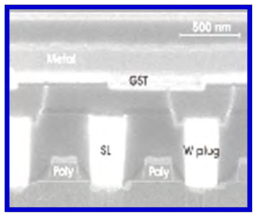
3.背景—持久性内存带来巨大机遇

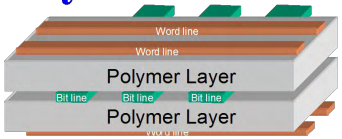
- 低功耗
  - ✓ 零泄露功耗
- 非易失、高可靠性、抗振动
  - ✓ 掉电数据保持
  - ✓ 抵抗软错误与物理振动
- 潜在的高性能
  - ✓ 更大容量高性能片上存储
  - ✓ 更快的外存读写速度

FERAM

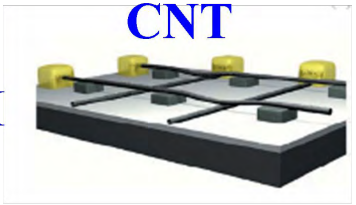
PCM



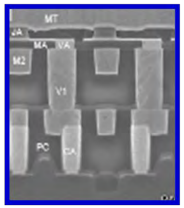
Polymer FeRAM



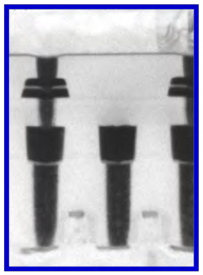
CNT



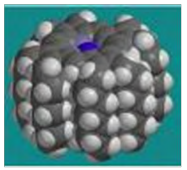
MRAM



MOx-RRAM

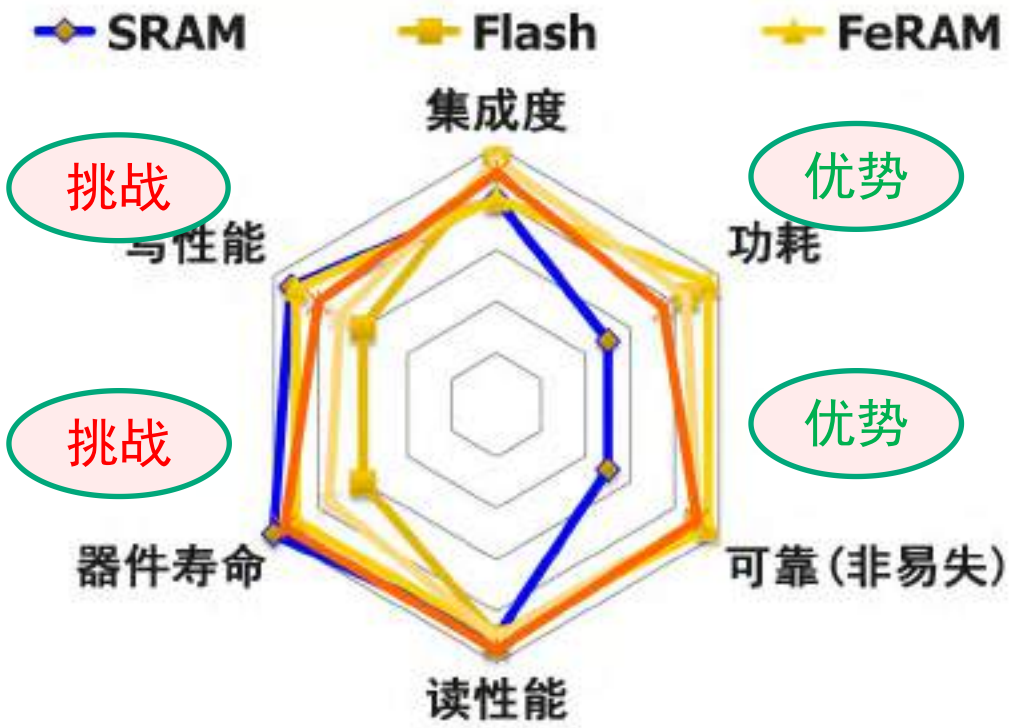


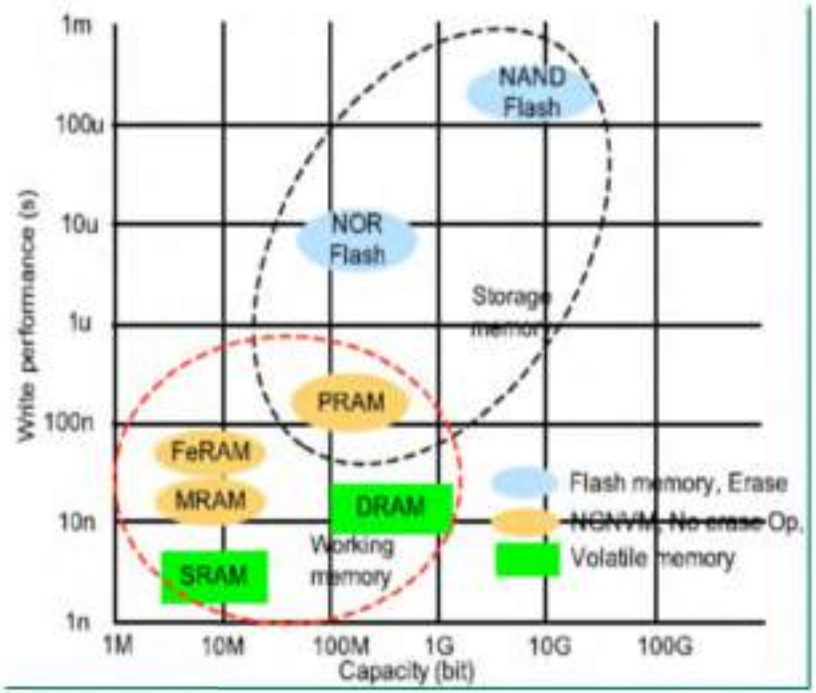
Molecular



解决传统存储系统面临的挑战，**新型存储器件提供了重要途径和机遇。**在Gartner发布的2011年十大战略技术中，**SCM是唯一位列其中的存储技术！**

3.背景—持久性内存的特点及其影响



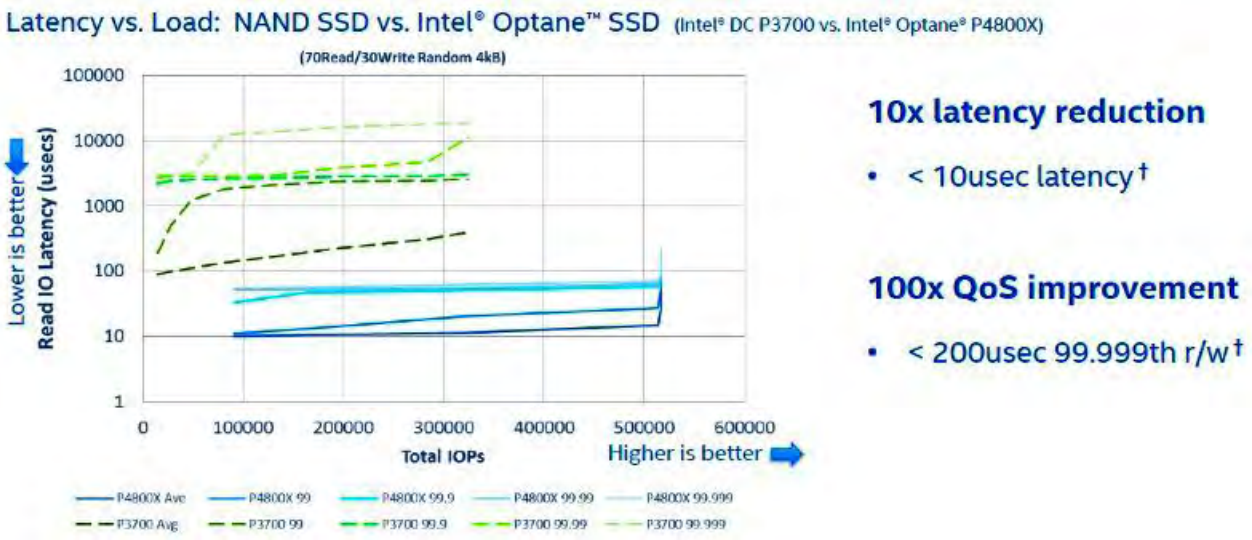


传统存储架构、软件及各层次都是针对传统器件设计的，难以发挥新型存储器件的特性；同时新型存储器件本身也有写性能和器件寿命等不足，这些都是需要解决的问题！

3.背景—持久性内存器件研发进展

- Intel & Micron: 3D Xpoint
- 2017 Q1: Optane SSD DC P4800X (PCI-E 3.0/NVMe)
- 2018年(预期)：3D XPoint DIMM板卡

