CHINA LINUX KERNEL
中国Linux内核开发者大会

華中科技大學
网络空间安全学院
School of Cyber Science and Engineering, HUST

第19届中国Linux内核开发者大会

赞助单位

华中科技大学 | HUAWEI | 阿里云 | FUJITSU | 腾讯云
oppo | 火山引擎 | vivo | 龙芯中科 LOONGSON TECHNOLOGY | 蚂蚁集团 ANT GROUP

支持单位

清华大学 Tsinghua University | 迪捷软件 DIGIPROTO | OpenEuler | OpenAnolis 龙蜥社区 | OpenCloudOS

支持社区&媒体

CSDN | 云巅论剑 | sf 思否 | 阅码场 yamcode | InfoQ 极客传媒
51CTO | 开源江湖

2024年10月　湖北 • 武汉

華中科技大學

SLS单层存储：零拷贝共享内存设计与应用

分享人：黎红波 （华为OS内核实验室）

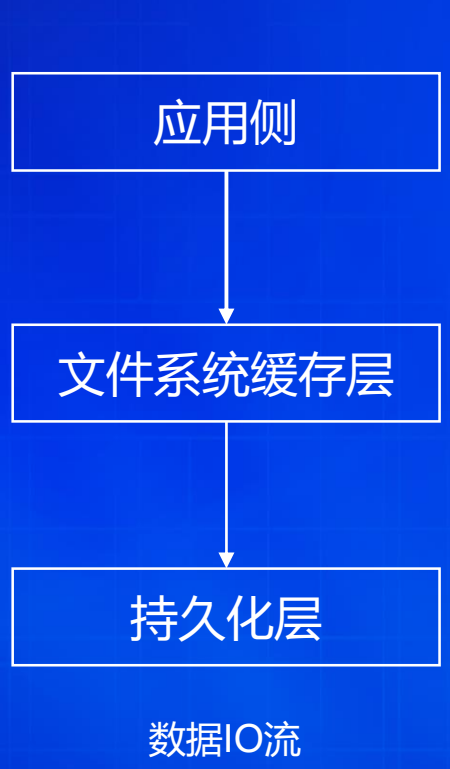
2024 CLK

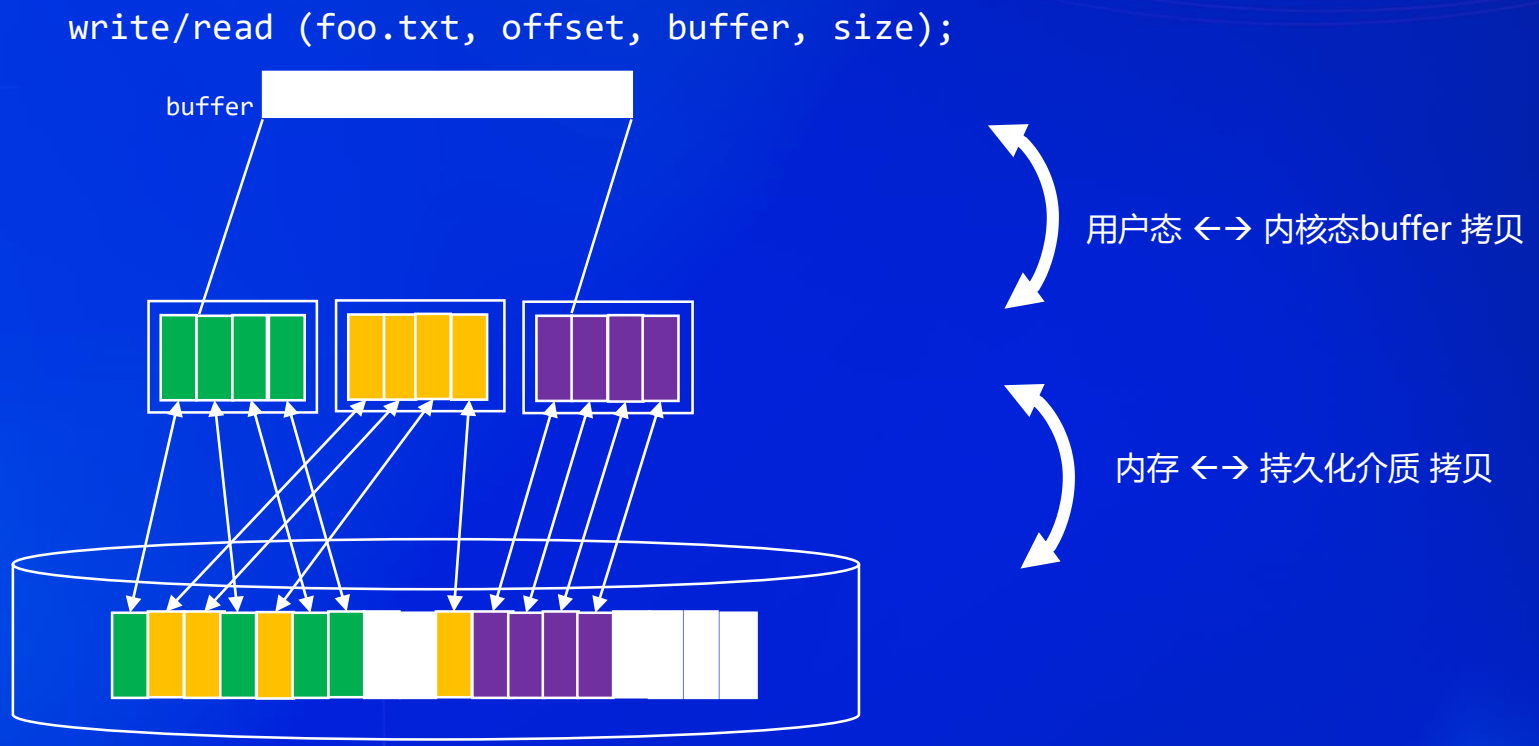
華中科技大學

目录

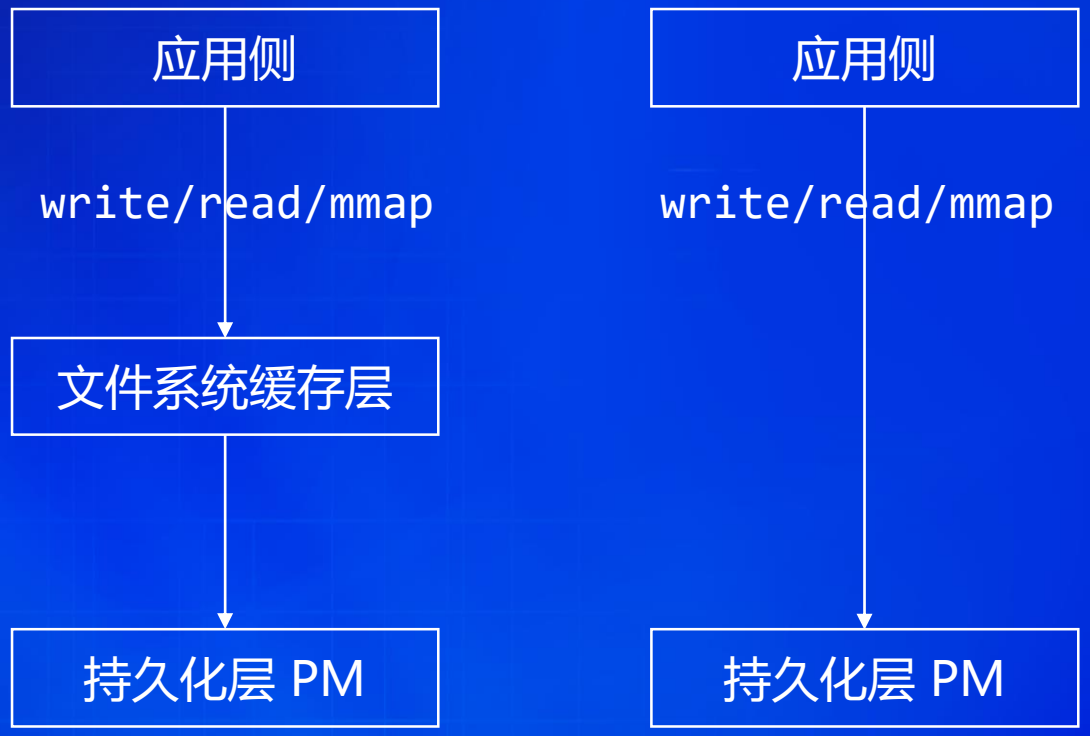
背景介绍

存储介质存储延迟：HDD ~10ms, SSD 10~100us, DRAM 80~100ns。
