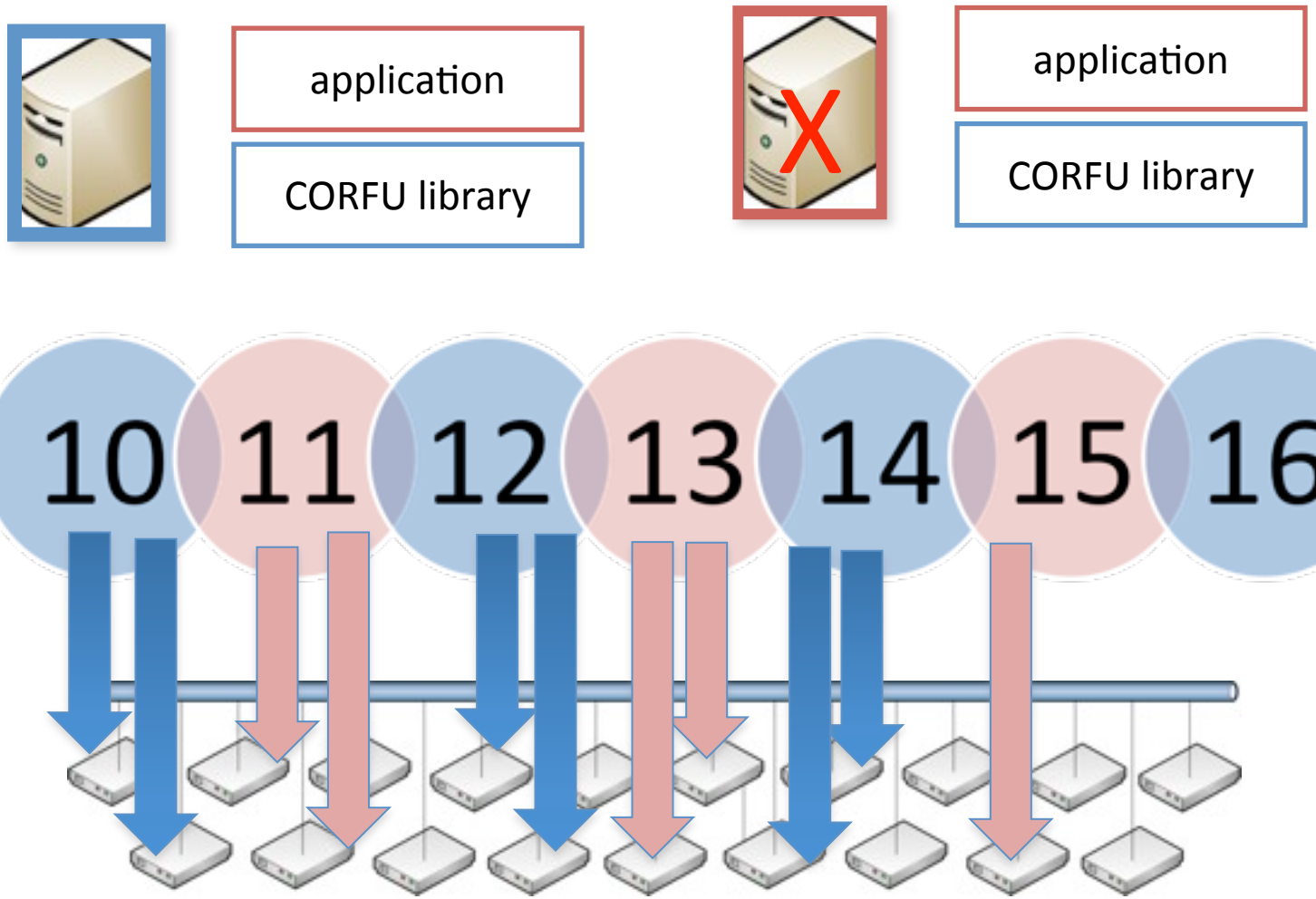
Holes


application
CORFU library
application
CORFU library
10
11
12
13
14
15
16

Holes


application
CORFU library
application
CORFU library
10
11
12
13
14
15
16

Holes


application
CORFU library
X
application
