WineFS: A hugepage-aware file system for persistent memory that ages gracefully

Rohan Kadekodi, Saurabh Kadekodi, Soujanya Ponnapalli, Harshad Shirwadkar, Gregory R. Ganger, Aasheesh Kolli, Vijay Chidambaram

TEXAS The University of Texas at Austin

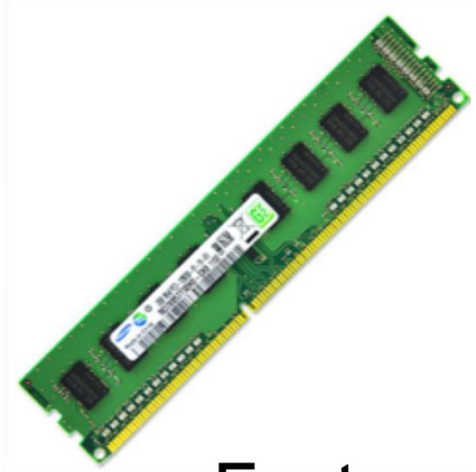
CARNEGIE MELLON UNIVERSITY PITTSBURGH PENNSYLVANIA 1900

Google

vmware®

Persistent Memory

Non-volatile

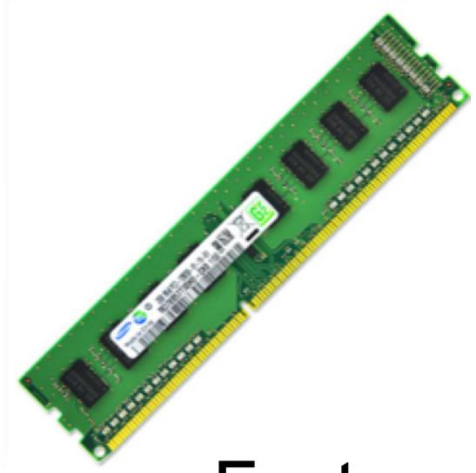


Fast

Persistent Memory

Non-volatile

+



Fast

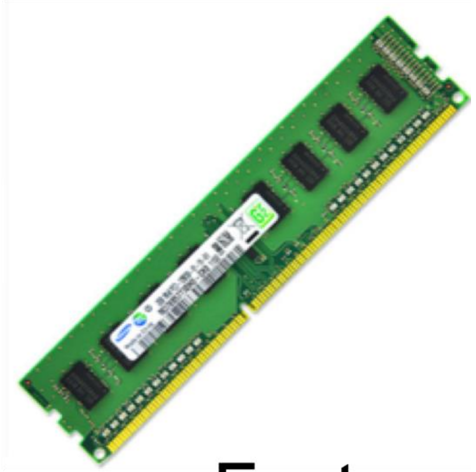
Retain data across power cycles

Persistent Memory

Non-volatile

Retain data across power cycles

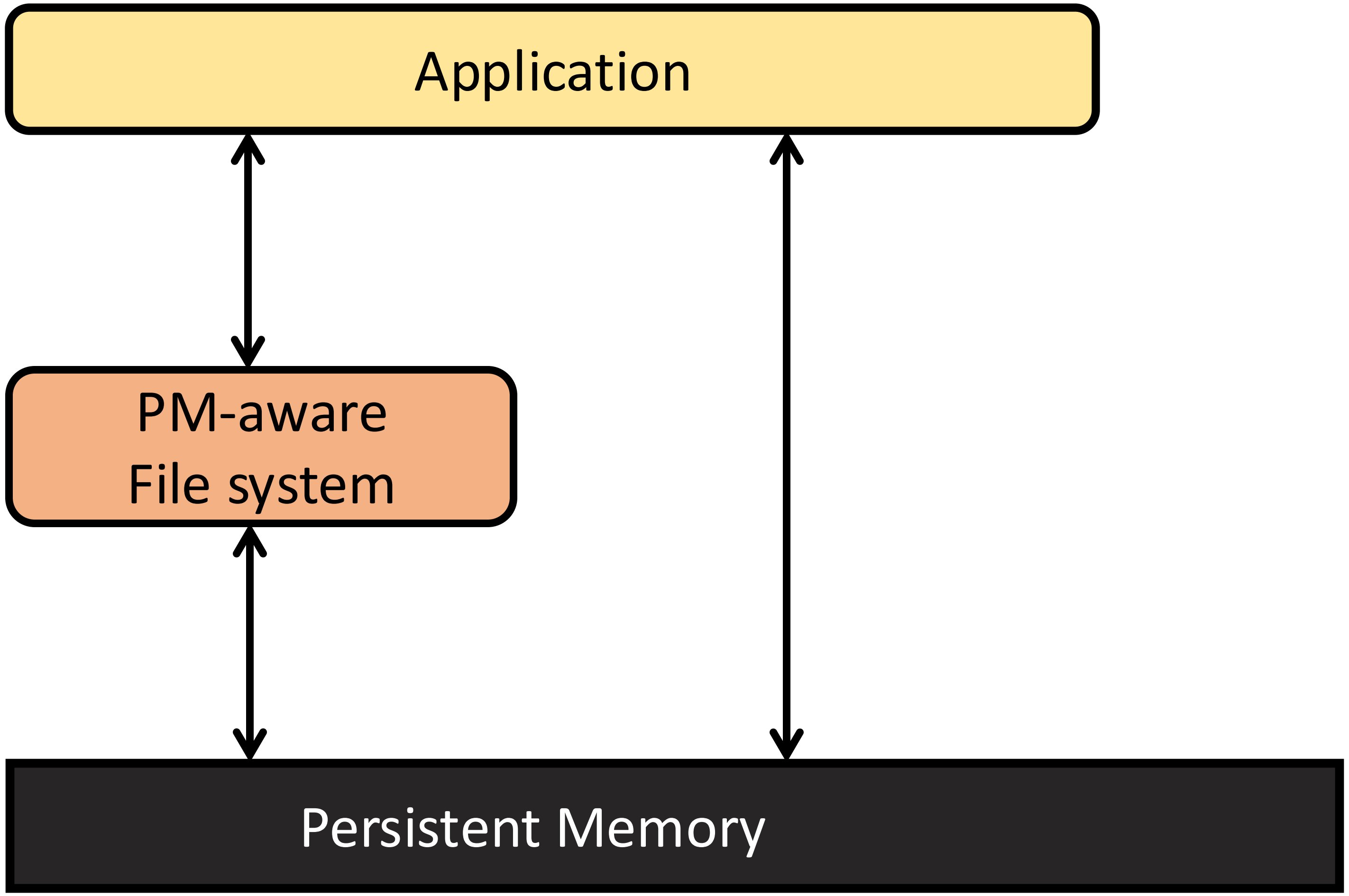
+



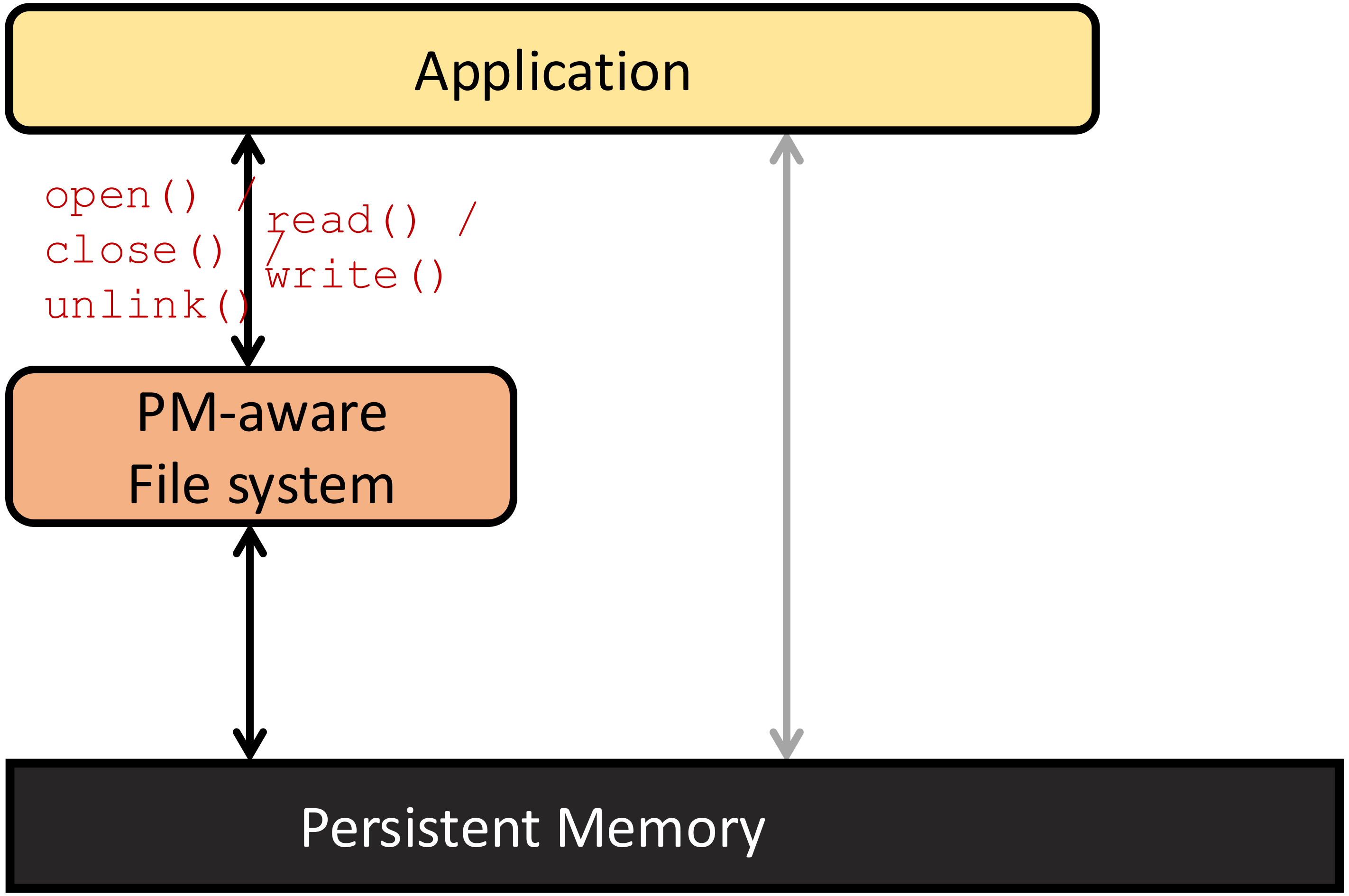
Fast

Access latencies similar to DRAM

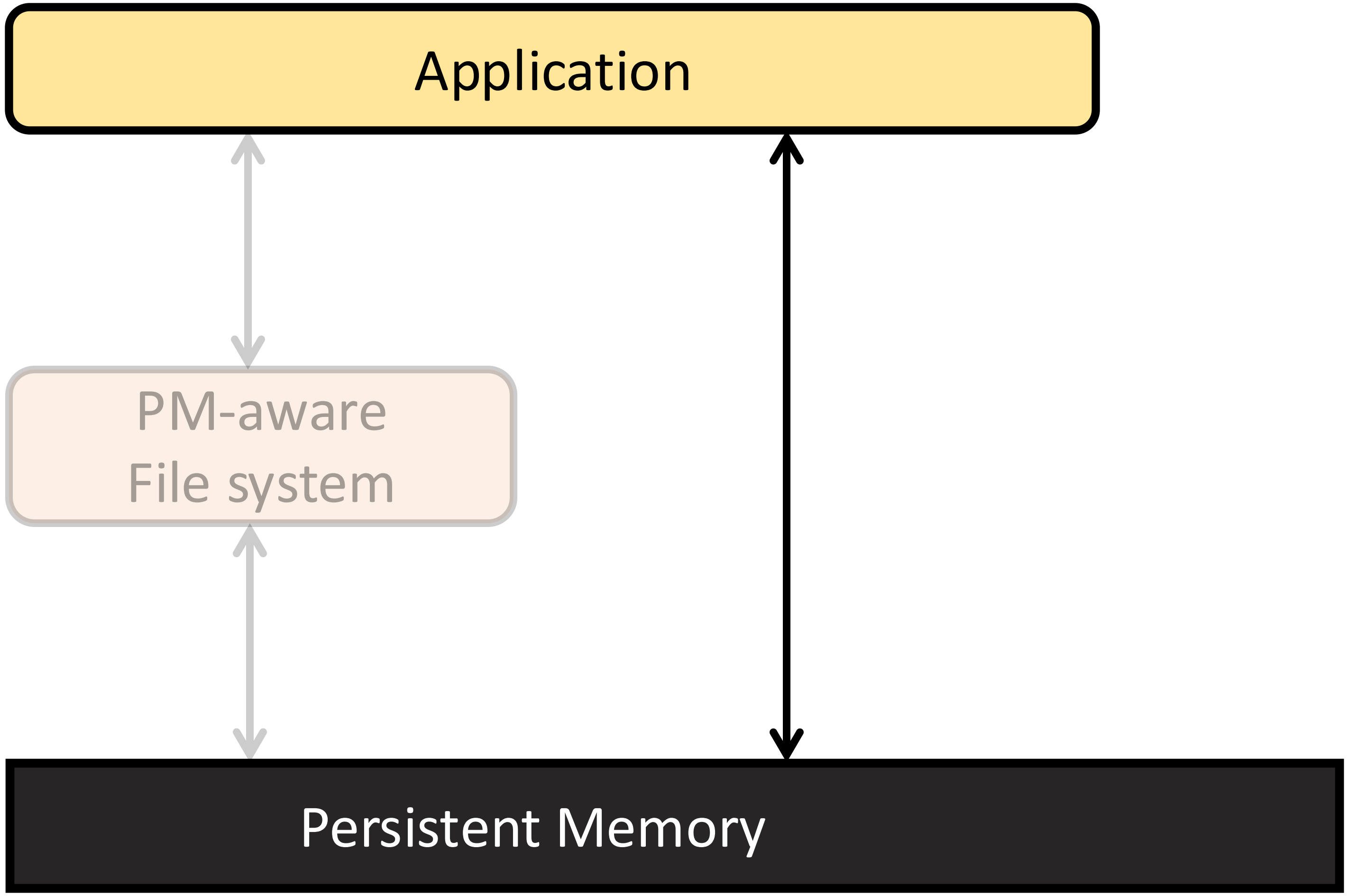
PM Applications



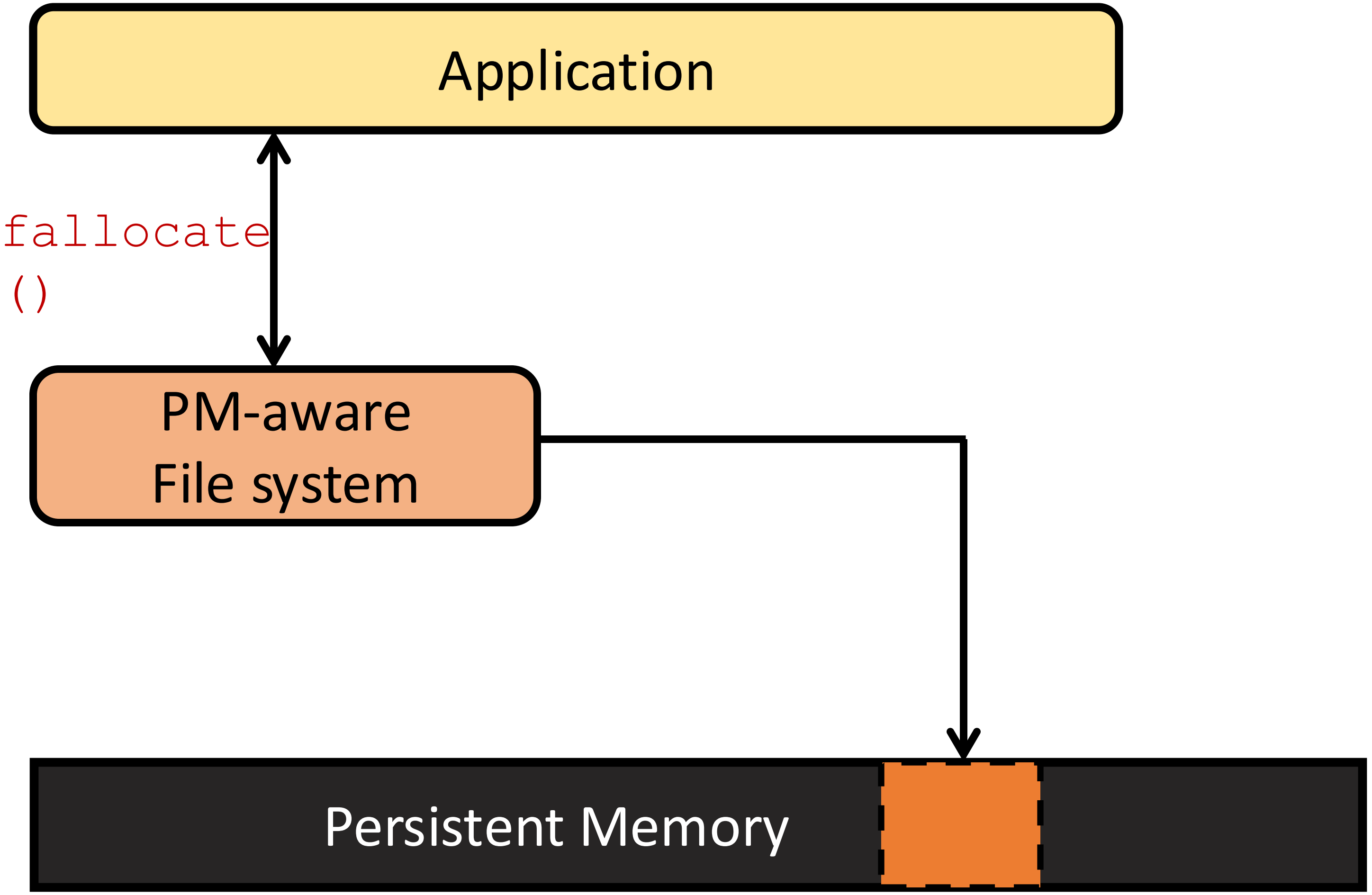
POSIX system-call applications



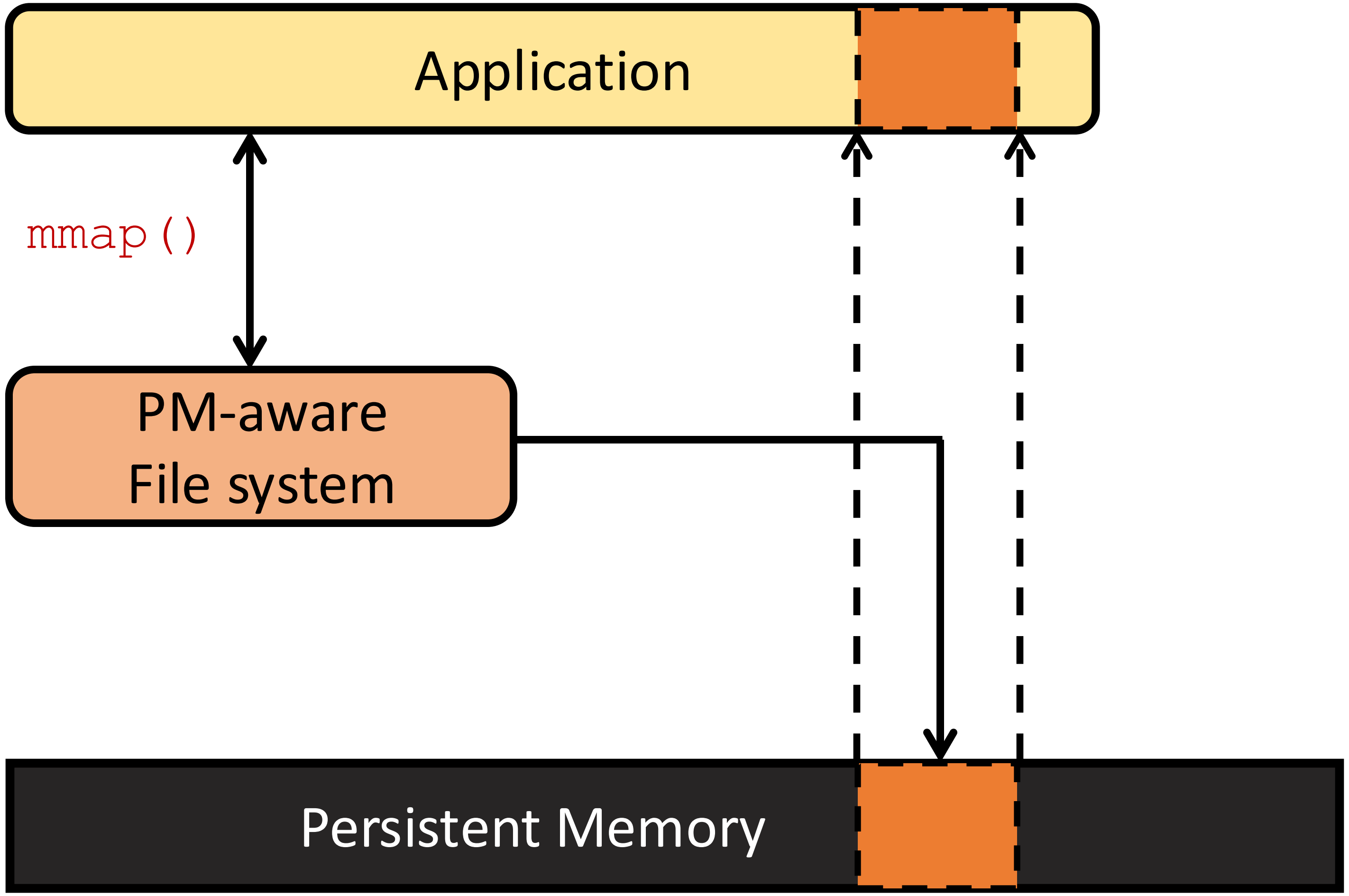
Memory-mapped Applications



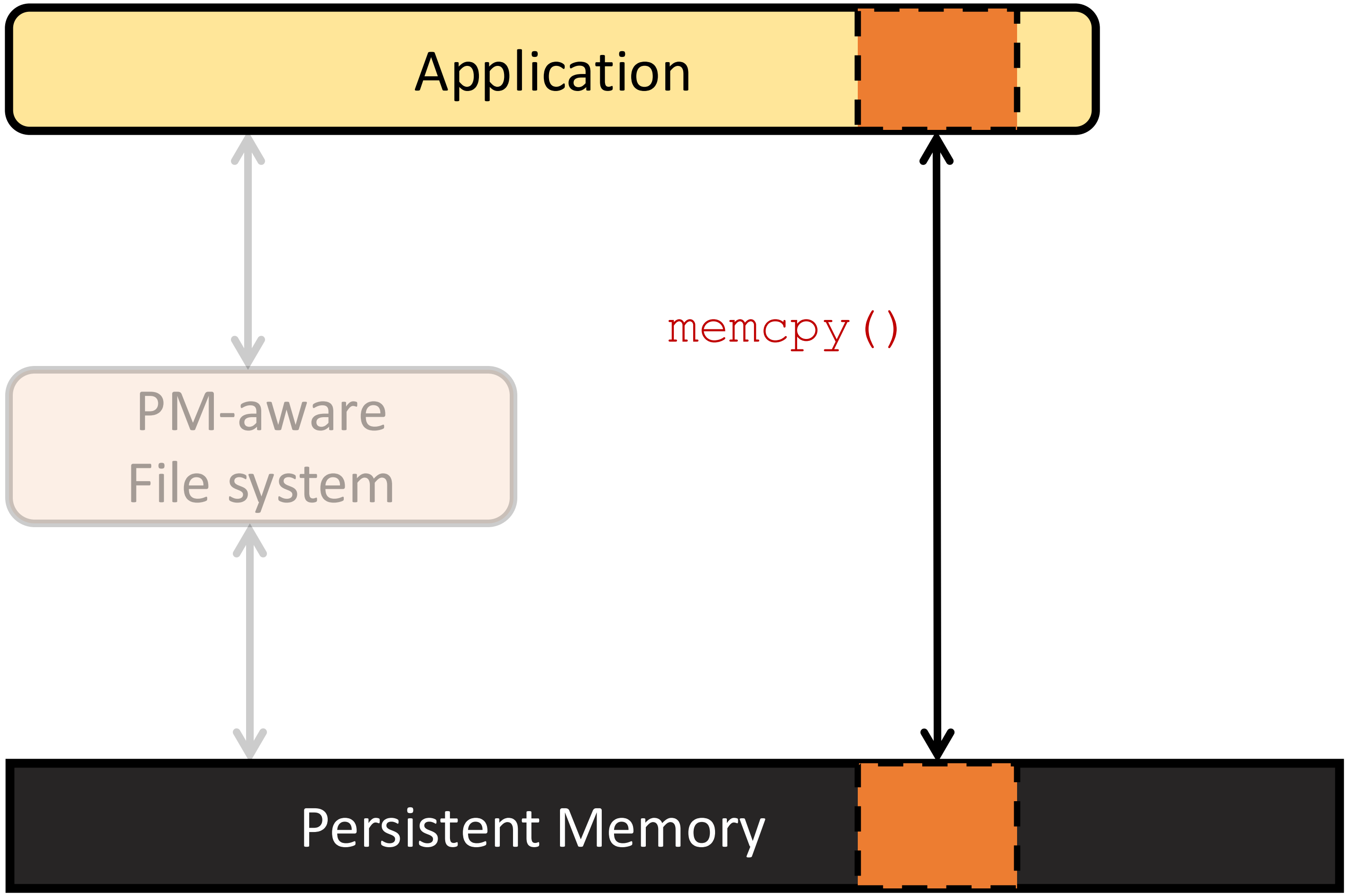
Memory-mapped Applications



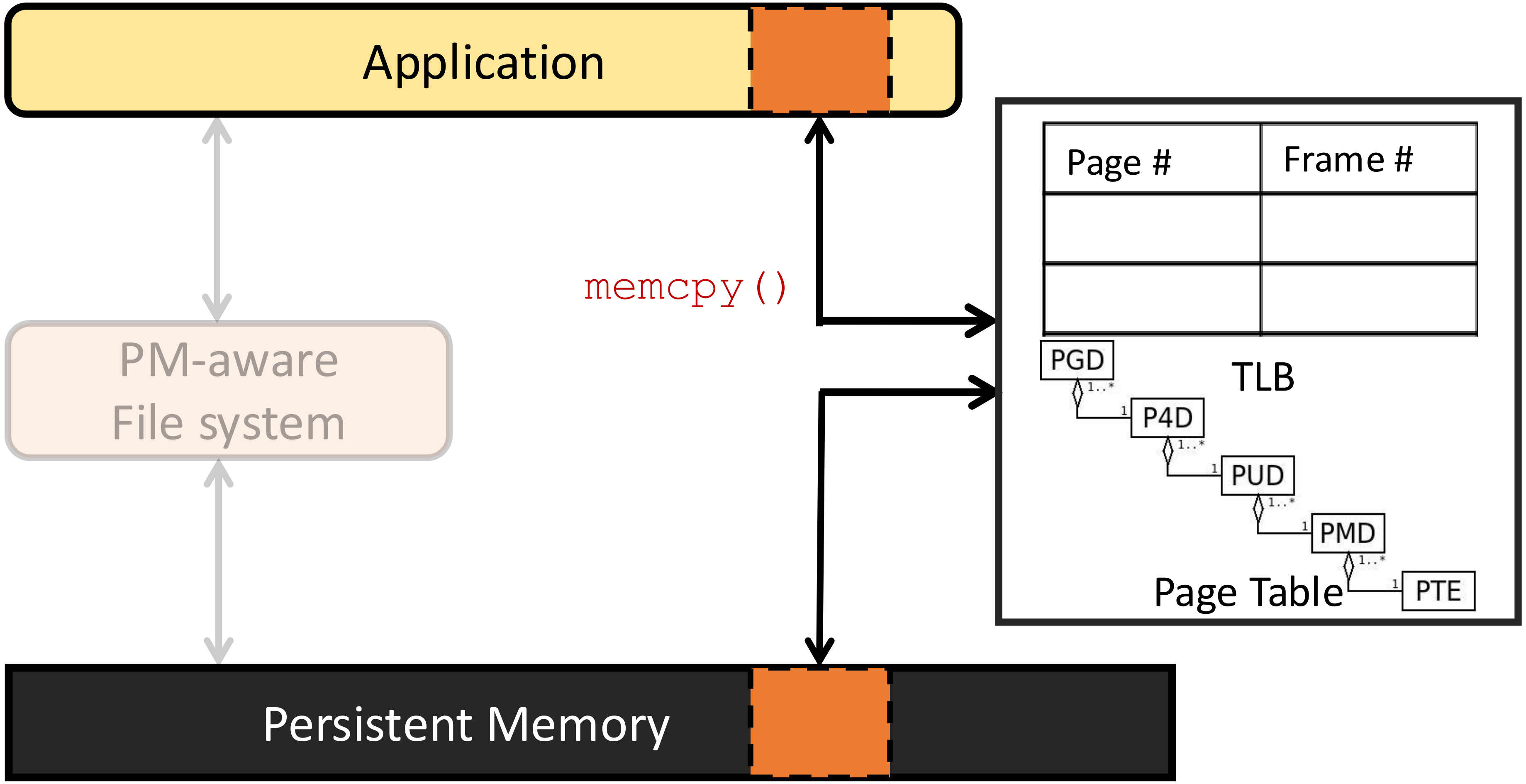
Memory-mapped Applications



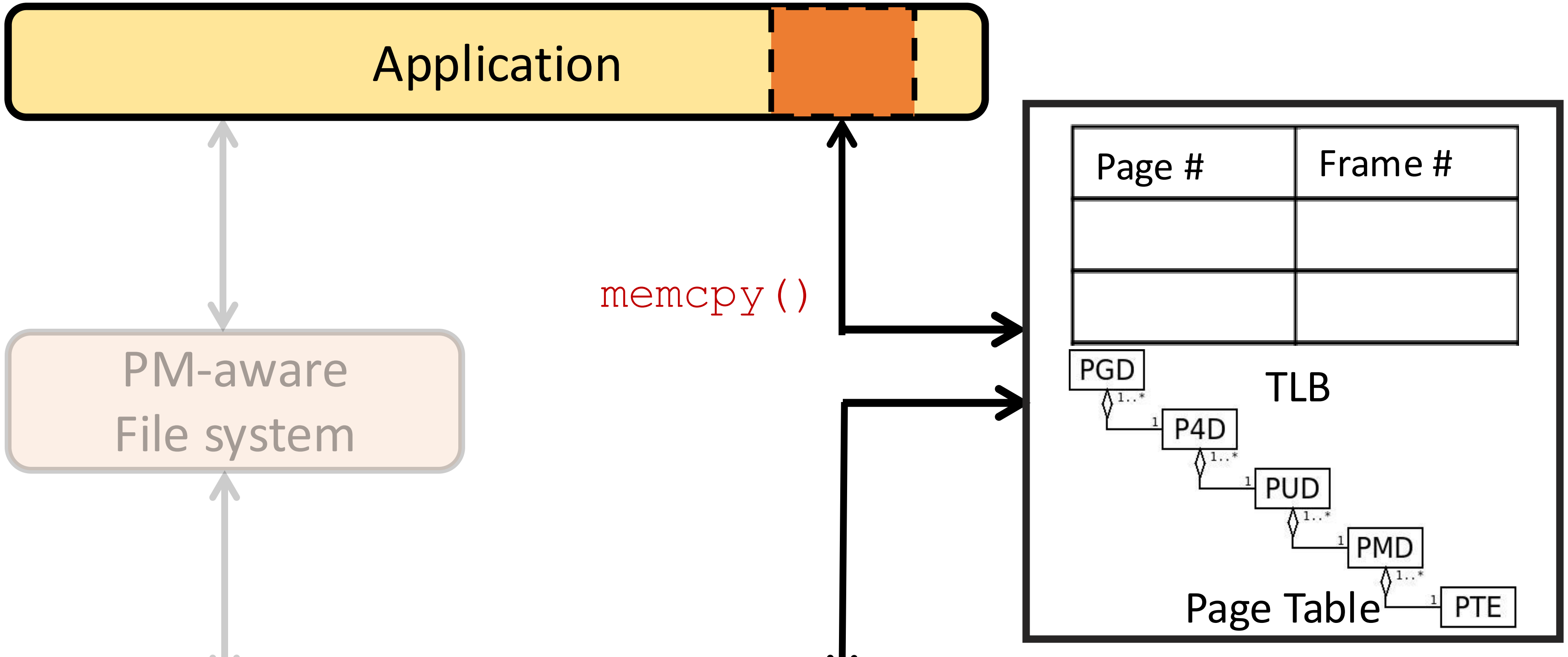
Memory-mapped Applications



Memory-mapped Applications



Memory-mapped Applications



Page #	Frame #

Performance of memory-mapped applications depends on page faults and TLB misses

Hugepages

Large pages (2MiB/1GiB)

Hugepages

Large pages (2MiB/1GiB)

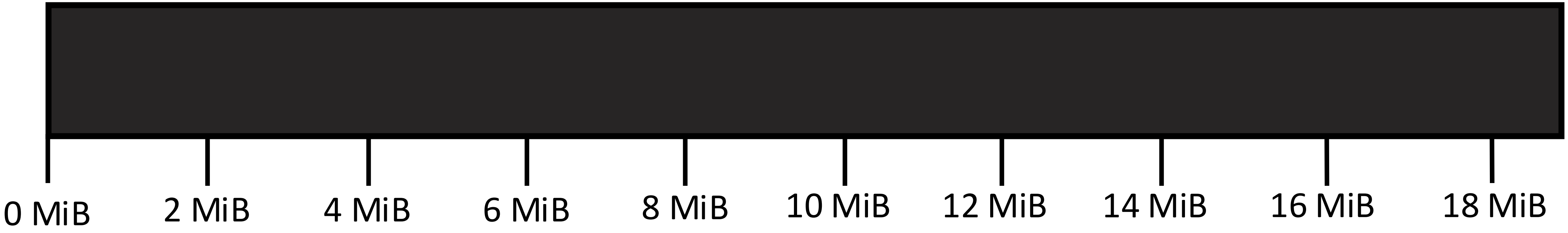
Reduce the number of page faults and TLB misses by up-to 500x

Hugepages

Large pages (2MiB/1GiB)

Reduce the number of page faults and TLB misses by up-to 500x

File systems need to allocate files using aligned & contiguous 2MiB extents