通过引入中间缓存能弥补存储介质时延差异大带来的开销。





背景介绍：当持久化内存设备作为存储介质



fsdax模式：EXT、XFS

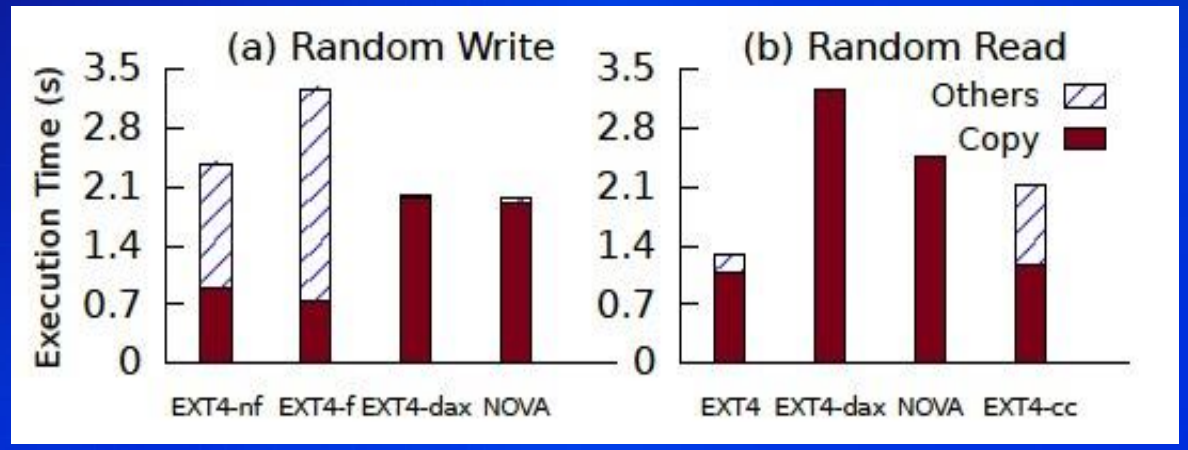
devdax模式：EXT-dax、NOVA

```
mkfs.ext4 /dev/pmem0
mount -o dax /dev/pmem0 /mnt/ext4

mkfs.ext4 /dev/pmem0
mount /dev/pmem0 /mnt/ext4
```

Cache VS Direct Access ?

背景介绍：Cache VS Direct Access



- 在Cache类型文件系统上APP-Cache数据拷贝占用了23%以上的 开销（EXT4-nf)
- 在DAX类型文件系统上APP-PM数据拷贝占用了96%以上的开销（EXT4-dax, NOVA）
- 数据同步(后台刷脏) 导致了37% 性能下降 （EXT4-nf, EXT4-f)
- 数据迁移(cache miss时数据加载) 导致了65% 的性能下降（EXT4-cc）

观察点 1:

存储设备仍然是层次结构的，ＰＭ比ＤＲＡＭ慢很多倍

观察点 2:

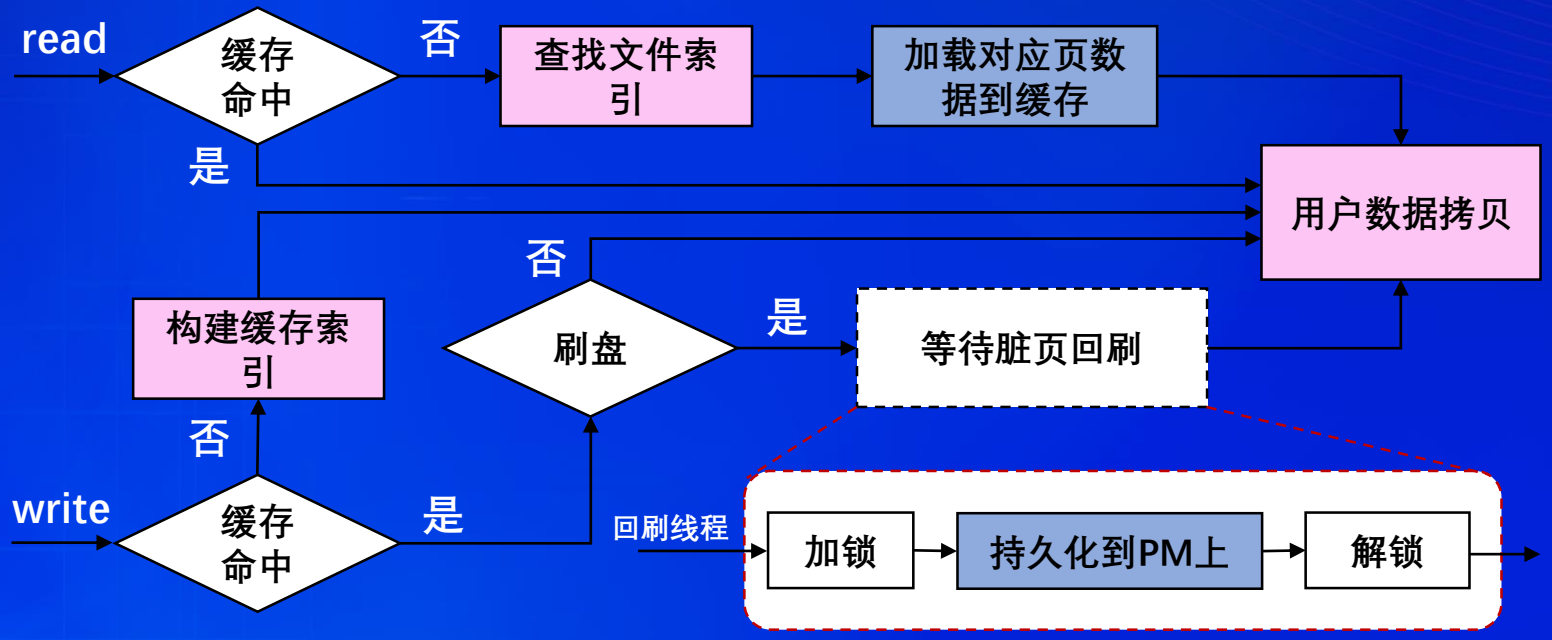
文件系统与APP buf 之间数据传输的开销是显著的

观察点 3:

缓存带来的数据同步与迁移的开销是厚重的

缓存有必要，但引入了额外开销!