CORFU library
10
11
12
13
14
15
16

Holes

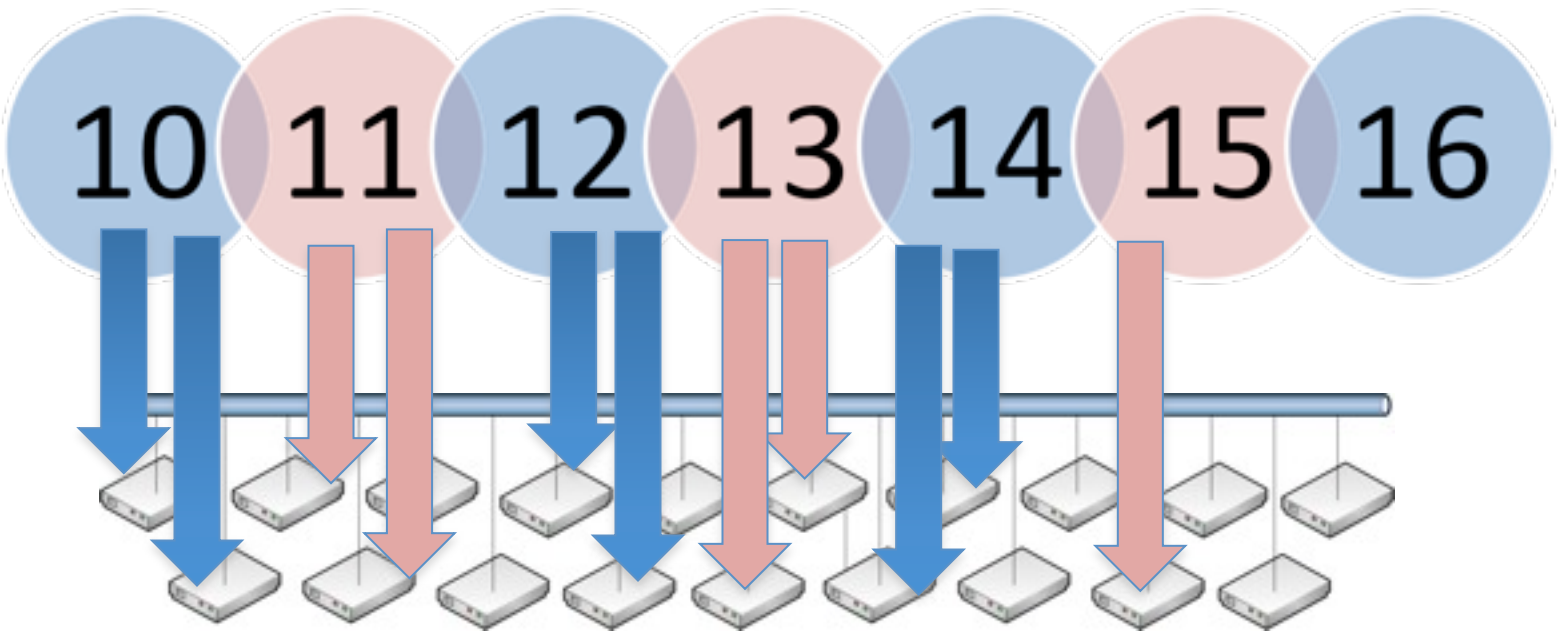
application

CORFU library

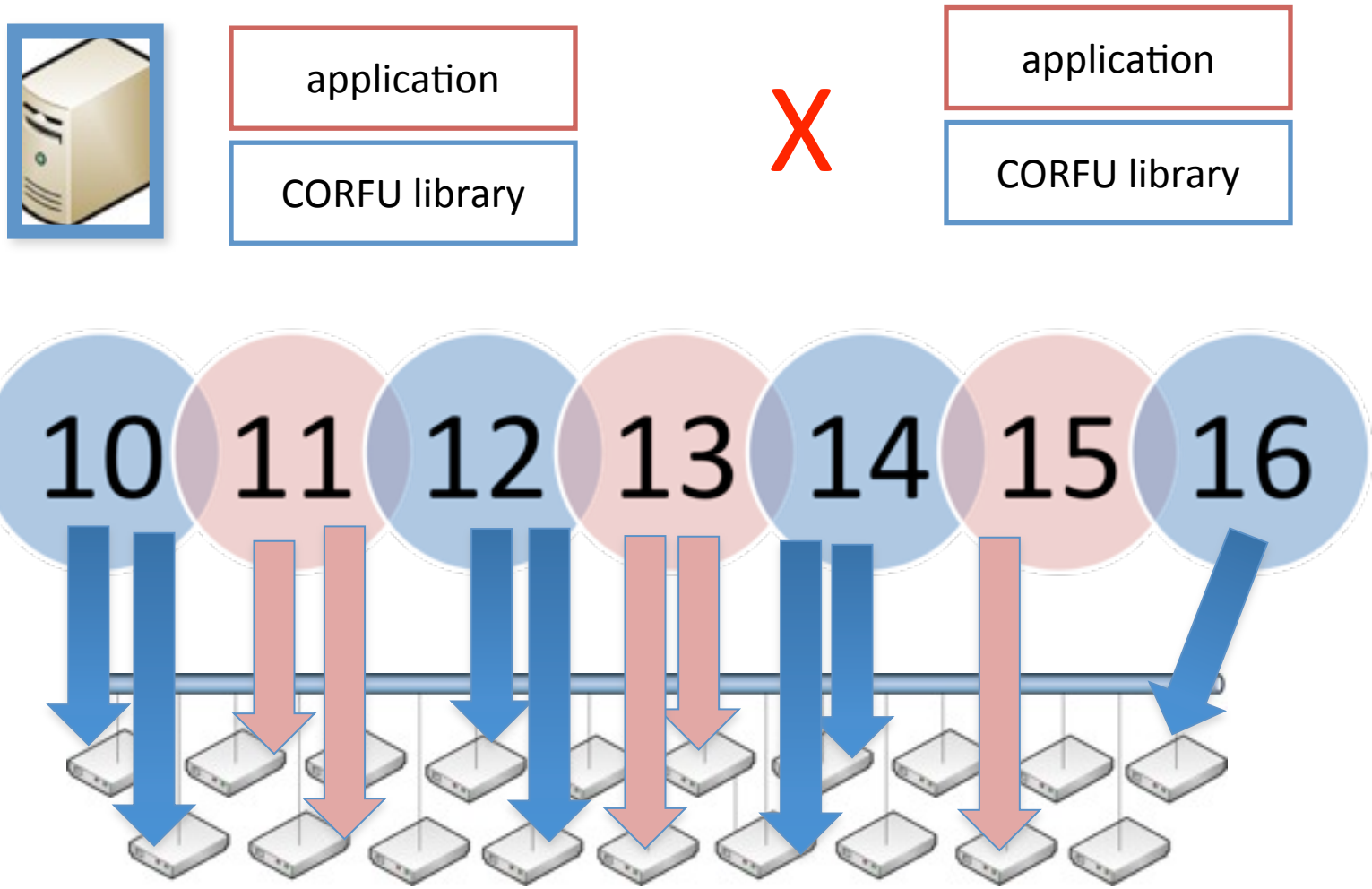
X

application

CORFU library



Holes


application
CORFU library
X
application
CORFU library
