

---

# An Application-Local Library File System for Persistent Memory

**Keiichi Matsuzawa**

Hitachi, Ltd.

**Takahiro Shinagawa**

The University of Tokyo

# Summary

---

A file system for persistent memory (PM) to improve performance

- Fully implemented as a library in user-space
- Files are shared between specific processes
- POSIX-compliant for backward compatibility

# Observation

- System-wide state management is not needed
- Modern applications use their own dedicated directory
  - Not accessed by other applications or system
  - Often accessed by huge number of metadata operations and small I/Os

 mongoDB® Data files of mongoDB

```
/var/lib/mongodb# ls -l
total 152
-rw----- 1 mongodb mongodb 48 Dec 2 20:44 WiredTiger
-rw----- 1 mongodb mongodb 21 Dec 2 20:44 WiredTiger.lock
-rw----- 1 mongodb mongodb 1036 Dec 2 20:45 WiredTiger.turtle
-rw----- 1 mongodb mongodb 24576 Dec 2 20:45 WiredTiger.wt
-rw----- 1 mongodb mongodb 4096 Dec 2 20:44 WiredTigerLAS.wt
-rw----- 1 mongodb mongodb 16384 Dec 2 20:45 _mdb_catalog.wt
```

Only mongod can access these files

 Build directory of glibc pkg.

```
~/glibc-2.31/build-tree# find | head -30
.
./amd64-libc
./amd64-libc/configparms
./amd64-libc/config.log
./amd64-libc/support
./amd64-libc/support/stamp.o
./amd64-libc/support/stamp.os
```

Only used for build procedures

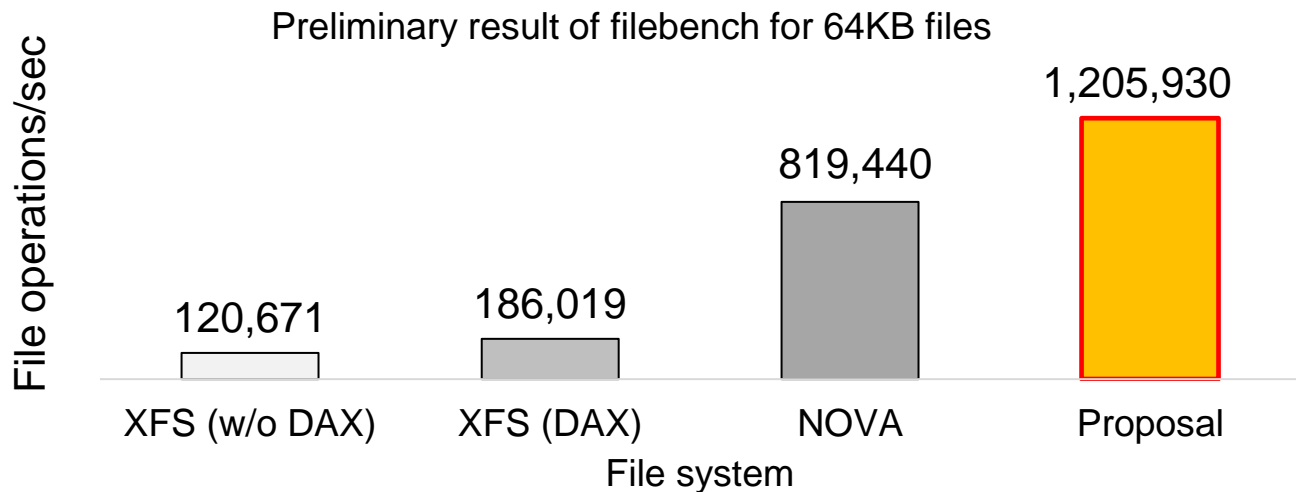
# Solution: Application-local file system

---

- Limited access to files to specific applications for faster operations
  - simplify state management
  - omit security features
- Provided as user-space library file system
  - Share PM among processes and each process access contents on PM directly without task switching
  - Run under glibc to achieve POSIX compliance

# Current Status

- Single-process, multi-thread version is working
  - performs better than kernel-based file systems for small files
- Multi-process support is ongoing
  - Designing data structure and lock strategies



Thank you!

Contact: [keiichi.matsuzawa.kd@hitachi.com](mailto:keiichi.matsuzawa.kd@hitachi.com)