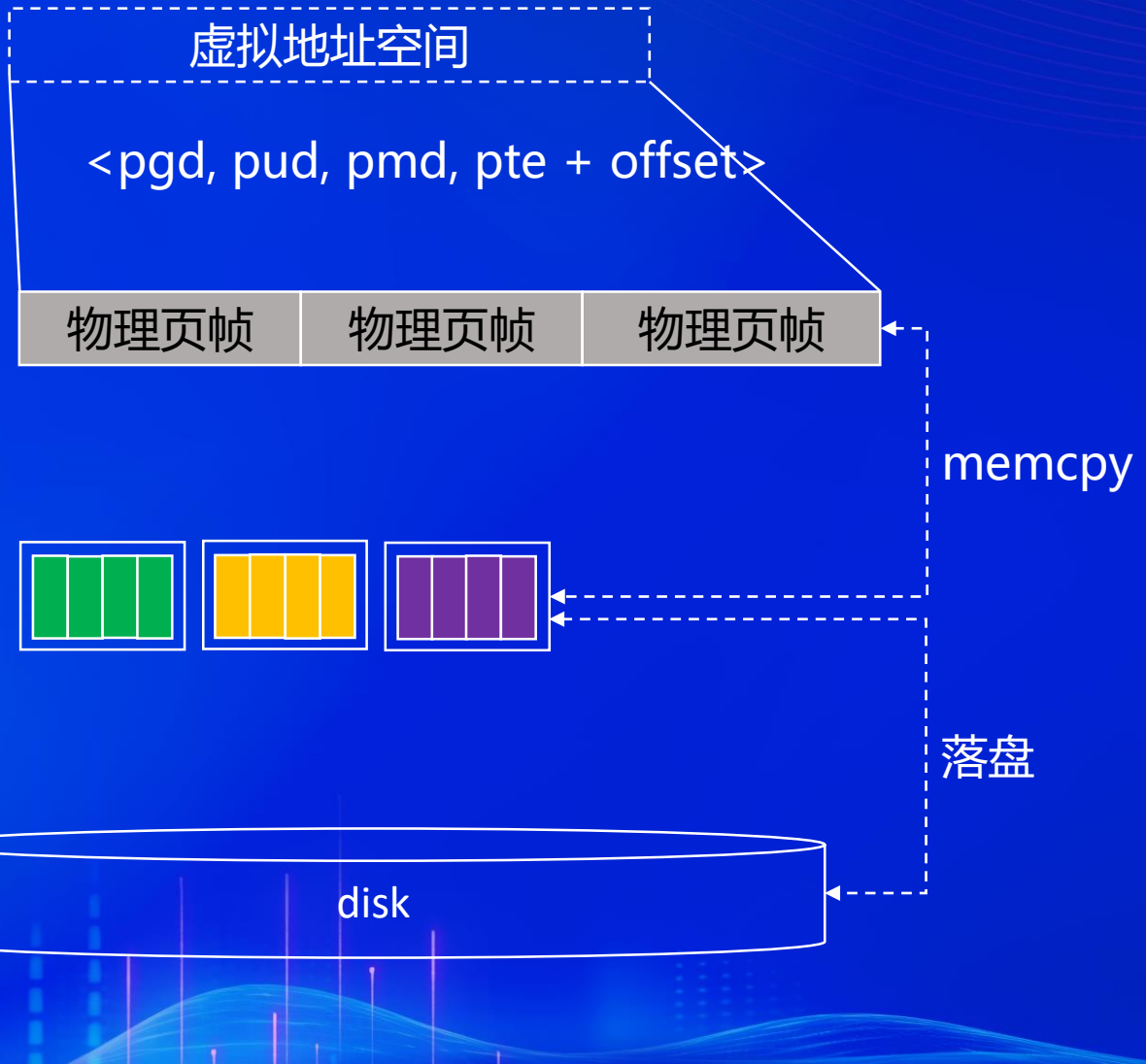
背景介绍：异构存储文件系统IO流



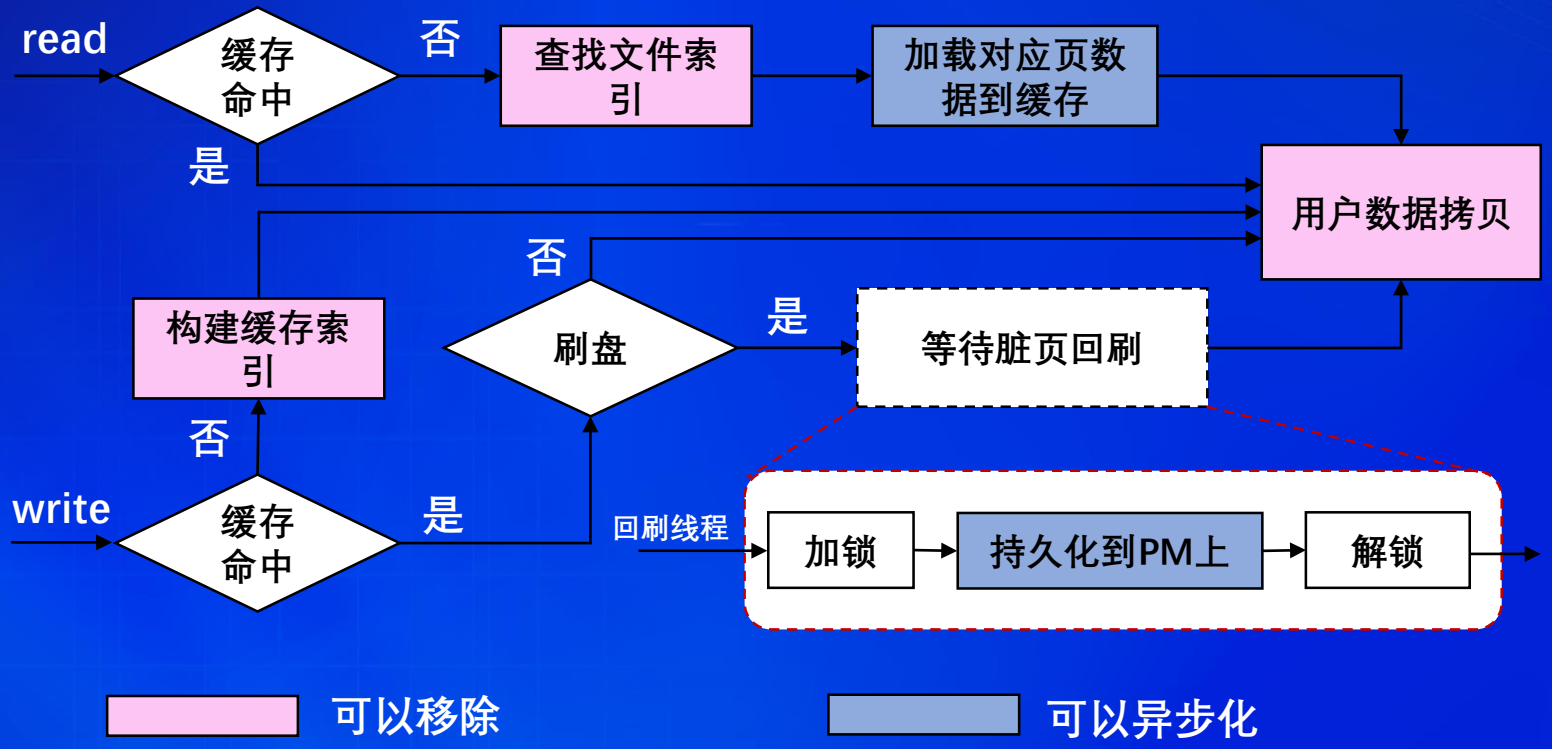
- 开销1：数据拷贝开销
- 开销2：数据同步、迁移和索引开销

背景介绍：从页表映射机制来看数据拷贝

buffer_addr: 0x000facabdxab



优化动机



动机 1:

用零拷贝技术优化APP与文件系统缓存之间的数据传输和索引的开销

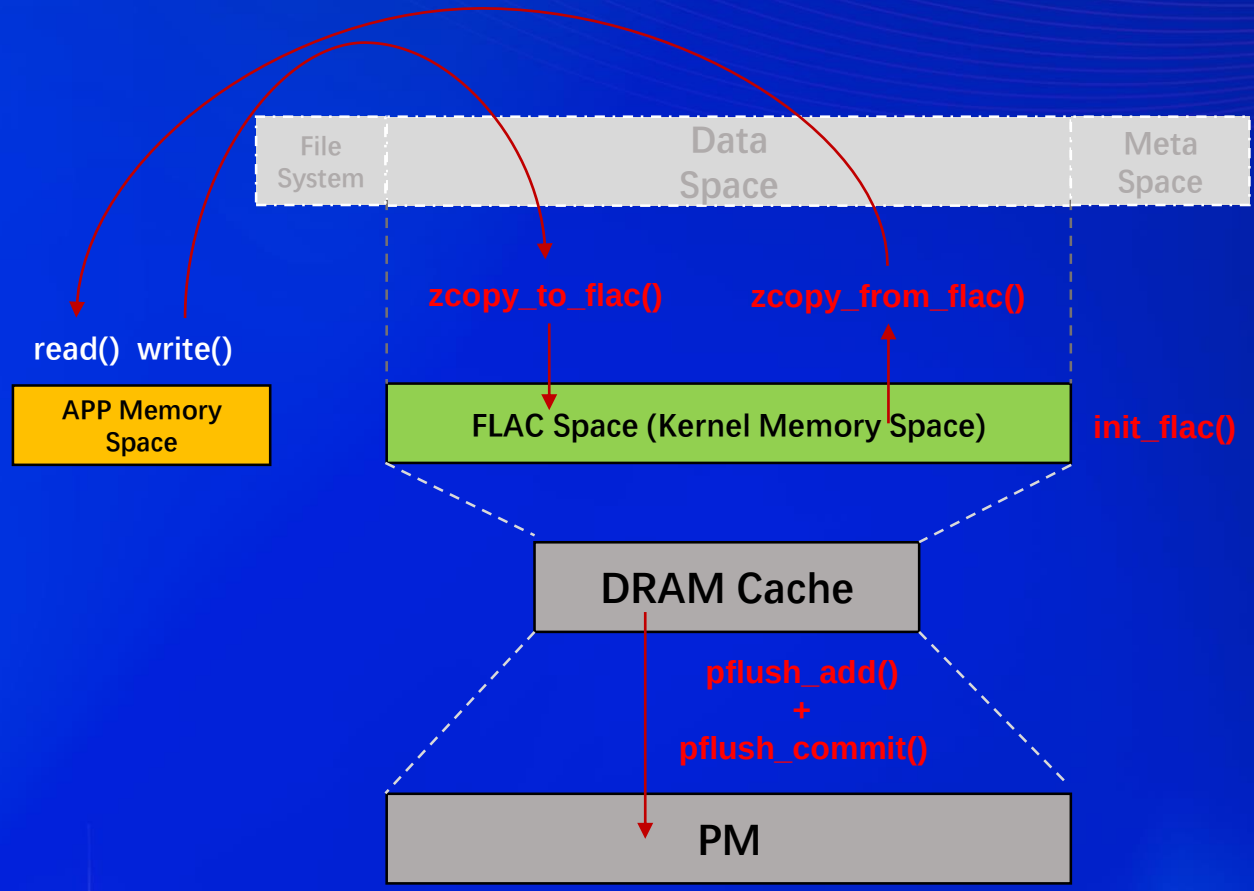
动机 2:

隐匿由于缓存带来的数据同步和迁移的开销

SLS单层存储设计： FLAC

FLAC提供异构内存统一管理框架，对外提供如下API：

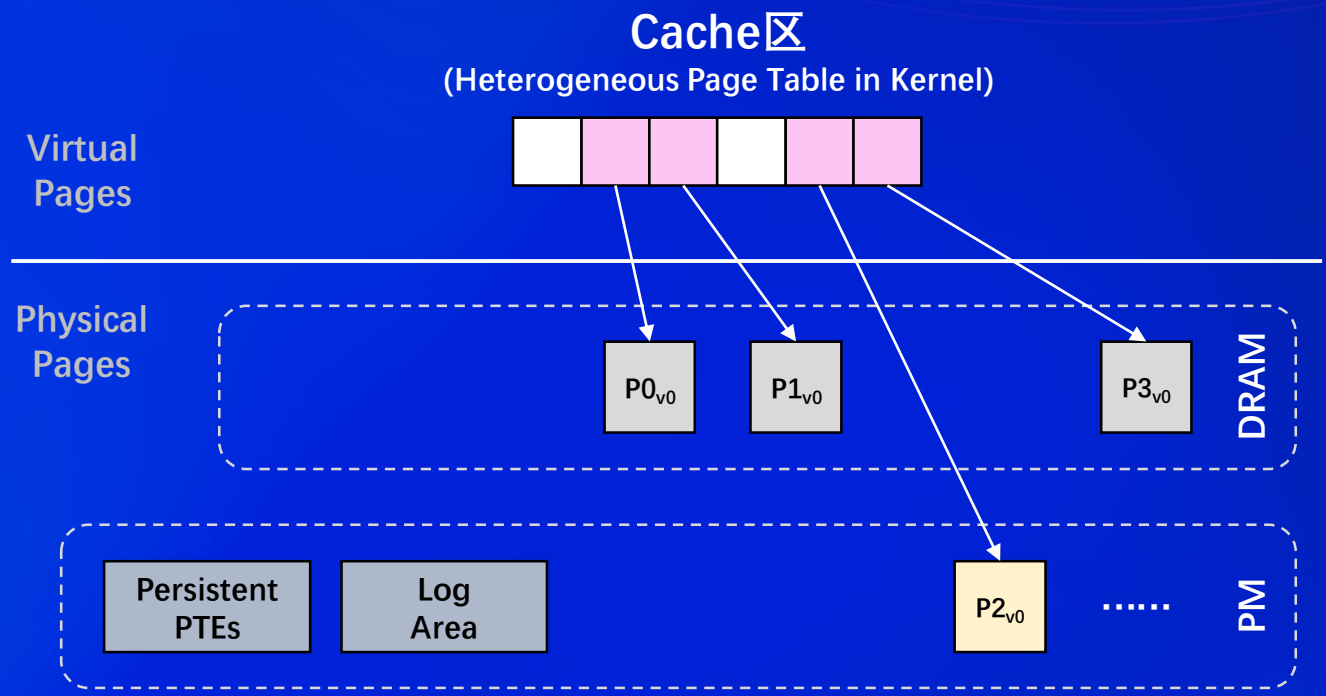
API	Main Para.	Description
`init_flac`	pm_path	Create/Recover the FLAC space
`zcopy_from_flac` `zcopy_to_flac`	from_addr to_addr size	Zero-copy transfer data between the application and the FLAC space
`pflush_add`	pflush_handle addr size	Attach (map) the pages to the flushing buffer and add to the handle
`pflush_commit`	pflush_handle fs_metalog	Flush the pages in the handle and update the metadata atomically
`pfree`	addr size fs_metalog	Reclaim the PM pages and update the metadata atomically



零拷贝缓存：优化APP-Cache之间的数据传输

技术点: 零拷贝缓存

- 异构内存页表
  - 提供了一个统一连续的虚拟内存地址空间（Cache区）给FS。
  - 动态将虚拟页映射到DRAM或PM上
  - FS开发者可以将文件数据存储在Cache区上。
  - Cache区的页表会持久化到PM上的，当故障发送时可以恢复。