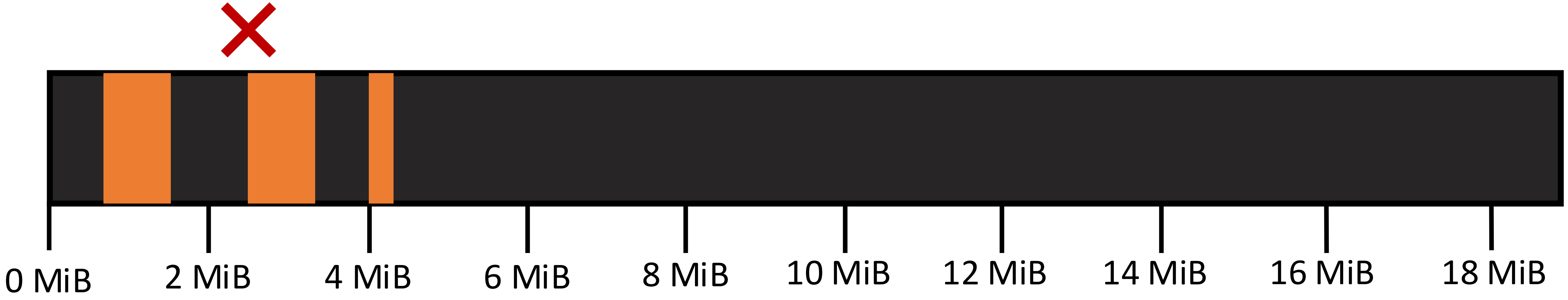


Hugepages

Large pages (2MiB/1GiB)

Reduce the number of page faults and TLB misses by up-to 500x

File systems need to allocate files using aligned & contiguous 2MiB extents

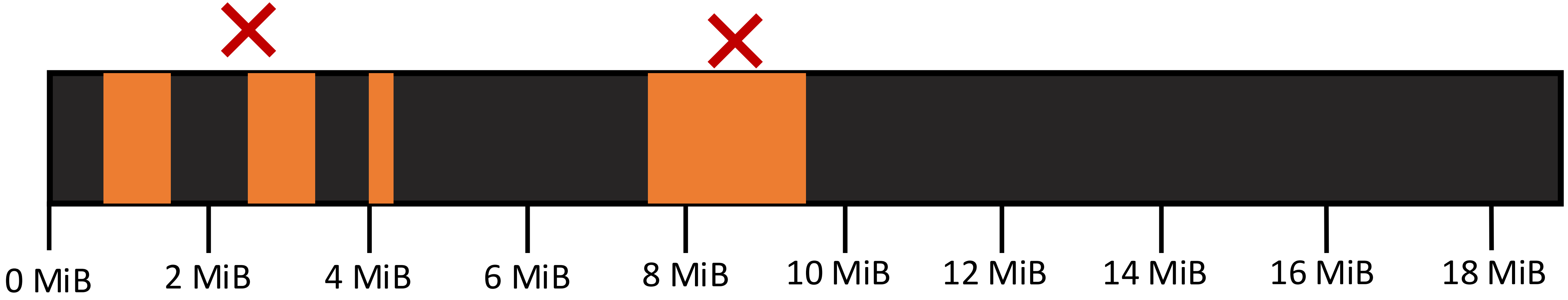


Hugepages

Large pages (2MiB/1GiB)

Reduce the number of page faults and TLB misses by up-to 500x

File systems need to allocate files using aligned & contiguous 2MiB extents

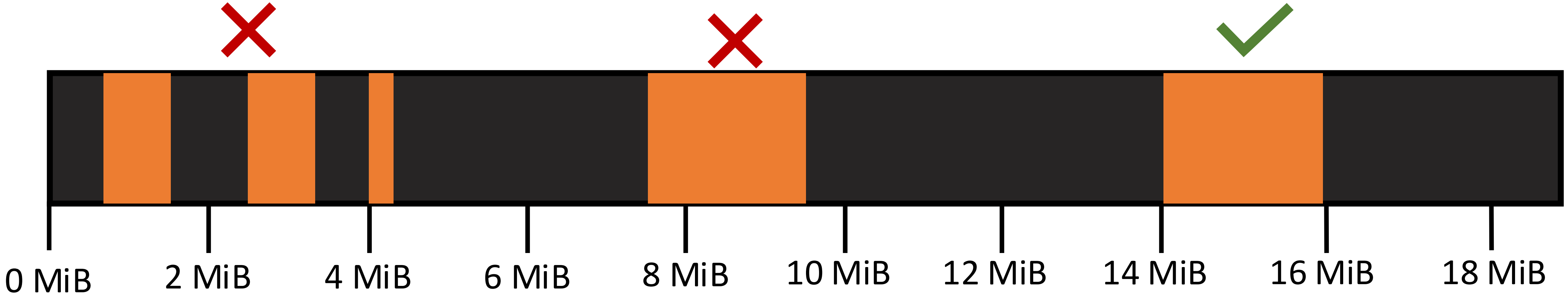


Hugepages

Large pages (2MiB/1GiB)

Reduce the number of page faults and TLB misses by up-to 500x

File systems need to allocate files using aligned & contiguous 2MiB extents



Hugepages

Large pages (2MiB/1GiB)

Reduce the number of page faults and TLB misses by up-to 500x

File systems need to allocate files using aligned & contiguous 2MiB extents

File systems are responsible in issuing hugepages for memory-mapped applications

Hugepages

Large pages (2MiB/1GiB)

Reduce the number of page faults and TLB misses by up-to 500x

File systems need to allocate files using aligned & contiguous 2MiB extents

File systems are responsible in issuing hugepages for memory-mapped applications

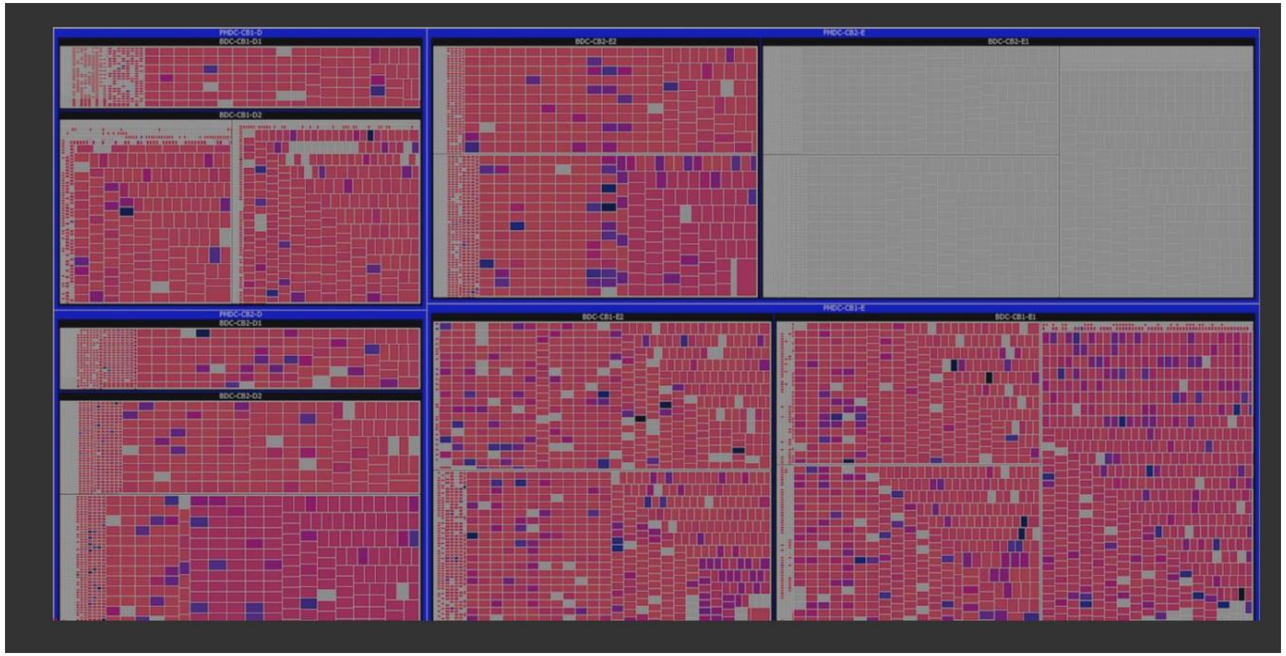
File systems must preserve hugepages with age

What is aging and why should we care?

State of file systems as a result of continuous allocations/deallocations, over time

What is aging and why should we care?

State of file systems as a result of continuous allocations/deallocations, over time

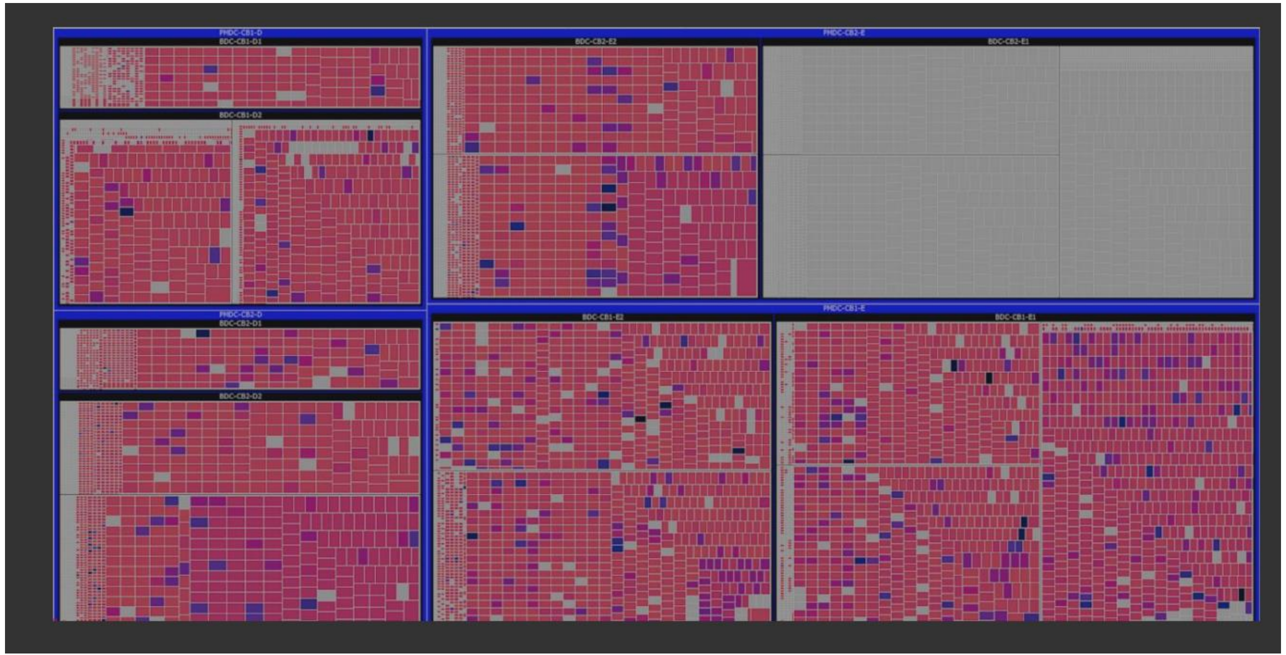


Google: “We want to keep disks full and busy to avoid excess inventory and wasted disk IOPs.”