10
11
12
13
14
15
16

Holes

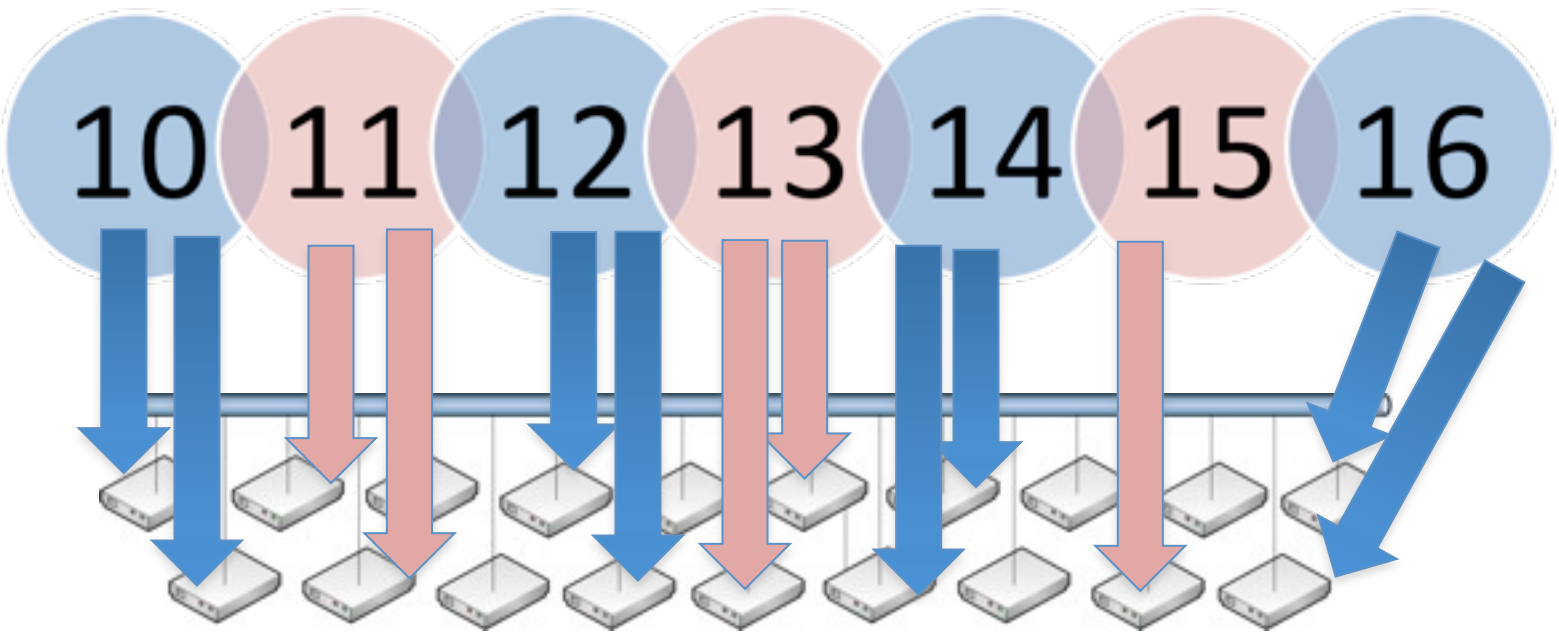
application

CORFU library

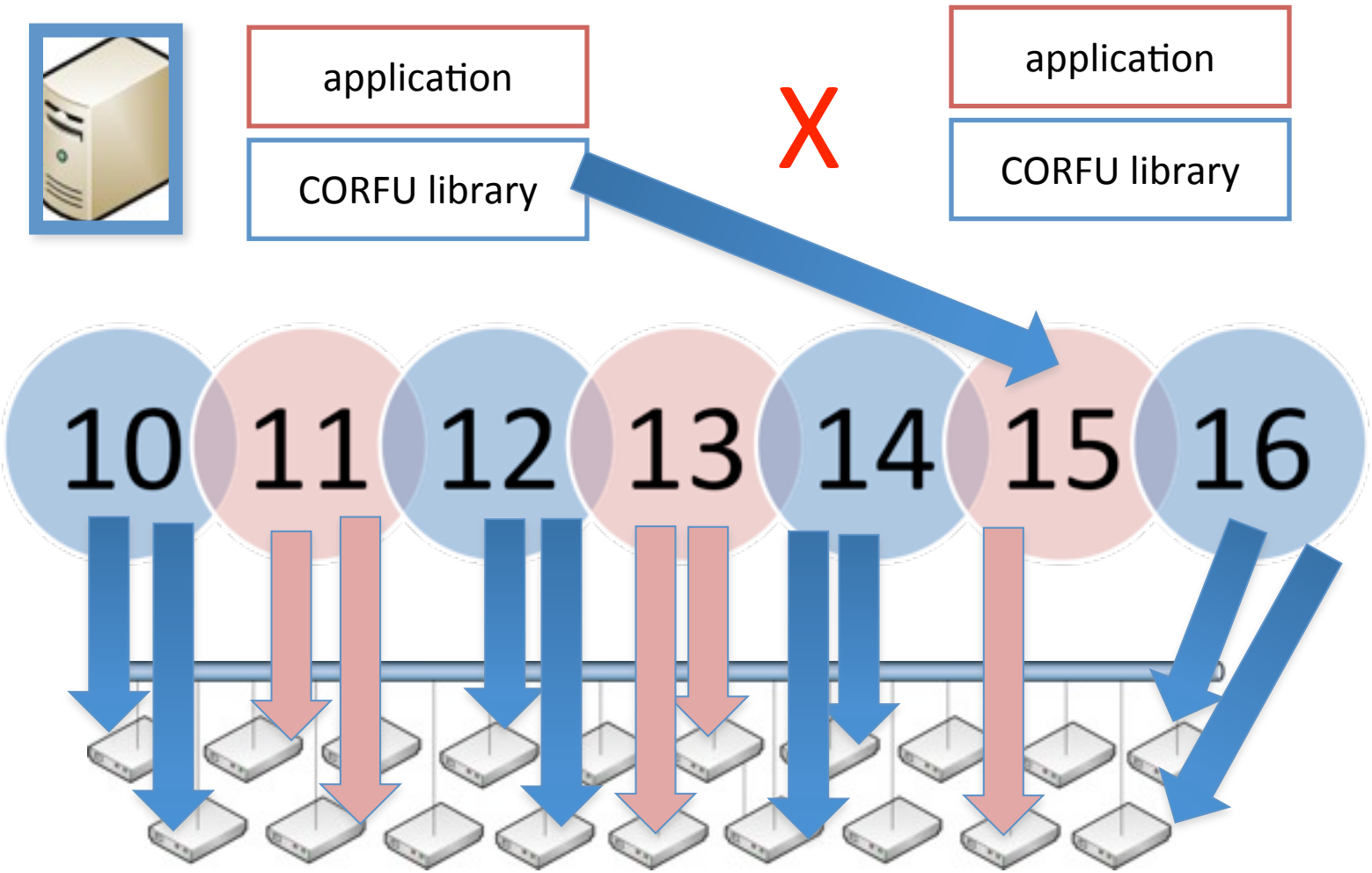
X

application

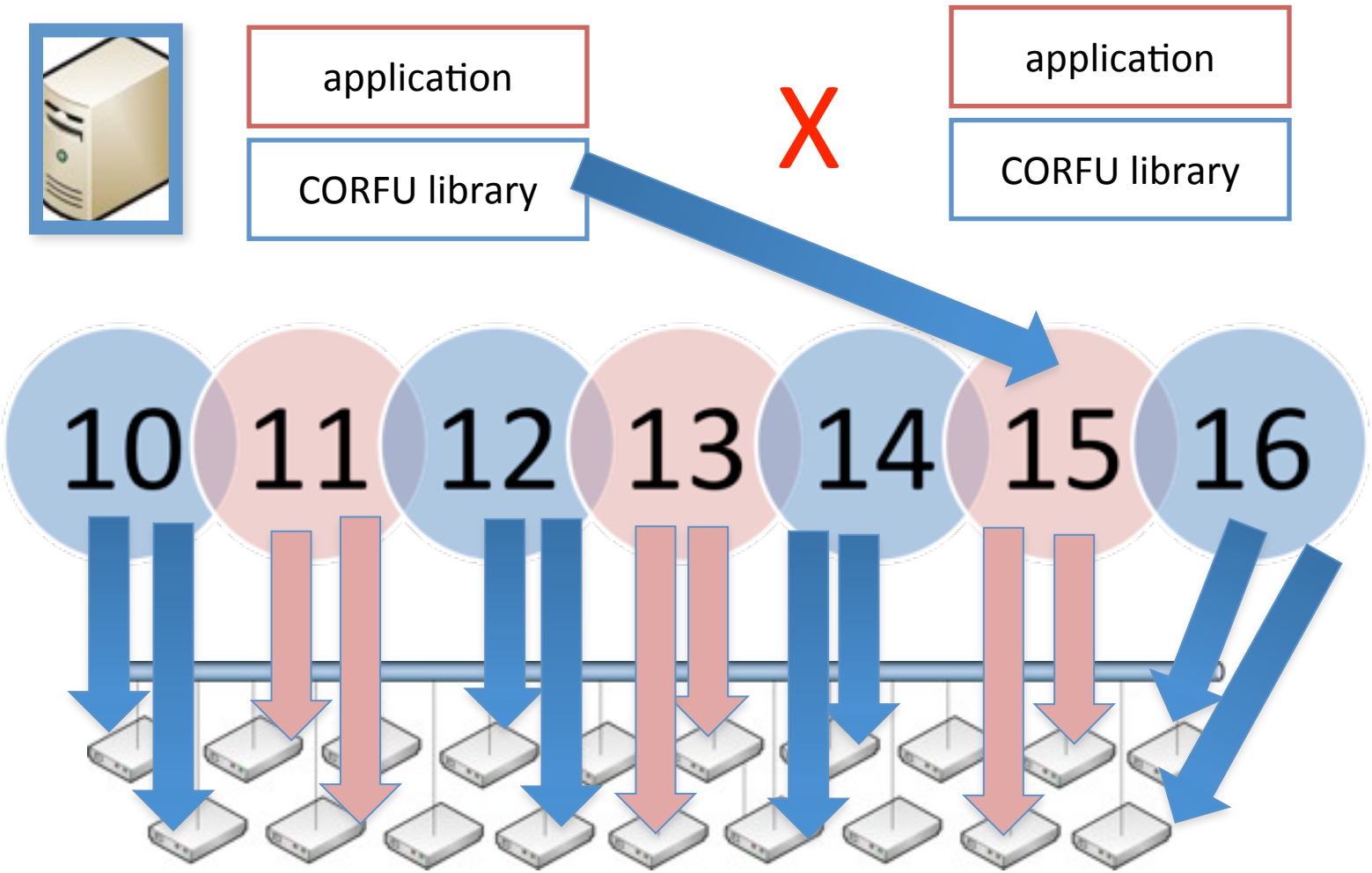
CORFU library



Holes

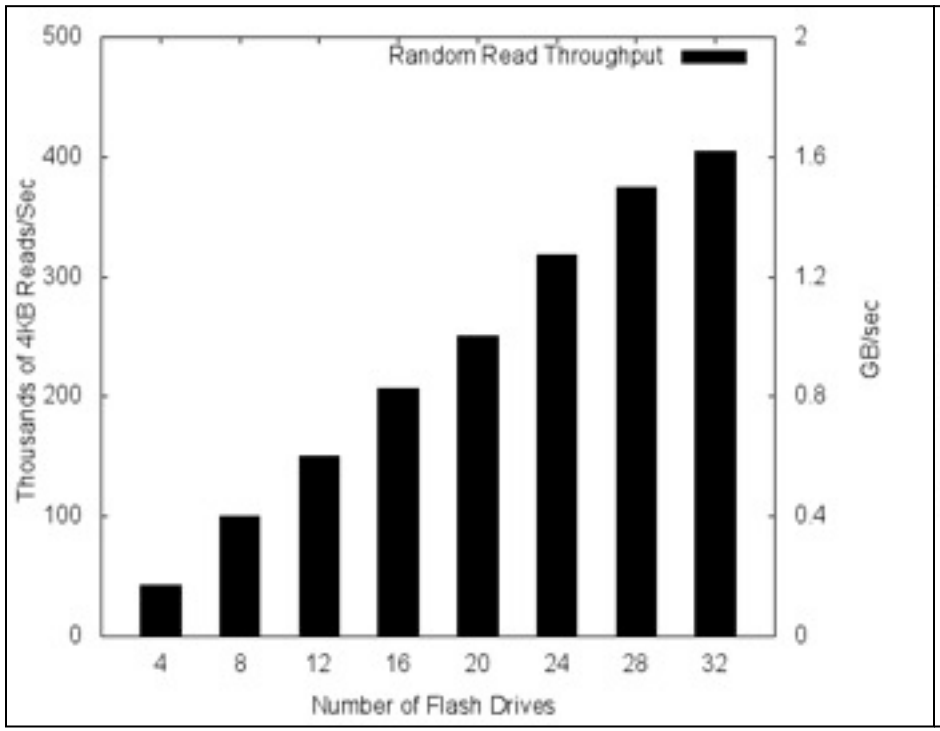

