西安交通大学
XI'AN JIAOTONG UNIVERSITY

基于QEMU的存储系统设计与实践

西安交通大学

聂世强

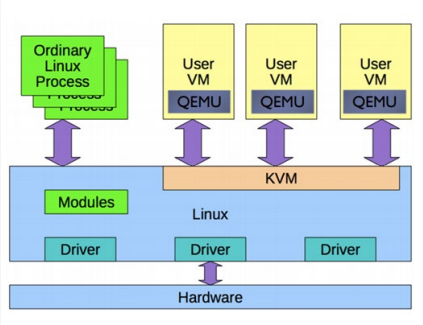
2024年8月20日

大纲

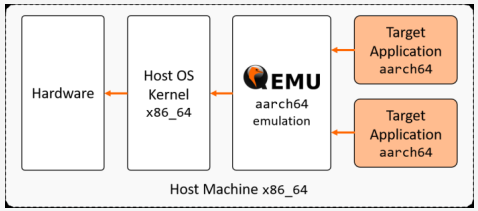
1. QEMU模拟器介绍
2. 研究内容一 键值存储系统数据安全删除
3. 研究内容二 基于DPU的交换空间设计与优化
4. 研究内容三 SPI Flash仿真器设计与实现

一、QEMU模拟器介绍

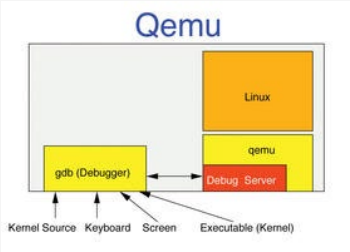
QEMU(Quick Emulator) :开源的虚拟化和仿真工具，使用动态二进制转换技术来模拟处理器，并且提供多种硬件和外设模型，能够提供对多种硬件平台的仿真和虚拟化支持。



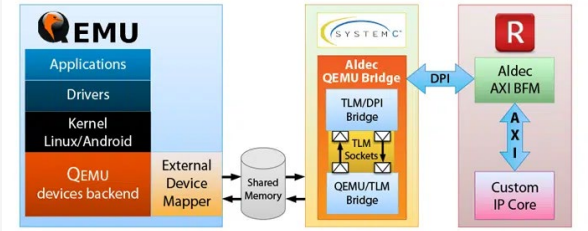
① Openstack和QEMU



② 跨平台开发



③ Linux kernel调试



④ 基于QEMU的外设联合调试

QEMU应用场景

使用场景:

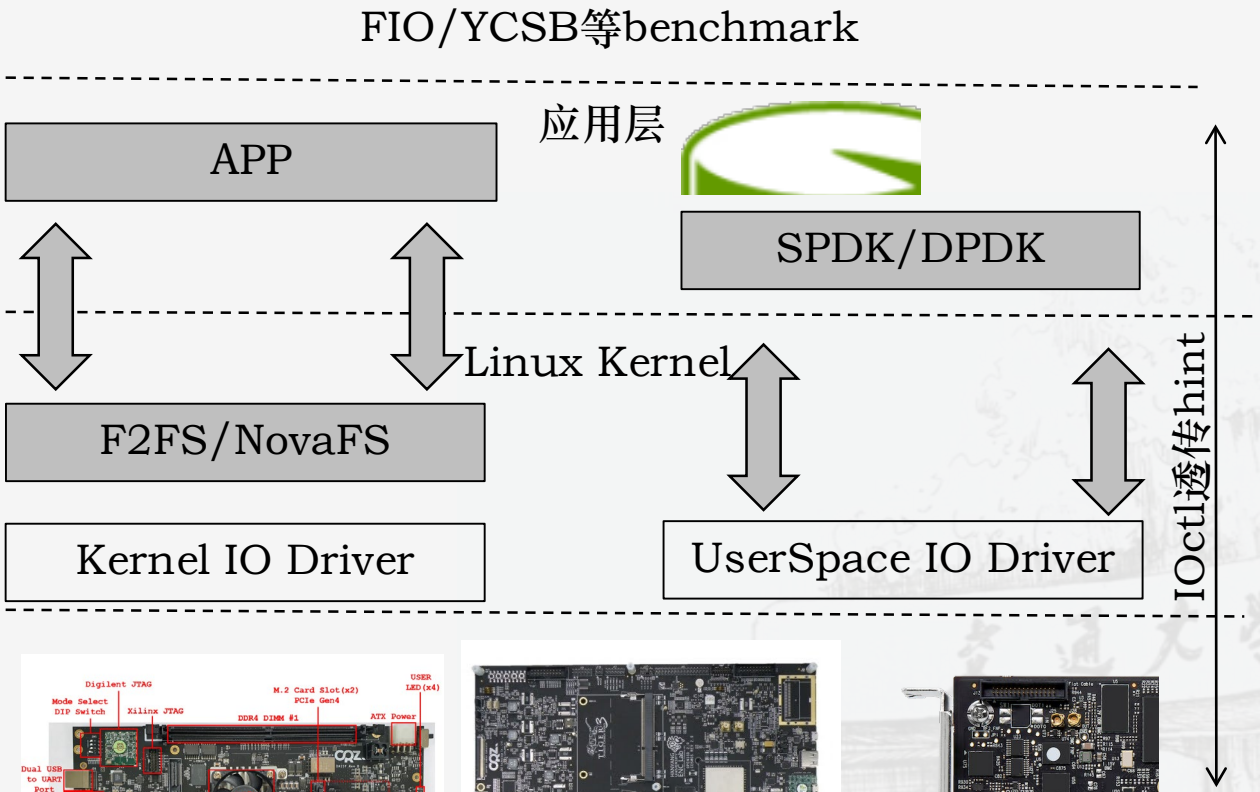
① 硬件/设备仿真: 模拟出完整的计算机系统，包括CPU、内存和外部设备，如硬盘、网络接口、USB设备等。

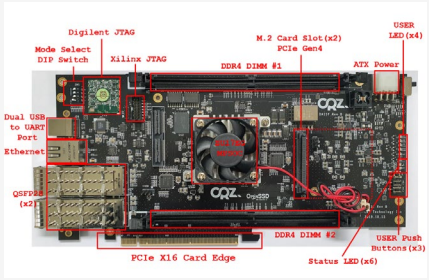
② 虚拟化: 结合KVM(Kernel-based Virtual Machine)内核模块，被广泛用于服务器虚拟化、桌面虚拟化、云计算等领域。

③ Linux内核调试: 提供安全、隔离内核运行环境，适用于操作系统内核的开发和调试。

④ 跨平台开发: QEMU支持多种CPU架构（如x86、ARM、MIPS、RISC-V等），例如，在x86主机上开发ARM架构的应用程序，而不需要实际的ARM设备。