Google. 2021. Colossus under the hood: a peek into Google’s scalable storage system. https://cloud.google.com/blog/products/storage- data-transfer/a-peek-behind-colossus-googles-file-system.

What is aging and why should we care?

State of file systems as a result of continuous allocations/deallocations, over time



Google: “We want to keep disks full and busy to avoid excess inventory and wasted disk IOPs.”

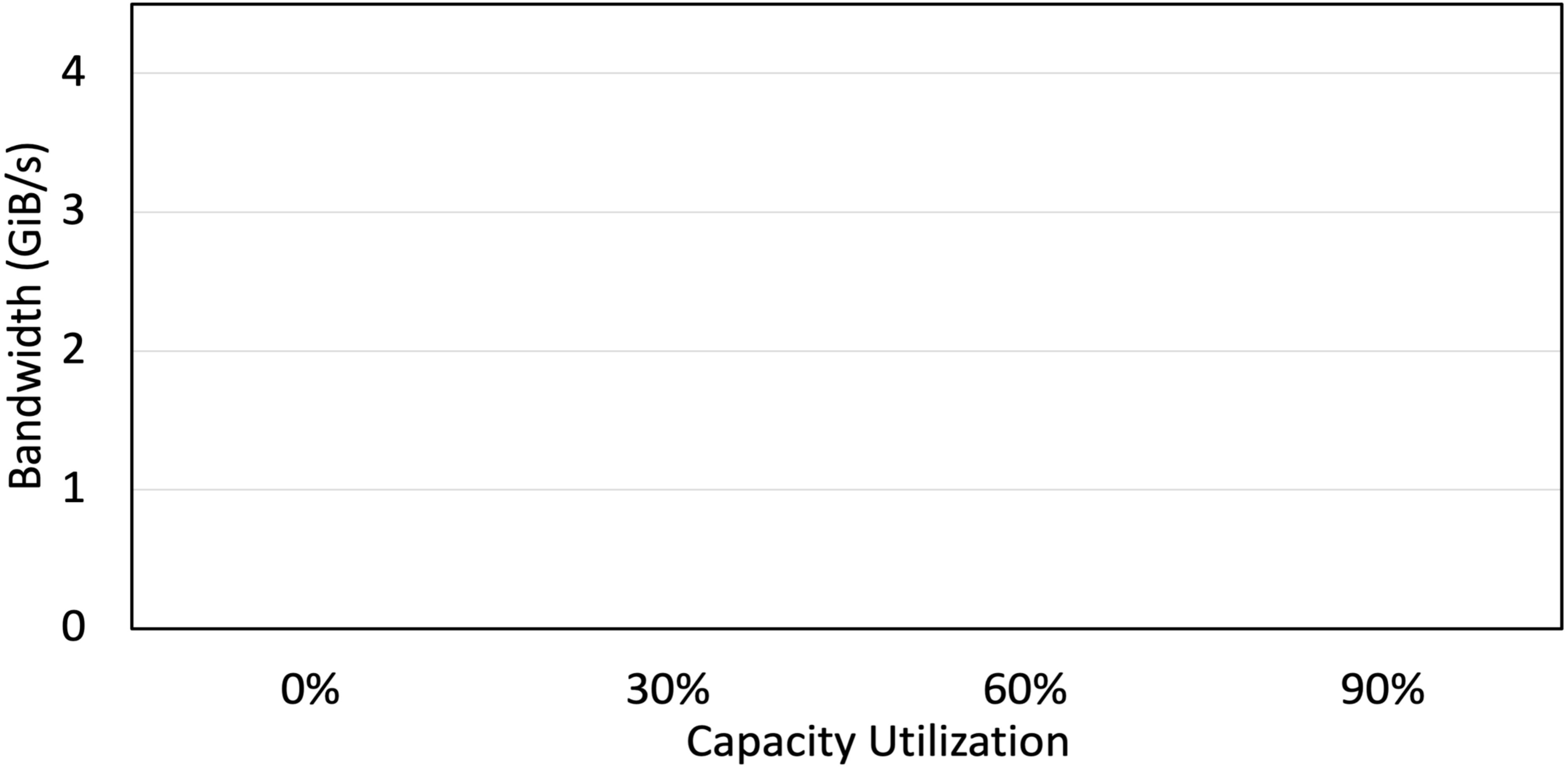
Google. 2021. Colossus under the hood: a peek into Google’s scalable storage system. https://cloud.google.com/blog/products/storage- data-transfer/a-peek-behind-colossus-googles-file-system.

File systems become fragmented over time due to frequent allocations and deallocations[1]

[1] Smith, Keith A., and Margo I. Seltzer. "File system aging—increasing the relevance of file system benchmarks." *Proceedings of the 1997 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. 1997.

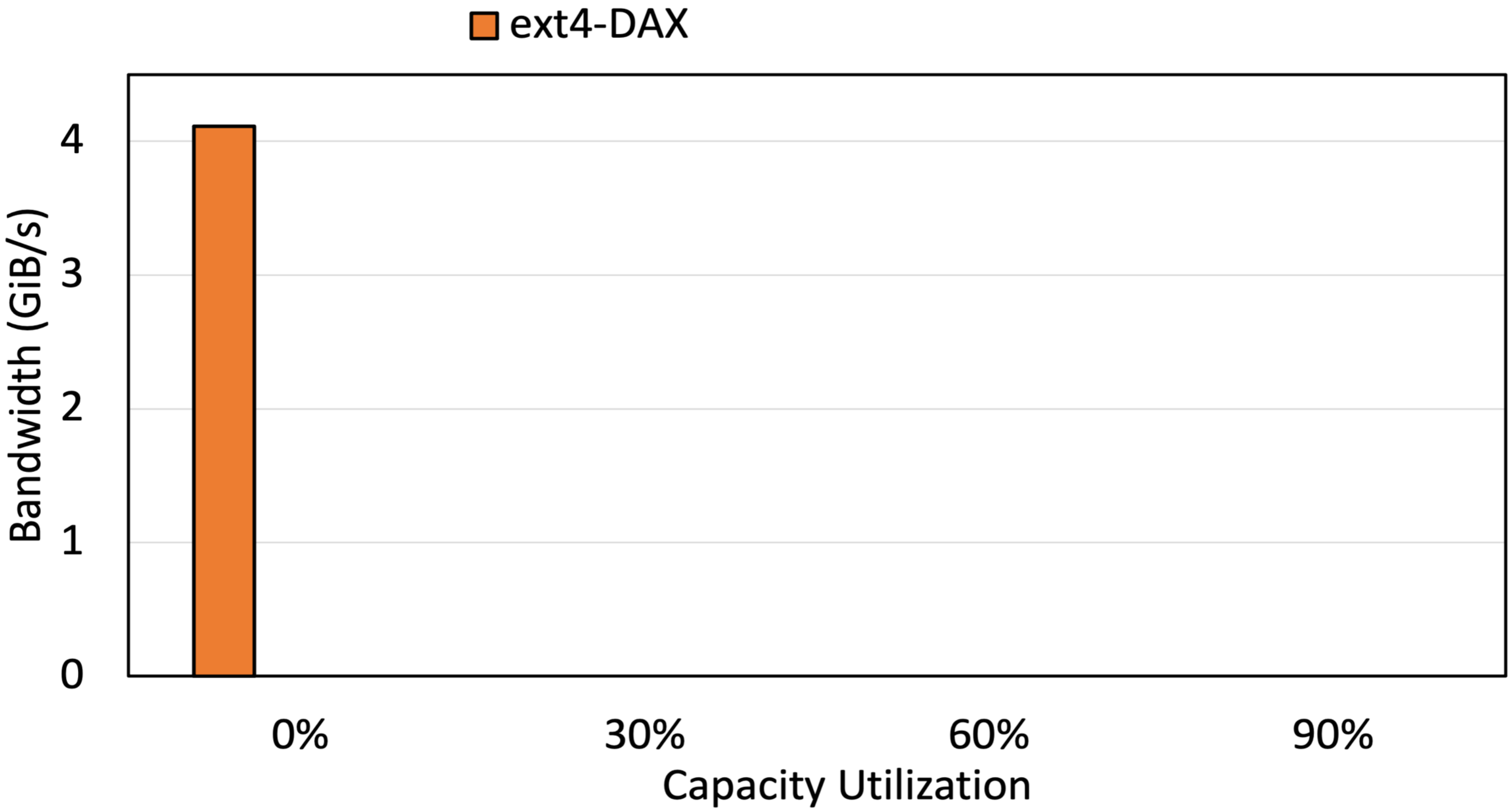
Performance impact of aging

Sequential write bandwidth using memcpy() on memory-mapped file



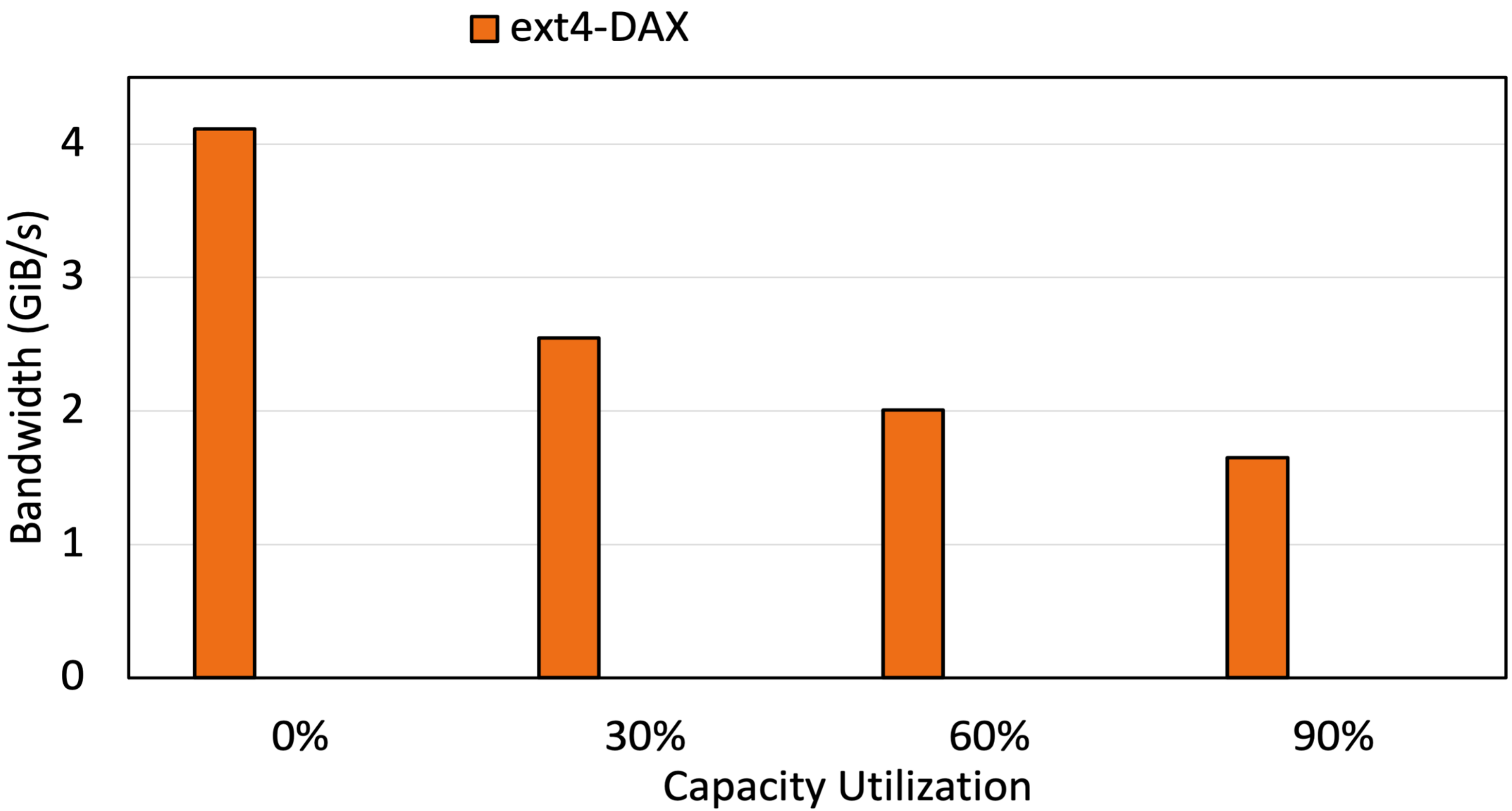
Performance impact of aging

Sequential write bandwidth using memcpy() on memory-mapped file



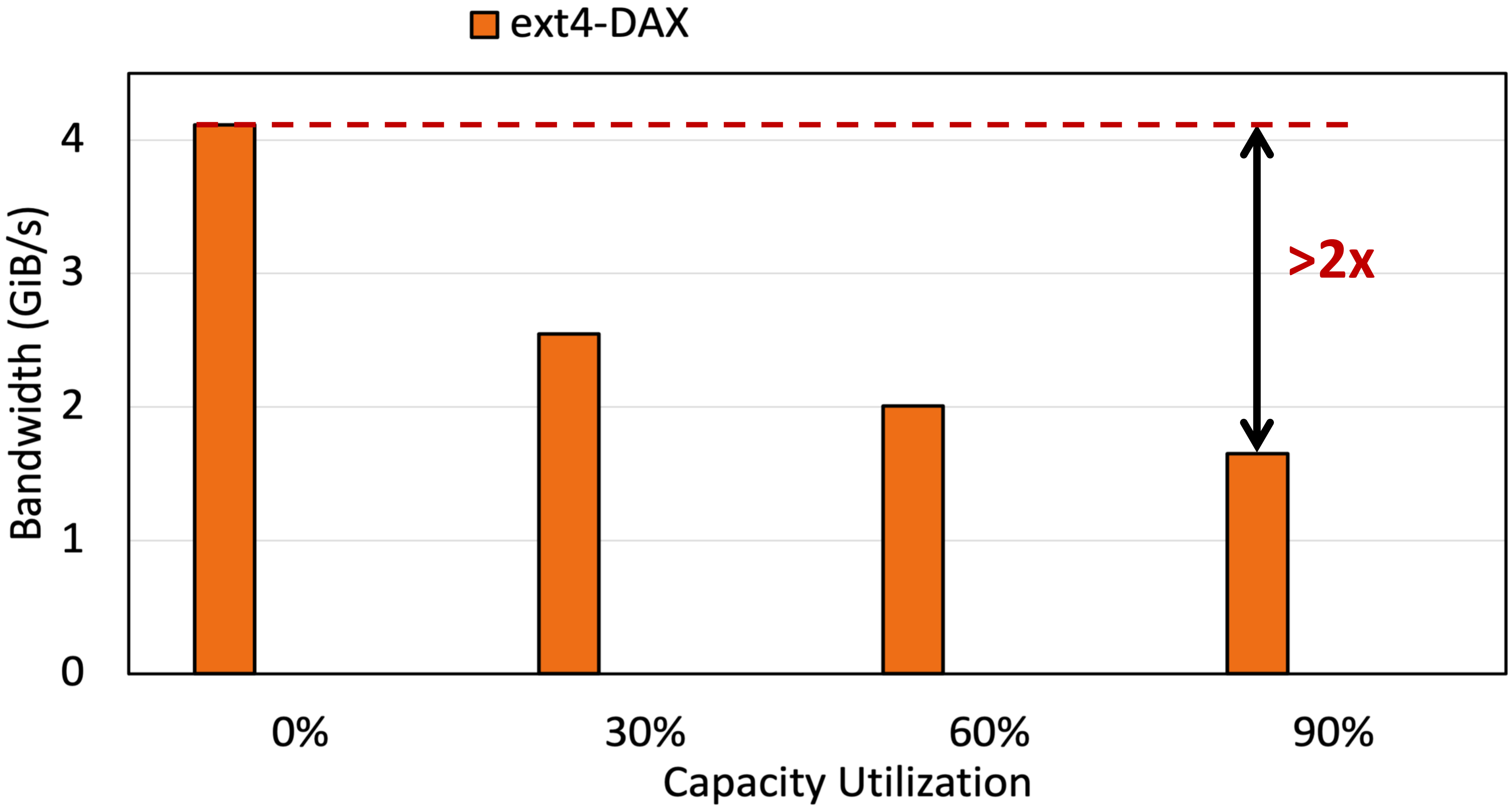
Performance impact of aging

Sequential write bandwidth using memcpy() on memory-mapped file



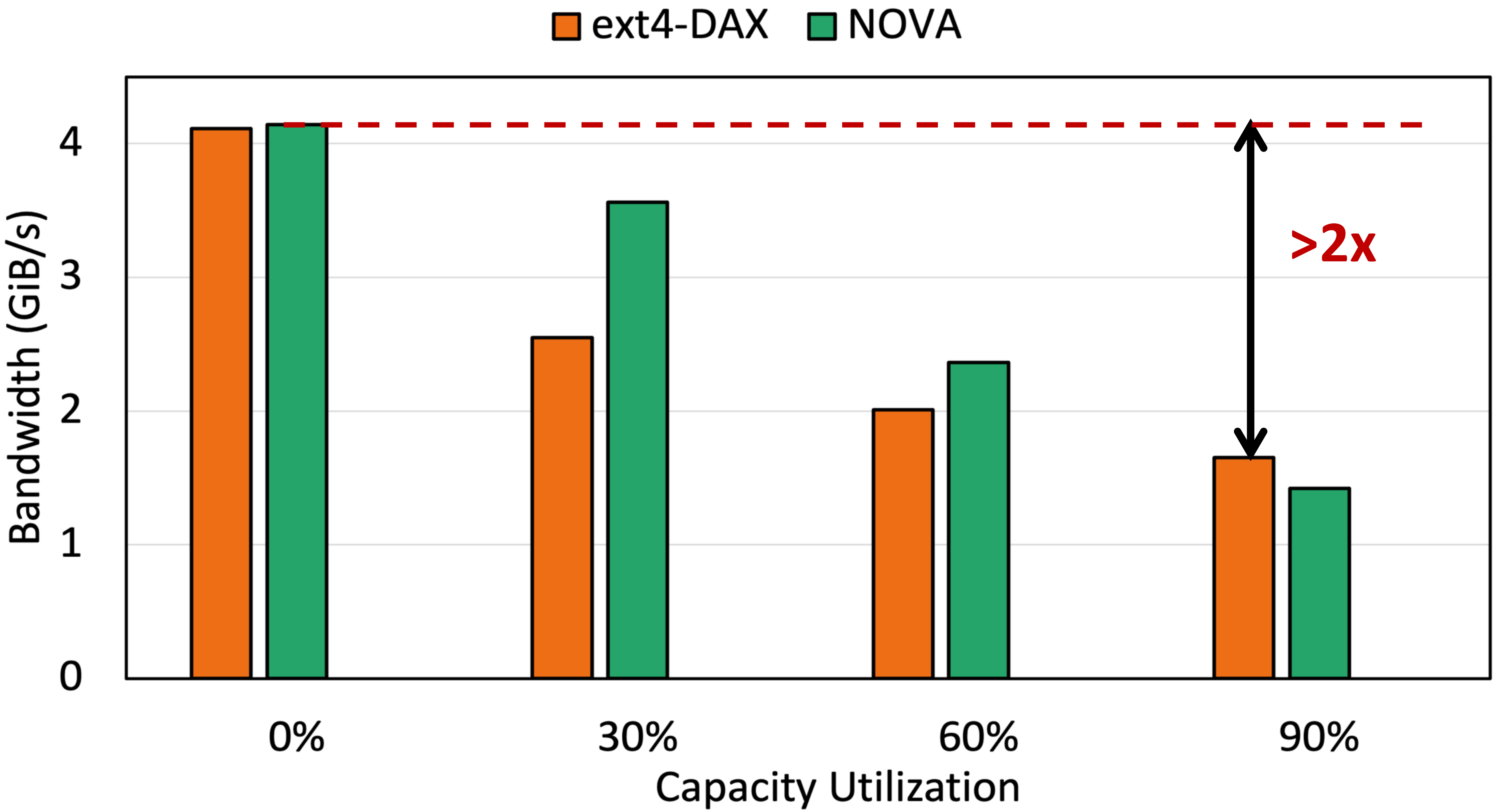
Performance impact of aging

Sequential write bandwidth using memcpy() on memory-mapped file



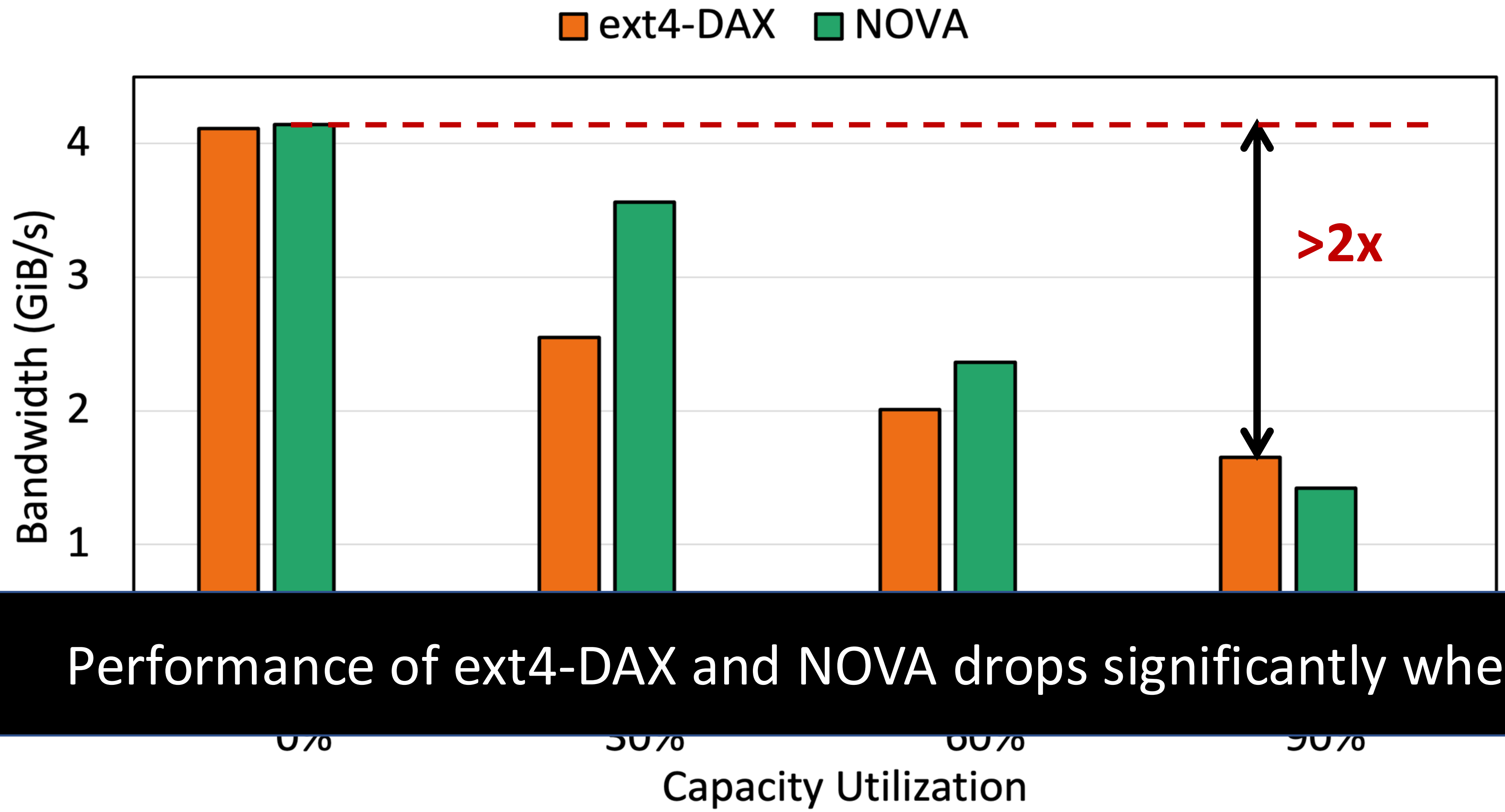
Performance impact of aging

Sequential write bandwidth using memcpy() on memory-mapped file



Performance impact of aging

Sequential write bandwidth using memcpy() on memory-mapped file



Performance of ext4-DAX and NOVA drops significantly when aged

WineFS

Hugepage-aware file system for PM that ages gracefully

WineFS

Hugepage-aware file system for PM that ages gracefully

WineFS uses a novel alignment-aware allocation policy to preserve hugepages

WineFS

Hugepage-aware file system for PM that ages gracefully

WineFS uses a novel alignment-aware allocation policy to preserve hugepages

WineFS achieves high performance for memory-mapped applications and POSIX system-call applications

