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

Authors: Rohan Kadekodi, Saurabh Kadekodi, SoujanyaPonnapalli, Harshad Shirwadkar, Gregory R. Ganger, AasheeshKolli, Vijay Chidambaram

Presented By :Rajendra Sahu

References

SOSP'21

DOI: https://dl.acm.org/doi/10.1145/3477132.3483567

Slides : https://www.cs.utexas.edu/~vijay/papers/winefs-sosp21-slides.pdf

Github: https://github.com/utsaslab/WineFS

UT Systems and Storage Lab: https://utsaslab.github.io/

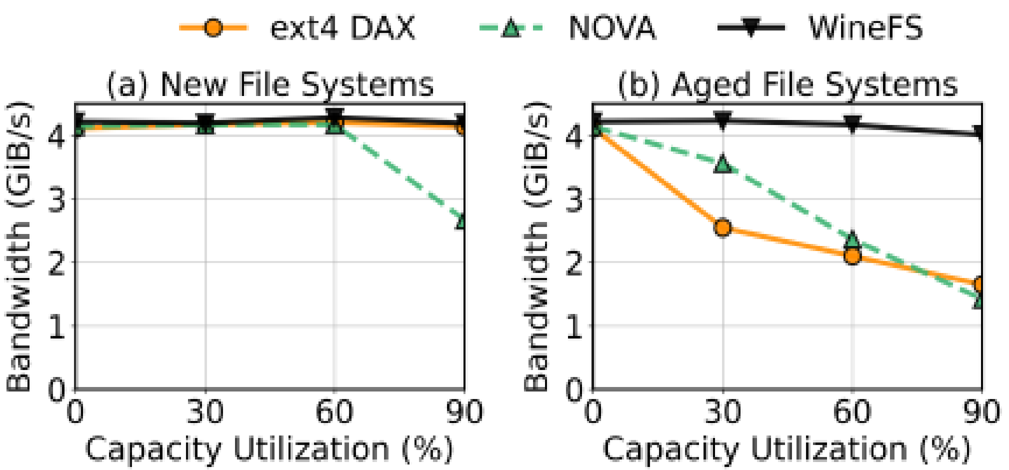
The same lab who produced SplitFS.

Agenda

- *Introduce the problem statement*
- *Background & Motivation*
- *Implementation of WineFS*
- *Evaluation*
- *Previous work comparison*

Problem Statement

- Modern PM file systems perform well when disk is fresh & newly created.
- But their memory-mapped performance degrades over time.



Write bandwidth to memory mapped files for three PM FSs.

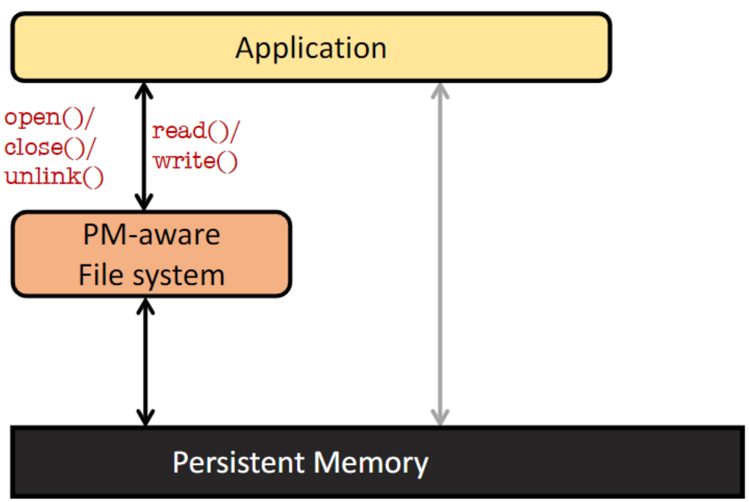
Agenda

- *Introduce the problem statement*
- *Background & Motivation*
- *Implementation of WineFS*
- *Evaluation*
- *Previous work comparison*