一、QEMU模拟器和存储设备研究





OpenSSD

Cosmos/Cosmos+ OpenSSD
FPGA Platform

BlueField DPU

基于DPU/SSD等存储设备的研究架构

存储软硬件协同优化研究思路：

① 根据存储设备的特性优化键值存储、文件系统；

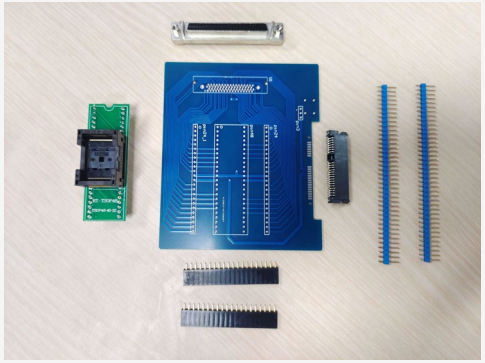
② 将应用特性透传至设备，设备主动辅助优化应用性能。

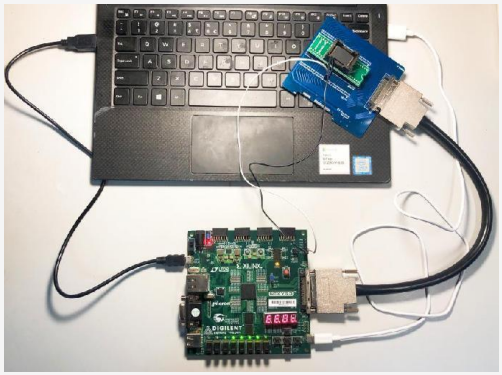
面临的挑战/问题：

① 存储设备固件可修改、可定制；

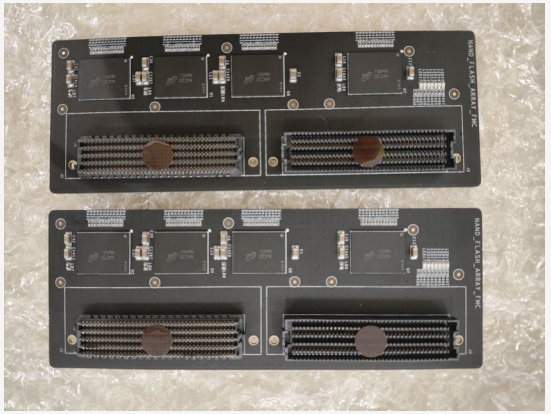
② Linux内核/应用二次修改，调试方便。

一、QEMU模拟器和存储设备研究





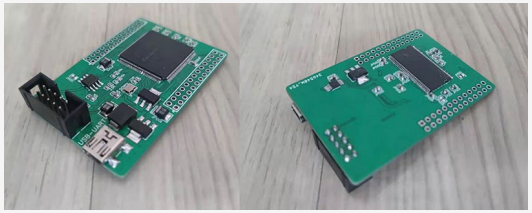
采用FPGA实现NAND闪存和mRAM的固件逻辑，采用状态机控制读写操作；
从可靠性、读写延迟等指标测试芯片存储机制。

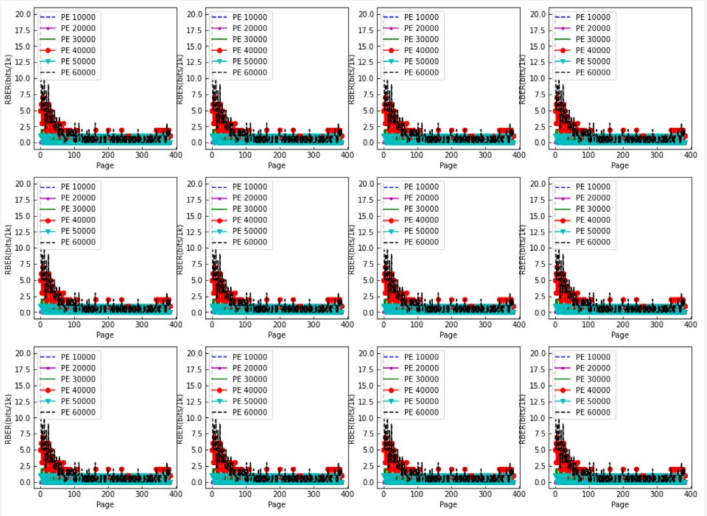


定制化PCB闪存和MRAM测试板

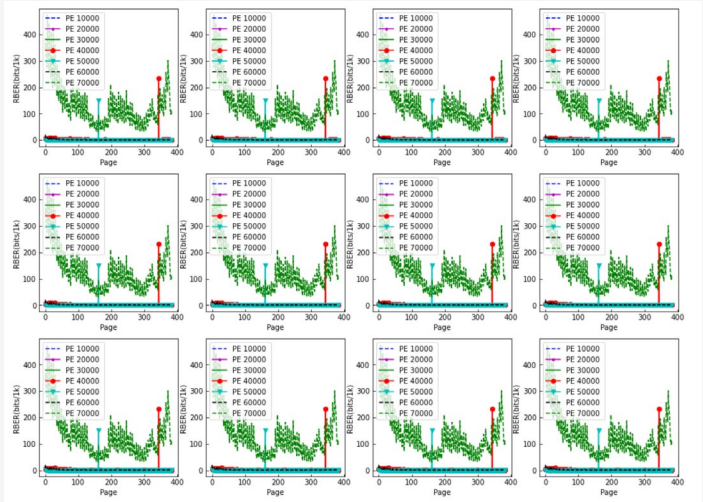
FeRAM测试板(左)

DRAM测试板(右)