WineFS

Hugepage-aware file system for PM that ages gracefully

WineFS uses a novel alignment-aware allocation policy to preserve hugepages
WineFS achieves high performance for memory-mapped applications and POSIX system-call applications

WineFS design achieves high scalability and works well on multiple NUMA nodes

WineFS

Hugepage-aware file system for PM that ages gracefully

WineFS uses a novel alignment-aware allocation policy to preserve hugepages
WineFS achieves high performance for memory-mapped applications and POSIX system-call applications

WineFS design achieves high scalability and works well on multiple NUMA nodes

WineFS performance almost stays the same when aged.
Aged WineFS performs better than freshly formatted NOVA

https://github.com/utsaslab/winefs

Insight behind WineFS

Insight behind WineFS

- Allocate memory-mapped files on aligned & contiguous extents

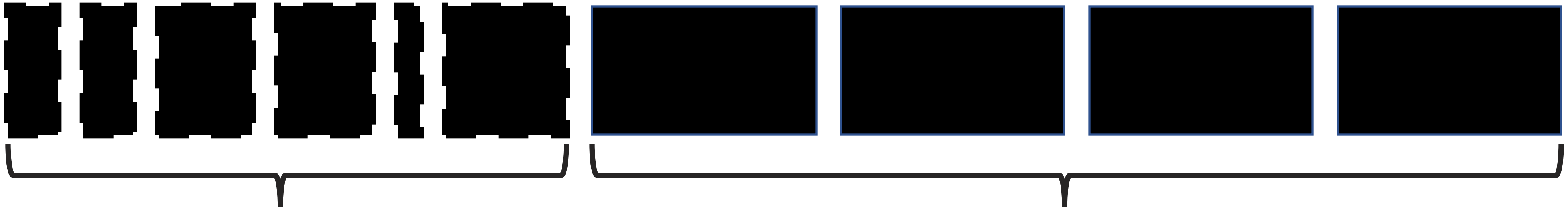
Insight behind WineFS

- Allocate memory-mapped files on aligned & contiguous extents

  Limitations of other file systems:

  - ext4-DAX and xfs-DAX preserve contiguity of free-space but not alignment

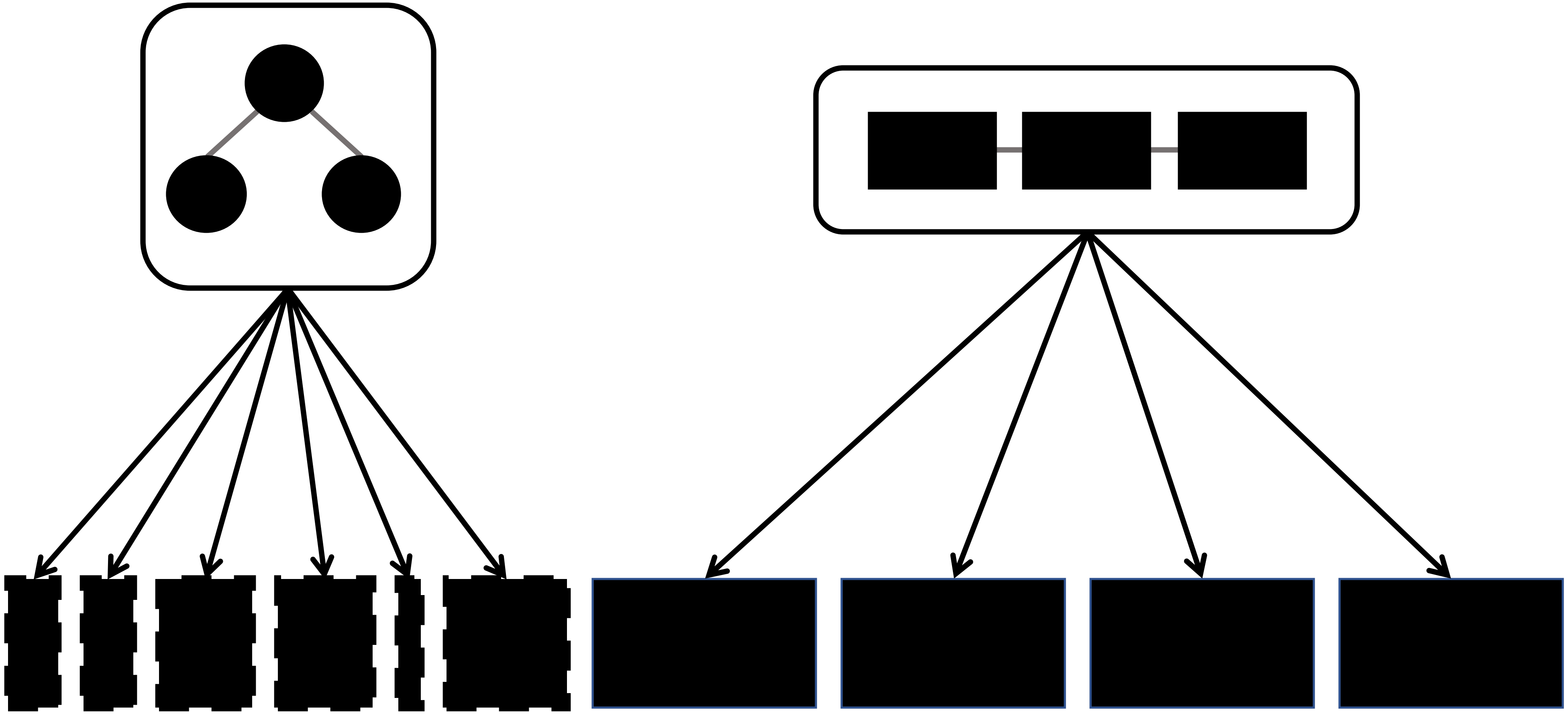
Alignment-aware allocation policy



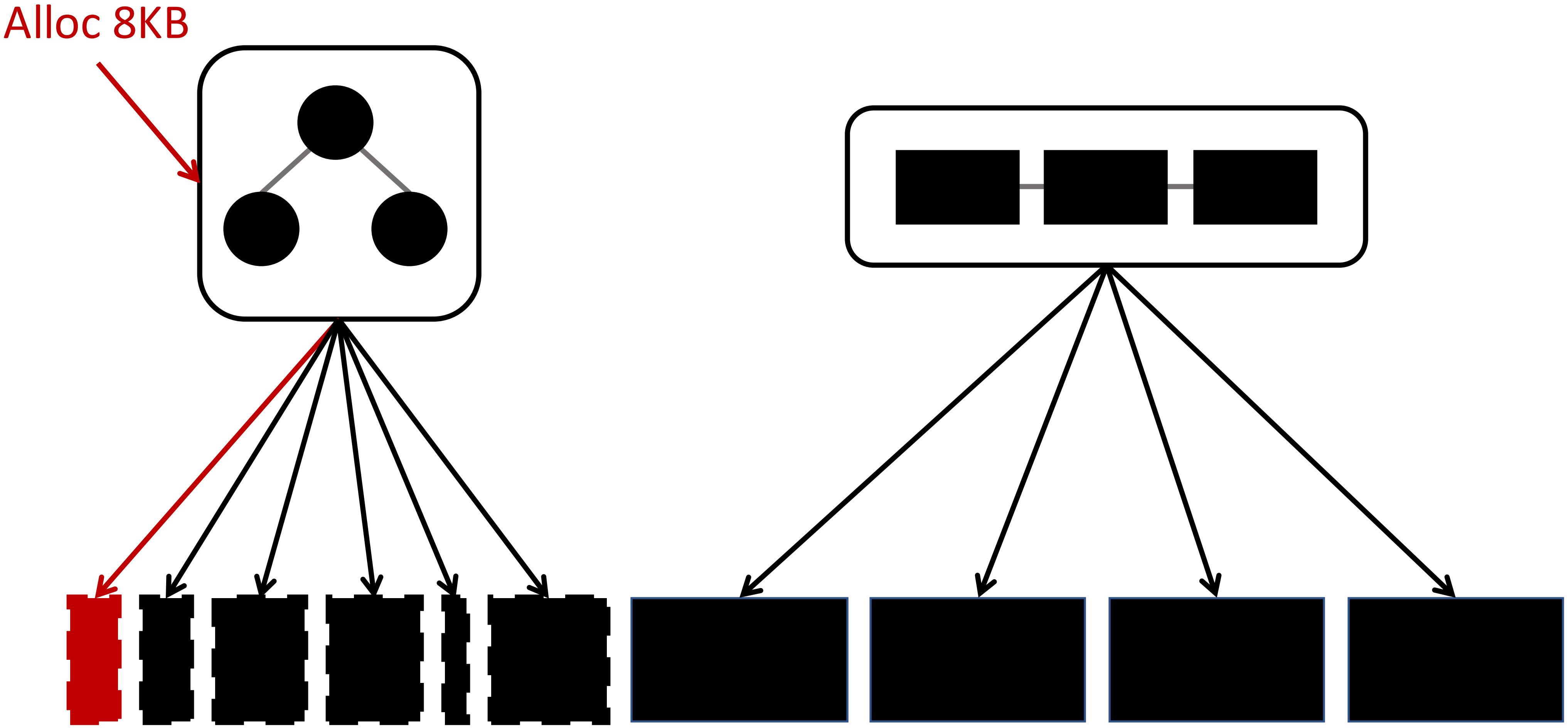
small/unaligned holes

2MiB aligned extents

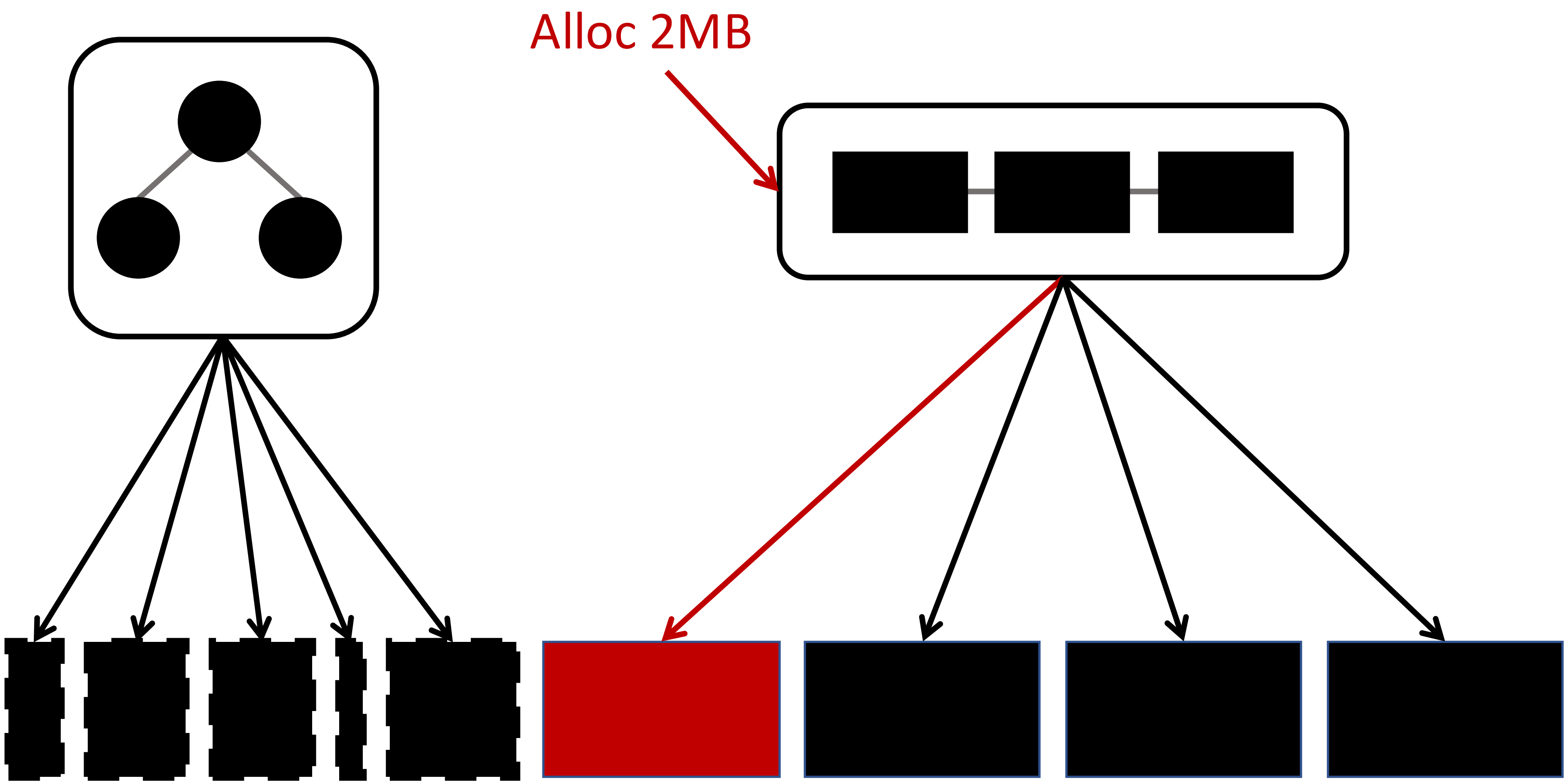
Alignment-aware allocation policy



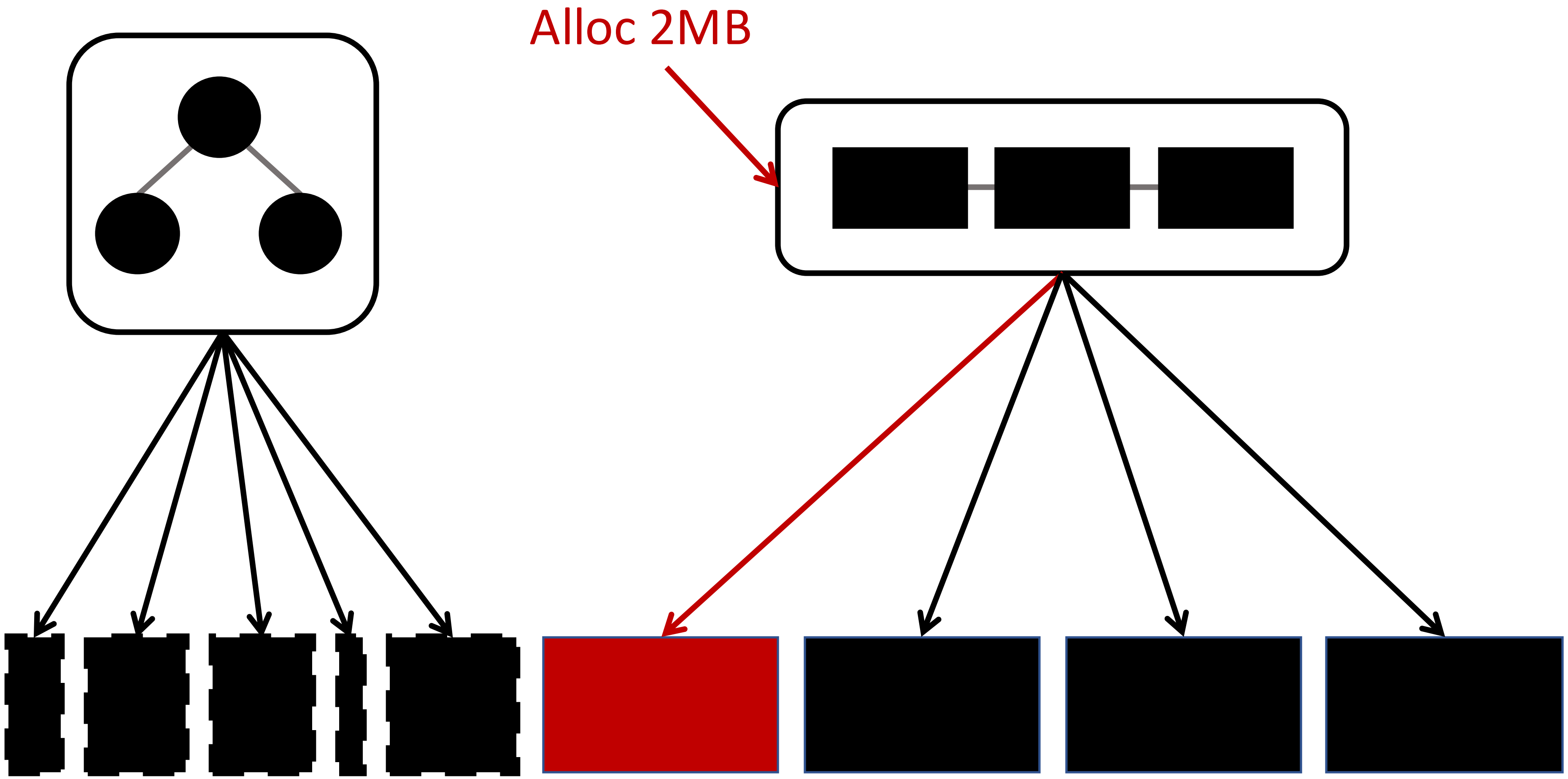
Alignment-aware allocation policy



Alignment-aware allocation policy

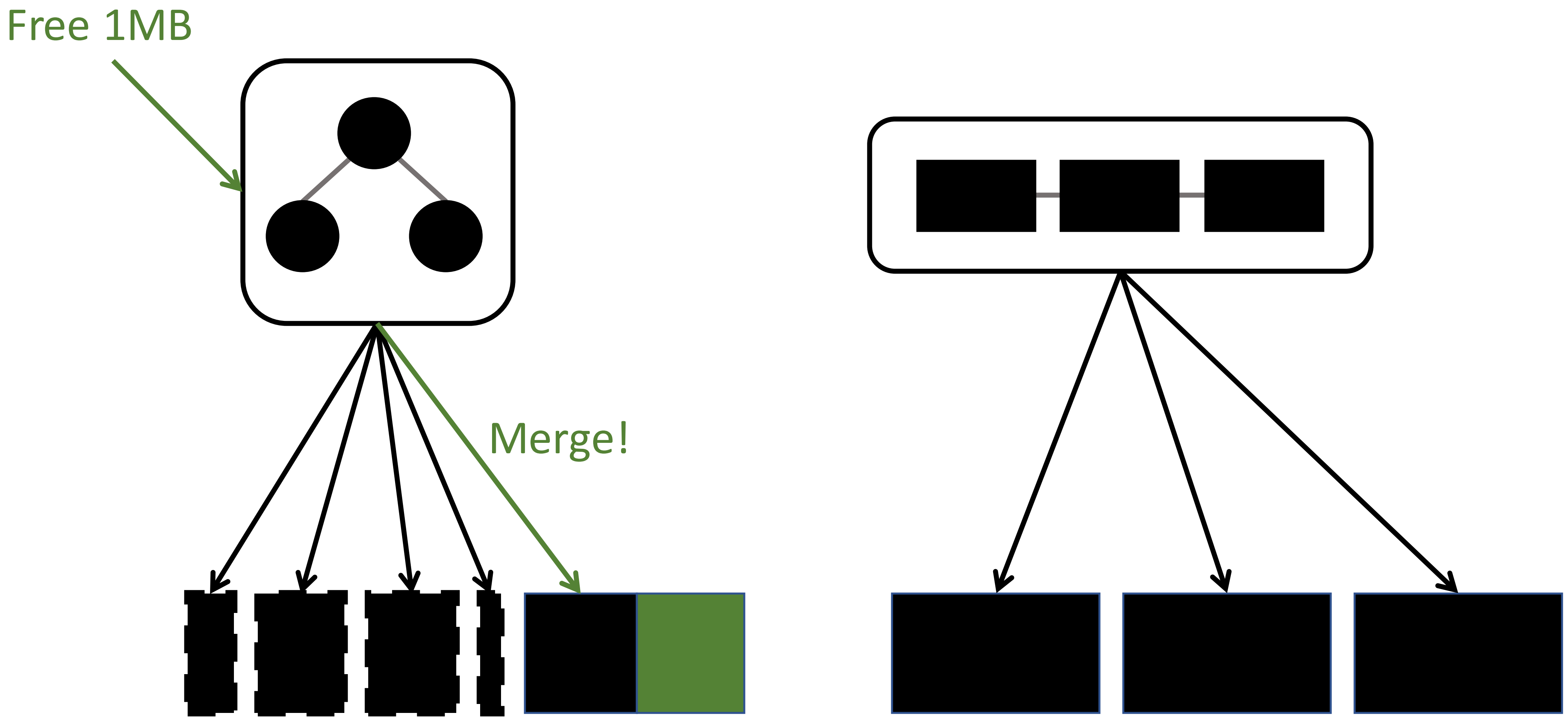


Alignment-aware allocation policy



Larger allocations are broken down into multiple 2MiB allocations and allocated from the hugepage list