application
CORFU library
X
application
CORFU library
10
11
12
13
14
15
16

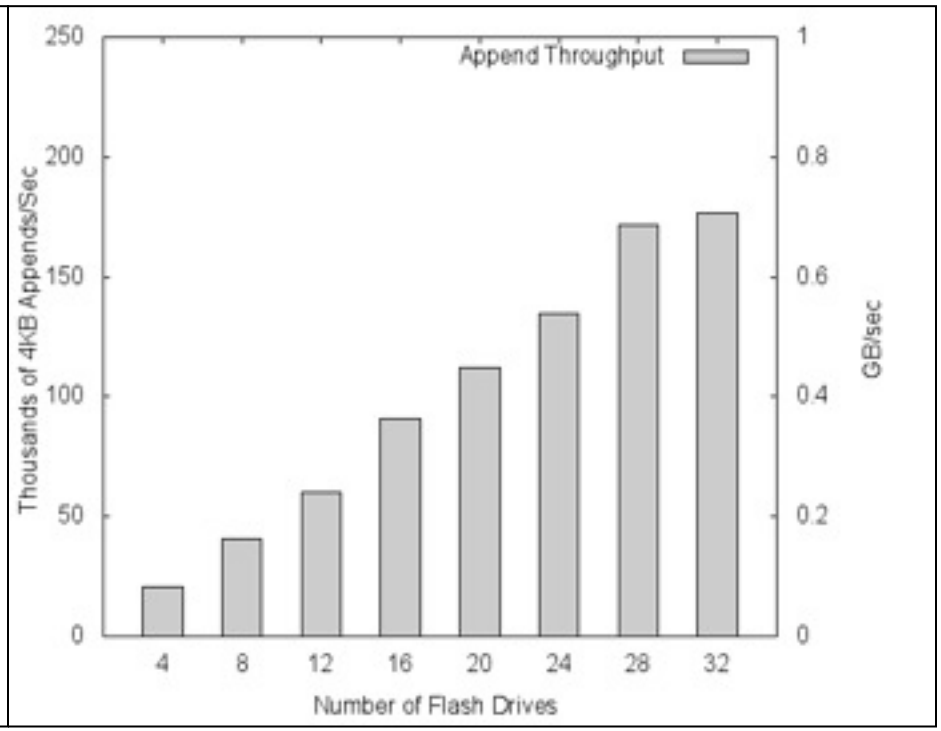
Holes


application
CORFU library
X
application
CORFU library
10
11
12
13
14
15
16

CORFU throughput (server+SSD)