Ways to access PM

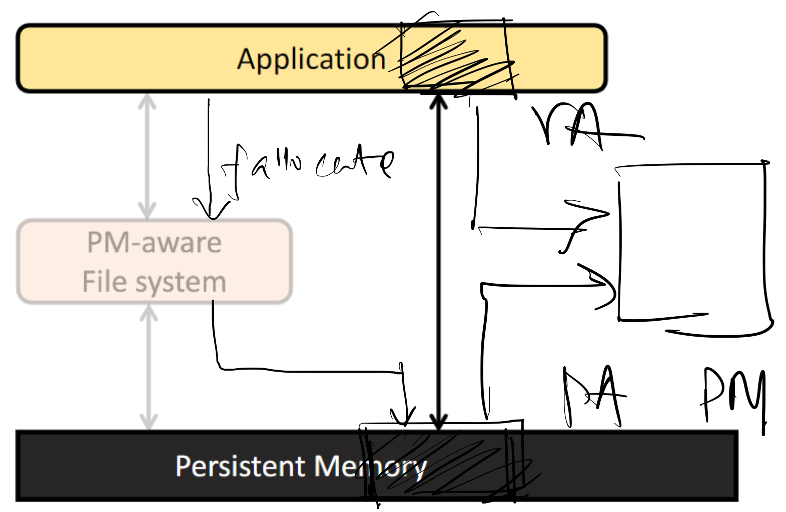
Writing sequentially to 1GB file is 2x faster using memory mapped files compared to system calls.

POSIX system-call applications



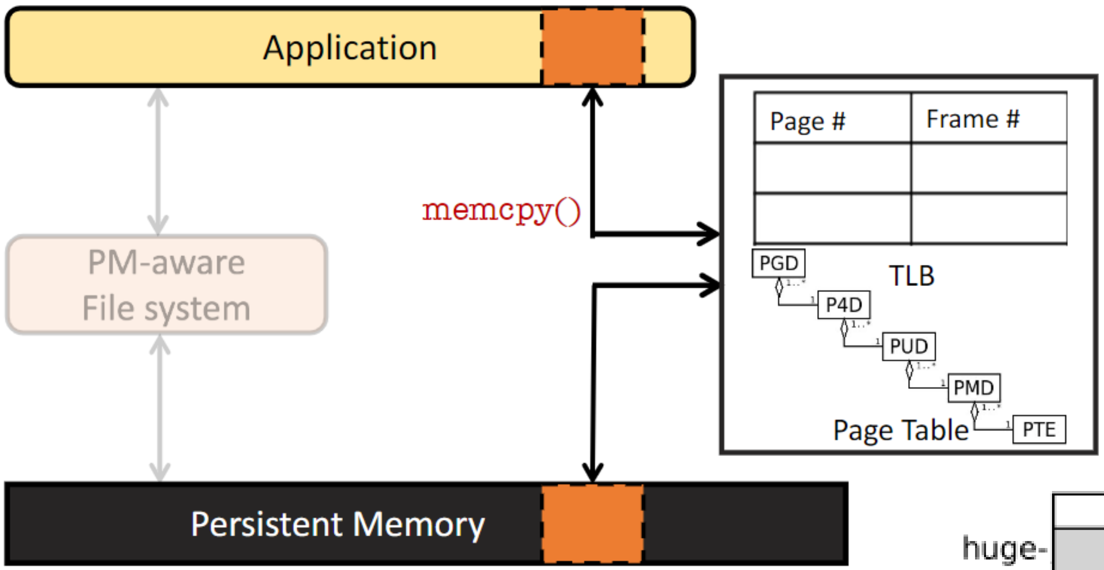
Legacy applications that uses POSIX system call which goes into the kernel. Not the most efficient way to access