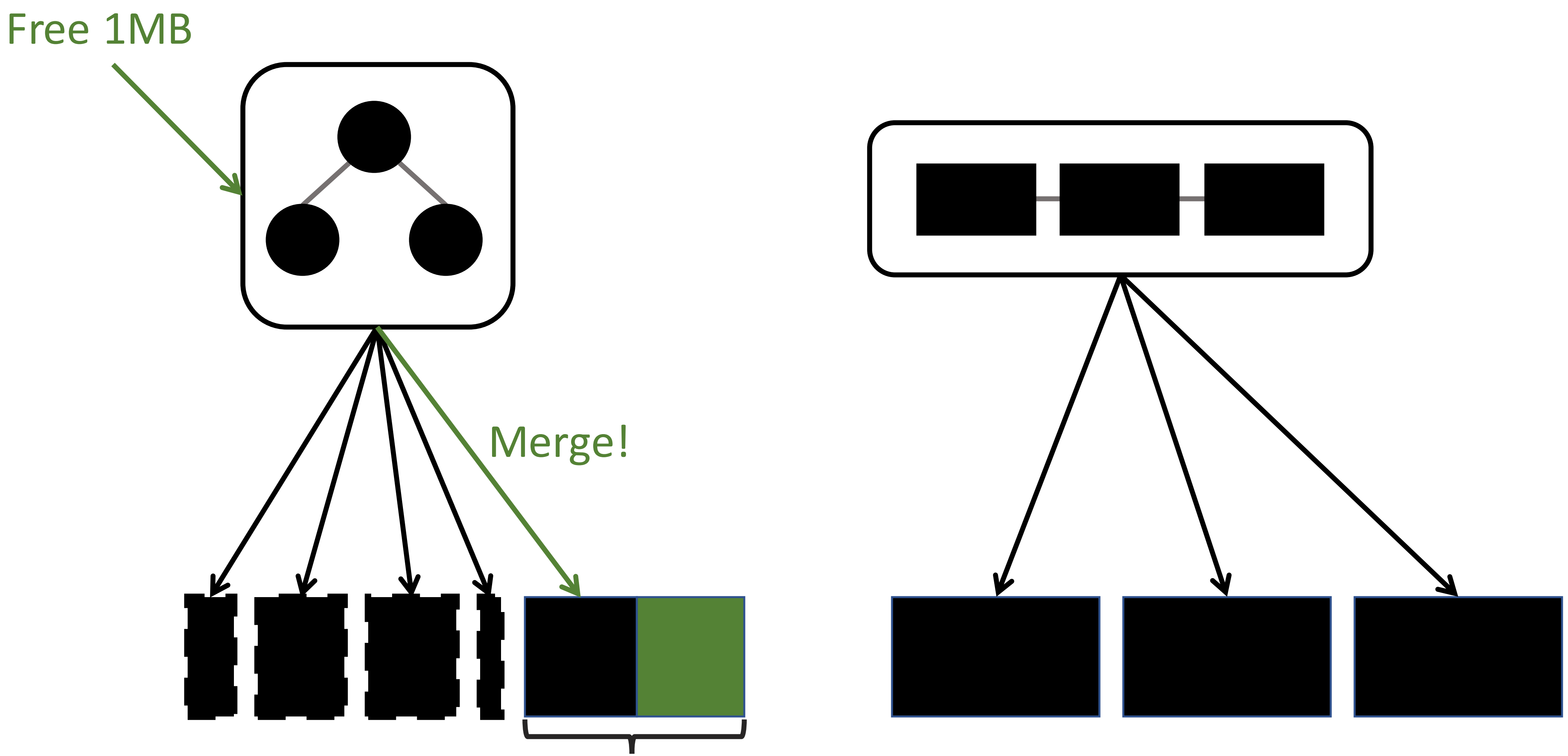
Alignment-aware allocation policy



Free 1MB

Merge!

Alignment-aware allocation policy

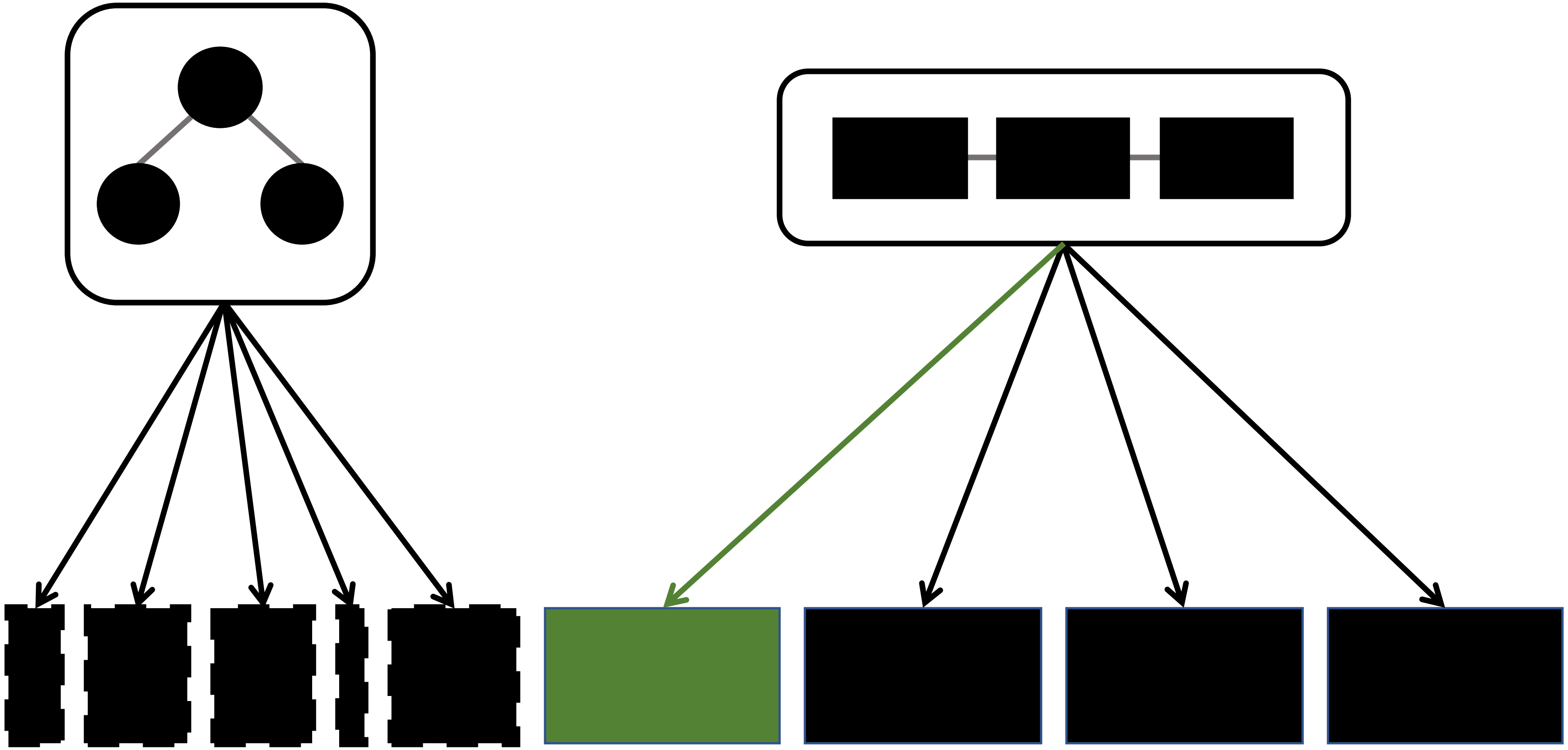


Free 1MB

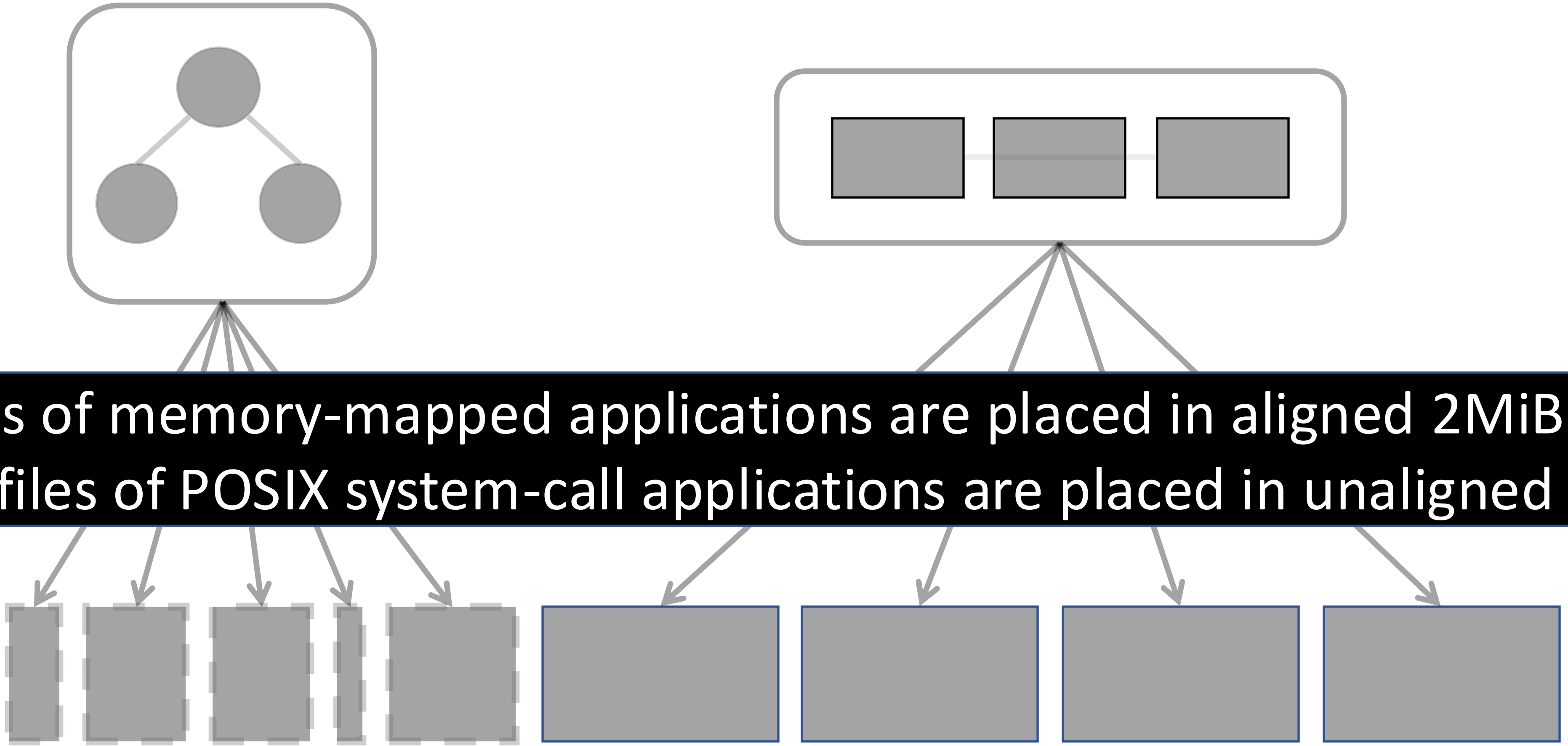
Merge!

2MB aligned extent!
Move to aligned extent pool

Alignment-aware allocation policy

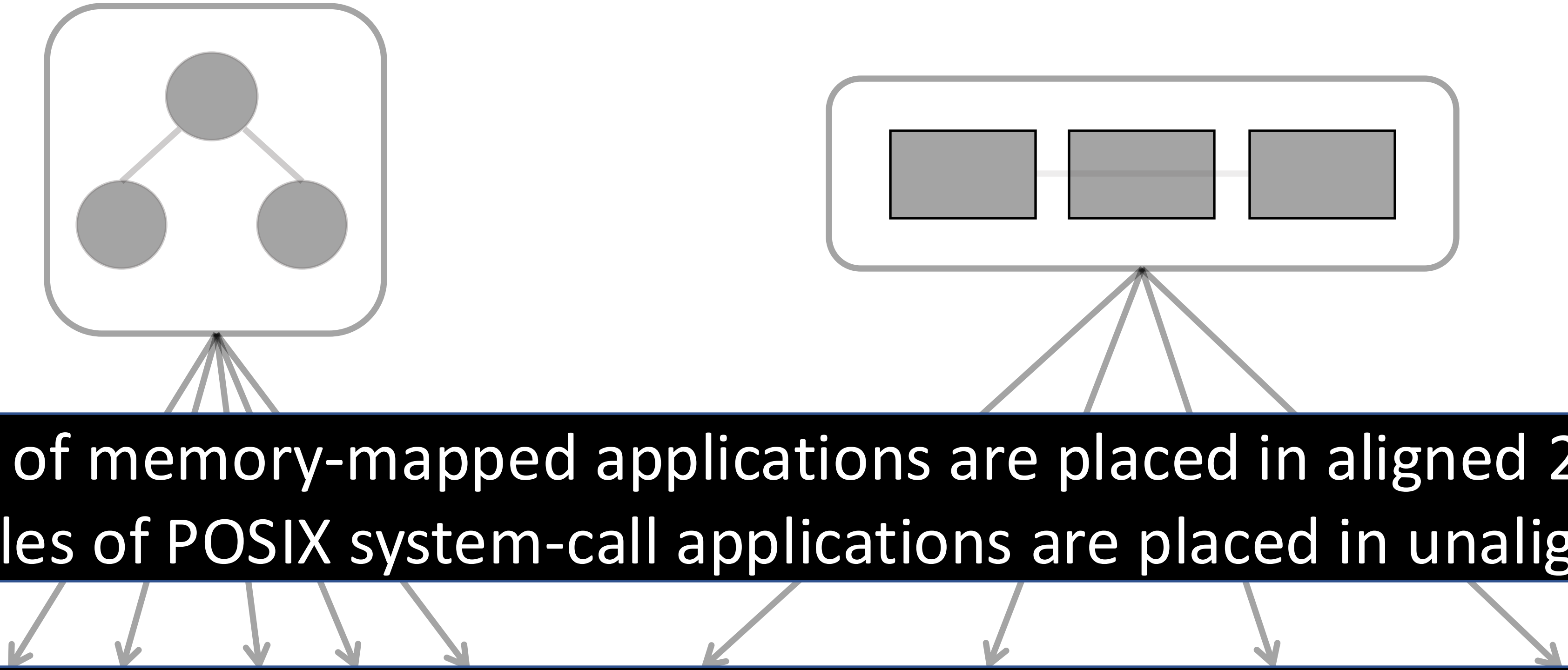


Alignment-aware allocation policy



Large files of memory-mapped applications are placed in aligned 2MiB extents, small files of POSIX system-call applications are placed in unaligned holes

Alignment-aware allocation policy



Large files of memory-mapped applications are placed in aligned 2MiB extents, small files of POSIX system-call applications are placed in unaligned holes

Hugepages are aggressively reclaimed on file deallocations

Insight behind WineFS

- Allocate memory-mapped files on aligned & contiguous extents

  Limitations of other file systems:

  - ext4-DAX and xfs-DAX preserve contiguity of free-space but not alignment

- Achieve high scalability while avoiding fragmentation of free space

Insight behind WineFS

- Allocate memory-mapped files on aligned & contiguous extents

  Limitations of other file systems:

  - ext4-DAX and xfs-DAX preserve contiguity of free-space but not alignment

- Achieve high scalability while avoiding fragmentation of free space

  Limitations of other file systems:

  - Per-inode log of NOVA fragments free space
  - Per-process log of Strata fragments files

Achieving high scalability while avoiding fragmentation

Achieving high scalability while avoiding fragmentation

WineFS uses per-CPU journals & allocation groups for achieving high scalability

Achieving high scalability while avoiding fragmentation

WineFS uses per-CPU journals & allocation groups for achieving high scalability

WineFS uses a hybrid data consistency mechanism for avoiding fragmentation of free space

Achieving high scalability while avoiding fragmentation

WineFS uses per-CPU journals & allocation groups for achieving high scalability

WineFS uses a hybrid data consistency mechanism for avoiding fragmentation of free space

WineFS constructs on-PM layout that avoids fragmentation of free space by containing metadata structures to specific regions on PM

Evaluation

Setup:

- 500 GB partition of Intel Optane DC Persistent Memory
- 28 cores, 112 threads, 32MB LLC

File systems compared:

- ext4-DAX, xfs-DAX, NOVA, Strata, SplitFS

Evaluation

What is the memory-mapped performance of WineFS after aging?

What is the POSIX system-call performance of WineFS?

Is WineFS scalable?

Evaluation

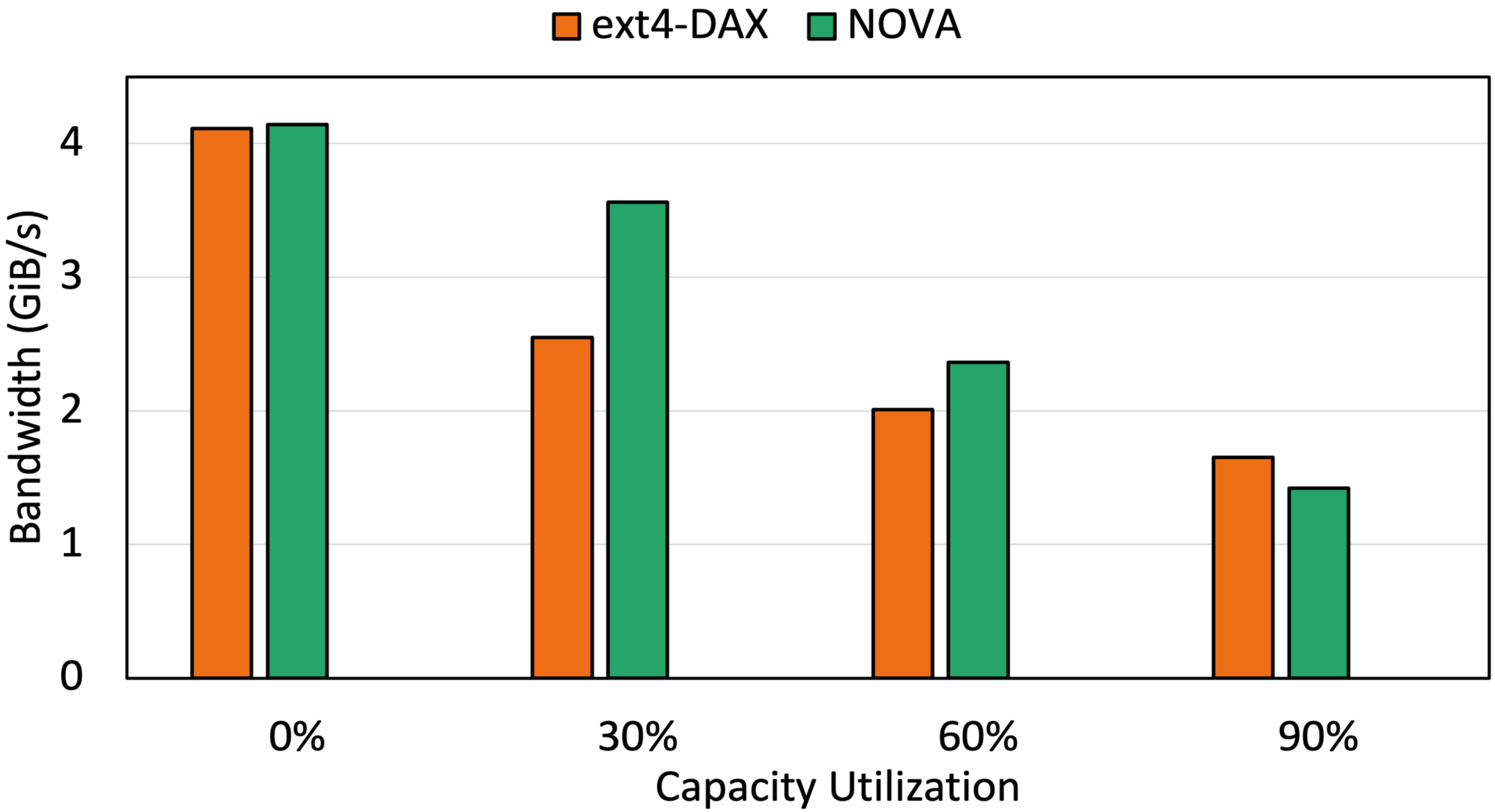
What is the memory-mapped performance of WineFS after aging?

What is the POSIX system-call performance of WineFS?

Is WineFS scalable?