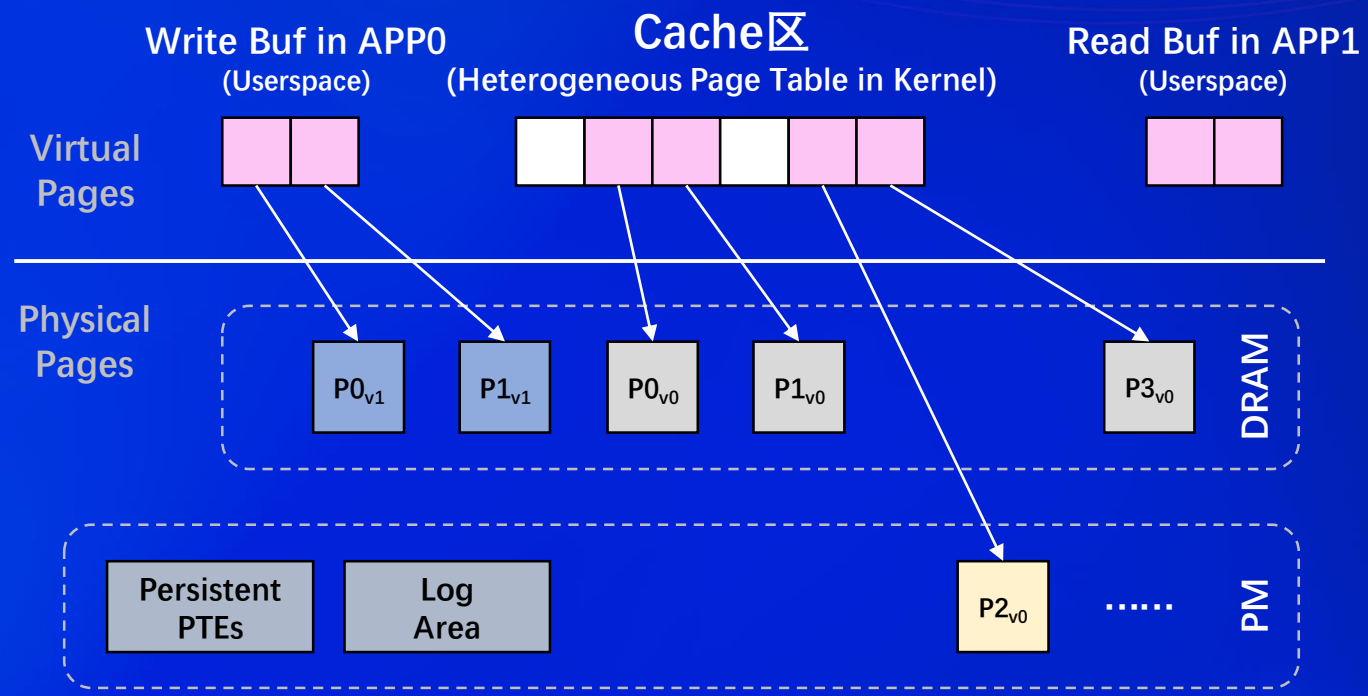


零拷贝缓存：优化APP-Cache之间的数据传输

技术点: 零拷贝缓存

- 异构内存页表
- page attach机制
  - 将物理页从源地址映射到目的地址上来实现数据的交互
  - 利用COW触发缺页来确保数据安全

```
attach(to_addr, from_addr, size, pmode);
```

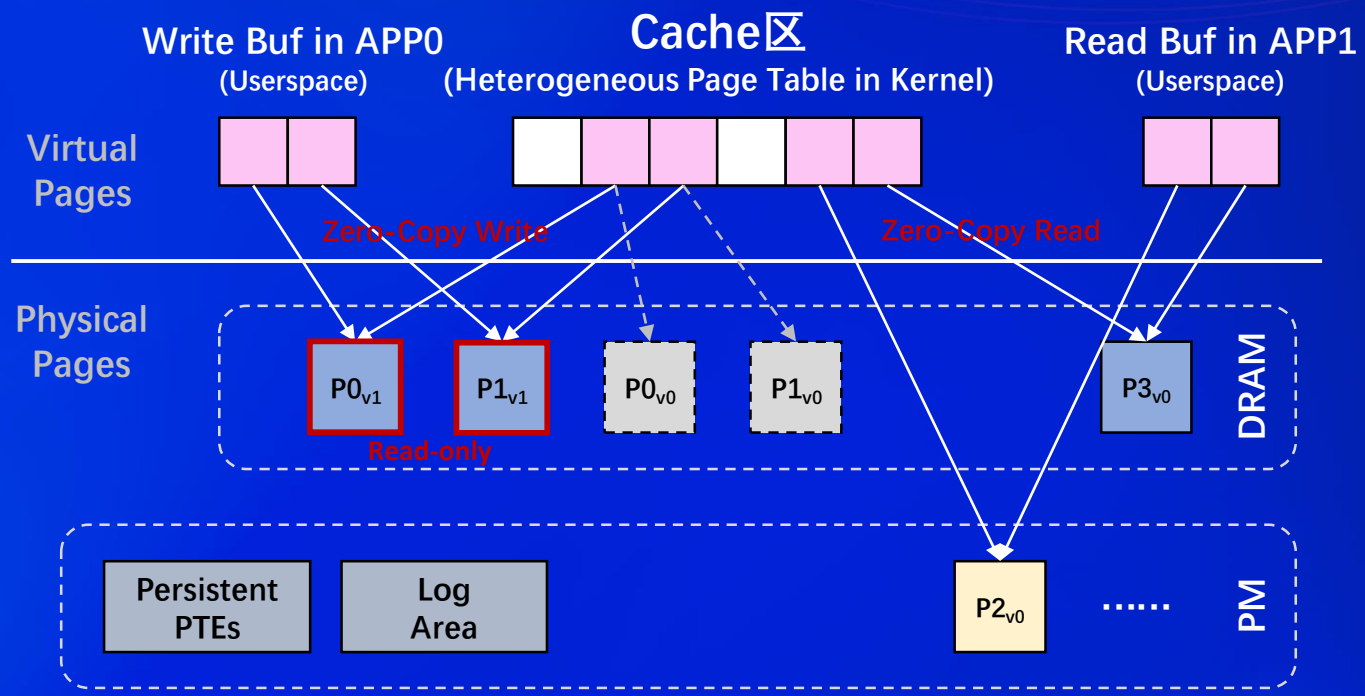


零拷贝缓存：优化APP-Cache之间的数据传输

技术点: 零拷贝缓存

- 异构内存页表
- page attach机制
  - **将物理页从源地址映射到目的地址上来实现数据的交互**
  - 利用COW触发缺页来确保数据安全

```
attach(to_addr, from_addr, size, pmode);
```