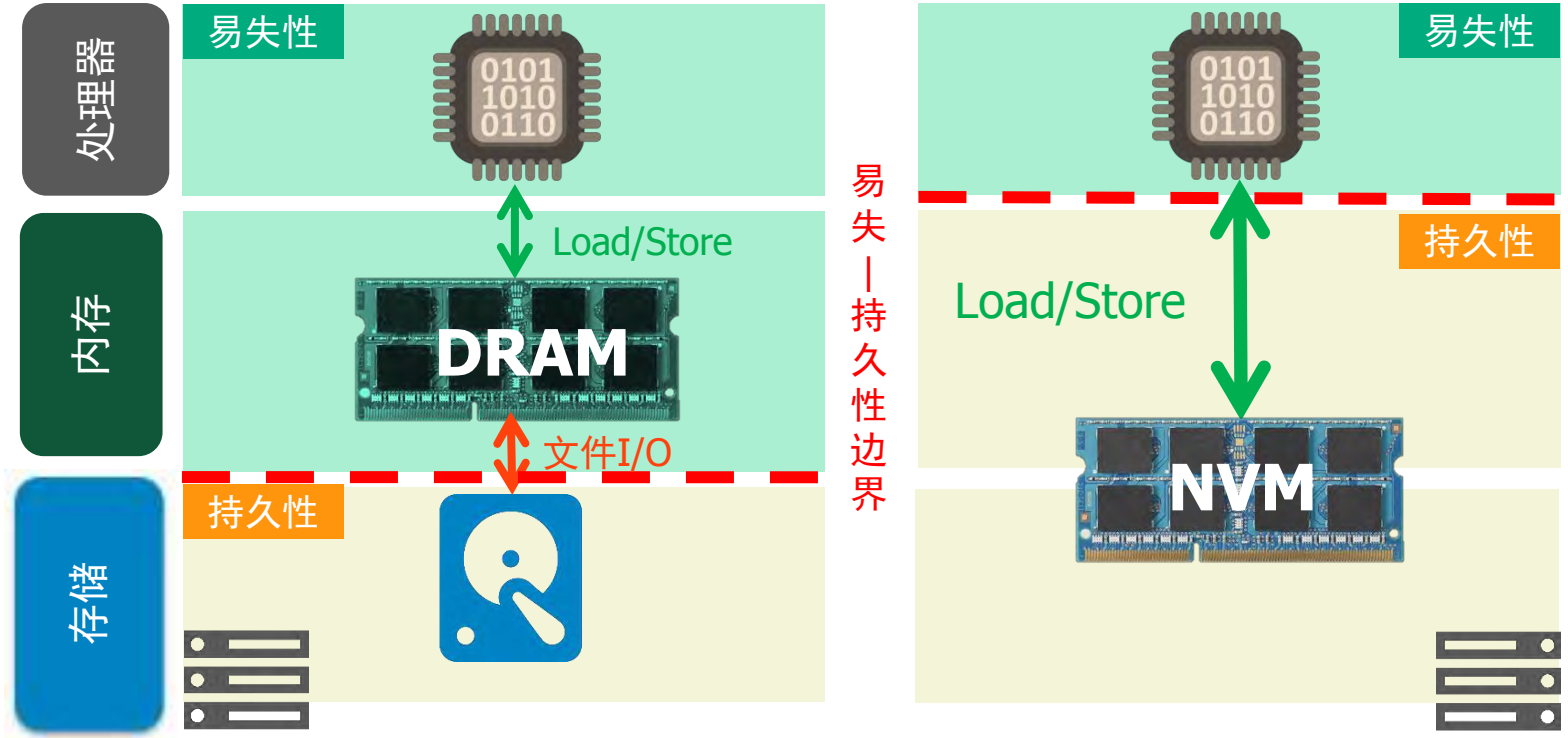
英特尔发布傲腾 ，
基于 技术打造



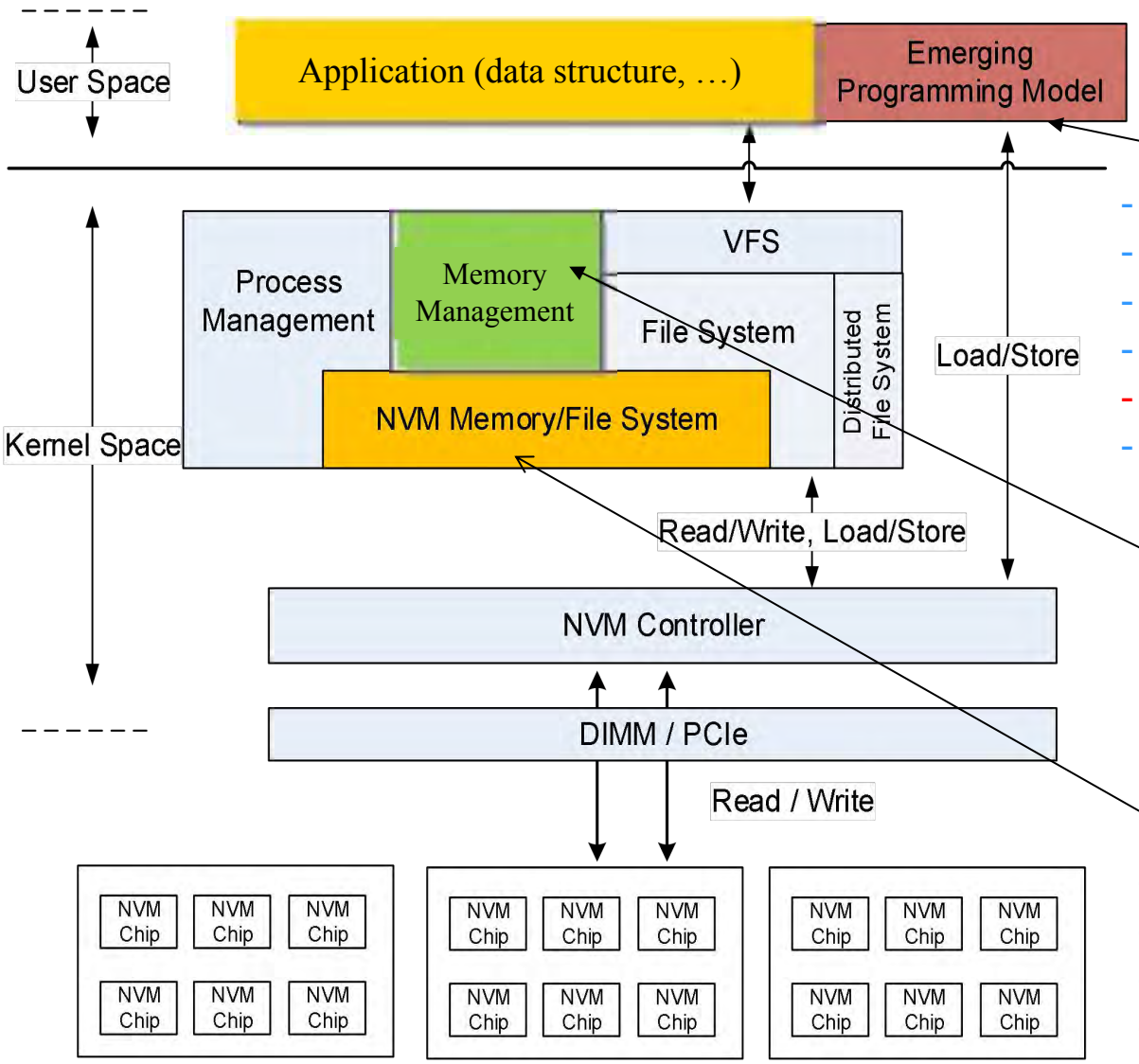
能随着工作负载增加而提供更快的写入速度 （来源：英特尔）

3.背景—持久性内存存储系统结构变化

- 易失性-持久性边界的变化
- 对存储介质的利用方式发生变化
  - ✓ 文件？对象？



3.背景—持久性内存存储系统研究现状



持久性内存编程模型

- SOSP'09
- ASPLOS'11
- ASPLOS'12
- MICRO'13
- ICCD'14
- TOS'14
- ISCA'14
- MSST'15
- ASPLOS'16
- MICRO'16
- Eurosys'17
- ASPLOS'17