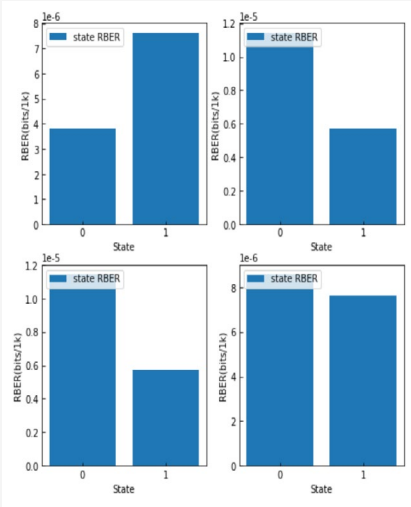




2D 不同block内page的RBER错误率分布情况

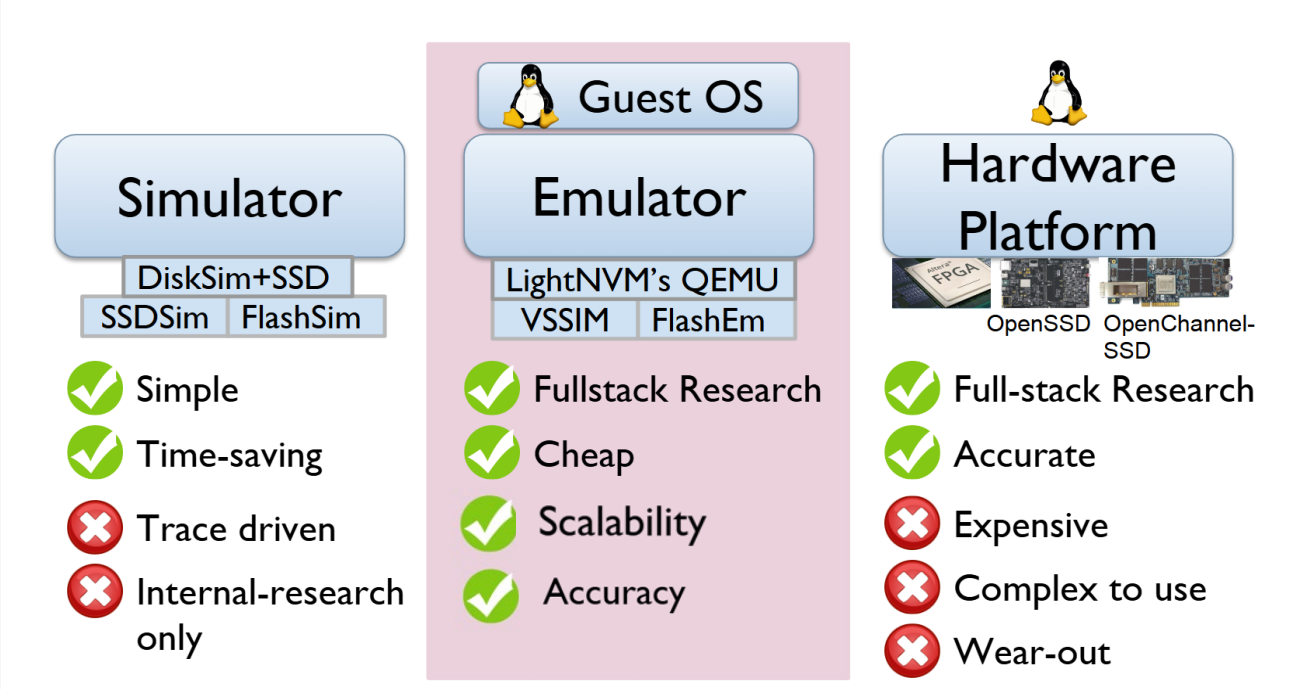


3D 不同block内page的RBER错误率分布情况



写入数值分布与RBER

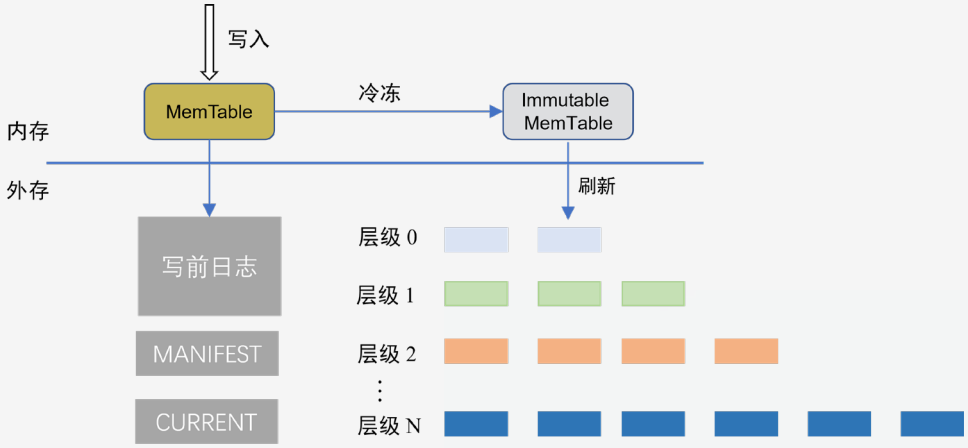
一、QEMU模拟器和存储设备研究



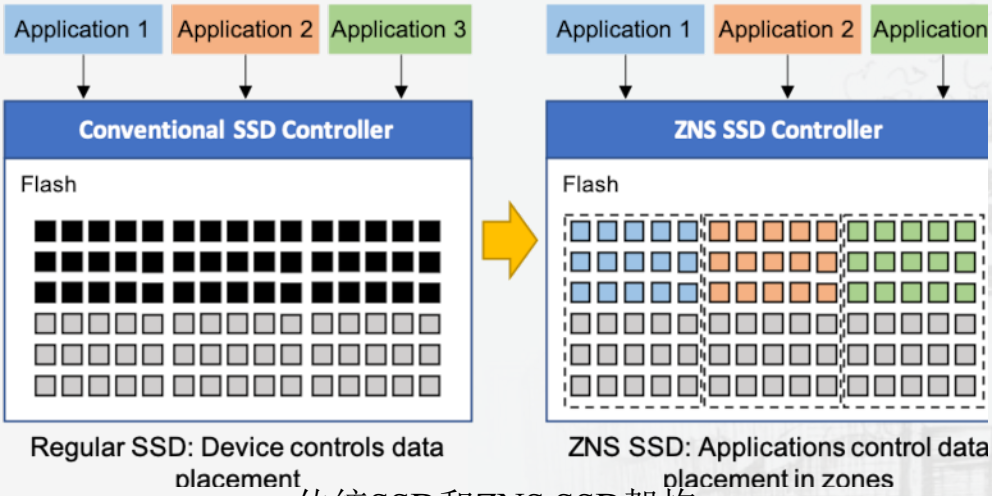
SSD设备实验平台方法比较

基于QEMU模拟器搭建实验平台，设计、优化、验证存储系统的算法/策略

二、基于ZNS SSD的键值存储系统数据安全删除



LSM日志结构合并树架构



传统SSD和ZNS SSD架构

日志结构合并树（Log-Structured Merge-Tree, LSM-Tree）是一种专为高效数据写入设计的数据结构，日志结构合并树的核心思想是先在内存中累积写操作，然后统一顺序写入到硬盘，从而减少硬盘I/O操作的频率、提高写入速度。

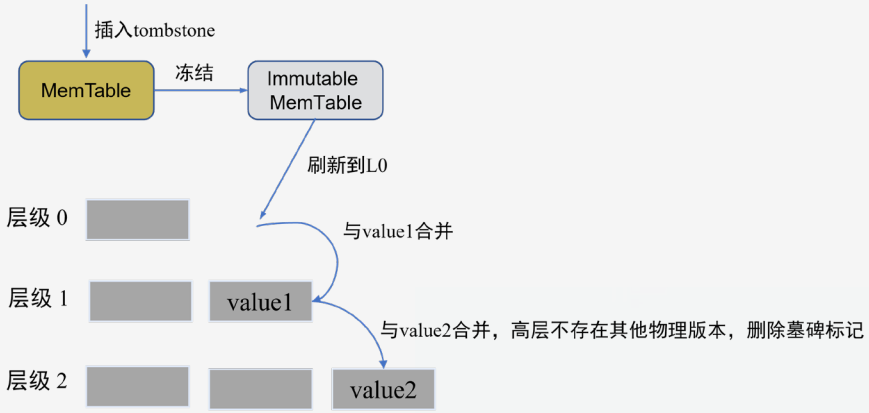
特点：顺序写入、异地更新、冷热分层

ZNS SSD分区内部只允许进行顺序写入，但可以进行随机读取。分区命名空间固态盘的设计允许主机端直接负责垃圾回收和数据布局。

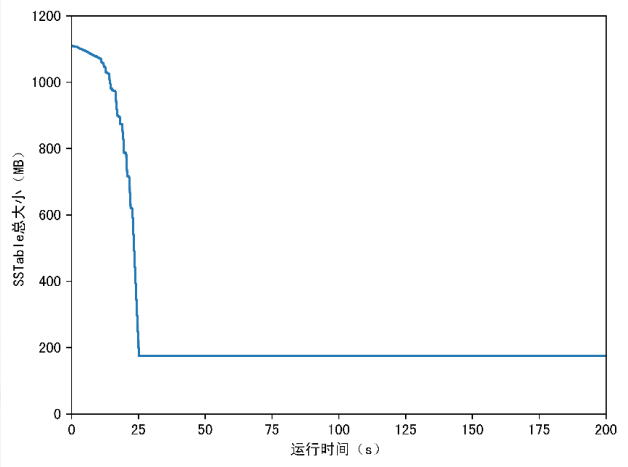
动机：日志结构合并树的外存追加更新思想符合分区命名空间固态盘的顺序写限制，其合并操作先顺序写入后集中删除，与分区命名空间固态盘的空间回收方式相一致。

Nie, Shiqiang and Lei, Tong and Li, Menghan and Niu, jie and Liu, song and Wu, Weiguo.Zone-Awared Persistent Deletion for LSM-tree Key Vaule Store Engine. The 13th IEEE Non-Volatile Memory Systems and Applications Symposium. 2024.