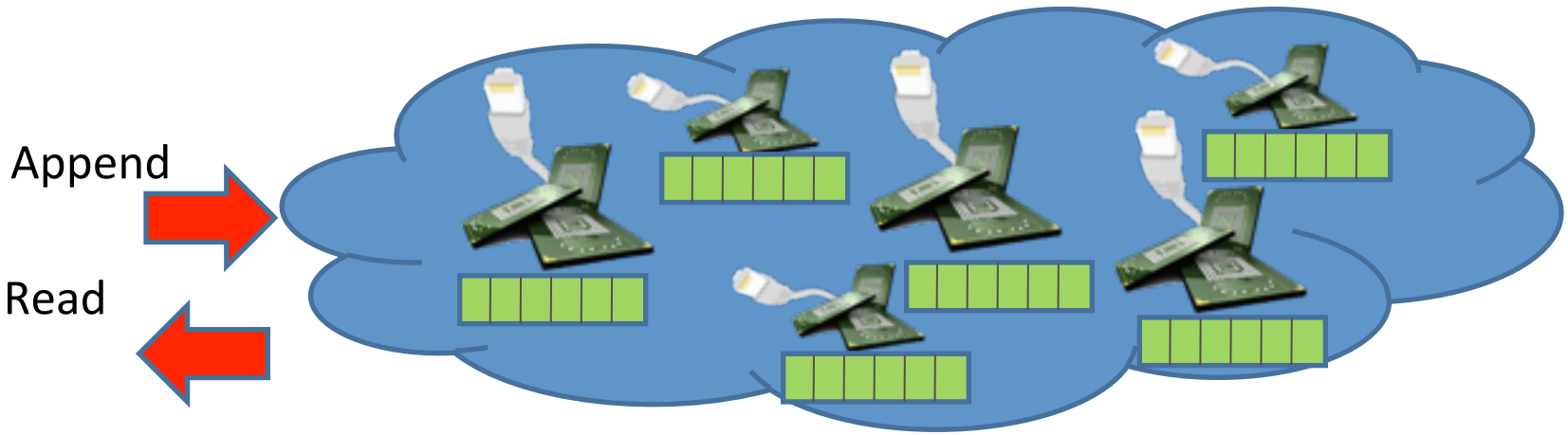
[*CORFU: A Shared Log Design for Flash Clusters*, NSDI 2012]



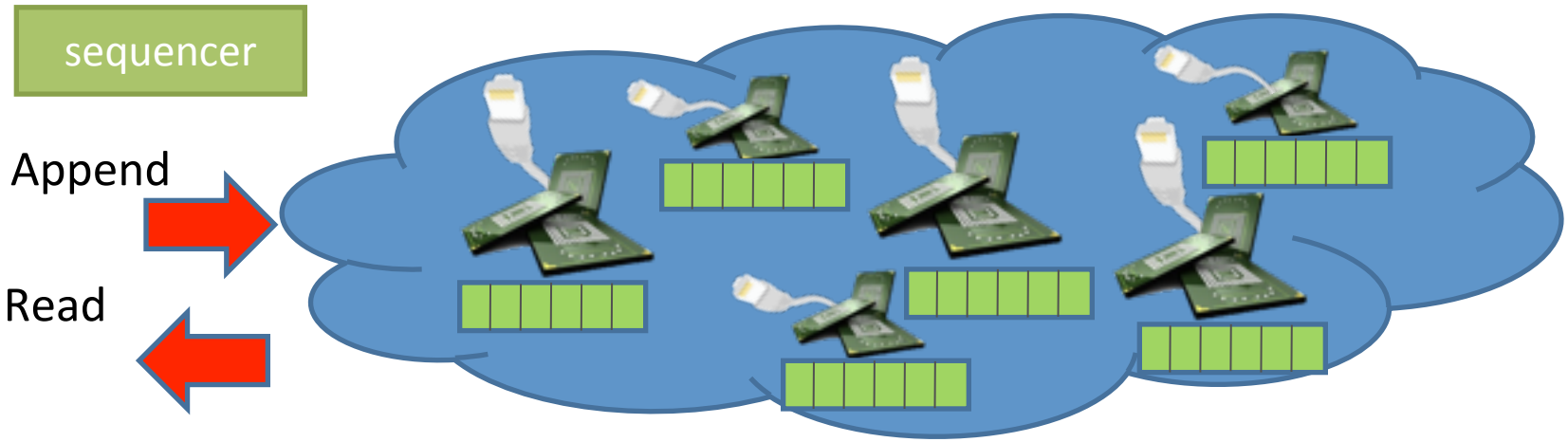


- 10 Gbps router X 16 servers (1 Gbps) X 2 SSDs per server
- Reads bounded by network bandwidth
- Writes are replicated, and bounded by sequencer throughput