Memory-mapped Applications

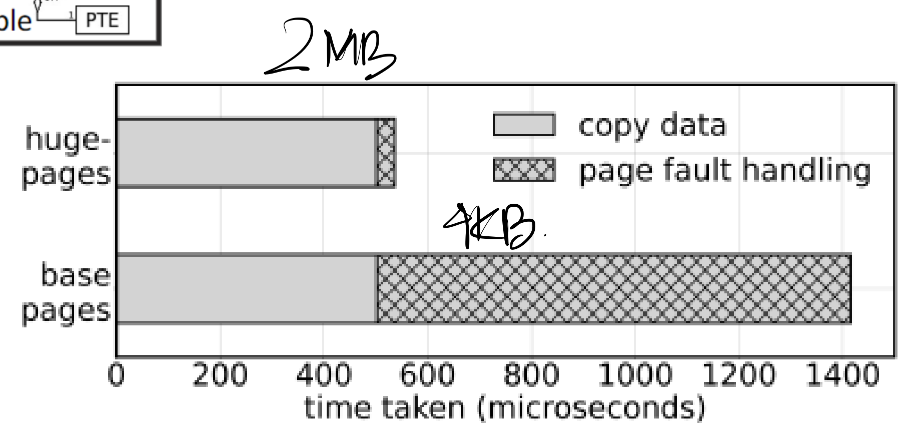


To leverage the low latency of PM, the user space application memory map the PM into the user space and makes load stores access. (DAX) Bypasses kernel

Basepages vs Hugepages

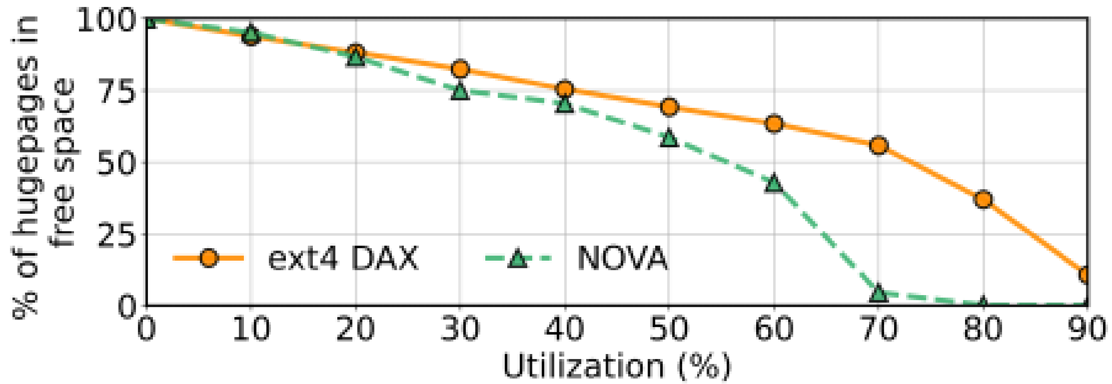


Performance of mem-mapped applications depends on page faults and TLB misses.



Time taken to memory-map and write a 2 MB file with and without hugepages

Aging Destroy Hugepages



- Aging disturbs the alignment & contiguity required for hugepages (2MB)
- Memmap applications performance take a hit.

- No impact of aging on performance of applications accessing PM through POSIX calls
- No Hugepage vs Basepage tradeoff incase of POSIX.

POSIX system call read/write = 25-30 ms
page fault handling time = (1-2 us)

More Motivation

- A background defragmentation utility is very costly on PM’s bandwidth. (Proactive over reactive)
- Allocating hugepages always will lead to internal fragmentation. (Take hybrid approach)
- Making changes to existing FS not feasible.

Agenda

- *Introduce the problem statement*
- *Background & Motivation*
- *Implementation of WineFS*
- *Evaluation*
- *Previous work comparison*

