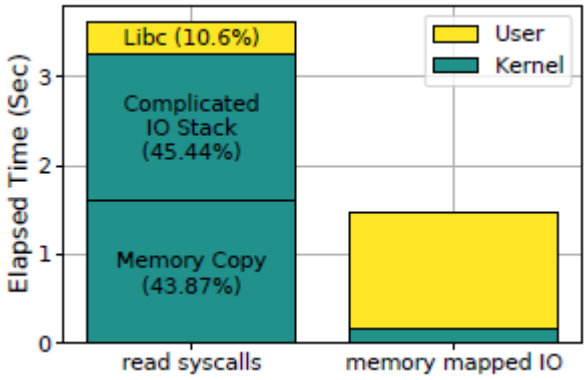
背景

传统的文件系统限制了NVM的性能（software overhead）



目的

充分发挥NVM的高性能

存在的问题

1. 基于kernel的数据访问具有较高的延迟，**mmap io**提供直接NVM访问，能够有效减低kernel开销。

   > 减少了用户空间和page cache之间的数据交换

2. mmap io不提供写数据的原子性，并且为了保证crash-safe，cache line应刷新以确保持久性，并应使用*内存隔离*以为NVM更新提供正确的持久顺序，这往往会带来大量开销，并且难以编程。

3. 现有的一致性保证机制中CoW存在写放大以及TLB-shootdown问题，journaling（logging）的两种方式有不同的适应场景。

   - redo log
     先将数据写入redo log，再将log持久化到目标文件。redo log中记录最新的数据。（适合写）
   - undo log
     先复制目标文件中的数据到undo log中，再对目标文件进行就地更新。目标文件中记录最新的数据。（适合读）

   对于可按字节读写的NVM设备，混合日志可显著减少写放大。

libnvmmio