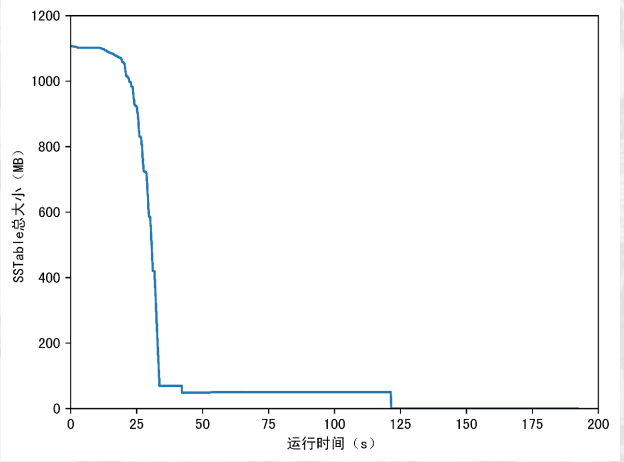
二、基于ZNS SSD的键值存储系统数据安全删除



删除操作示意图



SSTable文件总大小随时间变化关系



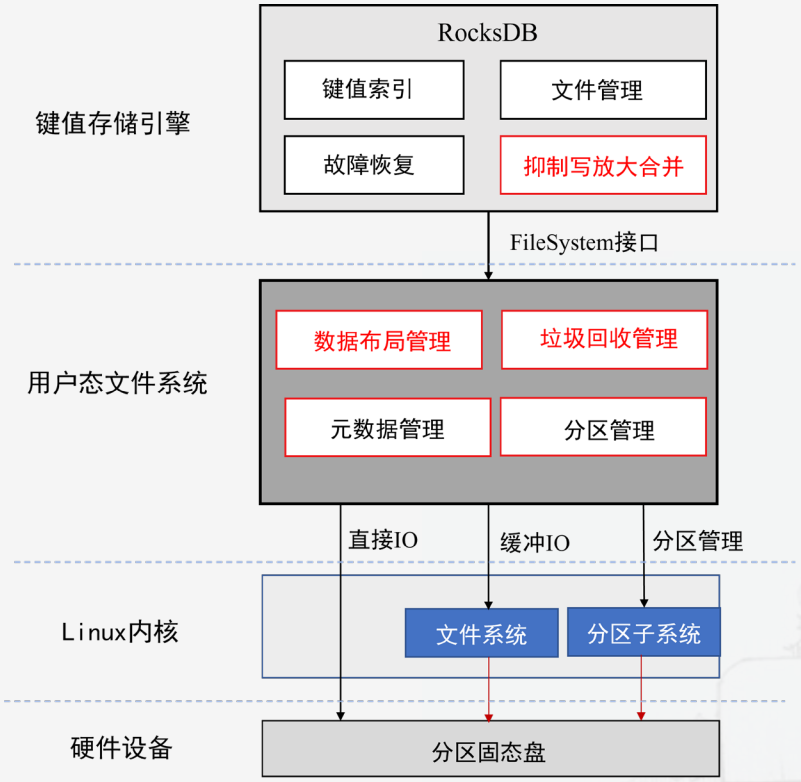
Fade策略下SSTable文件总大小随时间变化关系

对于删除操作，日志结构合并树通过插入一条特殊的删除记录来表示特定键值对的删除状态，即墓碑标记。持久化删除是指通过层层合并，在物理上删除所有过期版本和墓碑标记的操作。

RocksDB并未提供针对持久化删除延迟的保证，这可能引发一系列数据隐私问题。

根据Lethe论文的实验结果显示，在将持久化删除阈值设置为16%的实验运行时间，删除操作数占总操作数的10%时Fade策略相比于最小重叠比率算法增加了约30%的写入量。

二、基于ZNS SSD的键值存储系统数据安全删除



分区感知的键值存储持久化删除优化策略架构

➢ 抑制写放大的层级数据合并方法

- 对于过期墓碑文件
  移至队列前半部，以过期时间长短与墓碑标识数目确定内部排序
- 对于非过期墓碑文件的低层级SST文件
  使用最小重叠比率算法计算权值，确定队列后半部排序优先级
- 对于非过期墓碑文件的高层级SST文件
  考虑日志结构合并树、分区命名空间固态盘以及墓碑文件三方面的因素，确定队列后半部排序优先级

➢ 删除操作启发式数据布局和垃圾回收方法

使用合并策略赋予的存活时间辅助数据布局：
若存活时间小于Level 1的平均寿命，则将墓碑文件与低层级SSTable存放在一起；
若存活时间小于当前层级的第75个百分位数寿命，则将其存放独立的分区之中；
默认不对墓碑文件的存放位置进行特殊处理。

二、基于ZNS SSD的键值存储系统数据安全删除

实验环境：

使用RocksDB、ZenFS和ZNS SSD搭建实验环境，其中ZNS SSD使用Femu仿真器进行模拟；

工作负载：

使用Lethe论文提供的工作负载生成器生成所需测试数据；

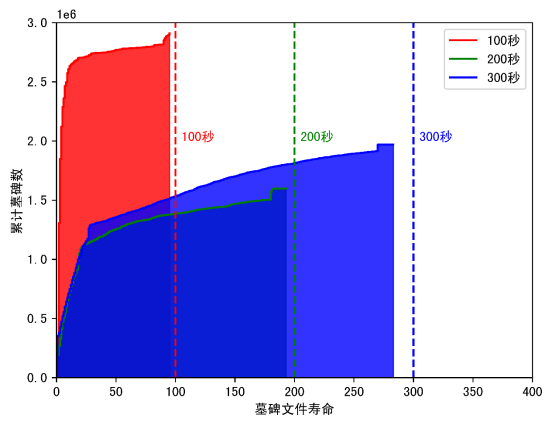
对比方案：

RocksDB+ZenFS：饱和度触发+最小重叠比率算法；

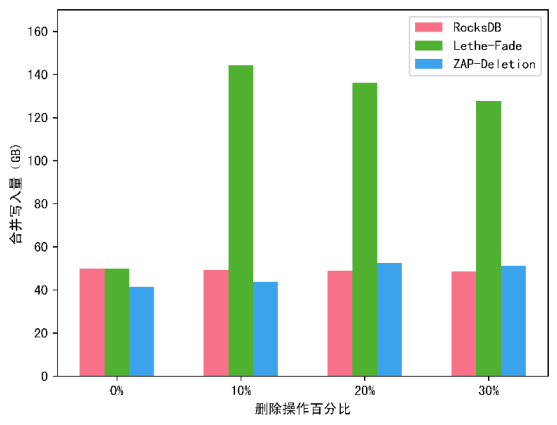
Lethe-Fade+ZenFS：饱和度触发/存活时间过期触发+最小重叠比率算法/过期时间排序/墓碑数量排序；