TEXAS A&M UNIVERSITY

WineFS Goals

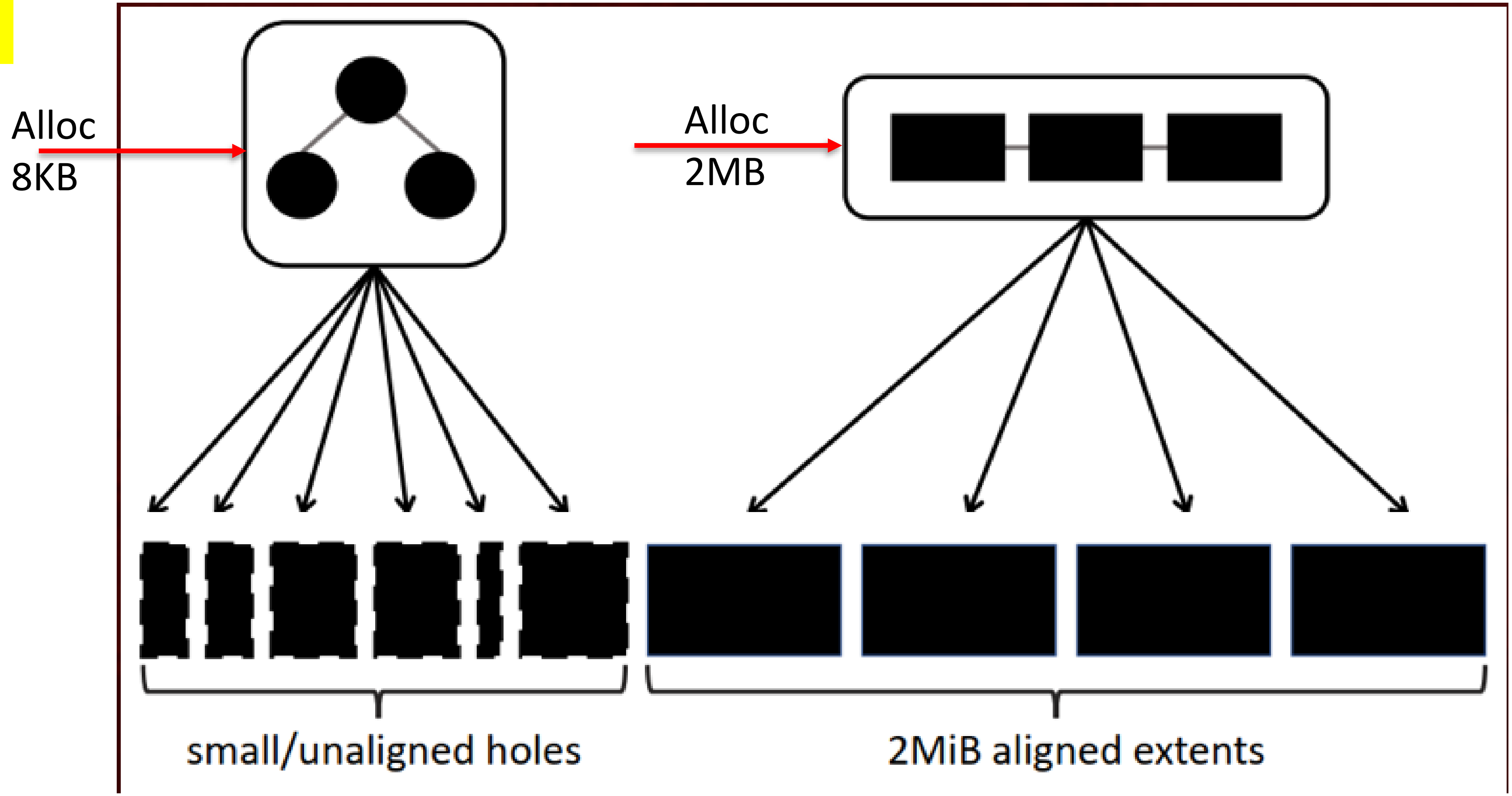
- Should be POSIX compliant.
- Hugepage alignment & contiguity must be preserved for memory mapped files.
- Must not sacrifice performance of POSIX system call applications.
- Mustn't sacrifice performance when FS is new
- Should preserve hugepages wherever possible.(especially when aged)

Design Choices

Hugepage Awareness

- Novel alignment aware allocator:
  hugepages -> aligned extents : basepages -> unaligned holes
- Journaling for crash consistency.
- Metadata structures and journals updated in-place.
- Per-CPU metadata structures for concurrency
- Hybrid Data Consistency Mechanism:
  data journaling for atomic update to aligned extents : copy-on-write for atomic update to unaligned holes.