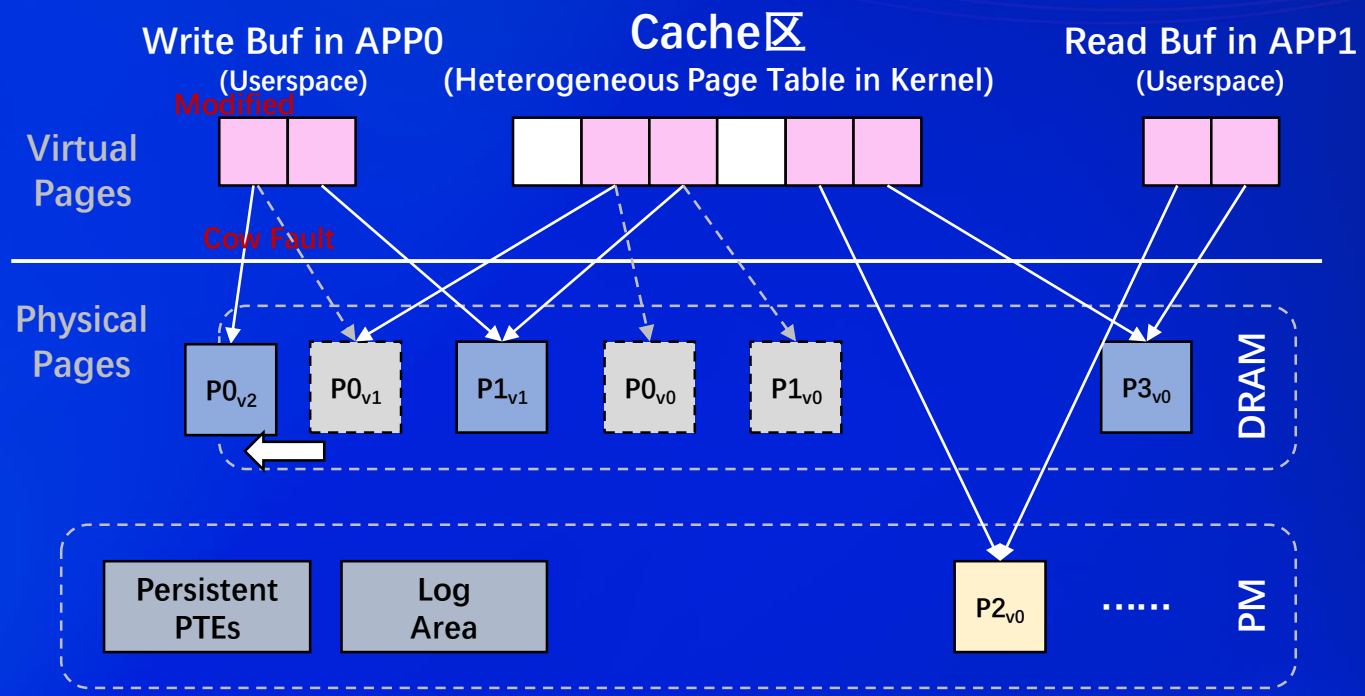


零拷贝缓存：优化APP-Cache之间的数据传输

技术点: 零拷贝缓存

- 异构内存页表
- page attach机制
  - 将物理页从源地址映射到目的地址上来实现数据的交互
  - **利用COW触发缺页来确保数据安全**

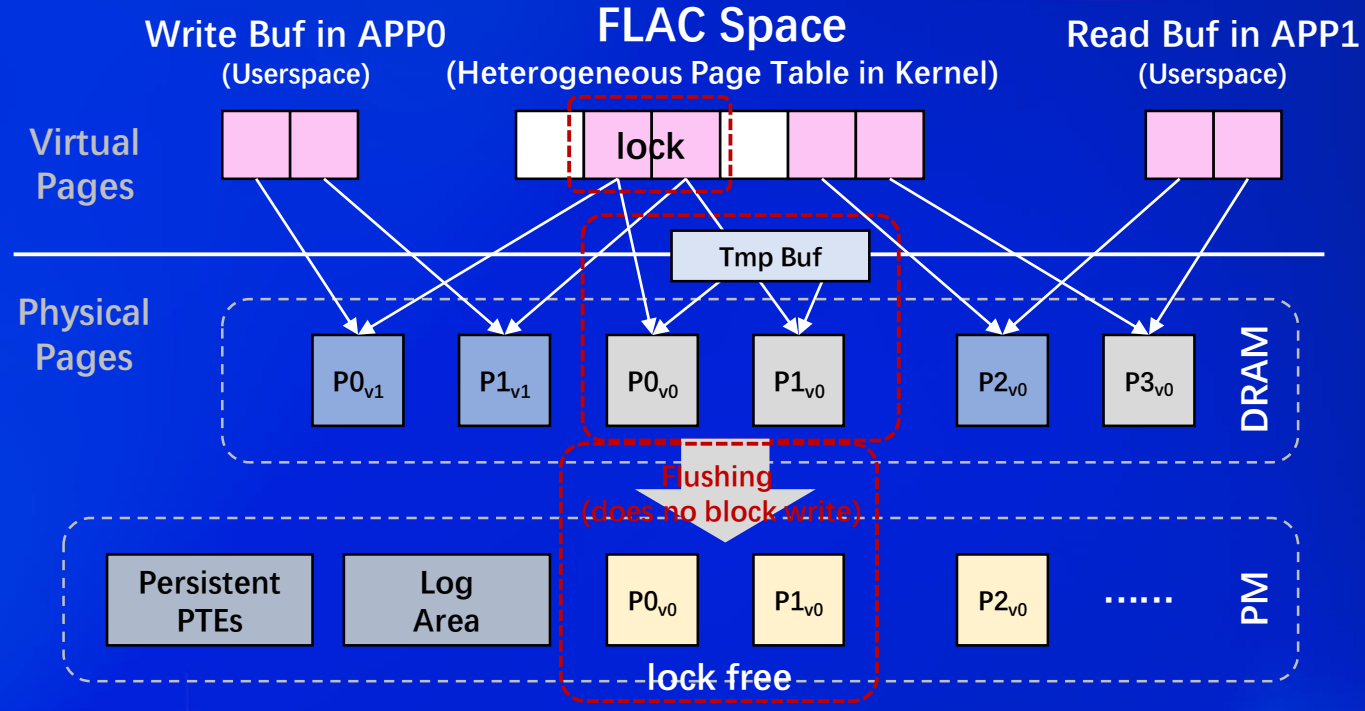
```
attach(to_addr, from_addr, size, pmode);
```



无锁刷脏与异步加载：消除缓存税

技术点: 并行优化的缓存数据同步与迁移

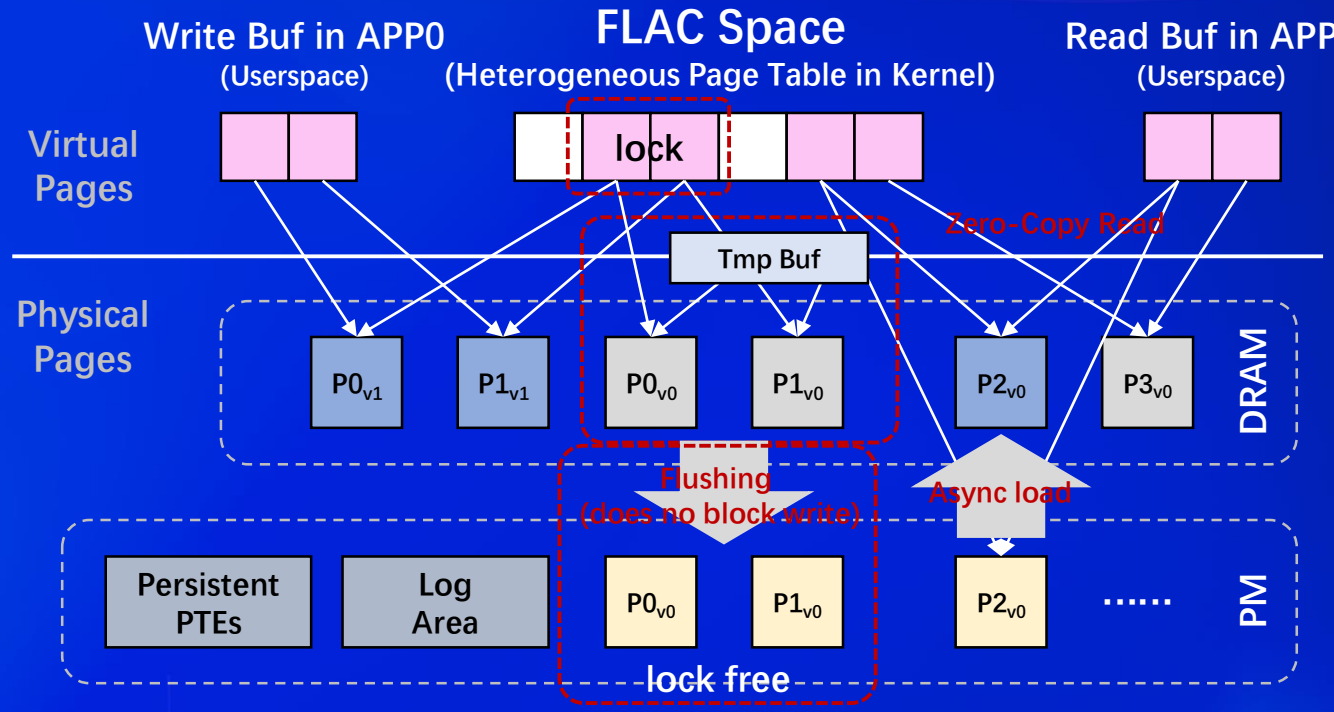
- 无锁刷脏
  - 映射到临时的buffer(lock)
  - 数据同步到PM(lock-free)