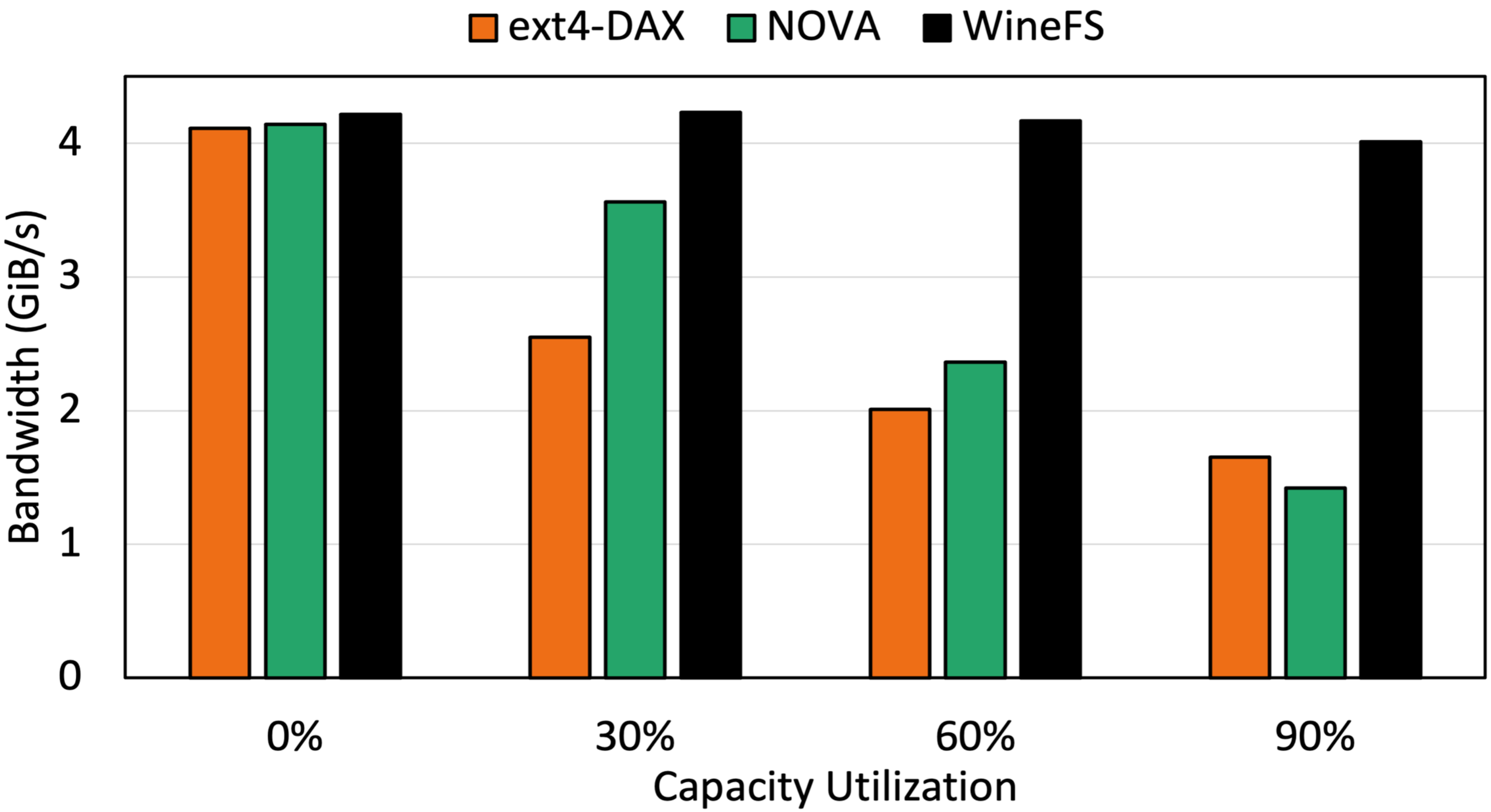
Performance impact of aging

Sequential write bandwidth using memcpy() on memory-mapped file



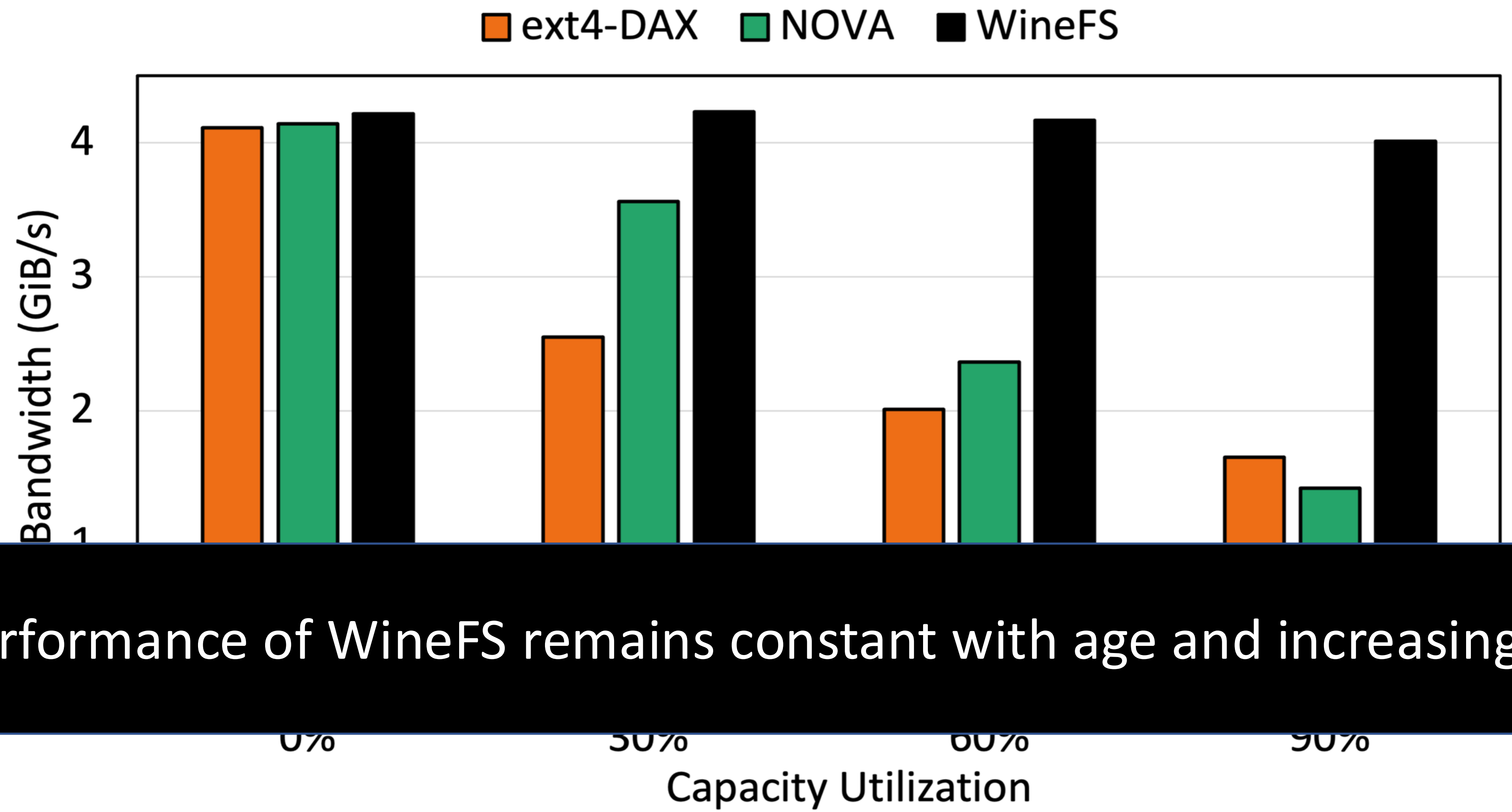
Performance impact of aging

Sequential write bandwidth using memcpy() on memory-mapped file



Performance impact of aging

Sequential write bandwidth using memcpy() on memory-mapped file

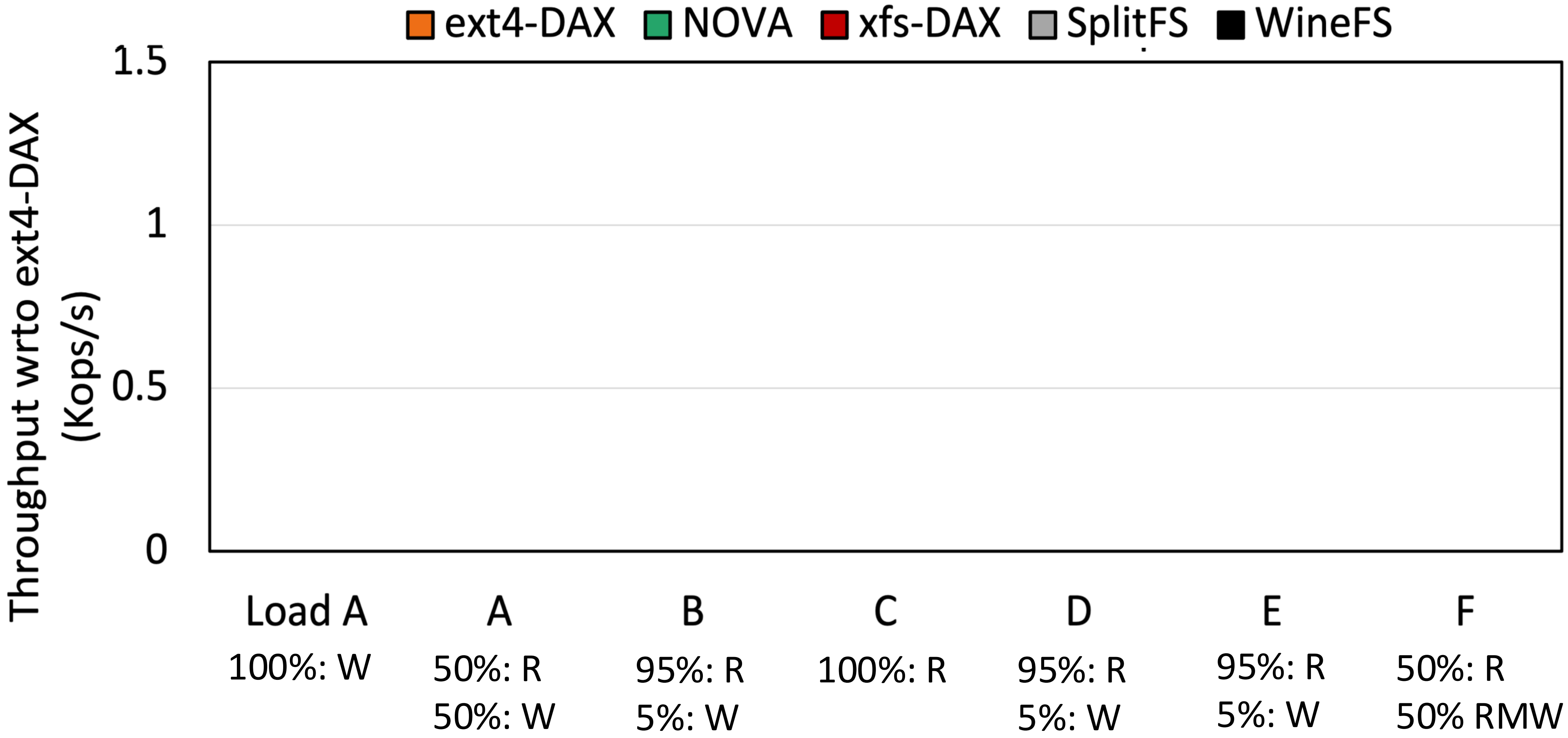


Performance of WineFS remains constant with age and increasing utilization

YCSB on RocksDB

Yahoo! Cloud Serving Benchmark - Industry standard macro-benchmark

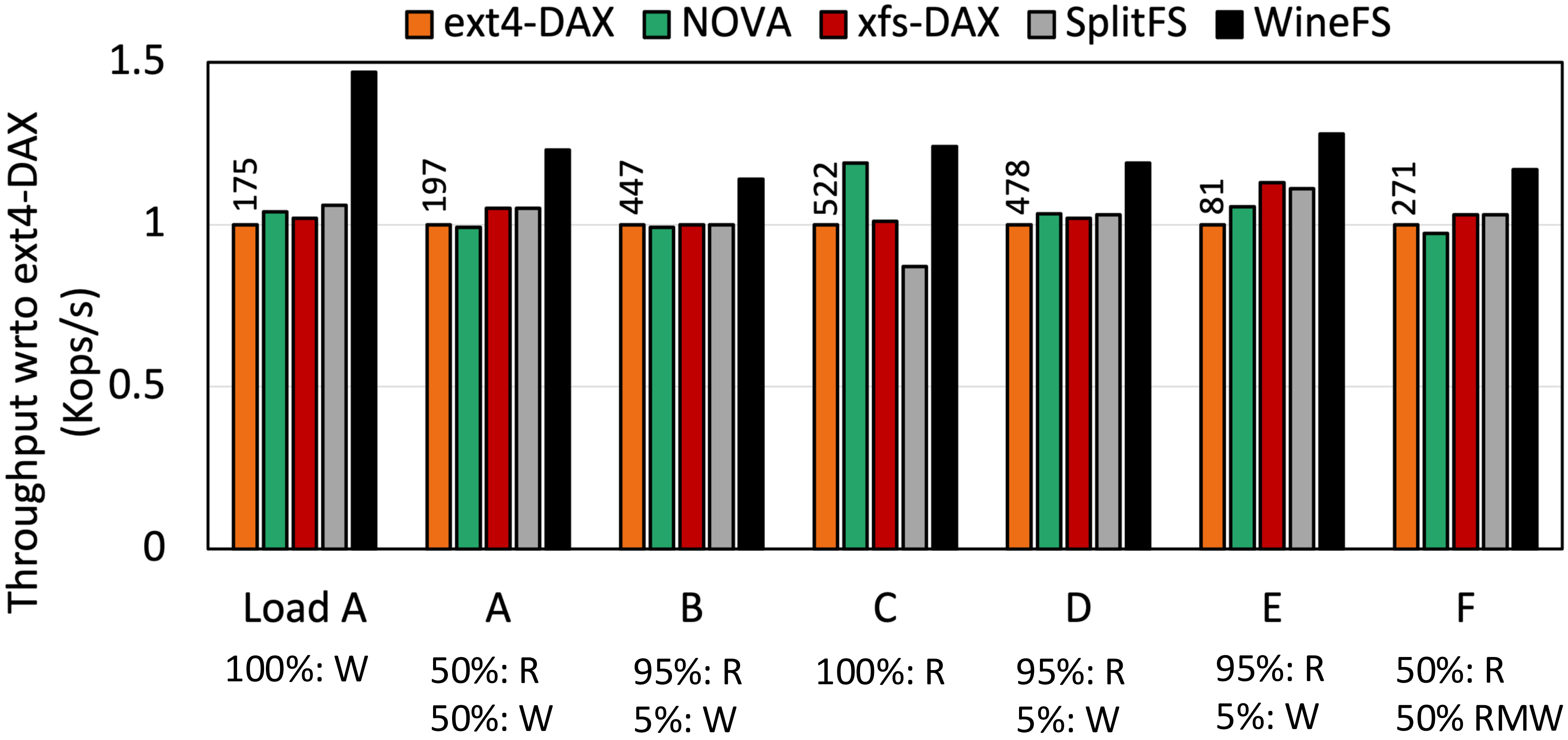
Insert 5M keys. Run 5M operations. Key size = 16 bytes. Value size = 1K



YCSB on RocksDB

Yahoo! Cloud Serving Benchmark - Industry standard macro-benchmark

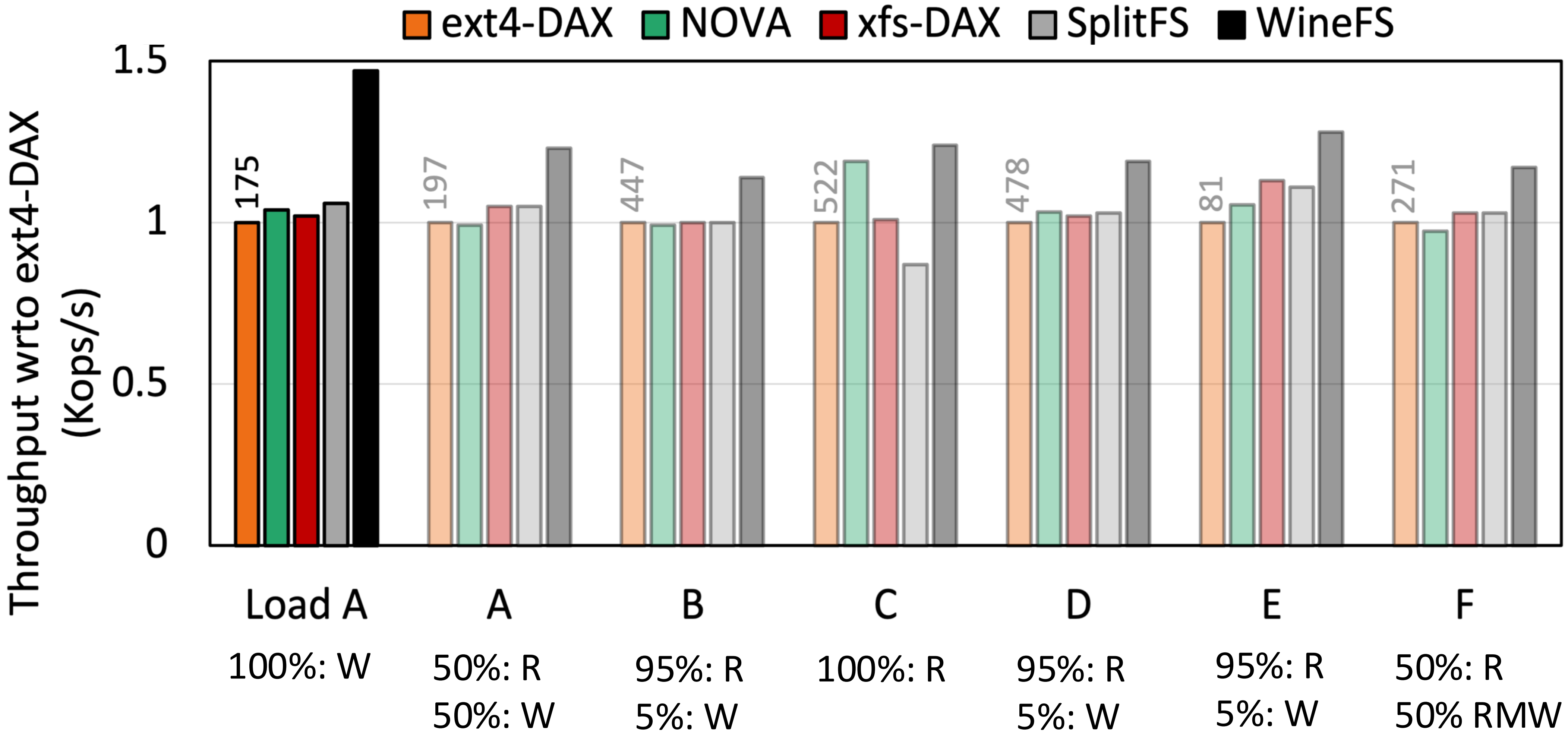
Insert 5M keys. Run 5M operations. Key size = 16 bytes. Value size = 1K



YCSB on RocksDB

Yahoo! Cloud Serving Benchmark - Industry standard macro-benchmark

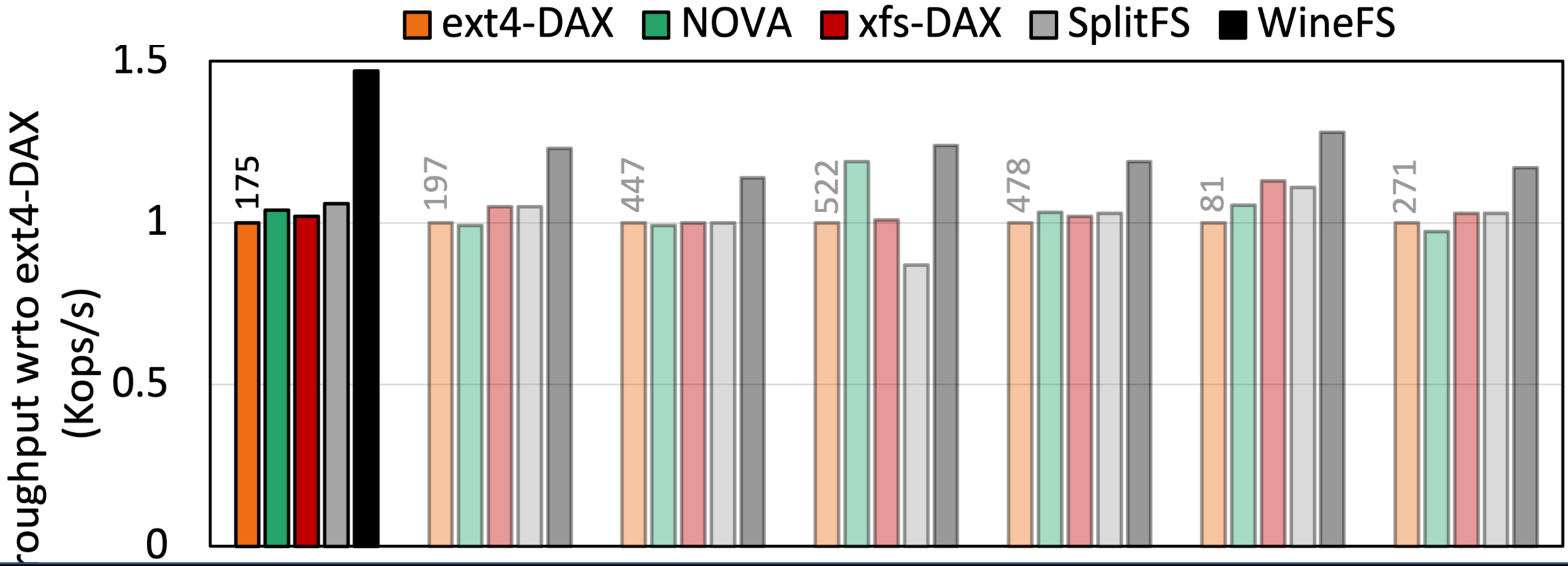
Insert 5M keys. Run 5M operations. Key size = 16 bytes. Value size = 1K