ZAP-Deletion：抑制写放大的层级数据合并方法 + 删除操作启发式数据布局和垃圾回收方法；

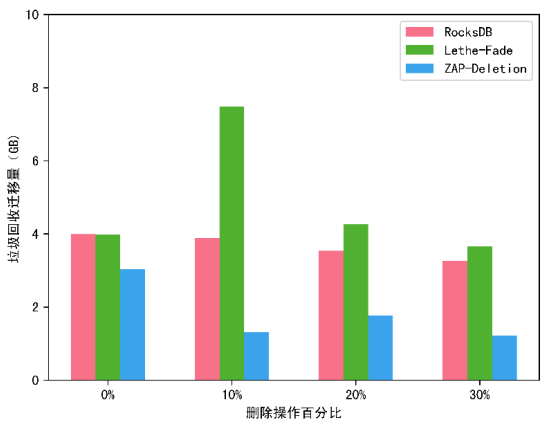
二、基于ZNS SSD的键值存储系统数据安全删除



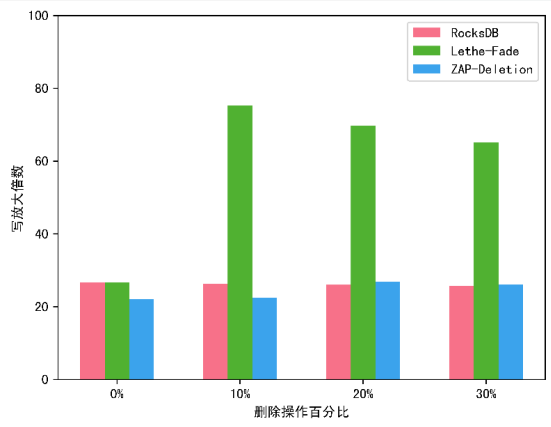
墓碑文件寿命与累计墓碑数变化关系



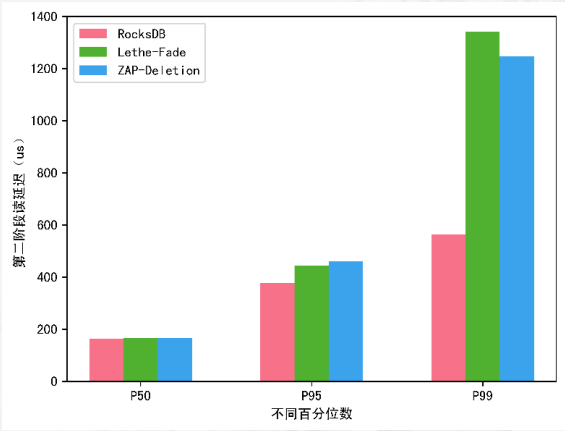
合并写入量与删除百分比关系



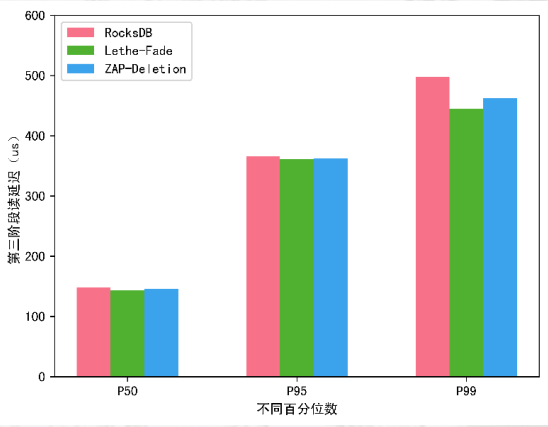
垃圾回收迁移量与删除百分比关系



写放大倍数与删除百分比关系

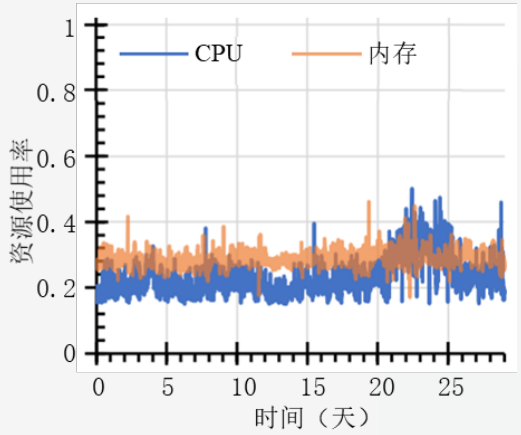


第二阶段不同百分位读延迟图

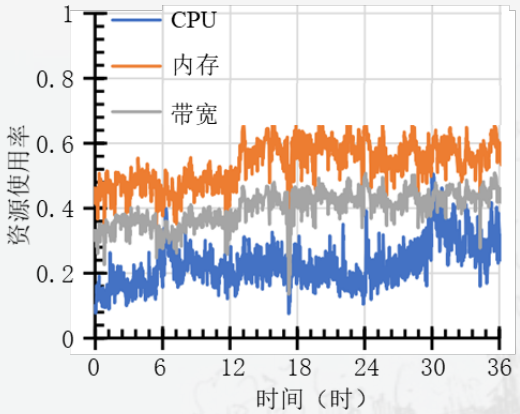


第三阶段不同百分位读延迟图

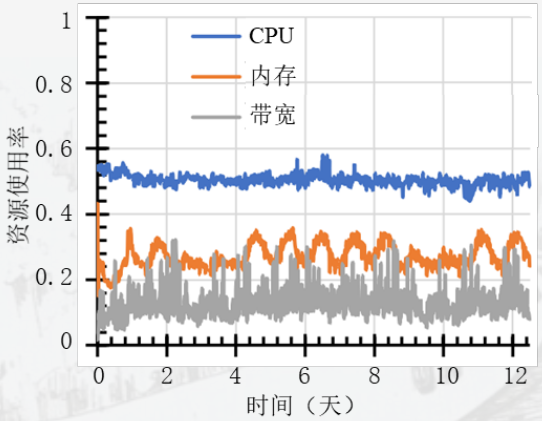
三、基于DPU的交换空间设计与优化



（a）Google Cluster



（b）Alibaba Cluster



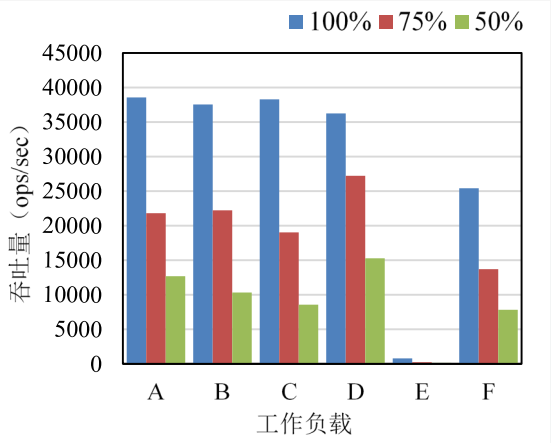
（c）Snowflake Cloud

谷歌和阿里巴巴数据中心及Snowflake云数据仓库的trace表明：

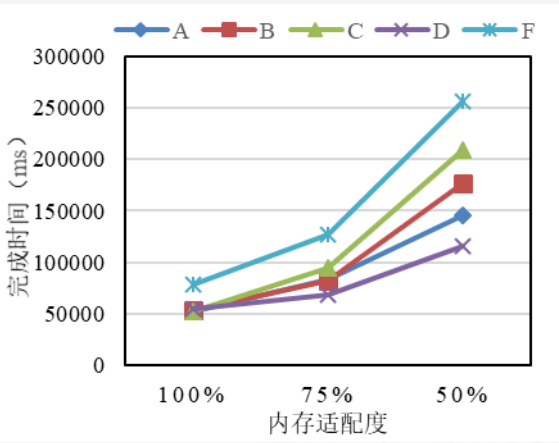
内存资源利用率甚至不足整个数据中心资源的一半，存在大量的未利用内存。

三、基于DPU的交换空间设计与优化

◆ 现有分离式内存的性能问题

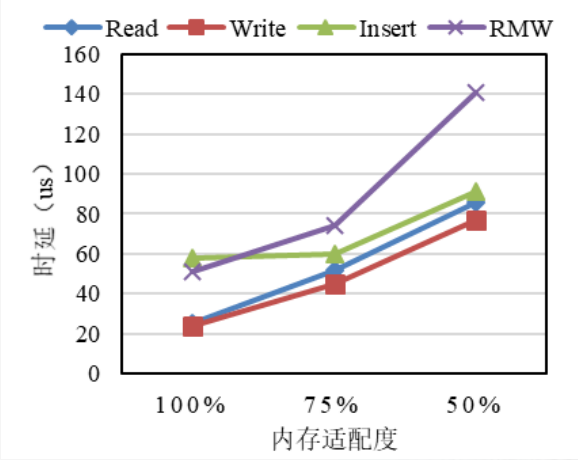


（a）吞吐量