无锁刷脏与异步加载：消除缓存税

技术点: 并行优化的缓存数据同步与迁移

- 无锁刷脏
- cache-miss时异步的数据加载
  - 直接将PM上的物理页映到用户BUF上
  - 异步将页从PM迁移到DRAM上

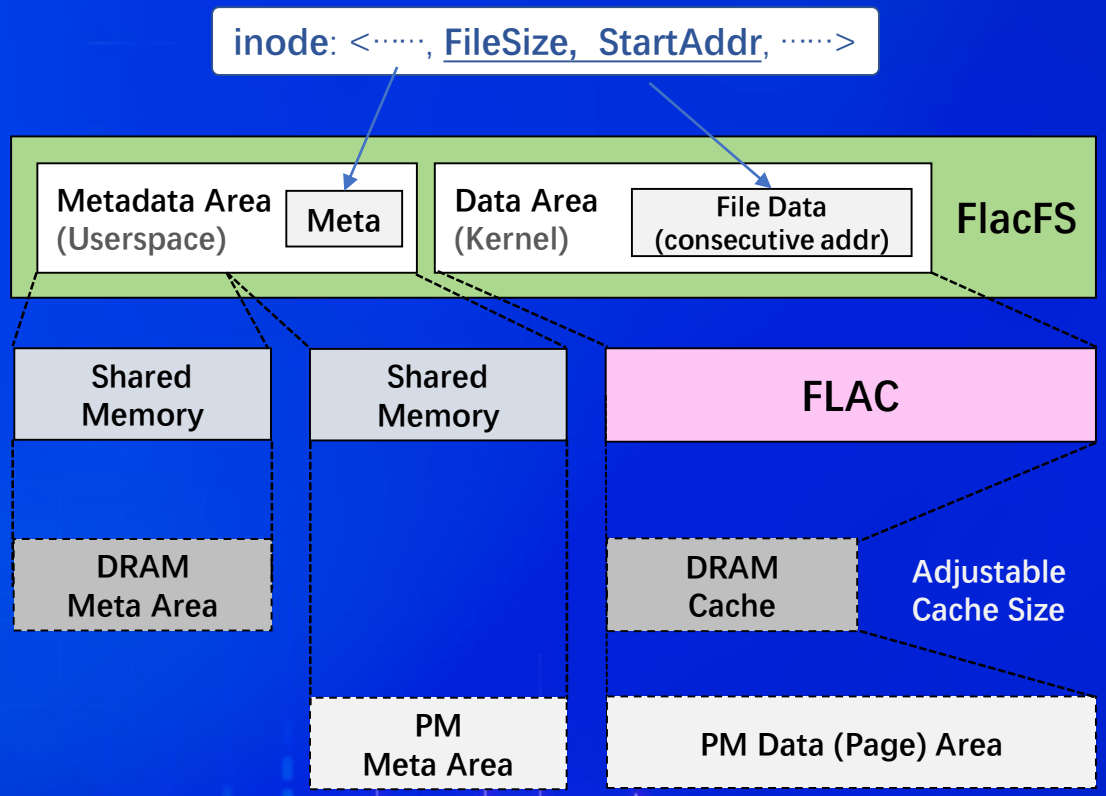


应用示例1： FlacFS

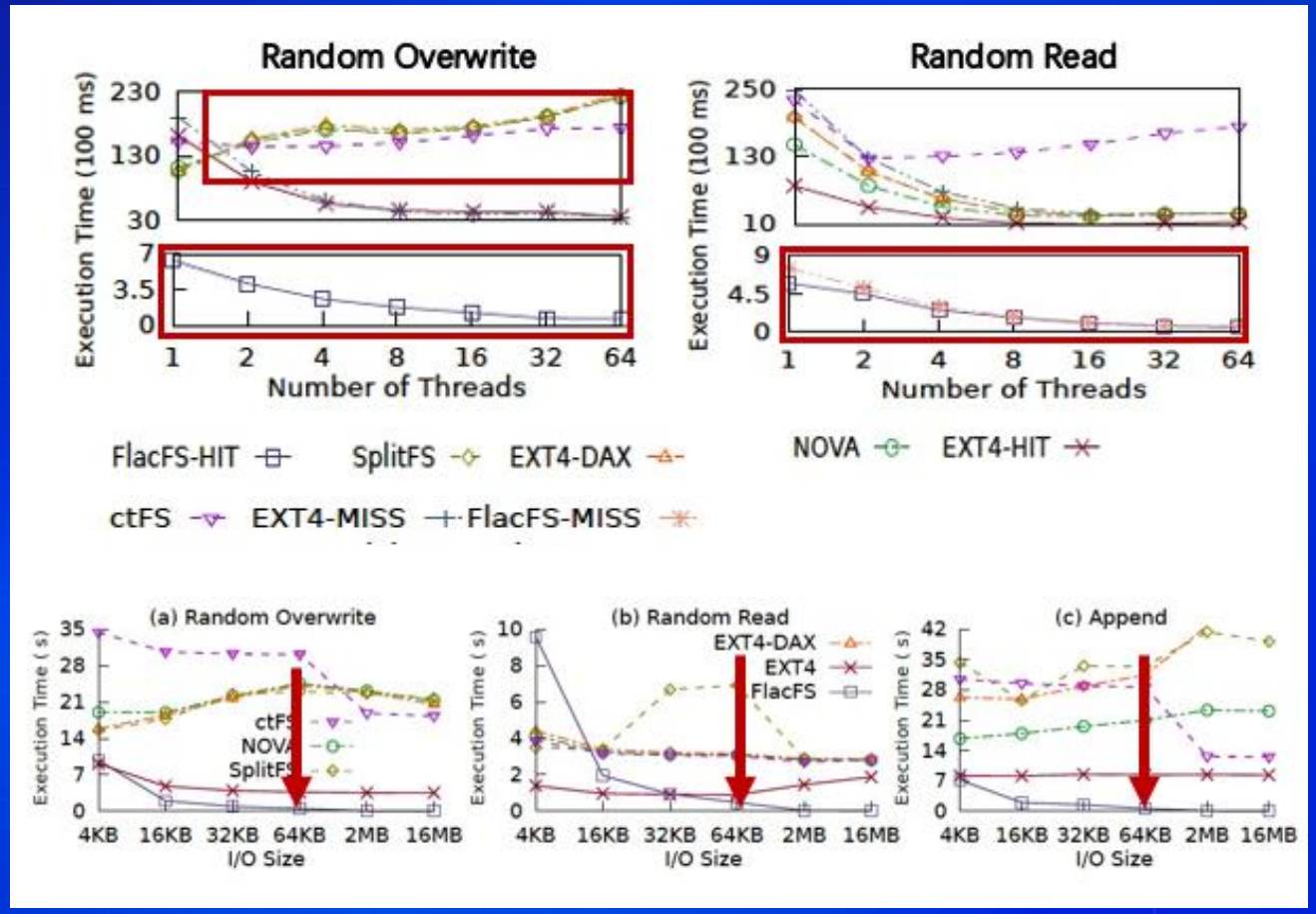
元数据管理：元数据存放在共享内存区域，最终持久化到PM中；元数据以KV对形式进行管理。