Alignment Aware Allocator



RB Tree Free List

Linked List Free List

Alignment Aware Allocator

- Larger allocations are broken down to multiple of 2MB allocation requests
- Large files of mem-map apps are placed in aligned 2MB extents, small files of POSIX apps are placed in unaligned holes.
  Ext4-DAX & xfs-DAX preserve contiguity of free space but not alignment
- Hugepages are aggressively reclaimed on deallocations.

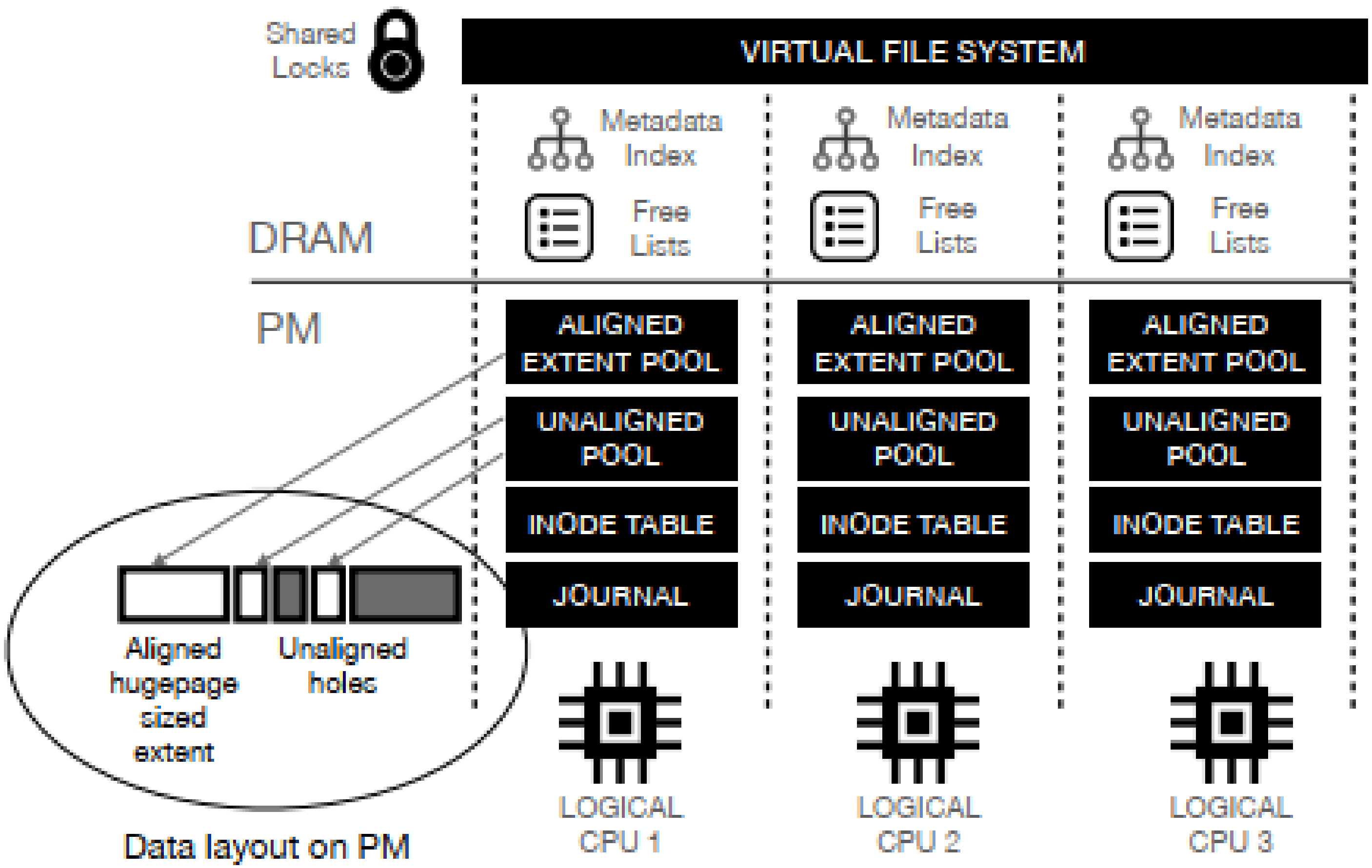
Concurrency -> Scalable (Concurrent but fragmentation)