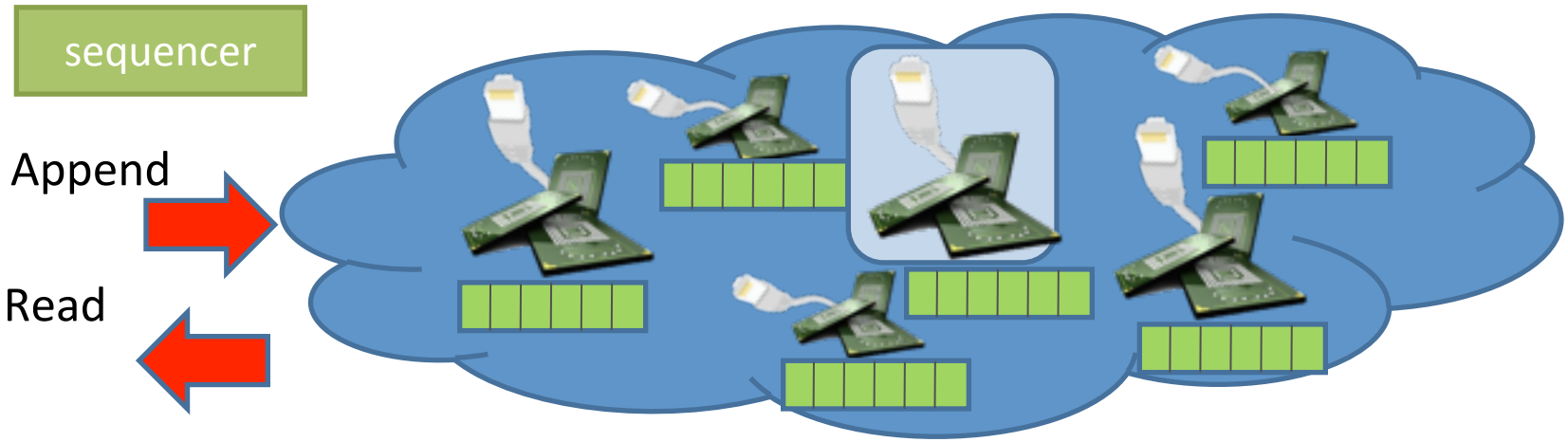
The CORFU Hardware



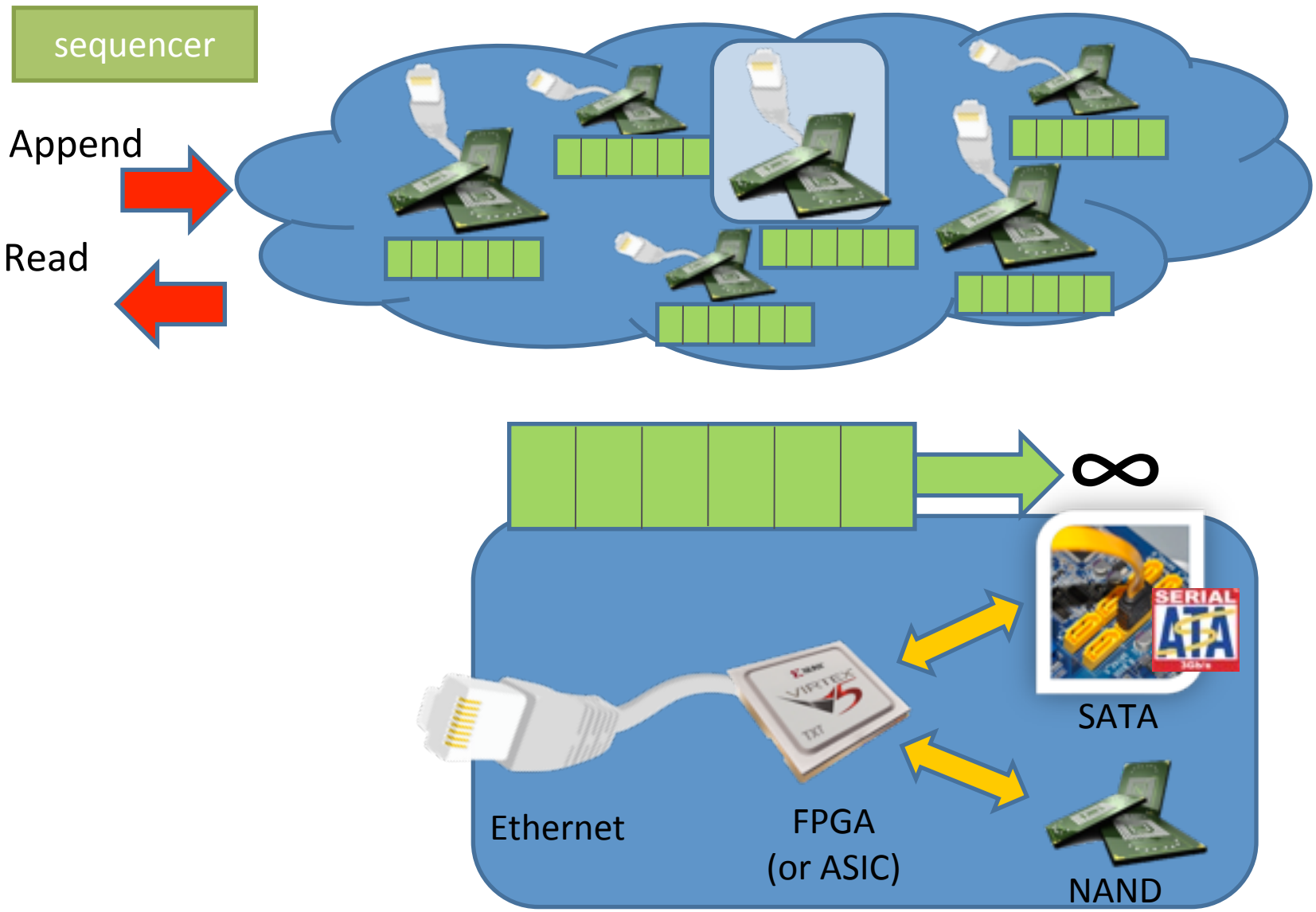
The CORFU Hardware



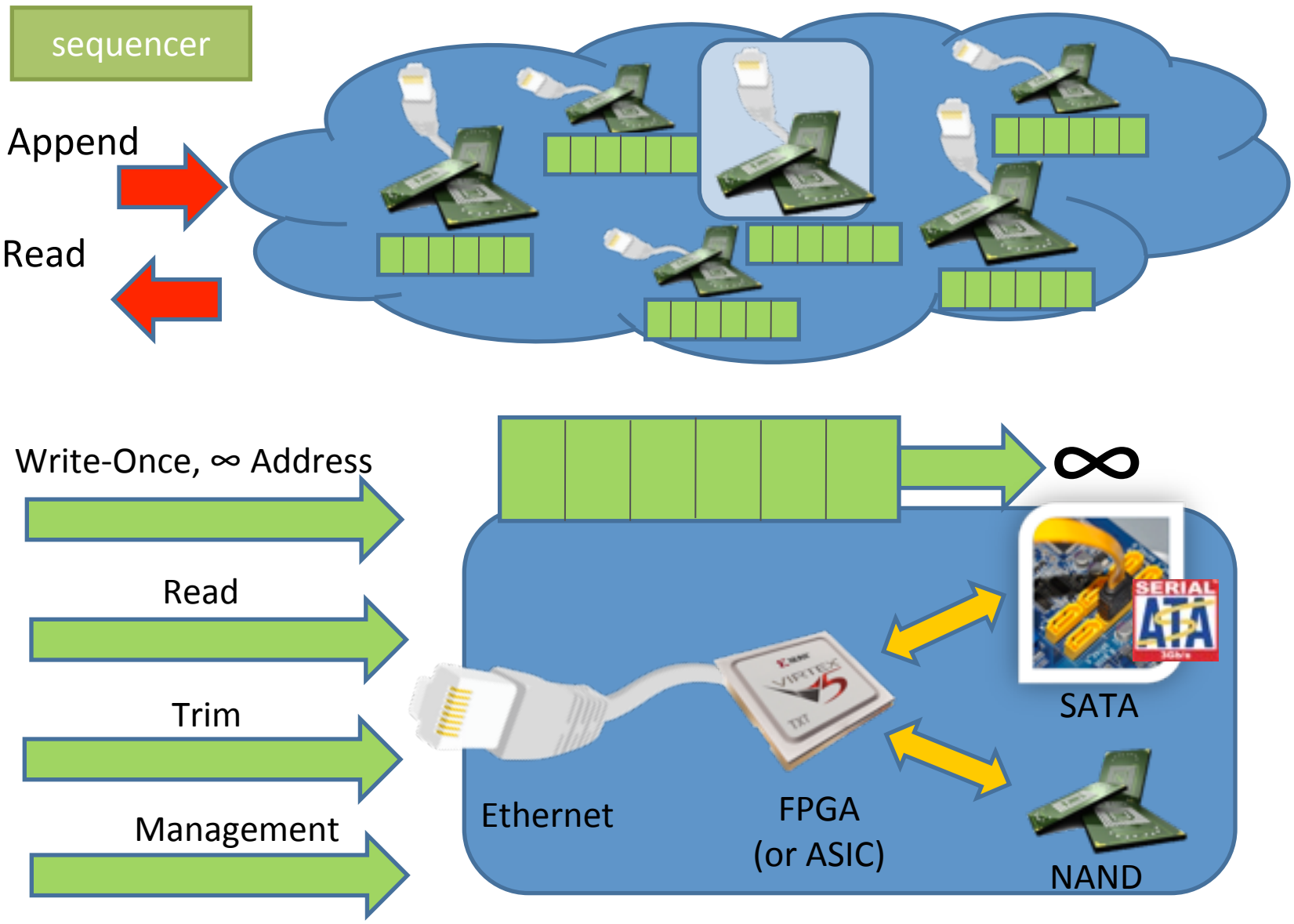
The CORFU Hardware



The CORFU Hardware



The CORFU Hardware



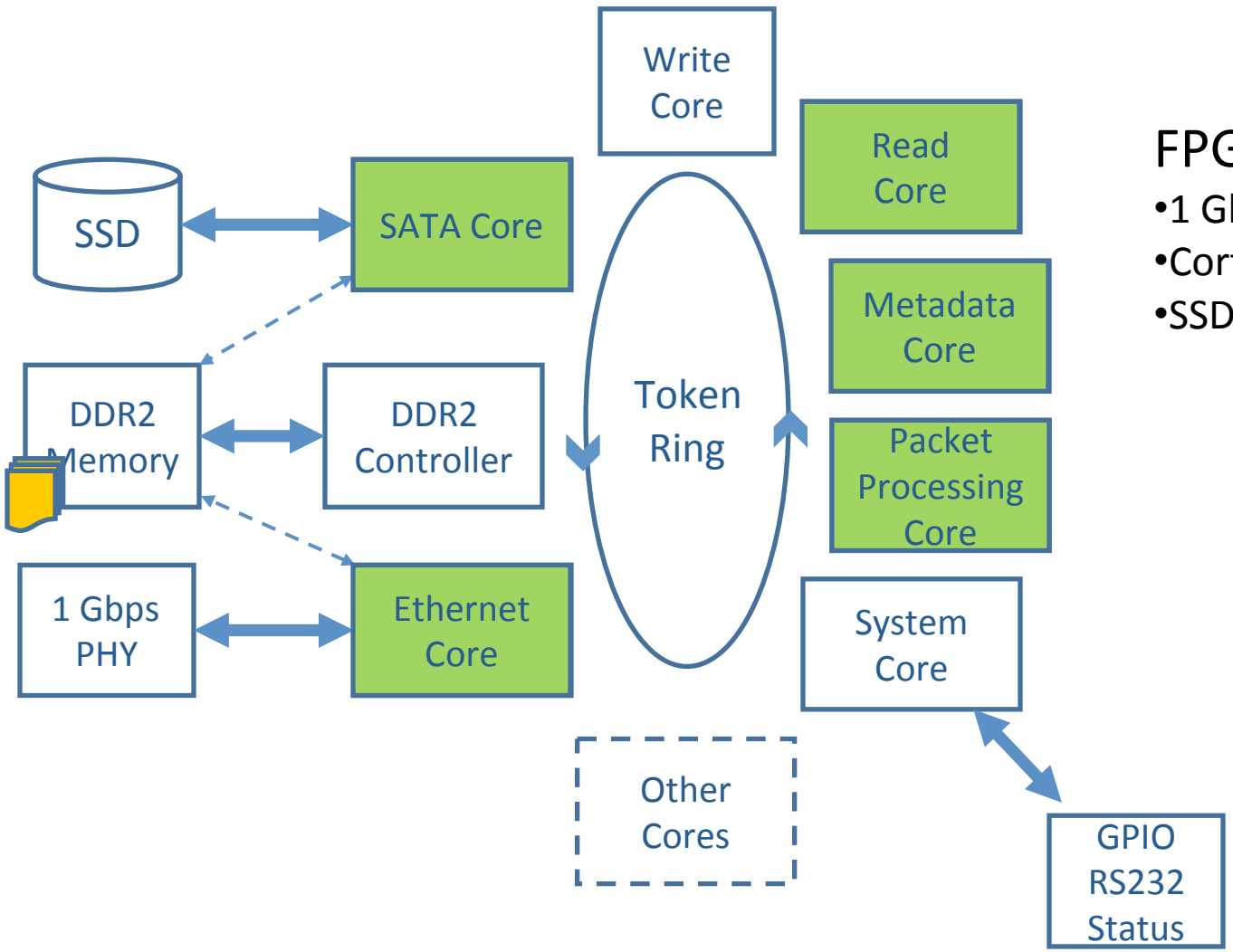
The CORFU Hardware Platform

- 2 Prototype Systems
  - XUPv5
    - Virtex5 XC5VLX110T
    - 2 GB DDR2 RAM
    - 2x SATA 2.0
  - BEE3
    - Virtex5 XC5VLX155T x4
    - 8GB DDR2 RAM
    - 8x SATA 2.0
    - 32/64GB Flash DIMM

The CORFU Hardware Platform

- 2 Prototype Systems
  - XUPv5
    - Virtex5 XC5VLX110T
    - 2 GB DDR2 RAM
    - 2x SATA 2.0
  - BEE3
    - Virtex5 XC5VLX155T x4
    - 8GB DDR2 RAM
    - 8x SATA 2.0
    - 32/64GB Flash DIMM

Hardware Design



FPGA prototype:

- 1 Gbps Ethernet
- Corfu protocol (UDP)
- SSD, not raw flash

Our hardware

Our hardware

- Simple, low power, cheap, naturally pipelined
- No interrupts; no thread synchronization; polling throughout
- No data copying
- Can implement time-sensitive functions in logic
- Frequency: **100 MHz**

Our hardware

- Simple, low power, cheap, naturally pipelined
- No interrupts; no thread synchronization; polling throughout
- No data copying
- Can implement time-sensitive functions in logic
- Frequency: **100 MHz**
- power: **15W**

Our hardware

- Simple, low power, cheap, naturally pipelined
- No interrupts; no thread synchronization; polling throughout
- No data copying
- Can implement time-sensitive functions in logic
- Frequency: **100 MHz**
- power: **15W**
- tput: **~27,000 reads /sec (4KB) ~ 920 Mbps**