持久性内存空间管理

- Eurosys'16
- OOSPLA'16
- ASPLOS'17
- ATC'17

持久性内存文件系统

- SOSP'09
- SC'11
- Eurosys'14
- DATE'17
- FAST'16
- Eurosys'16
- MSST'16

主要内容

一 背景

二 面向大数据的闪存存储系统

三 面向大数据的持久性内存存储系统

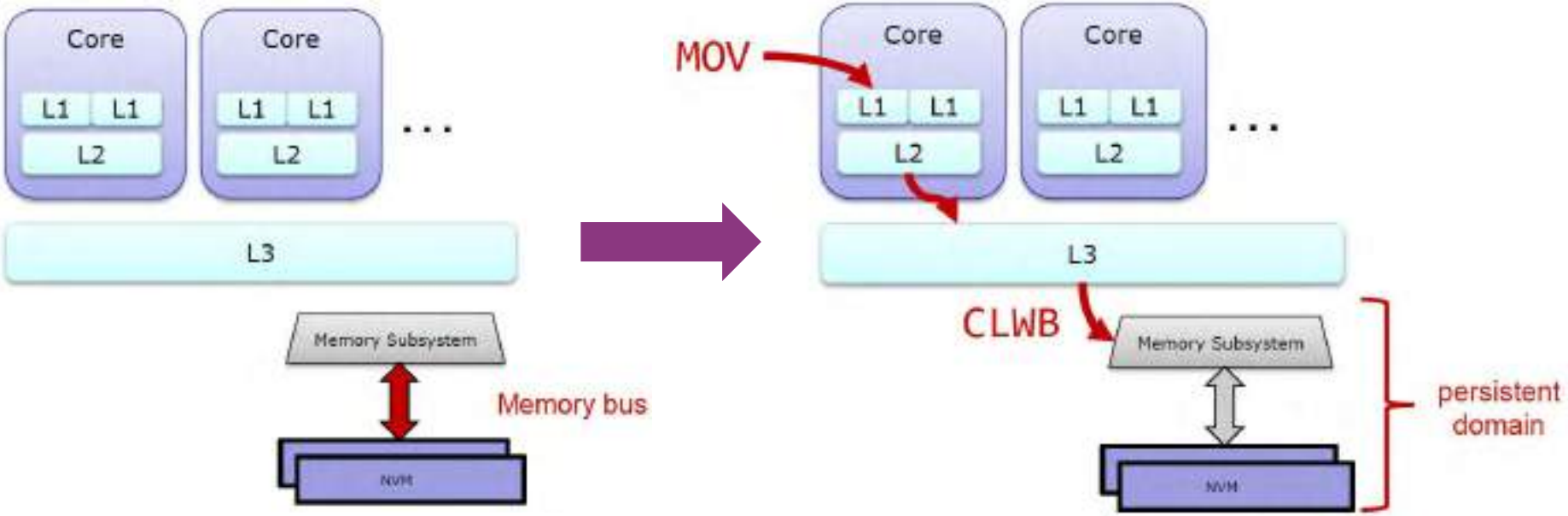
- 3.1 持久性内存编程模型
- 3.2 持久性内存系统软件

四 新型分布式存储系统

五 展望

3.1 持久性内存编程模型面临的挑战（1）

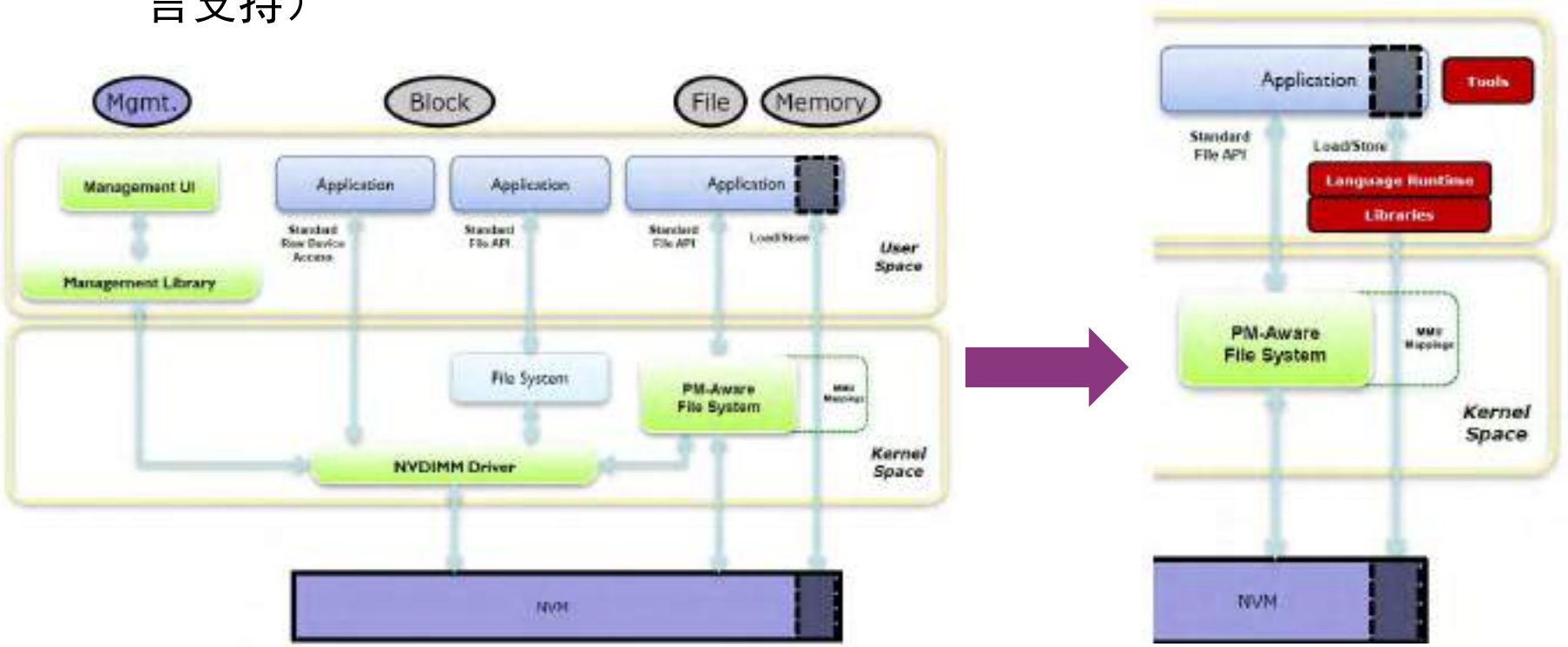
- 新型编程模型涉及的内容：
  - ✓ 软件访问硬件接口：以访问内存的方式访问NVM (cache coherent).
  - ✓ Instruction Set Architecture（ISA）：Store操作数据需从CPU Cache中刷出，全局可见性→数据持久化



Source: Andy Rudoff, Intel Corporation. The SNIA NVM Programming Model: Latest Developments and Challenges.

3.1 持久性内存编程模型面临的挑战（2）

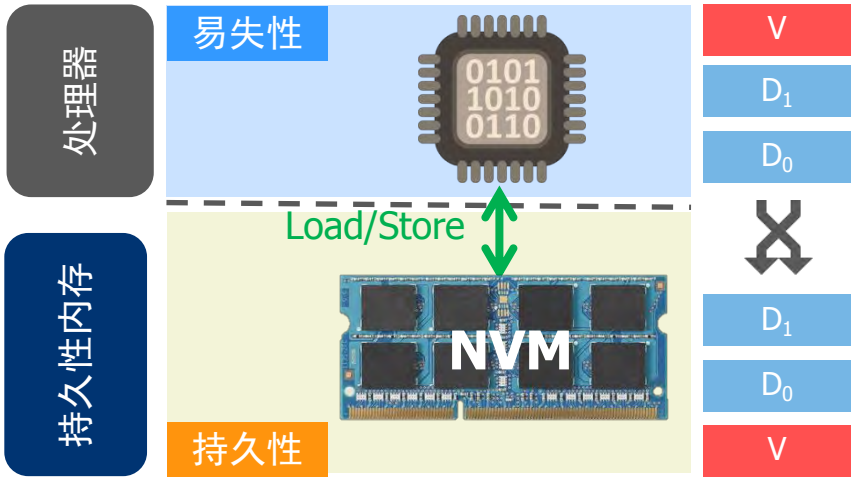
- 新型编程模型涉及的内容：
  - ✓ 暴露给应用程序的方式：
    - SNIA NVM编程模型（Block/File/Memory）
    - Memory-Mapped File：Direct Access（DAX）
  - ✓ 编程复杂度：更安全、更不易出错、熟悉的编程范式（常见编程语言支持）



3.1 持久性内存编程模型面临的挑战（3）

- 新型编程模型面临的问题：
  - ✓ 程序执行空间的数据已经持久化，如何提供应用程序自管理的持久化的功能？
  - ✓ 需要确保程序数据在异常掉电和系统崩溃之后能够被正确地恢复，如何实现低开销的一致性机制？

```
STORE data[0] = 0xFOOD
STORE data[1] = 0xBEEF
STORE valid = 1
```



一致性挑战：持久性顺序约束

- ❑ 必须确保数据保存到NVM才可恢复
- ❑ 高速缓存模式(如write-back)对写操作重排序，破坏了数据的可恢复性
- ❑ 强制缓存刷新(通过使用clflush 等指令)引入高额开销