YCSB on RocksDB

Yahoo! Cloud Serving Benchmark - Industry standard macro-benchmark

Insert 5M keys. Run 5M operations. Key size = 16 bytes. Value size = 1K



WineFS suffers from 30x fewer page faults compared to NOVA

Conclusion

WineFS demonstrates that it is possible to design a file system that…

Achieves high performance for new memory-mapped applications

Achieves high performance and scalability for legacy POSIX applications

Maintains high performance in the presence of aging and under high utilization

Artifacts Available acm V1.1

Artifacts Evaluated acm Functional V1.1

Results Reproduced acm V1.1

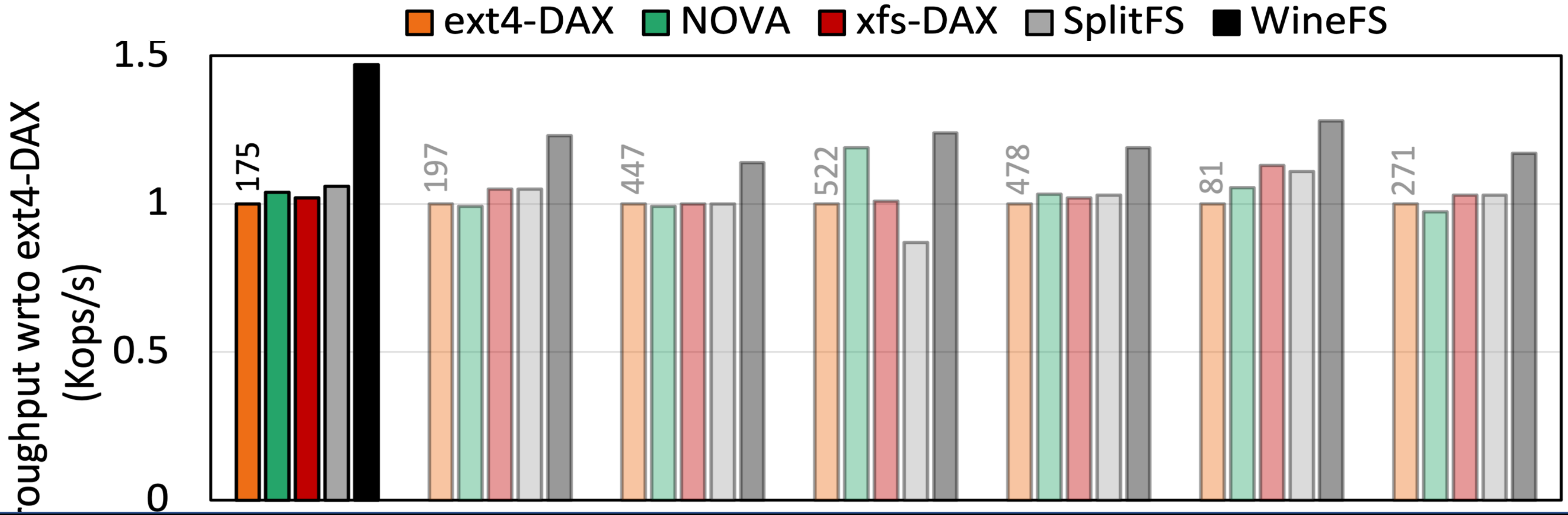
https://github.com/utsaslab/winefs

Backup Slides

YCSB on RocksDB

Yahoo! Cloud Serving Benchmark - Industry standard macro-benchmark

Insert 5M keys. Run 5M operations. Key size = 16 bytes. Value size = 1K

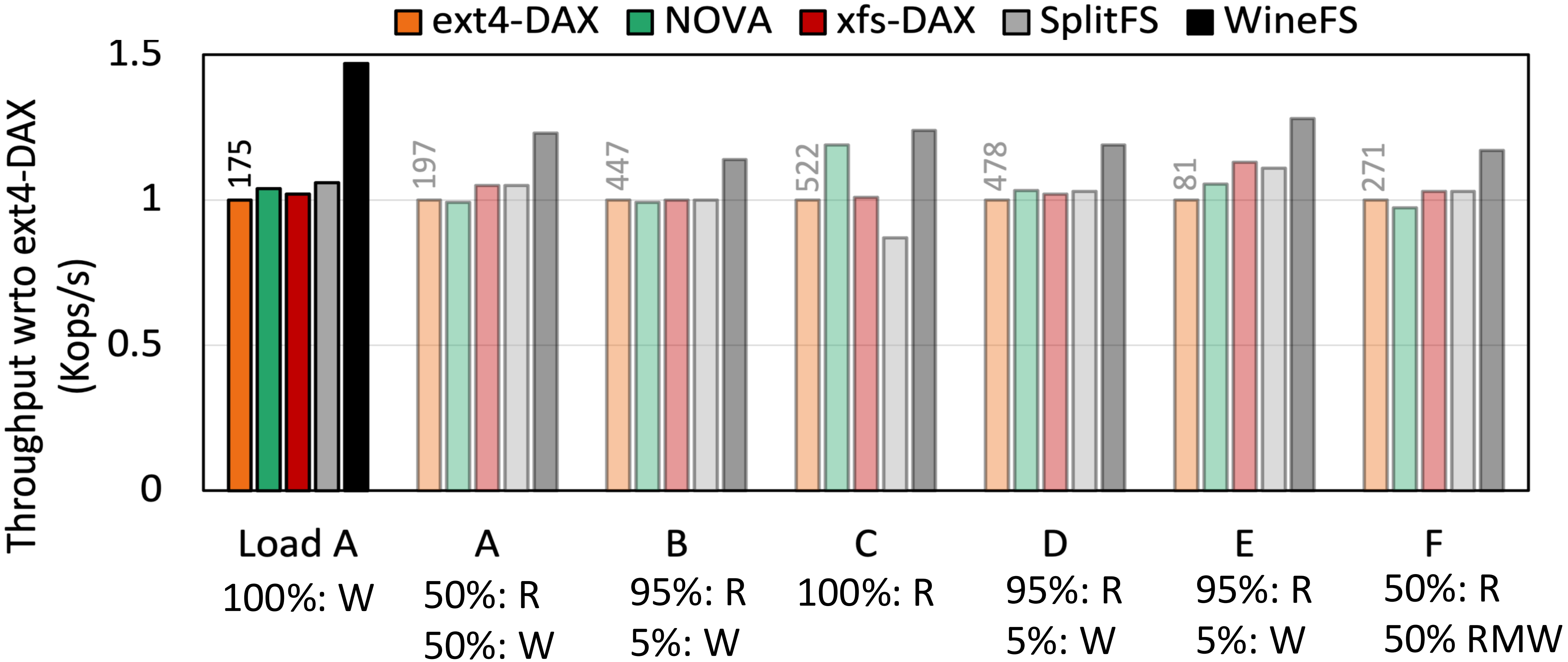


WineFS suffers from 30x fewer page faults compared to NOVA

YCSB on RocksDB

Yahoo! Cloud Serving Benchmark - Industry standard macro-benchmark

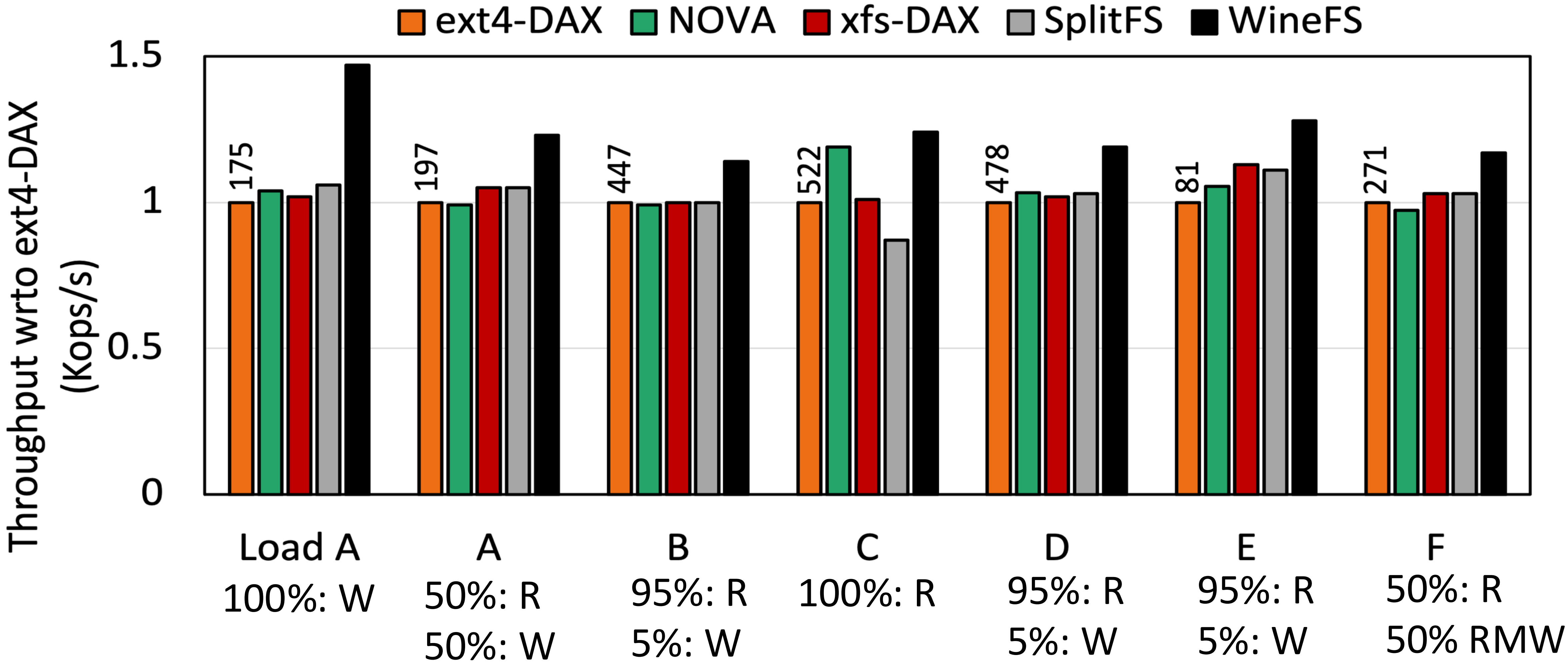
Insert 5M keys. Run 5M operations. Key size = 16 bytes. Value size = 1K



YCSB on RocksDB

Yahoo! Cloud Serving Benchmark - Industry standard macro-benchmark

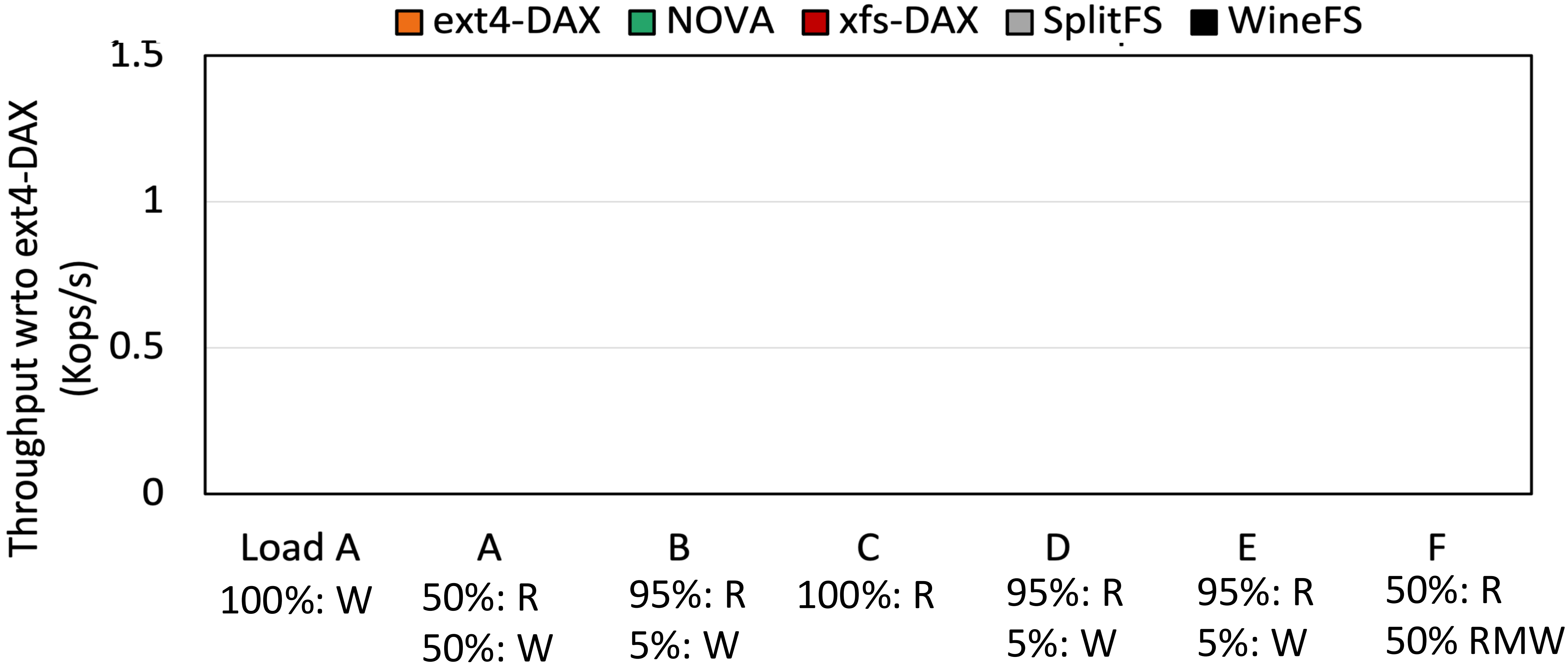
Insert 5M keys. Run 5M operations. Key size = 16 bytes. Value size = 1K



YCSB on RocksDB

Yahoo! Cloud Serving Benchmark - Industry standard macro-benchmark

Insert 5M keys. Run 5M operations. Key size = 16 bytes. Value size = 1K



Alignment-aware allocation policy

Places large files of memory-mapped applications in aligned hugepage extents

Alignment-aware allocation policy

Places large files of memory-mapped applications in aligned hugepage extents

Places small files of other POSIX applications in unaligned holes, avoiding fragmentation of free space

Alignment-aware allocation policy

Places large files of memory-mapped applications in aligned hugepage extents

Places small files of other POSIX applications in unaligned holes, avoiding fragmentation of free space

Aggressively reclaims hugepages on file deallocations

Insight behind WineFS

Hugepages require contiguity as well as alignment of free space.

Existing file systems such as ext4-DAX and xfs-DAX try to preserve contiguity of free-space but not its alignment!

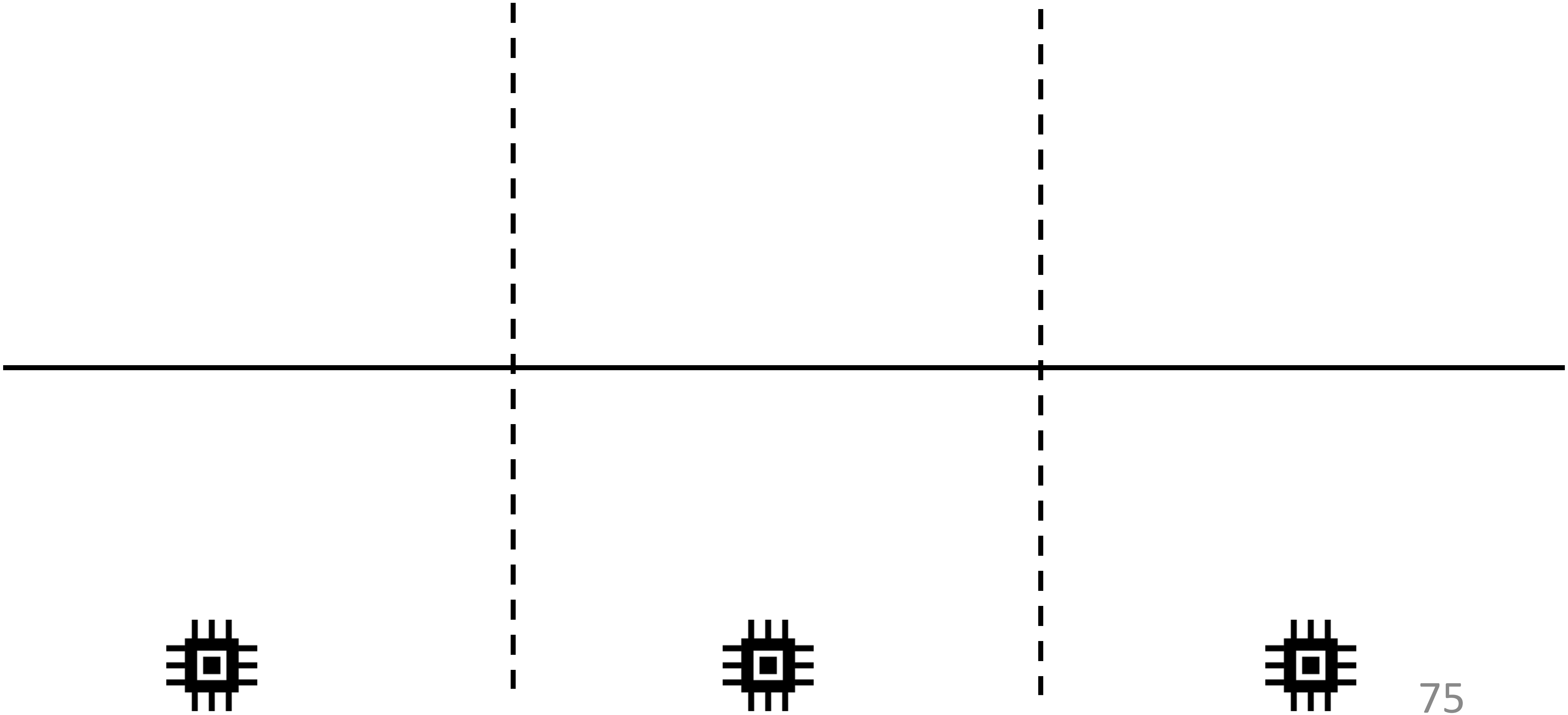
Hugepage preservation with age affects on-disk layout, allocation policies, crash consistency and concurrency.

Per-inode log of NOVA and copy-on-write of data fragments free-space

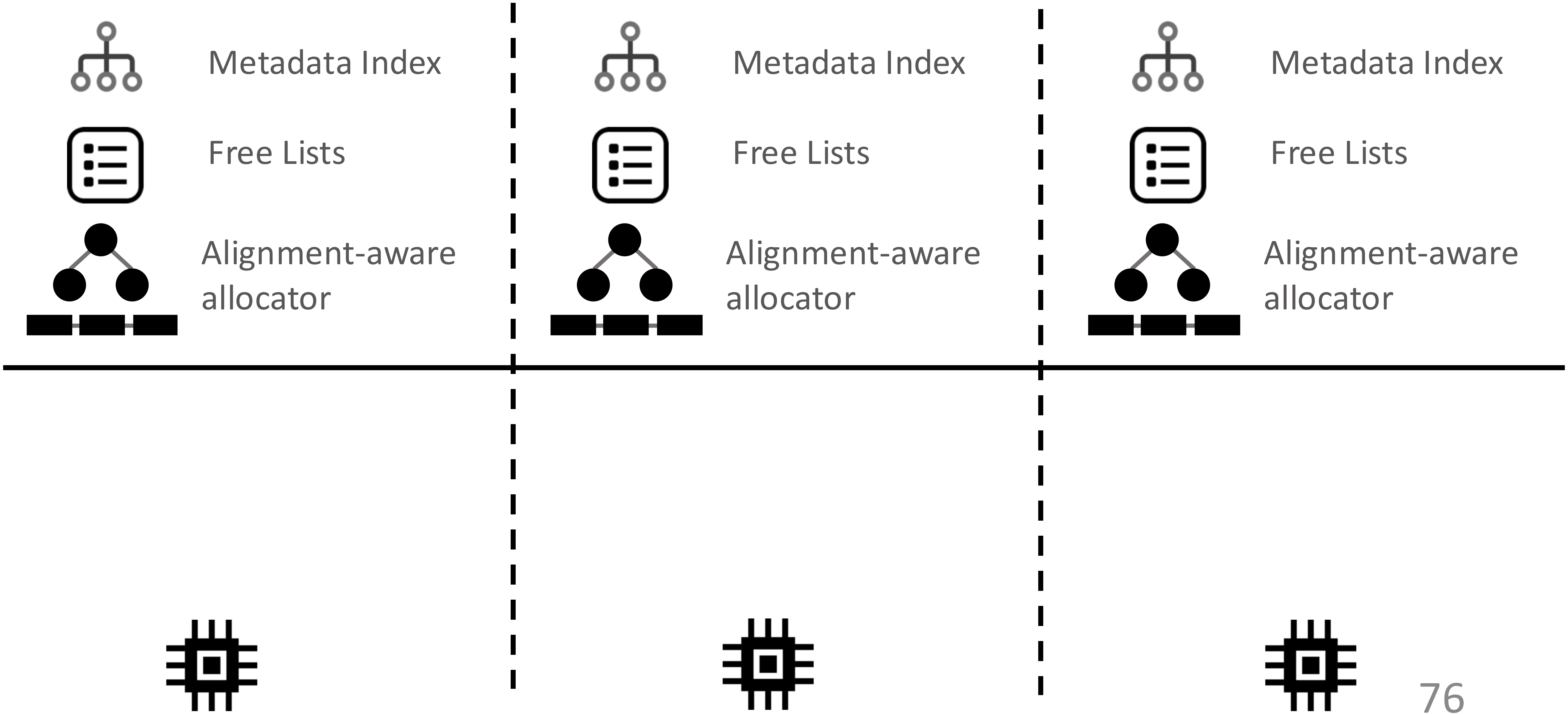
Per-process log of Strata fragments files