数据管理：文件映射一段连续虚拟地址空间，数据直接存储在FLAC空间，利用FLAC层提供接口完成应用数据到文件系统的读写。

安全性与一致性：在内核层面通过页表状态机制确保数据安全；对修改操作进行日志追加，利用FLAC层接口pflush_add/pflush_commit/pfree保证数据和元数据一致性。



实验数据



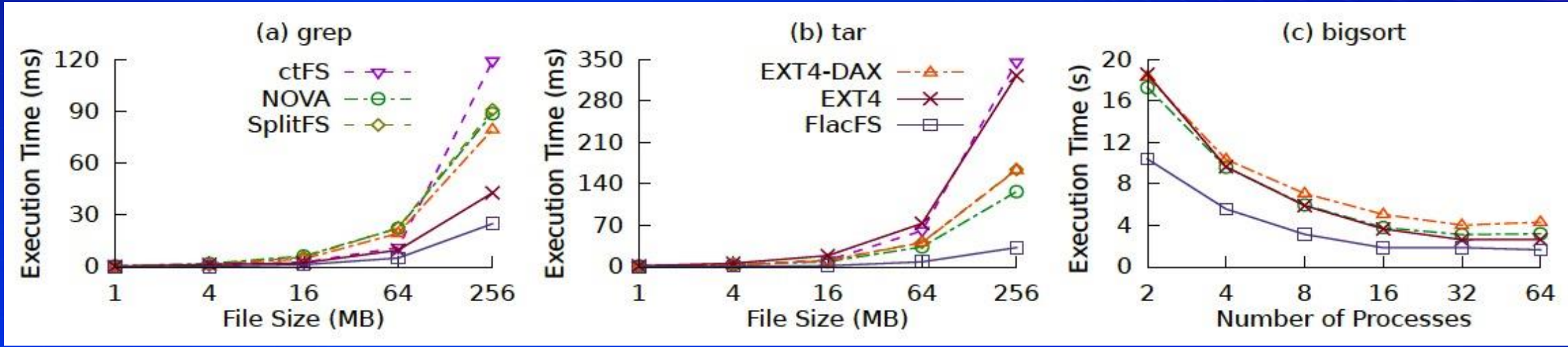
实验环境: microbench 2MB I/O; 64GB data

	NOVA	SplitFS	ctFS	EXT4-DAX	EXT4	FlacFS
Mode	sync	POSIX				
Cons.	Meta					Meta+Data
Cache Flush	N/A				100 ms	10 ms

总结:

- 大IO下读写操作较其他文件系统在microbench情况下有数量级提升
- 良好可扩展性
- 对于64KB及以上IO更友好

实验数据



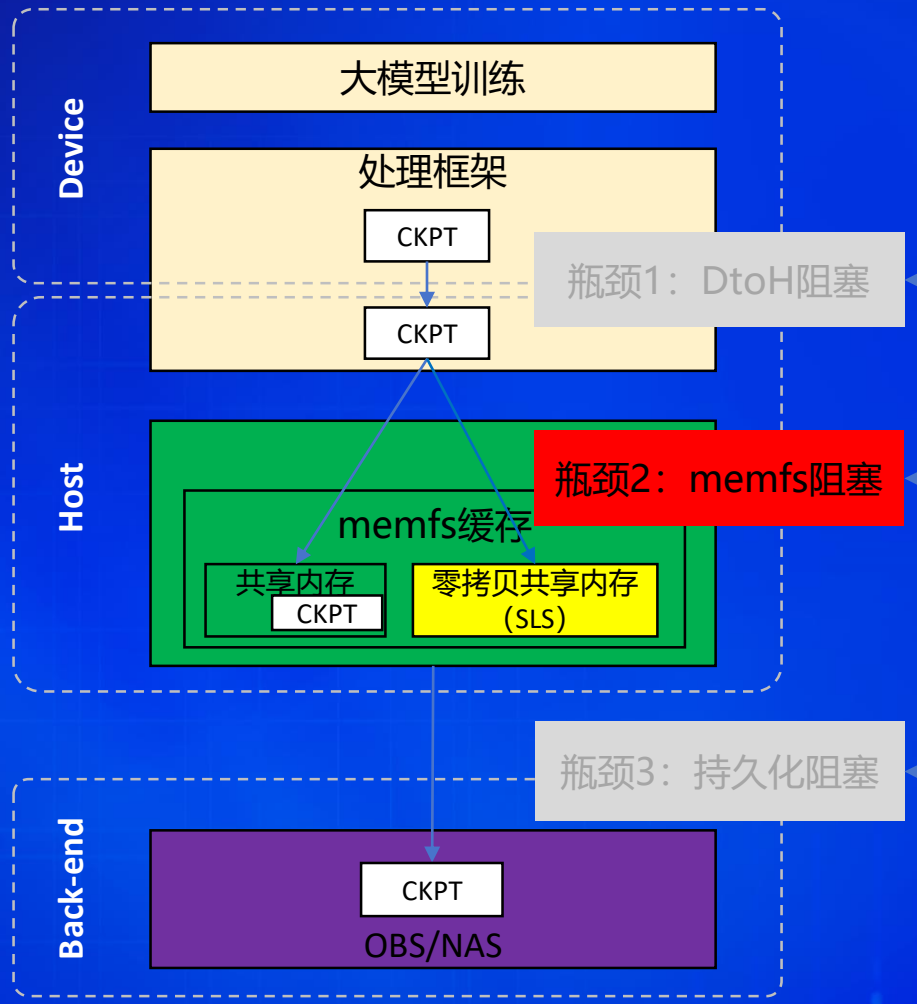
真实应用

- ***grep:*** 读密集型应用
- ***tar:*** 读写密集型应用
- ***bigsort:*** 读写密集、计算密集型应用 (134 million integers)

总结

- 对比DAX-based文件系统至少 6.7 倍性能提升
- 对比Cache-based文件系统可以获得至少9.4倍性能提升

应用示例2：零拷贝优化大模型Checkpoint Save



瓶颈1：DtoH阻塞

瓶颈2：memfs阻塞

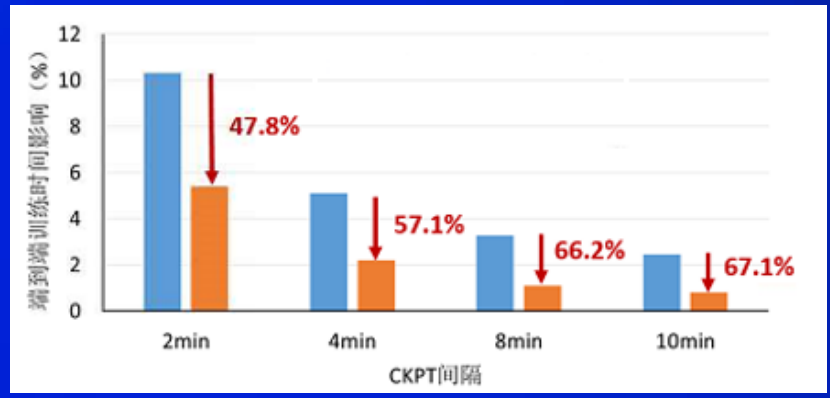
瓶颈3：持久化阻塞

训练过程中异步DtoH拷贝

CKPT从训练框架到memfs无需拷贝

memfs空间需足够大从而掩盖持久化同步开销

- GPT3-NEOX 20B，单机8卡
- memfs空间256GB
- 单个CKPT数据量约8GB
- OBS采用本地FS模拟



- 零拷贝加速CKPT SAVE过程
- 天然去重，节省内存空间

关于我们

融合存储团队（深圳）：致力于打造面向新型存储与互联技术的高性能IO栈，成果应用于大模型训推、大规模容器平台，大数据处理等场景。多个相关工作成果在顶级会议（OSDI、FAST）上进行发表。

华为内核技术交流

Q & A