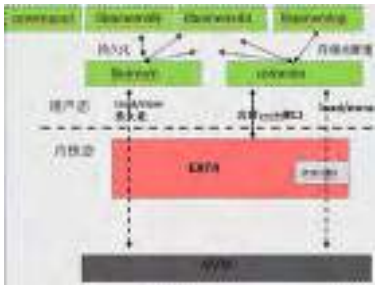
3.1 新型编程模型概述

- 存储层次变革，需要定义新的内存编程模型：
  - ✓ 基于单级持久性存储系统的非易失性编程接口
    - 如：Mnemosyne [ASPLOS'11], NV-Heaps [ASPLOS'11], NVML [Intel], Heapo [TOS'14]
  - ✓ 新型编程模型的一致性开销的优化
    - 如：Epoch [SOSP'09], WSP [ASPLOS'12], Kiln [MICRO'13], Strand Consistency [ISCA'14], LOC [ICCD'14], BPPM [MSST'15], Eager Sync [ASPLOS'16], Sync Ordering / Delegated Ordering [MICRO'16], DUDETM [ASPLOS'17], Kamino-TX [Eurosys'17]

SNIA NVMP

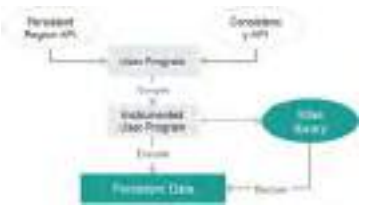
成立时间：2012.6
主导厂商：intel
旨在定义新的内存访问模型
定义软件的行为规范，不定义真正的API

NVML



Intel提供开源库
组件包括：
Libpmemobj
Libpmemblk
Libpmemlog
Libpmem
Libvmem
mempool

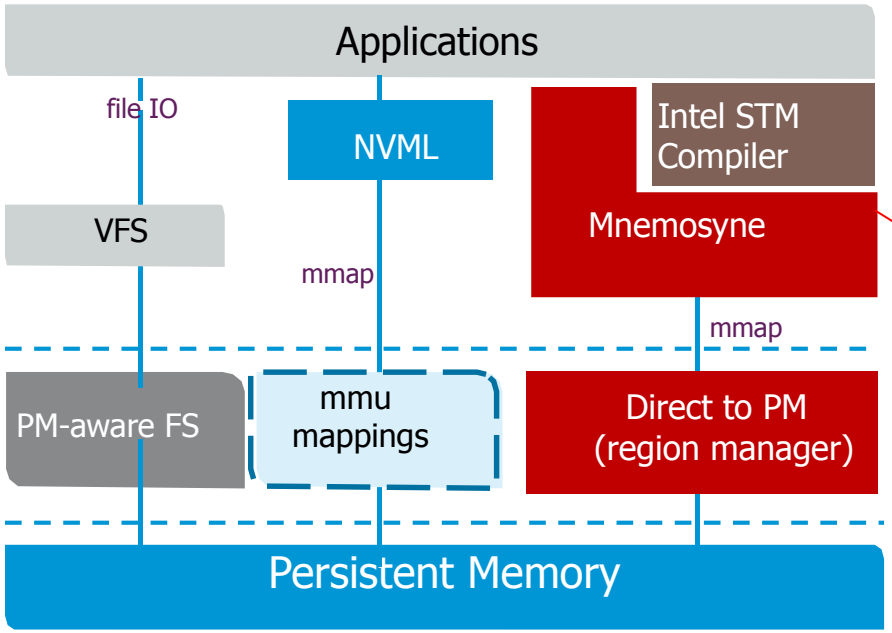
Atlas



C/C++11 __atomic
区分持久性和临时性数据
HP开源

3.1 新型编程模型实现关键技术

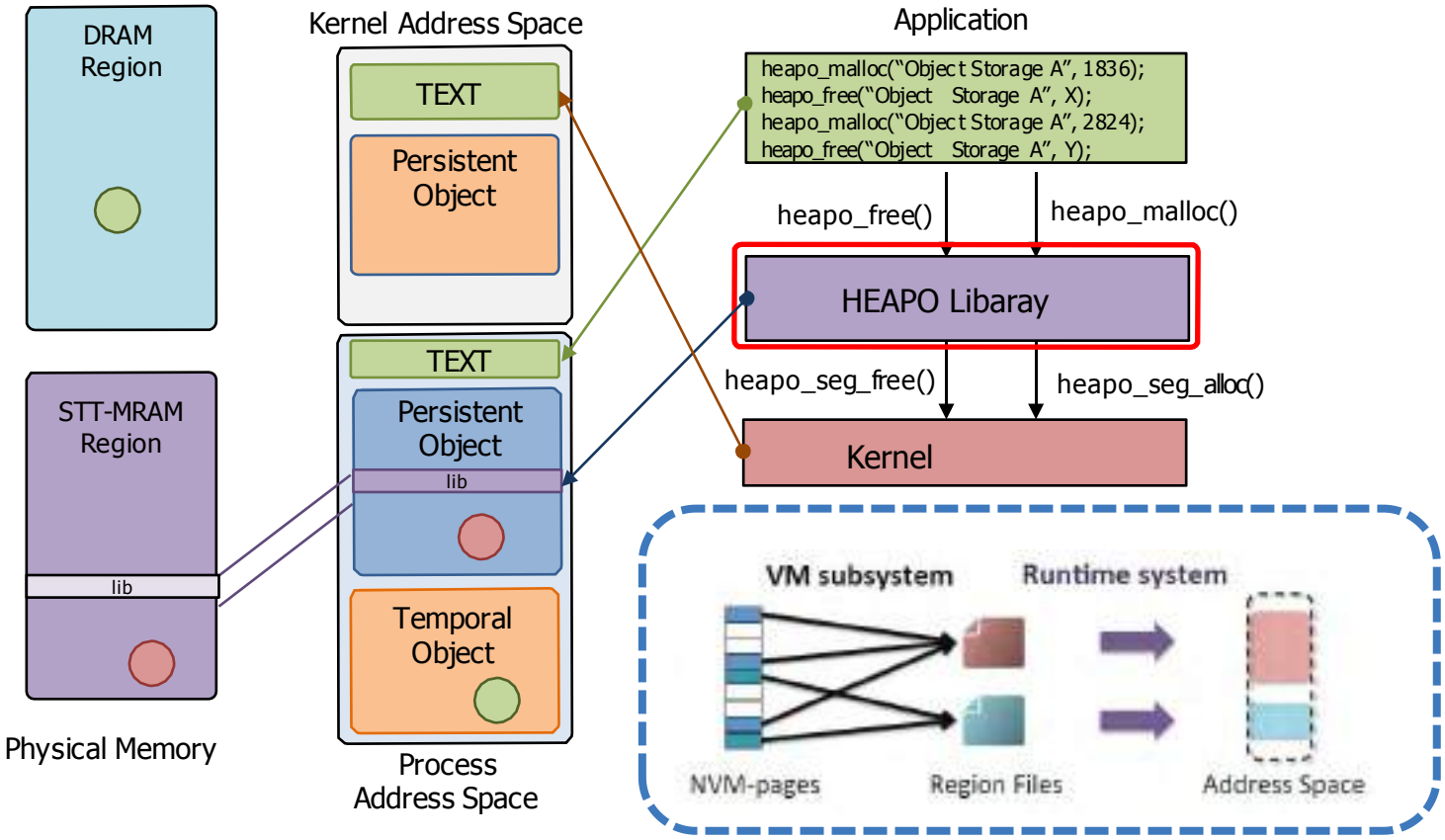
- 基于内存映射文件的持久性堆结构
  - ✓ 通过持久性内存文件系统（如PMFS、EXT4-DAX）和内存映射文件导出load/store 访问接口
    - Mnemosyne [ASPLOS'11]、NV-Heaps [ASPLOS'11]
    - NVM Library [Intel]
  - ✓ 提供持久化事务支持，保证数据的一致性




```
begin_transaction();
stm_store(&R.value,
0xC0F);
stm_store(&R.valid, 1);
commit_transaction();
```

3.1 新型编程模型实现关键技术

- 基于原生堆（Native heap）的持久性堆结构
  - ✓ 内在进程虚拟地址空间中预留一段持久性堆空间
    - 如：Heapo：基于堆的持久性对象存储[TOS'2014]