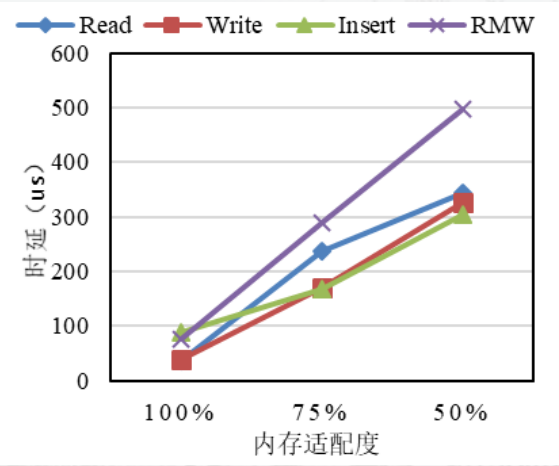


（b）完成时间

75%内存：
吞吐量减少46.7%
完成时间增加59.9%
50%内存：
吞吐量下降69.9%
完成时间增加2.1倍



（c）平均延迟

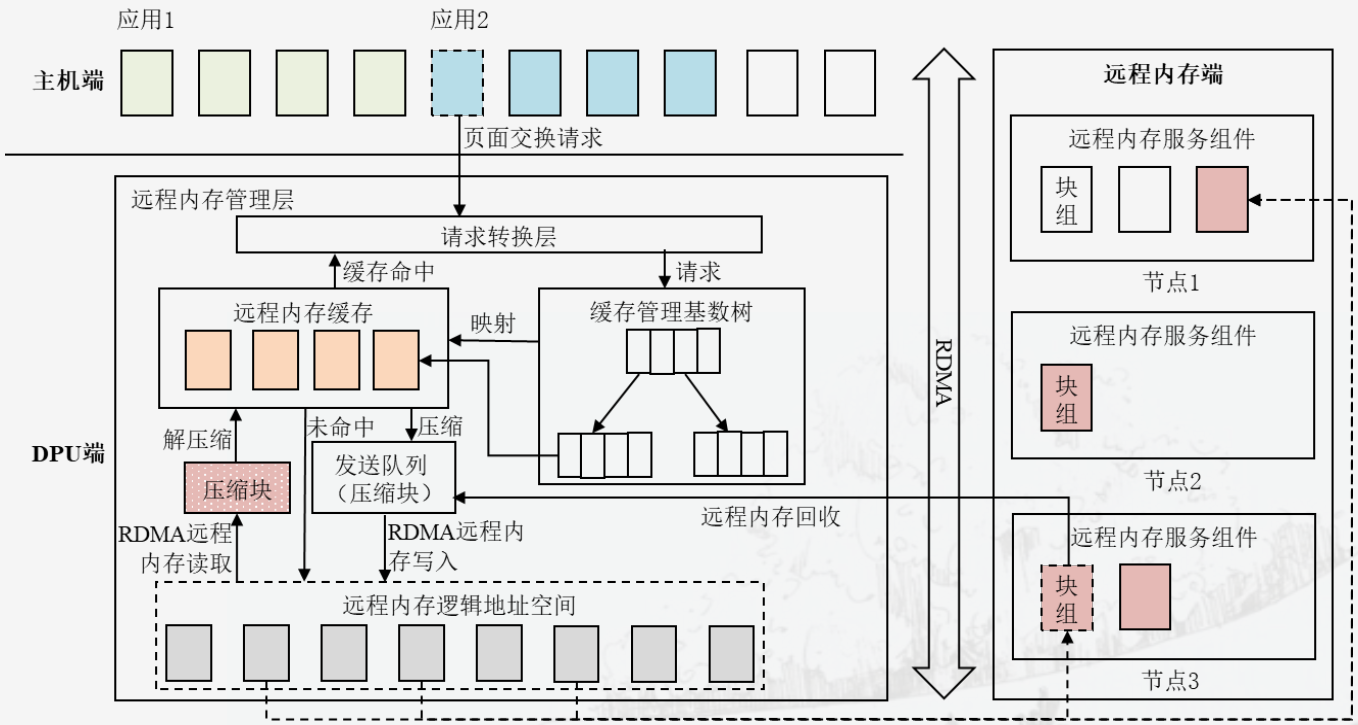


（d）P99延迟

75%内存：
平均延迟增加56.6%
P99分位延迟增加3.6倍
50%内存：
平均延迟增加2.1倍
P99分位延迟增加6.5倍

网络链路延迟限制分离式内存的效率(RDMA us　PCIe Link ns)

三、基于DPU的交换空间设计与优化



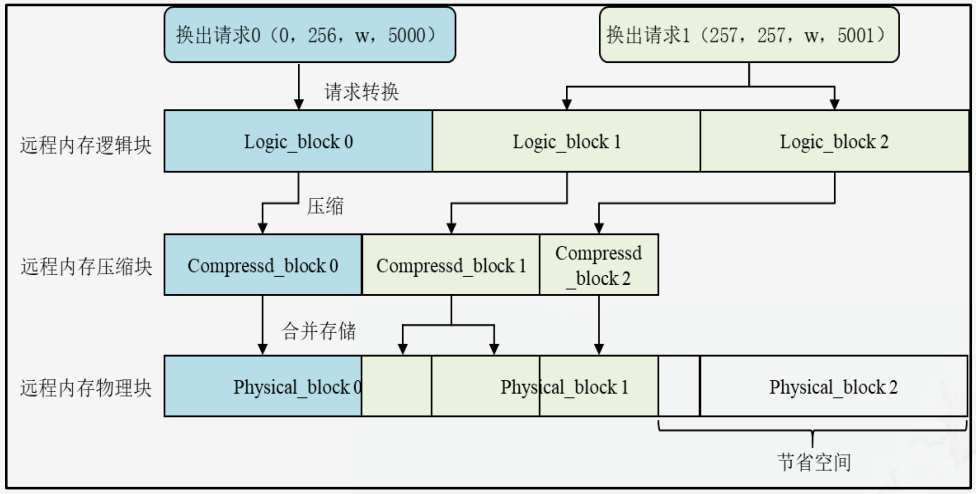
本地DPU侧：

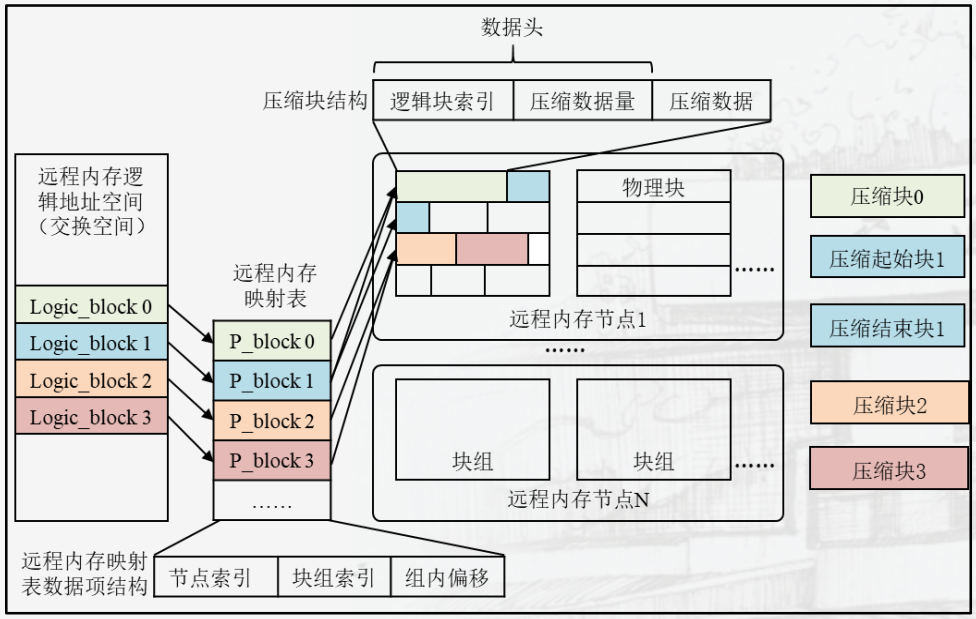
运行远程内存管理层（Remote Memory Management Layer，RMML），实现远程内存的地址映射和无效块回收以及集群节点负载均衡，另外RMML使用本地DPU内存用作远程内存缓存，以加快访问速度。 RMML通过DPU向主机提供虚拟连续存储空间，可以被配置为主机交换区。

远程内存节点侧：

运行远程内存服务组件，监控和管理本地内存使用情况，响应RMML请求。

三、基于DPU的交换空间设计与优化



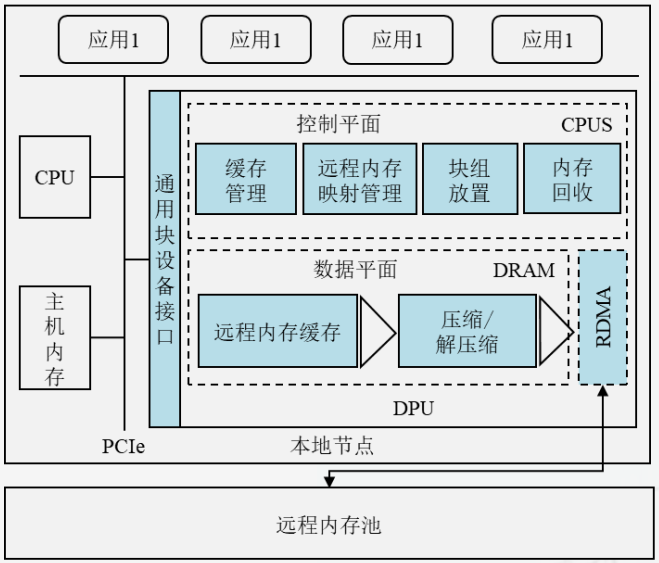


- **粗粒度**：使用1MiB的远程内存块为基本粒度组织远程内存，达到寻址表项占用空间与灵活寻址的折中
- **动态压缩**：内存块通过压缩拼接紧凑存放于远程内存，节省远程内存占用；每个压缩块需要添加额外的数据头维护其元数据信息，包括逻辑块地址，压缩长度
- **局部性布局**：N个远程逻辑块组成一个块组，以块组为单位映射至一块连续的远程内存中