High-level Design & On-PM Layout

High-level Design & On-PM Layout



High-level Design & On-PM Layout



Metadata Index

Free Lists

Alignment-aware allocator

Metadata Index

Free Lists

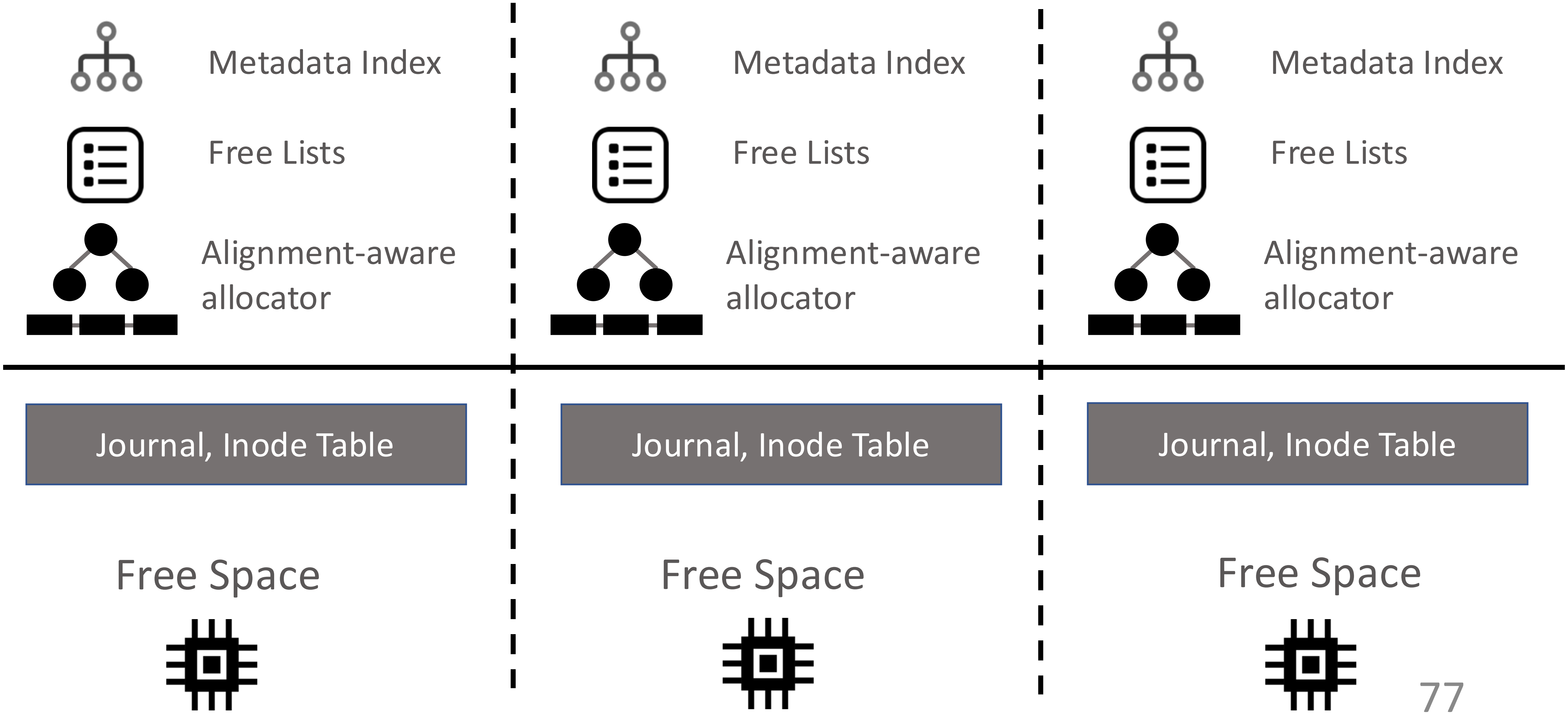
Alignment-aware allocator

Metadata Index

Free Lists

Alignment-aware allocator

High-level Design & On-PM Layout



Metadata Index

Free Lists

Alignment-aware allocator

Journal, Inode Table

Free Space

Metadata Index

Free Lists

Alignment-aware allocator

Journal, Inode Table

Free Space

Metadata Index

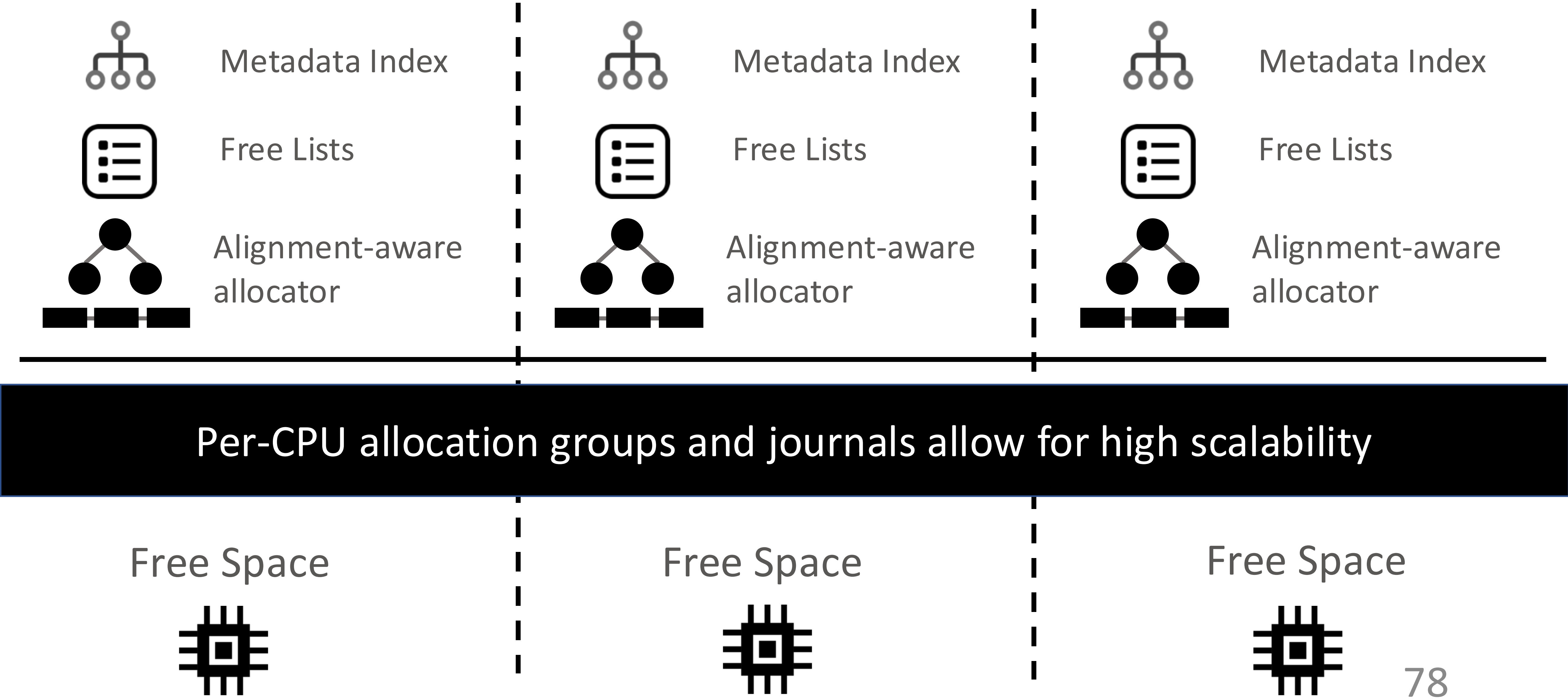
Free Lists

Alignment-aware allocator

Journal, Inode Table

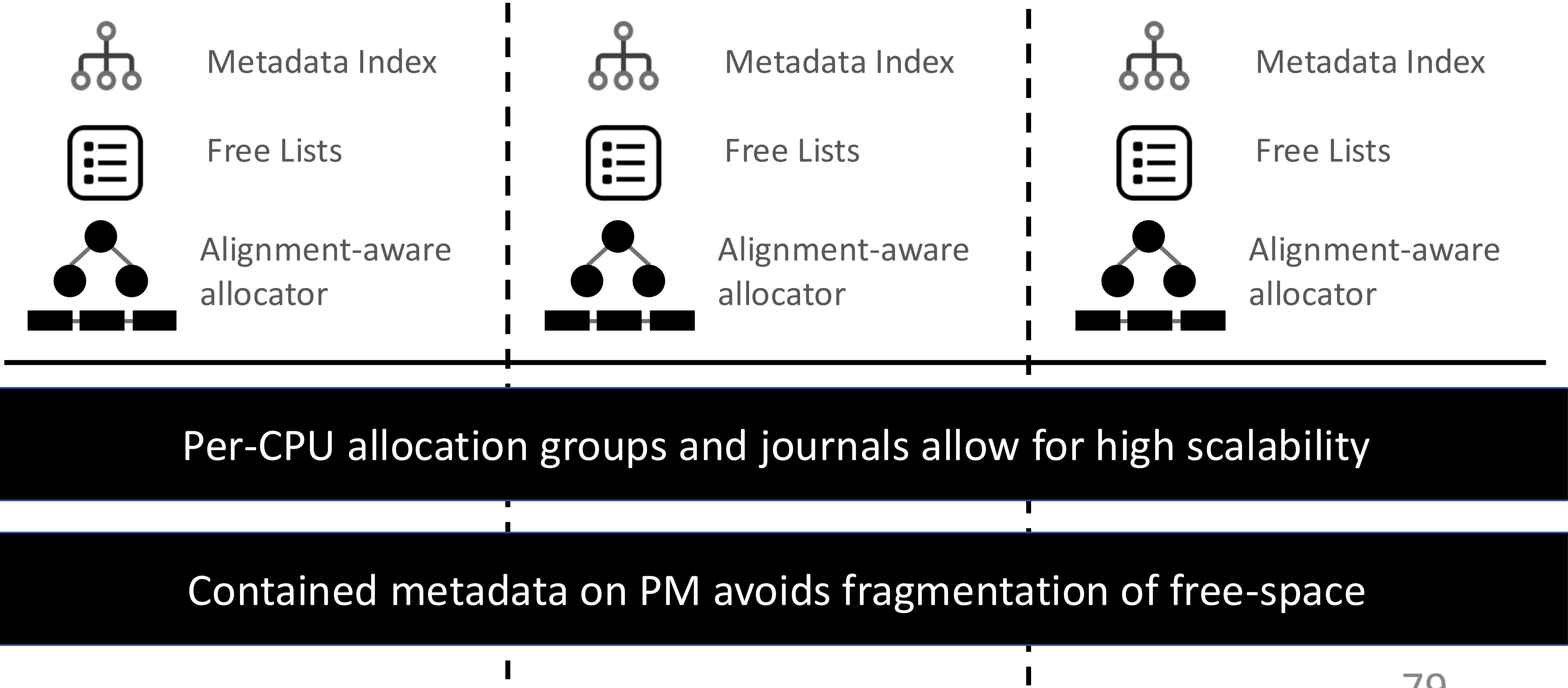
Free Space

High-level Design & On-PM Layout



Per-CPU allocation groups and journals allow for high scalability

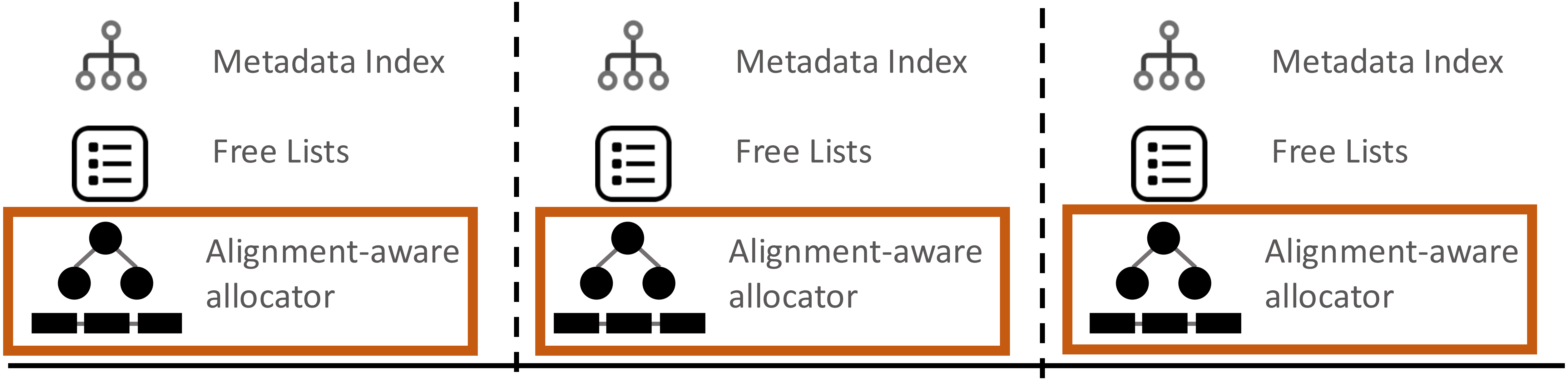
High-level Design & On-PM Layout


Metadata Index
Free Lists
Alignment-aware allocator
Metadata Index
Free Lists
Alignment-aware allocator
Metadata Index
Free Lists
Alignment-aware allocator

Per-CPU allocation groups and journals allow for high scalability

Contained metadata on PM avoids fragmentation of free-space

High-level Design & On-PM Layout



Metadata Index

Free Lists

Alignment-aware allocator

Metadata Index

Free Lists

Alignment-aware allocator

Metadata Index

Free Lists

Alignment-aware allocator

Per-CPU allocation groups and journals allow for high scalability

Contained metadata on PM avoids fragmentation of free-space

Impact of Aging

NOVA:

Per-inode log leads to fragmentation

Copy-on-write for data consistency leads to fragmentation

ext4-DAX:

Suboptimal allocation decisions only consider contiguity but not alignment of free space

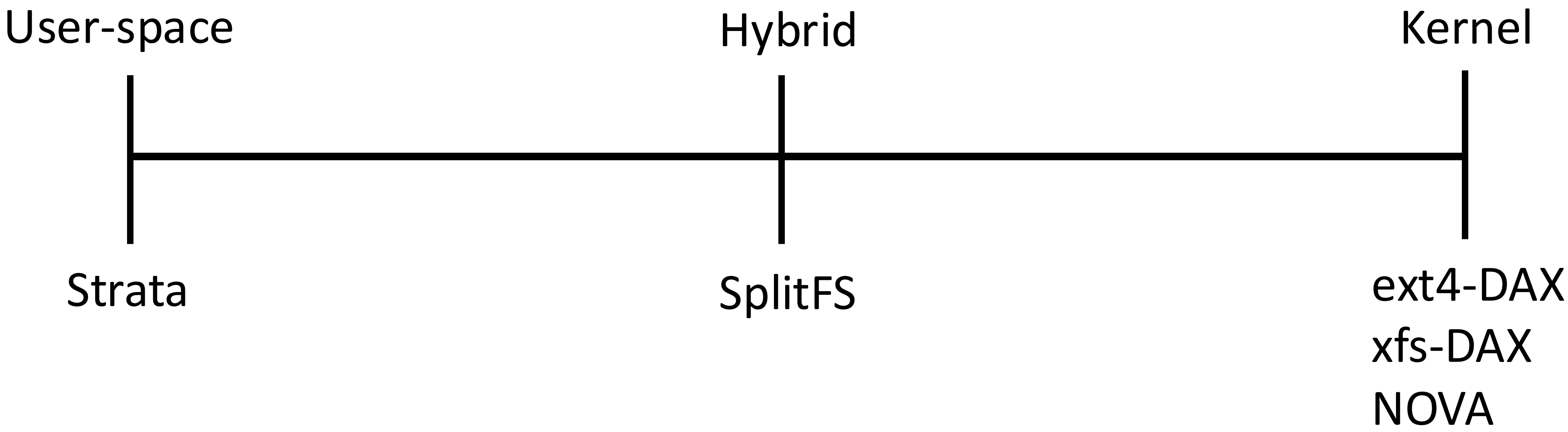
Alignment-aware allocation policy

Goals:

Allocate large files in hugepage-aligned extents

Preserve hugepage-aligned extents in the presence of aging

Persistent Memory file systems



User-space

Hybrid

Kernel

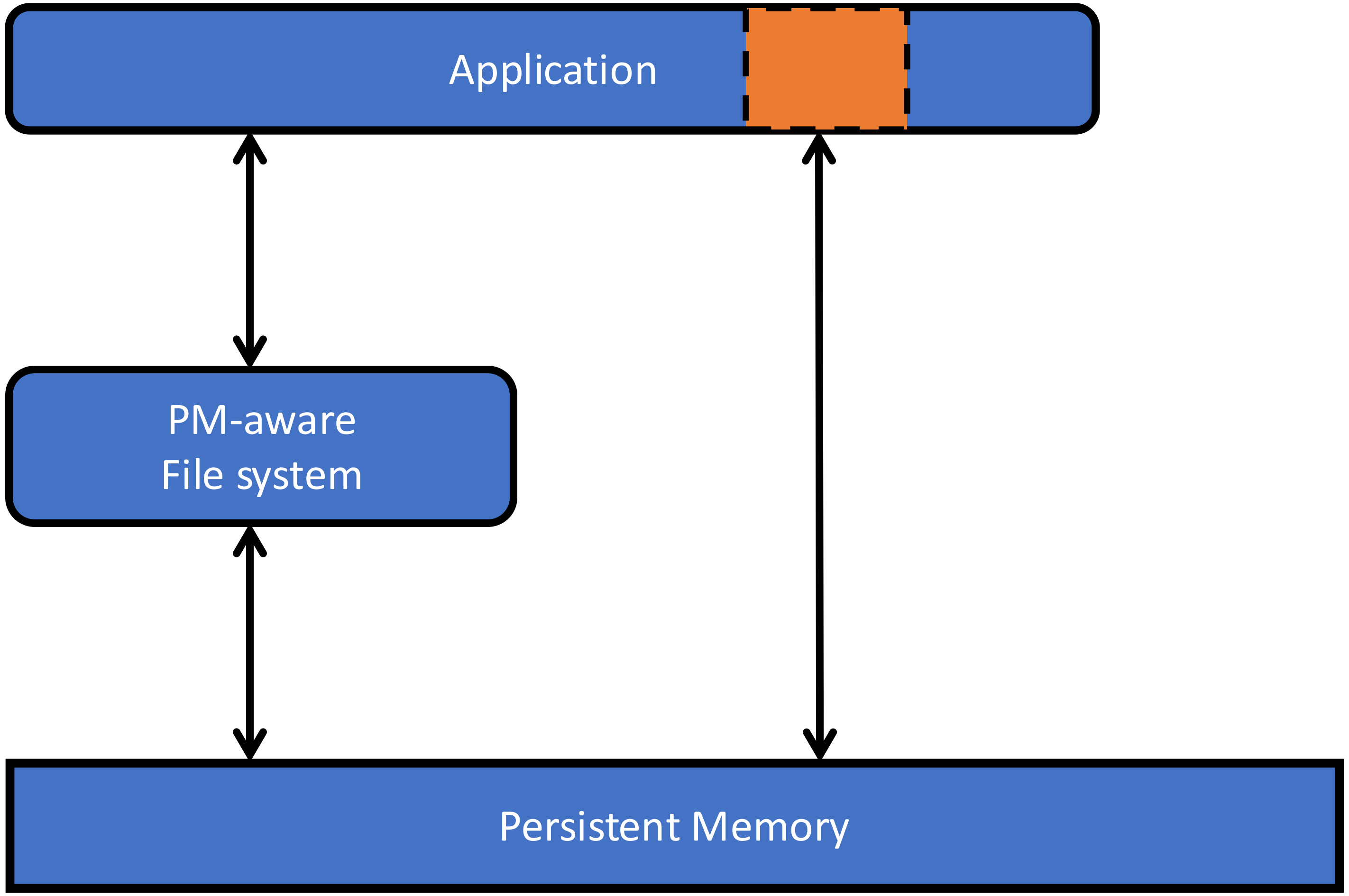
Strata

SplitFS

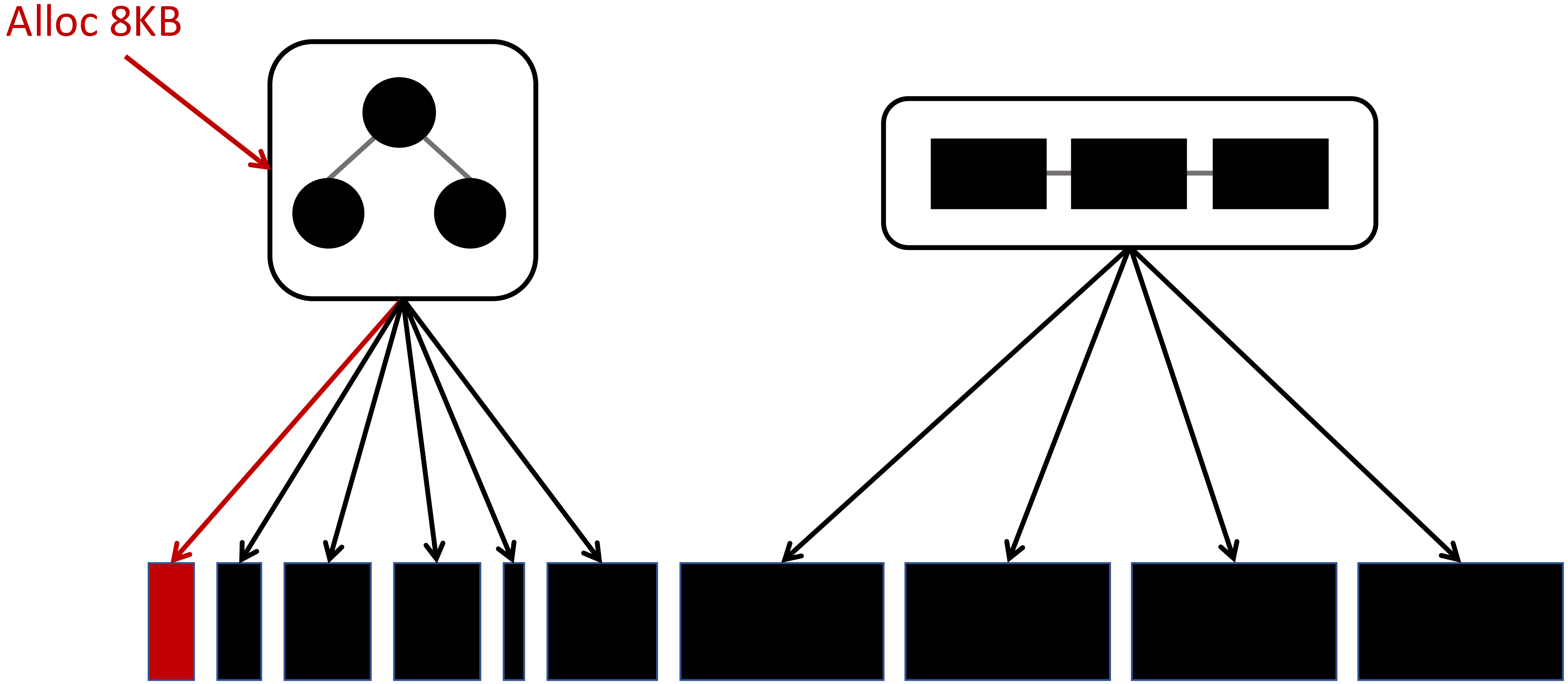
ext4-DAX
xfs-DAX
NOVA

Existing file systems degrade in performance when aged

Memory-mapped Applications



Alignment-aware allocation policy



Small allocations / files do not fragment free space, and are deliberately allocated in holes