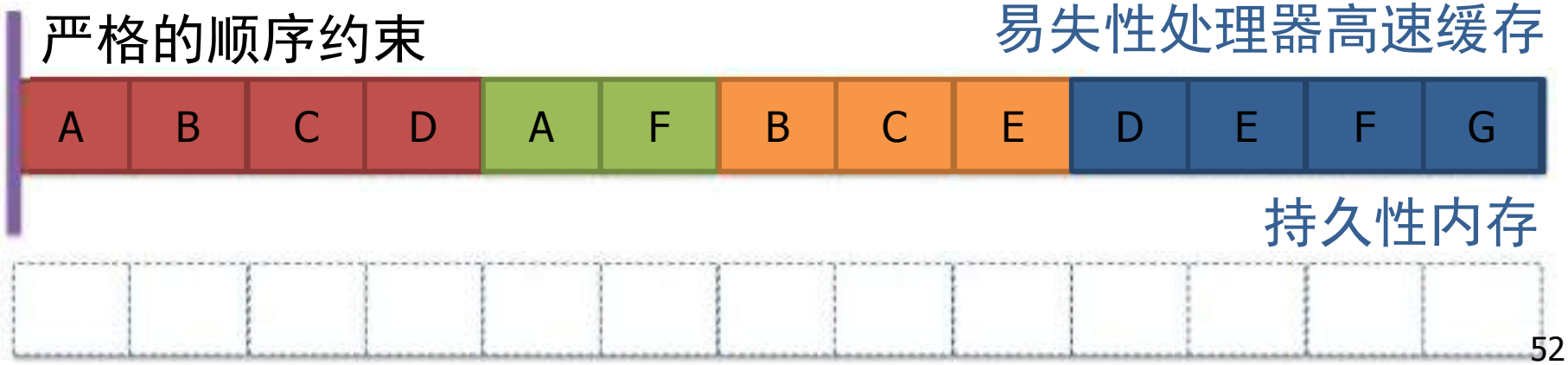
基于原生堆的持久性堆结构实现

基于内存映射文件的持久性堆的实现

3.1 新型编程模型—致性机制优化

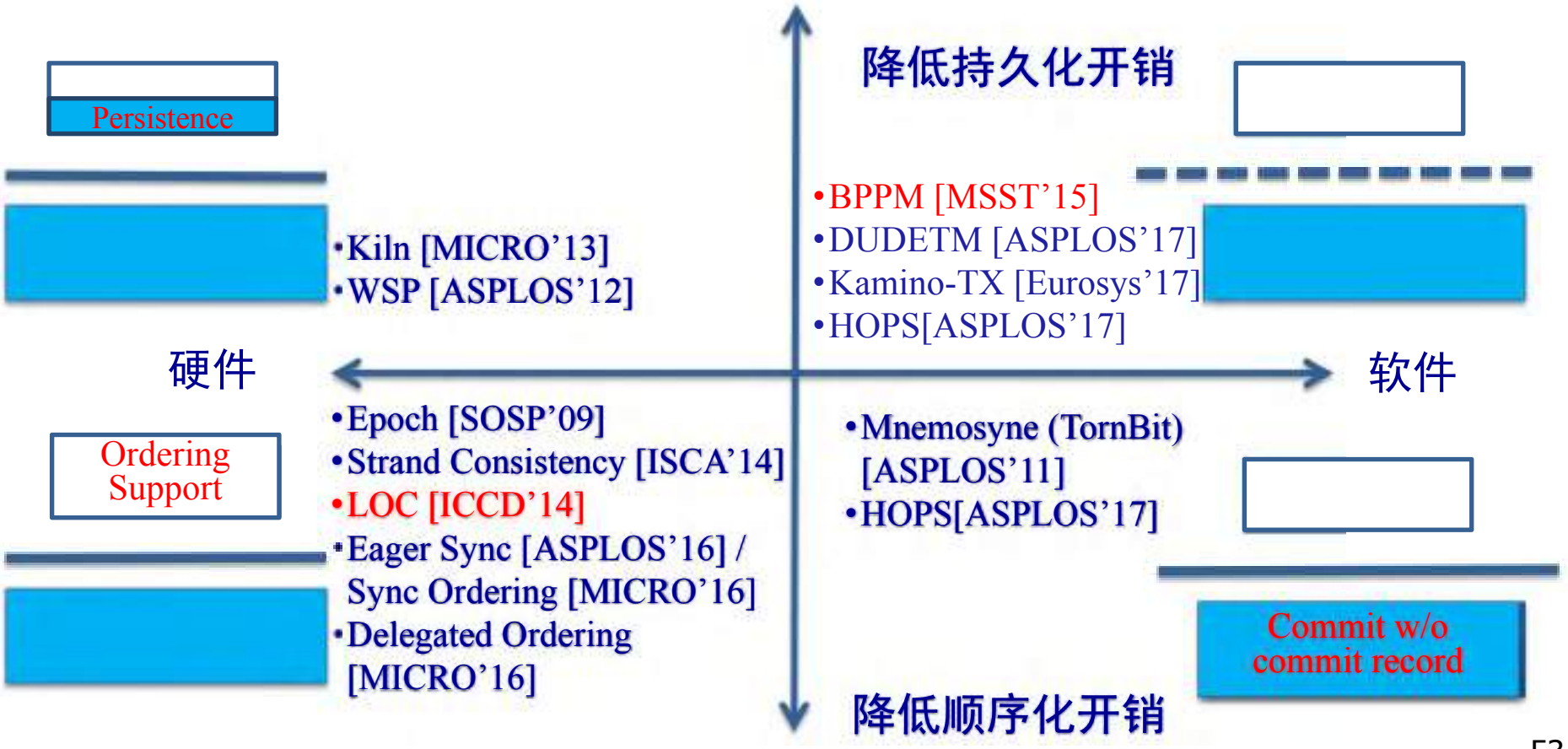
- 在内存级维护持久性数据结构带来一致性问题
- 持久性内存的一致性机制
  - ✓ 顺序性：处理器数据需要按照数据依赖关系顺序写回持久性内存
  - ✓ 持久性：处理器数据从 L1、L2 等多级易失性缓存中替换到持久性内存
- 一致性机制引起额外的性能开销

例：Tx1: (A, B, C, D) -> Tx2: (A, F) -> Tx3: (B, C, E) -> Tx4: (D, E, F, G)



3.1 新型编程模型—致性机制优化

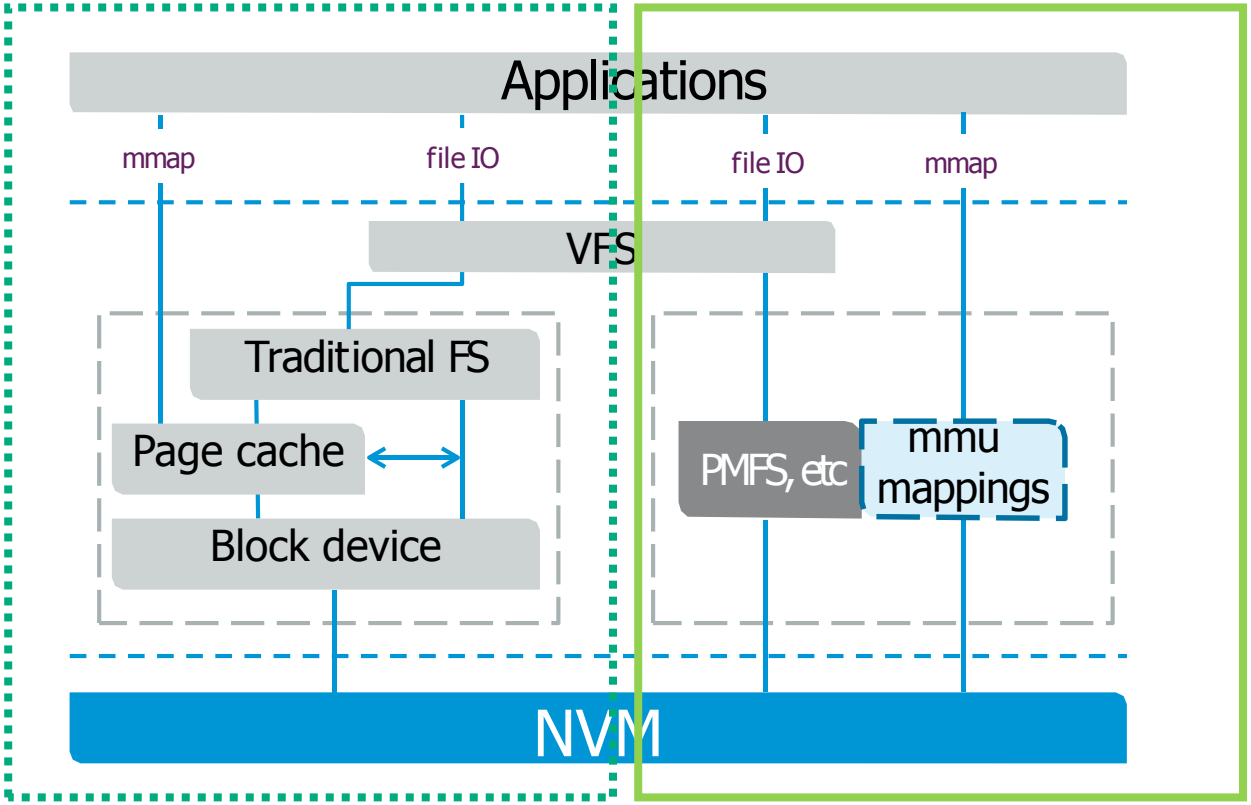
- 持久性内存的一致性机制优化
  - ✓ 维度：软件和硬件×降低持久化和降低顺序化开销
    - 如：非易失CPU缓存、CPU硬件功能扩展、软件层放松一致性等