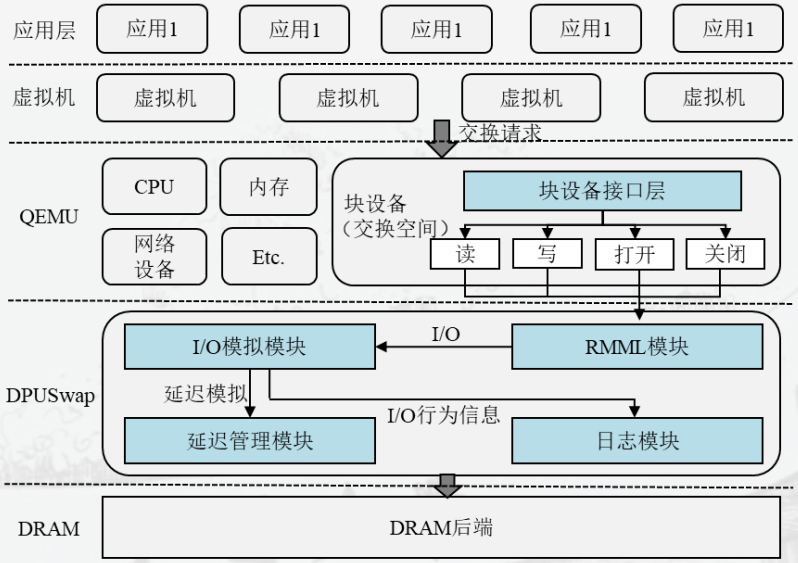
三、基于DPU的交换空间设计与优化

实验设置

- DPUSwap的软硬件原型包括应用程序+物理主机操作系统+远程内存池
- 模拟环境实现为应用程序+虚拟机操作系统+DPUSwap模拟块设备



（a）软硬件原型



（b）模拟环境

对比方案

- 使用本地高速固态硬盘作为交换空间的ZNSwap
- 现有分离式远程内存交换空间Infiniswap

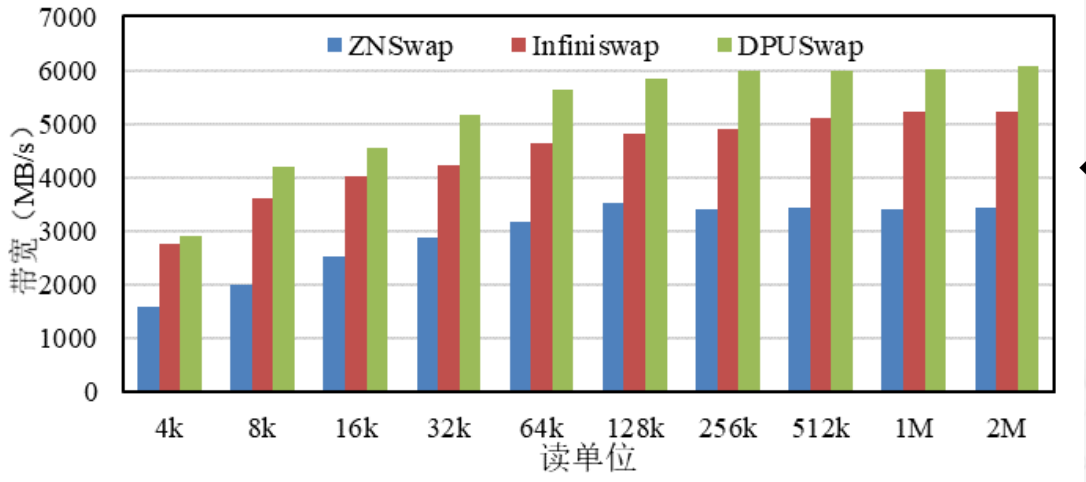
评价指标

- 带宽、吞吐量、完成时间、延迟、压缩率、缓存命中率

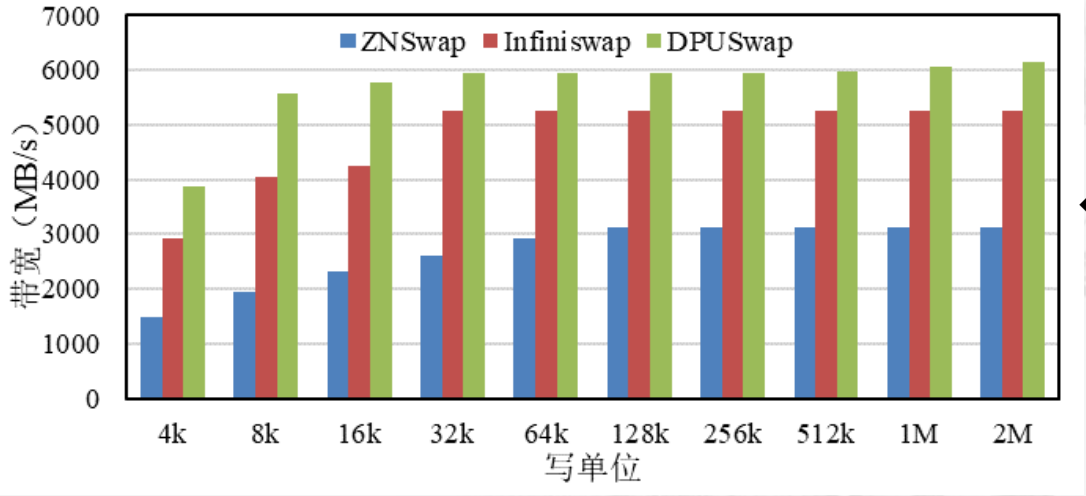
测试工具

- Fio、YCSB

三、基于DPU的交换空间设计与优化



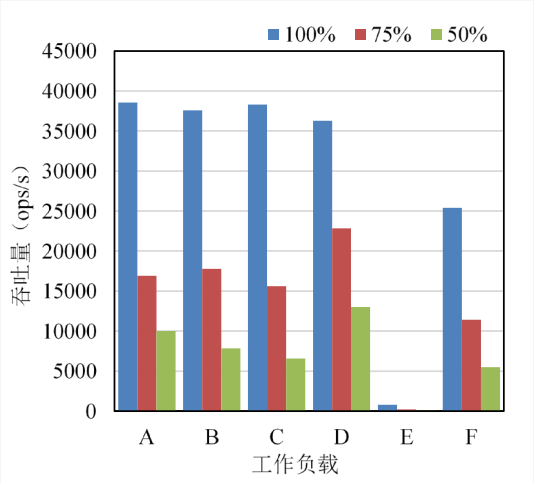
- 随机读
  - 相对ZNSwap提升81.7%
  - 相对Infiniswap提升为17.9%



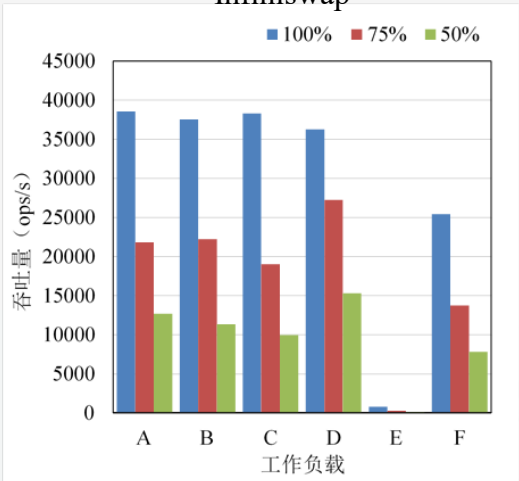
- 随机写
  - 相对ZNSwap提升1.28倍
  - 相对Infiniswap提升20.4%