Our hardware

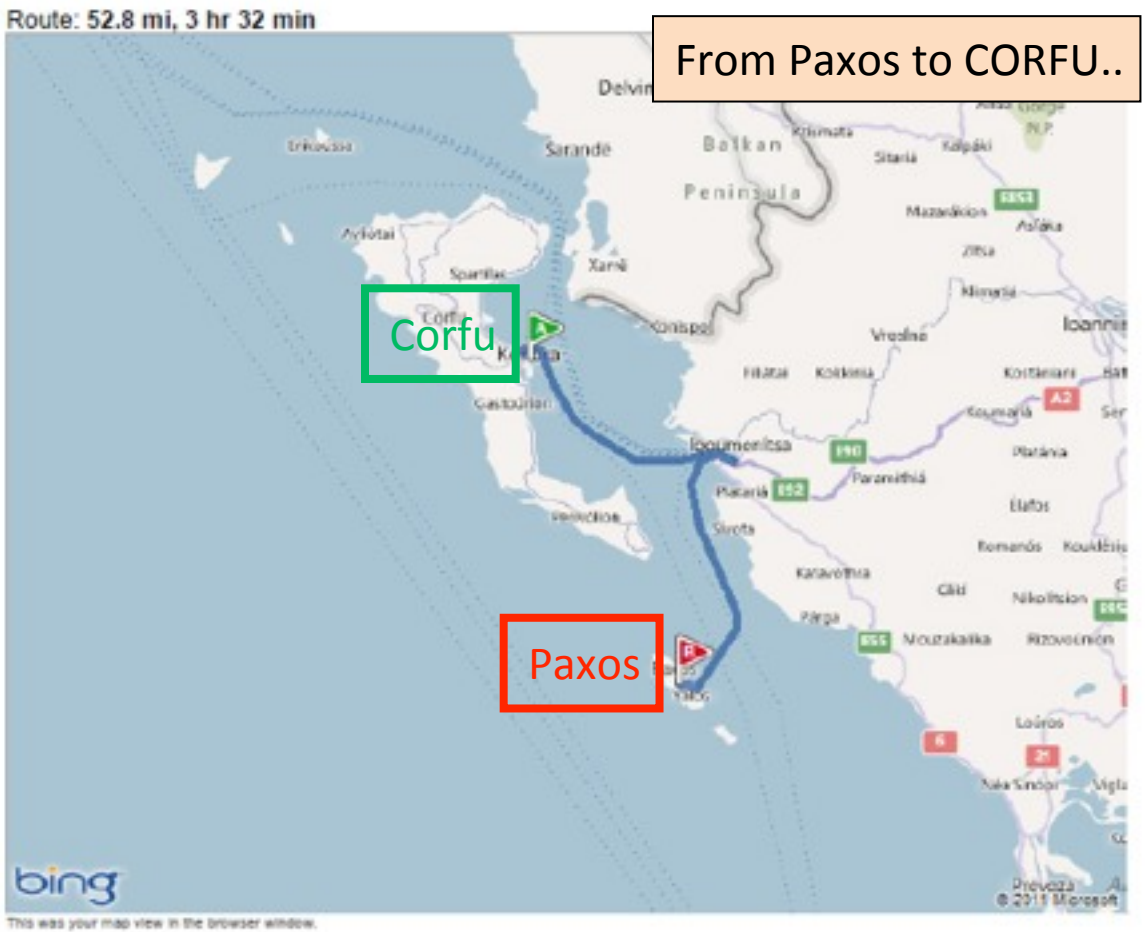
- Simple, low power, cheap, naturally pipelined
- No interrupts; no thread synchronization; polling throughout
- No data copying
- Can implement time-sensitive functions in logic
- Frequency: **100 MHz**
- power: **15W**
- tput: **~27,000 reads /sec (4KB) ~ 920 Mbps**
- end-to-end latency (measured at NIC):
  - reads: **75 µsecs**
  - non-mirrored appends: **200 µsecs**

Applications

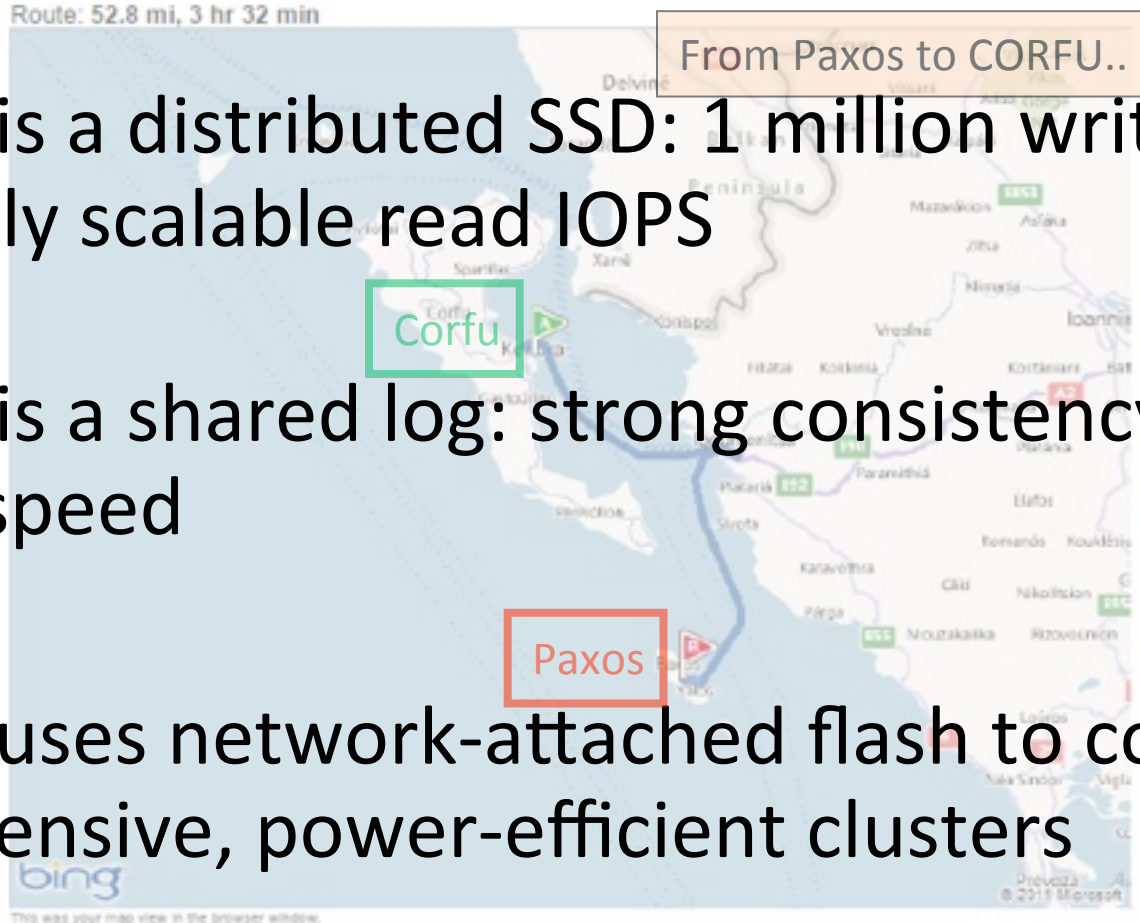
- Key-Value Store
- Virtual Block Device
  - Shared
  - Full rollback
- Distributed Synchronization (ZooKeeper)

Conclusion

Conclusion



Conclusion



CORFU is a distributed SSD: 1 million write IOPS, linearly scalable read IOPS

CORFU is a shared log: strong consistency at wire speed

CORFU uses network-attached flash to construct inexpensive, power-efficient clusters