3.2 持久性内存文件系统（1）

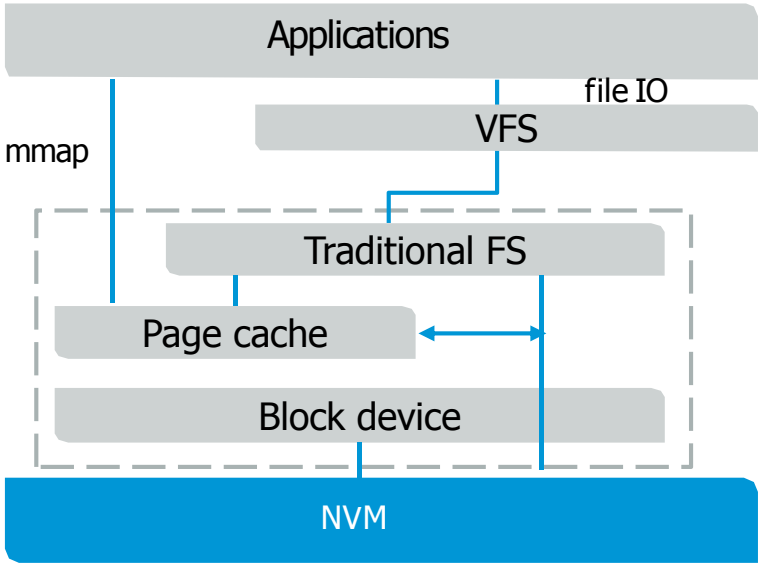
通过RAMDISK模拟块设备的形式兼容传统文件系统

基于持久性内存重新构建字节粒度的持久性文件系统



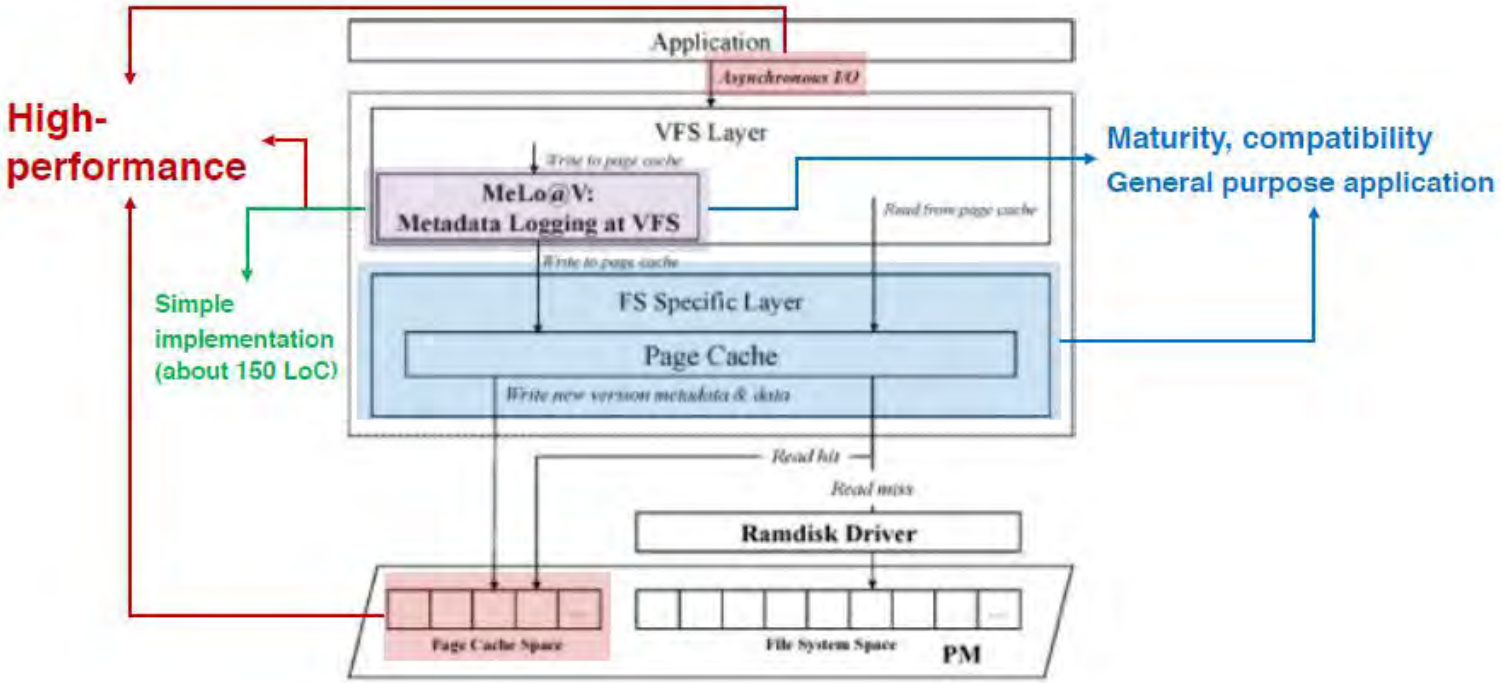
3.2 持久性内存文件系统（2）

- 通过RAMDISK模拟块设备兼容传统文件系统
  - ✓ 传统文件系统无需修改，可直接构建于以持久性内存模拟的RAMDISK块设备之上，如 EXT4，BtrFS
  - ✓ RAMDISK形式使得传统文件系统快速受益于内存级的数据持久化，相比于外存性能有数量级的提升
  - ✓ 不足：软件层的开销巨大，无法充分利用持久性存储介质优势



3.2 持久性内存文件系统（3）

- 改造传统的文件系统
  - ✓ 提升经过时间考验的（成熟）的传统文件系统性能[NVMW'17]
    - 移除Page Cache中同步数据更新：异步I/O隐藏page cache flush开销
    - 将Page Cache作为multi-versioning 区域
    - 优化文件系统一致性机制：轻量级的VFS层的元数据日志机制 MeLo@V



Source: Hyunsub Song,et al. "Transforming Legacy File Systems into Persistent Memory Exploiting File Systems with MeLo@V." NVMW2017

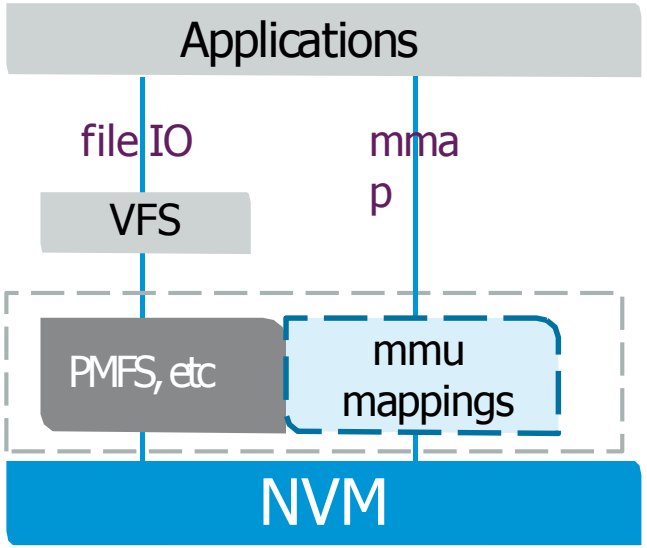
3.2 持久性内存文件系统（4）

- 基于持久性内存重新构建字节粒度文件系统
  - ✓ 细粒度的数据访问
  - ✓ 融合内外存管理方式
  - ✓ NVM直写，避免双重拷贝和块层开销

更高效地发挥了字节寻址 NVM 的性能优势

代表性工作：

1.字节寻址的持久性内存文件系统BPFS [SOSP'09]
2. 融合虚拟内存管理技术的持久性内存文件系统SCMFS[SC'11] / SIMFS [NVMSA'15,TC'16]
3. 轻量级持久化内存文件系统PMFS [EuroSys'14]
4. 日志结构持久性文件系统NOVA [FAST'16]
5. I/O路径优化的高性能文件系统HiNFS [EuroSys'16]
6. 支持应用层事务的文件系统FCFS [MSST'16]
7. 针对文件系统老化问题的文件系统SanGuo [DATE'17]