- Per-file log of NOVA fragments space.
- Per-process log of Strata wastes aligned pool.
- PMFS, ext-4 DAX uses single journal. (not concurrent)
- Journal, inode table, aligned & unaligned pools per logical CPU. (concurrency , less fragmentation)

Contained Fragmentation

- Metadata structures are small-> fragmentation
- Assigns dedicated location on PM.
- In-place update & recycling of space.

WineFS Architecture



Crash Consistency

- Journaling creates 2 writes– journal & in-place but preserves data layout.
- Logging creates 1 write but grbge collection disturbs data layout. (NOVA,Strata)
- WineFS chooses journaling (not entirely).

Hybrid Data Consistency Mechanism

- Journaling for aligned extents.
- Copyonwrite(Logging) for unaligned extents.
- Somewhat compensates the extra writes.

Design Choices

Ensure good performance for POSIX apps

- Fine grained journaling
- DRAM metadata indexes to accelerate directory lookups

(RBL trees to traverse directory
entries & inode free lists.
metadata indexes helps in fast
metadata operation.

Agenda

- *Introduce the problem statement*
- *Background & Motivation*
- *Implementation of WineFS*
- *Evaluation & Previous work*

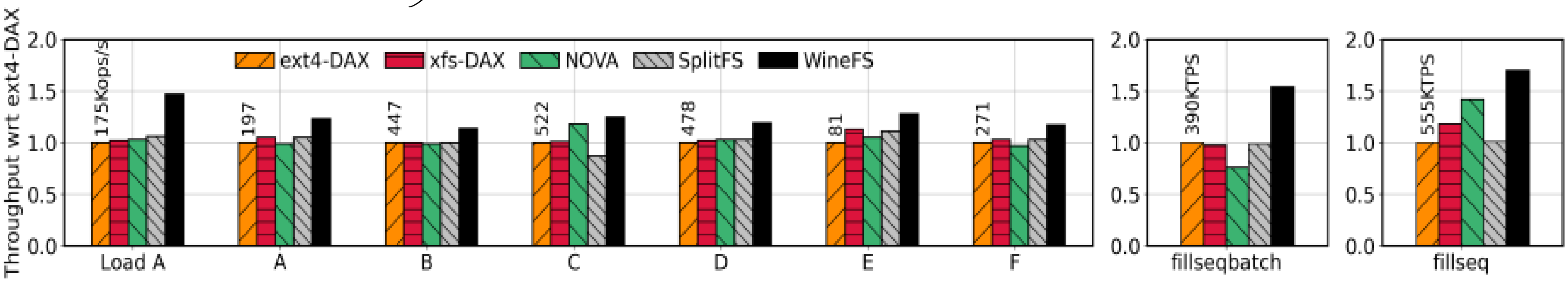
Evaluation

- 500 GB partition of Intel Optane Memory
- 28 cores, 112 threads, 32 MB LLC
- Geriatrix Aging Setup (Agrawal Profile)
  165TB write activity, creating & deletion of files (small & large(56%))
- File systems compared : ext4-DAX, xfs-DAX, NOVA, Strata, SplitFS