主要内容

一 背景

二 面向大数据的闪存存储系统

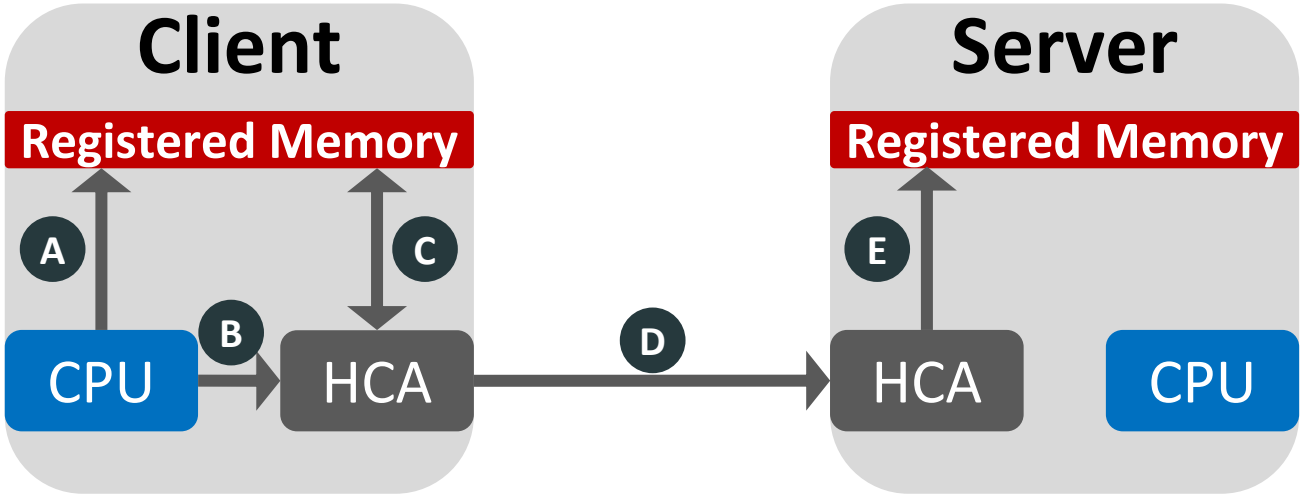
三 面向大数据的持久性内存存储系统

四 新型分布式存储系统

五 展望

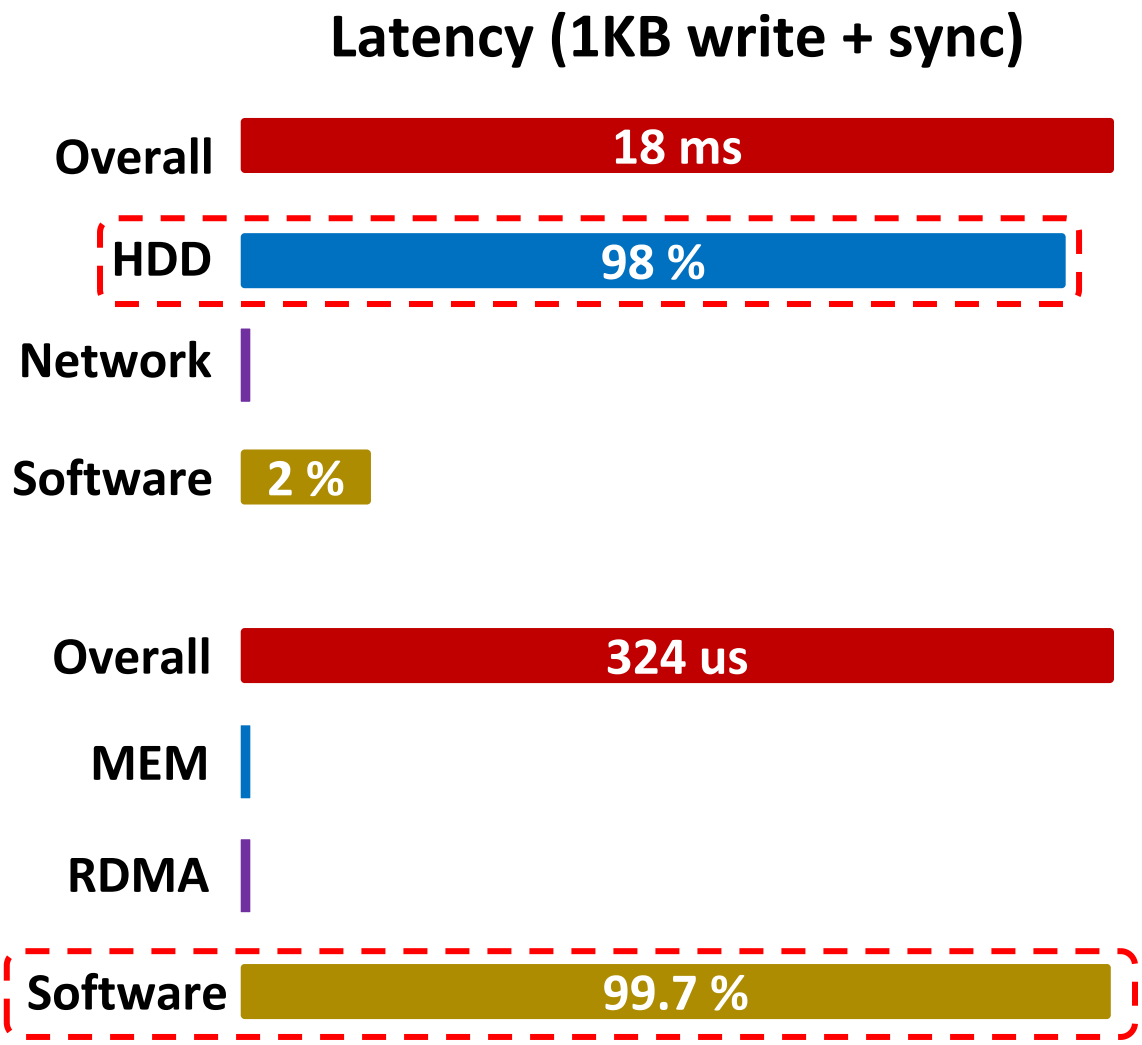
4. 背景—RDMA

- 大数据存储和处理的时效性要求越来越高
  - ✓ 网络对大数据系统的时效性影响非常大
- Remote Direct Memory Access (RDMA)
  - ✓ 之前广泛应用于HPC环境，随着大数据时代的快速发展，也开始应用到大数据处理与分析的场景中
- RDMA的优势：
  - ✓ 远程内存直接访问，绕过远端内核，低延迟、高吞吐



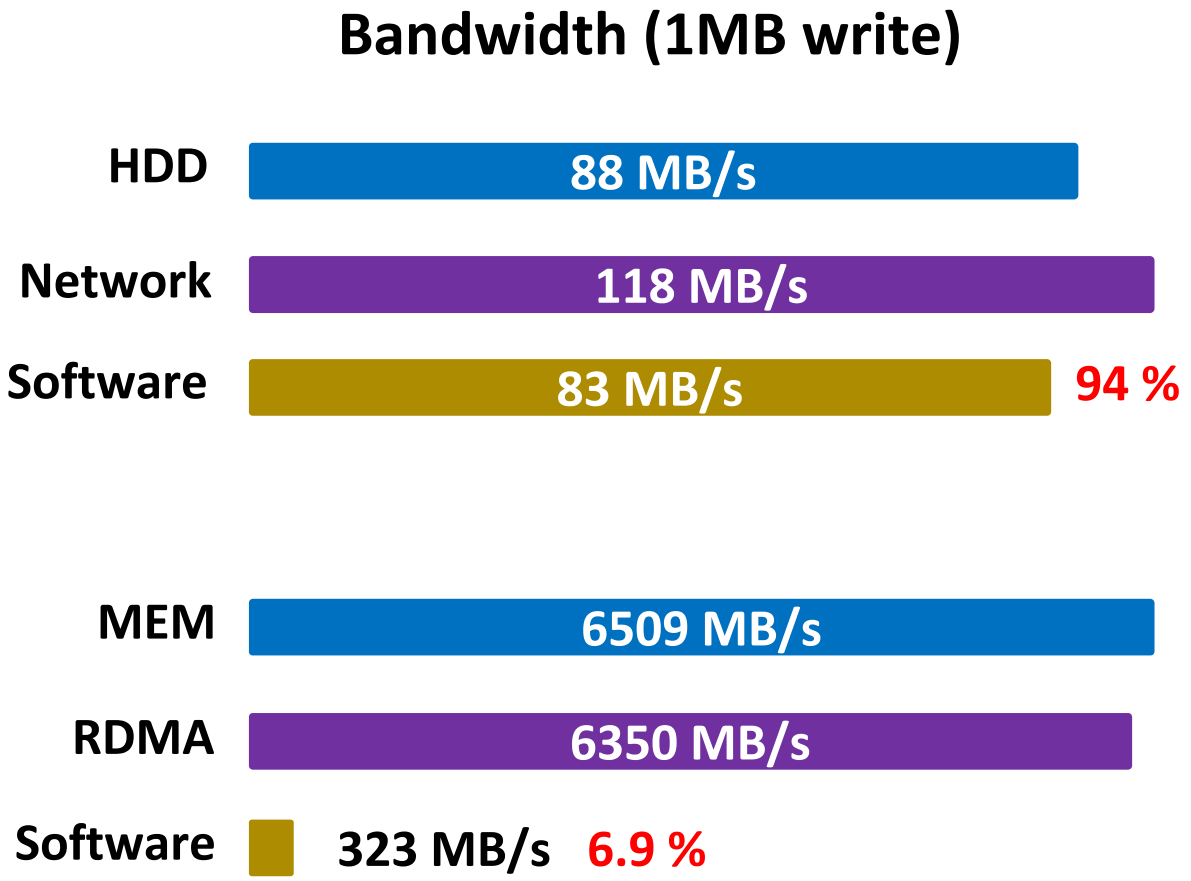
4. 背景—模块化分布式文件系统的挑战

- DiskGluster
  - ✓ 存储：磁盘
  - ✓ 通信：GbE
- MemGluster
  - ✓ 存储：内存
  - ✓ 通信：RDMA



4. 背景—模块化分布式文件系统的挑战

- DiskGluster
  - ✓ 存储：磁盘
  - ✓ 通信：GbE
- MemGluster
  - ✓ 存储：内存
  - ✓ 通信：RDMA



4. 持久性内存和RDMA的机遇

- NVM和RDMA的新特性，也为新设计带来了可能性
  - ✓ NVM可字节寻址
  - ✓ One-Sided RDMA
  - ✓ CPU成了新瓶颈
  - ✓ RDMA立即写的语义
  - ✓ RDMA原语

NVM可字节寻址 → 共享数据管理

One-Sided RDMA, CPU成了新瓶颈 → 新的数据流策略

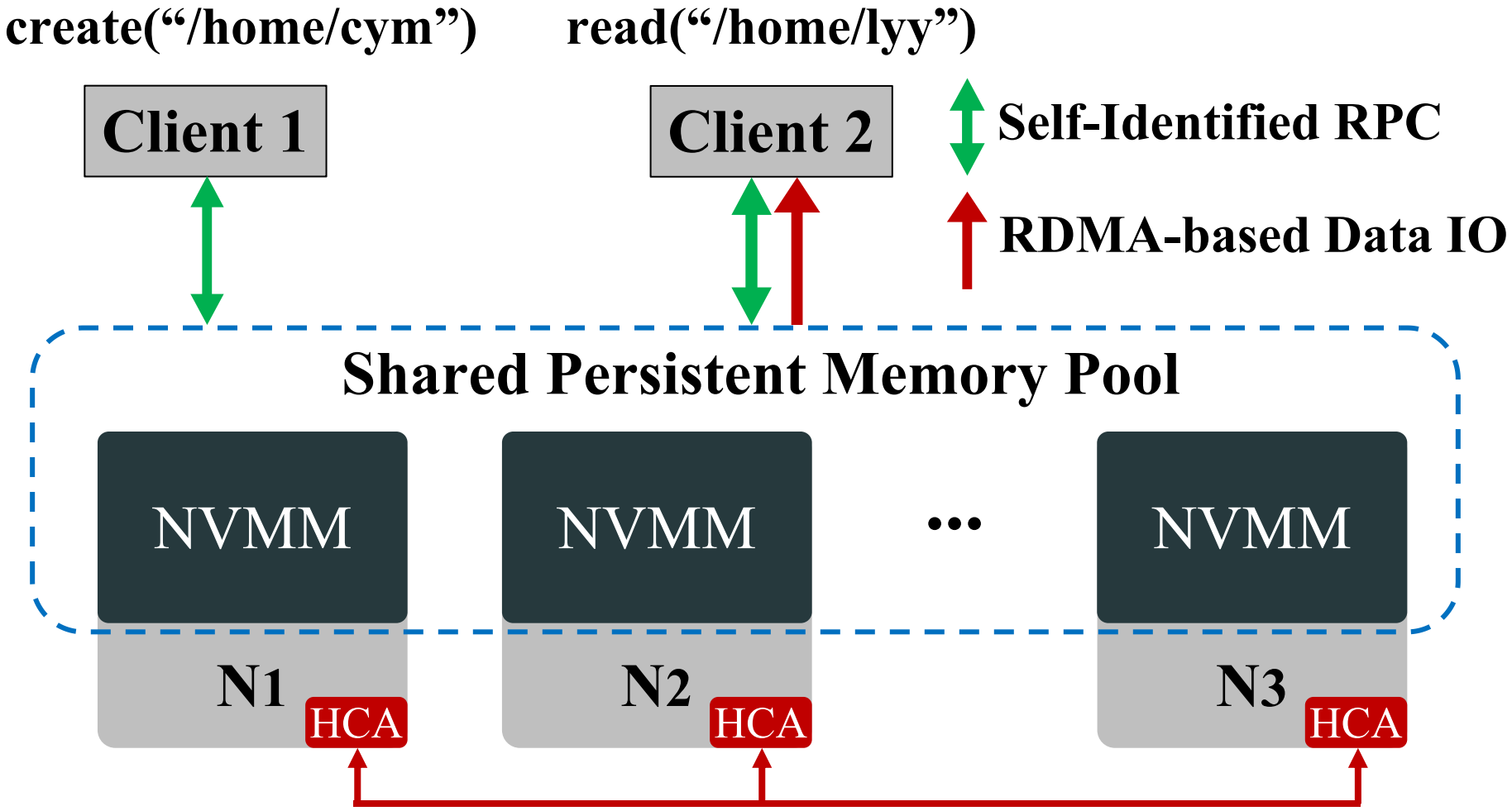
RDMA立即写的语义 → 高效的RPC设计

RDMA原语 → 并发控制

- 重新设计分布式文件系统

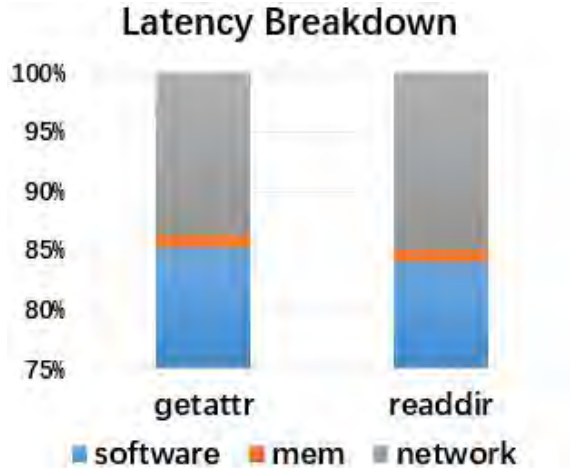
4. Octopus分布式文件系统

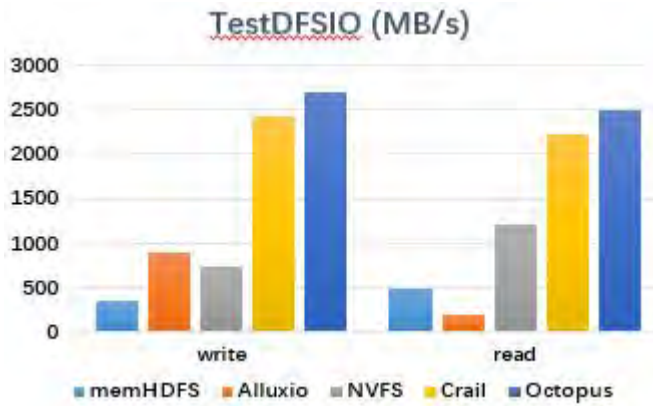
- Octopus—基于RDMA设计的分布式持久性内存文件系统

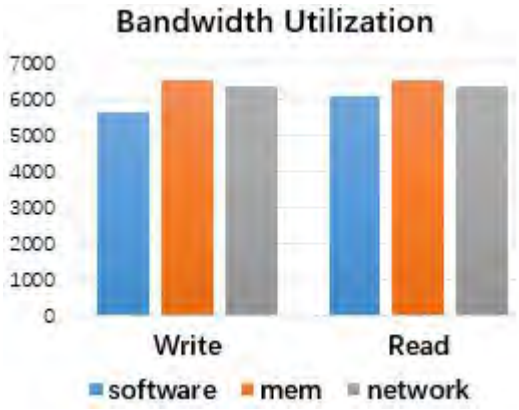


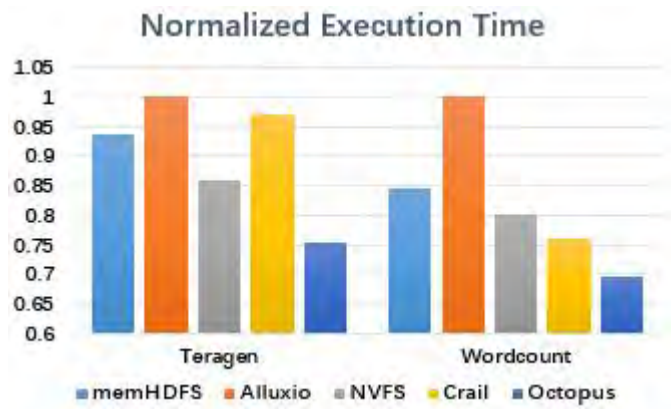
4. Octopus分布式文件系统

- 在大数据应用场景中，Octopus能够明显降低软件开销，比其他分布式内存文件系统提供更好的性能









主要内容

一 背景

二 面向大数据的闪存存储系统

三 面向大数据的持久性内存存储系统

四 新型分布式存储系统

五 展望

5. 大数据时代的存储系统展望

- 大数据时代对存储系统有着更加苛刻的要求
  - ✓ 随着硬件性能的提升，软件开销可高达**21.90%至94.09%**

- 新型分布式存储

谢谢！