Sowrabh
Kadekodi

Memory-mapped Application

YCSB on RocksDB



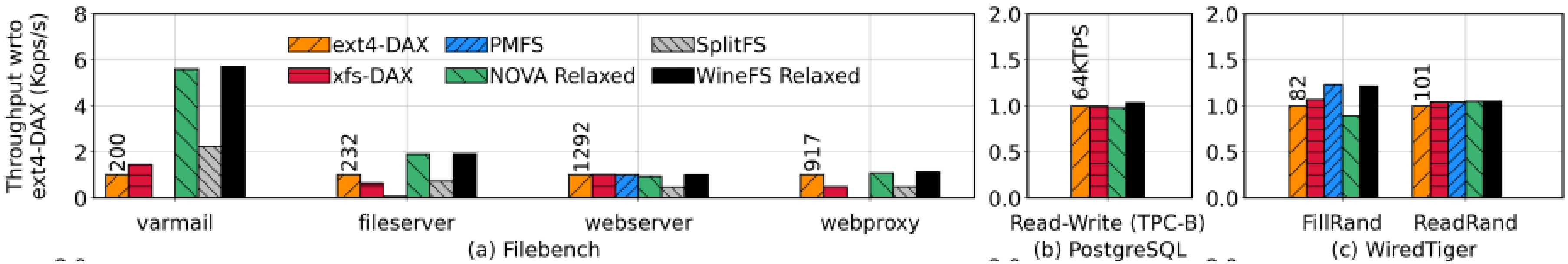
(a) RocksDB (MMAP reads and writes) (b) LMDB (c) PmemKV

All configurations running on 75% capacity utilization. Aged through Geriatrix

	YCSB							LMDB	PmemKV
	Load A	A	B	C	D	E	F	fillseqbatch	fillseq
WineFS	**1.59 Mn**	**3.83 Mn**	**3.83 Mn**	**1.34 Mn**	**1.36 Mn**	**0.48 Mn**	**4.85 Mn**	**0.06 Mn**	**0.01 Mn**
ext4-DAX	11.85×	17.86×	6.20×	10.60×	18.36×	45.38×	14.57×	205×	292×
xfs-DAX	28.26×	23.24×	7.04×	11.21×	20.27×	56.38×	17.70×	280×	455×
SplitFS	16.46×	20.52×	6.73×	10.93×	19.94×	50.58×	16.15×	208×	296×
NOVA	32.03×	1.57×	7.65×	1.05×	23.23×	1.15×	22.30×	261×	399×

No of page faults comparison; WineFS faults the least

System Call Application



All of the performance of a clean FS setup as aging has no impact on system call path

Scalability

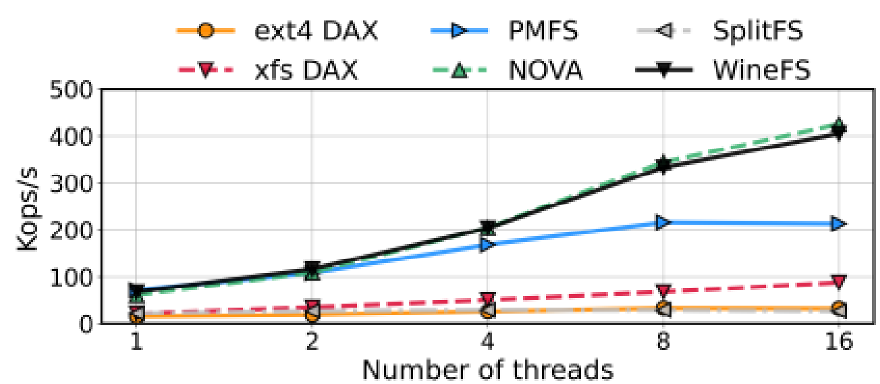


Figure 10. Microbenchmark: Scalability. WineFS throughput scales with increasing #threads on metadata-heavy workloads.

Previous work Comparison

- Hugepage Support : Prior work (Intel PMDK) suggests to enable hugepages always on ex4-DAX & xfs-DAX -> space amplification
  NOVA needs allocation request to be multiples of 2MB.
- NOVA, Strata per file logging -> fragments space; PMFS, ext4-DAX single journal-> no concurrency
- Prior work (Aging research by the same group) only studies aging on emulated PM.

Personal Critique

- No strong enough data on aging doesn't impact POSIX applications.
- What if POSIX applications want to use hugepages ? Paper doesn't talk on this.
- Results might be a little biased towards Agrawal profile of Geriatrix.

Thank You!

Any questions ?