三、基于DPU的交换空间设计与优化

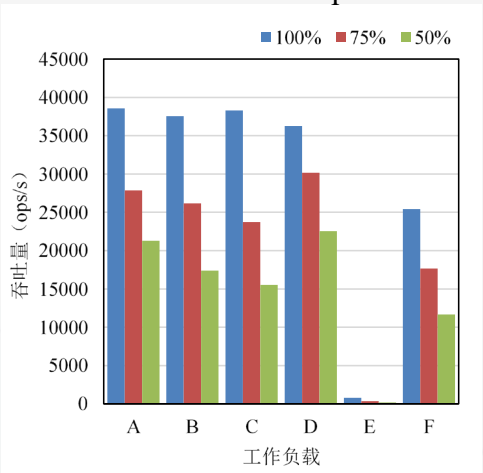
ZNSwap

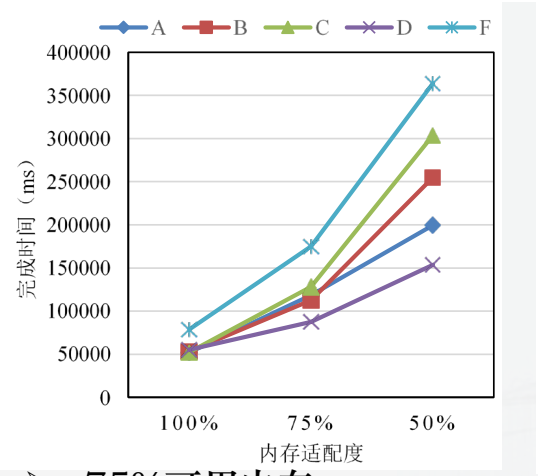


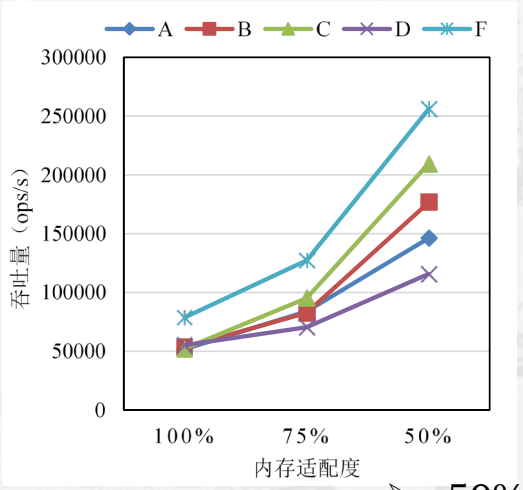
Infiniswap

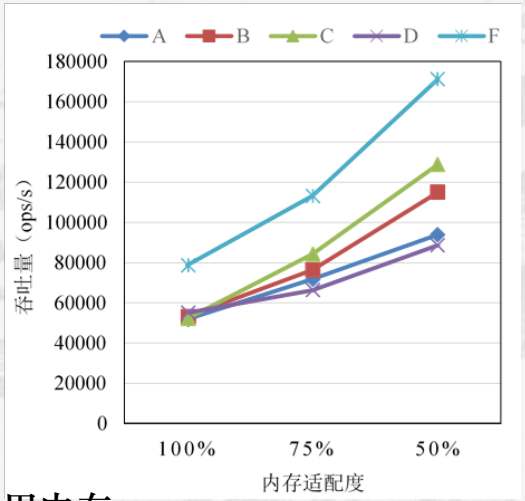


DPUSwap









- 75%可用内存

相较Infiniswap有26%的吞吐量提升，完成时间降低10%

相较ZNSwap平均有53.3%的吞吐量提升，完成时间降低34.1%

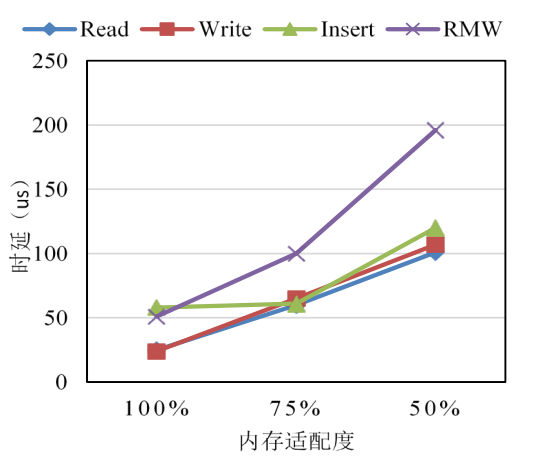
- 50%可用内存

相较Infiniswap吞吐量提升55.5%，完成时间减少33.1%

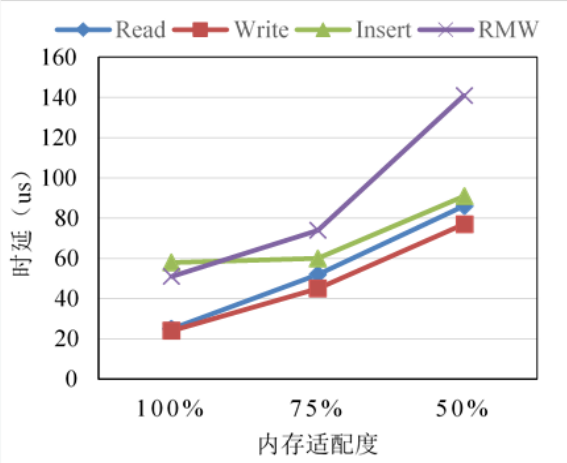
相较ZNSwap吞吐量提升1.1倍，完成时间减少52.1%

三、基于DPU的交换空间设计与优化

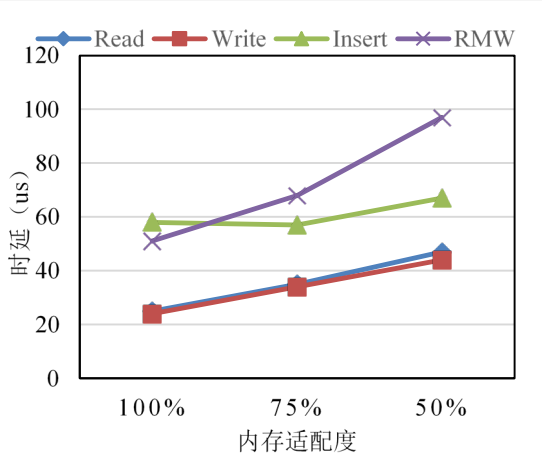
ZNSwap

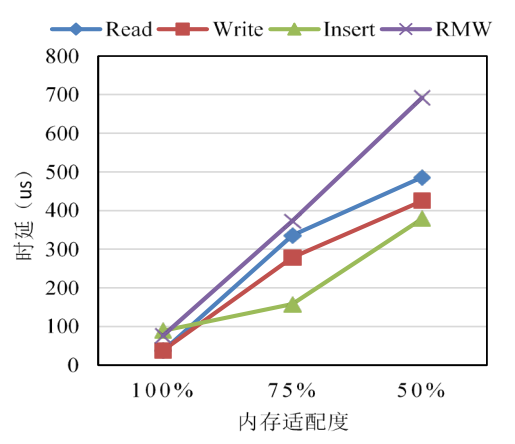


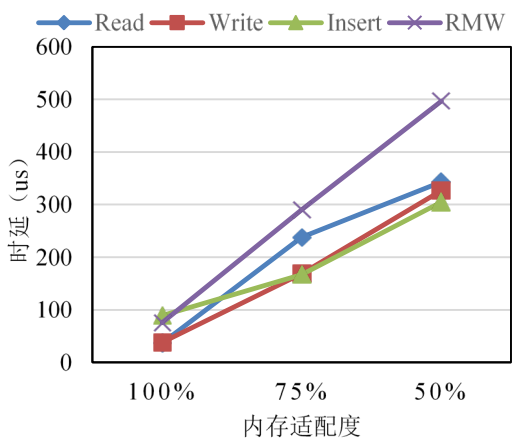
Infiniswap

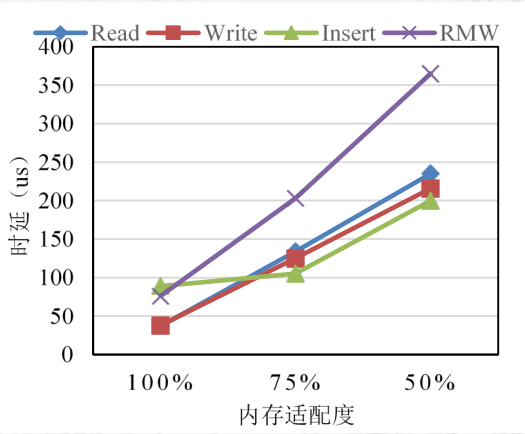


DPUSwap









- 75%可用内存

相较Infiniswap平均延迟减少20%，P99延迟则减少33.4%

相较ZNSwap平均延迟减少31.9%，P99延迟减少了47.6%

- 50%可用内存

相较Infiniswap平均延迟减少38.8%，P99延迟减少31.8%

相较ZNSwap平均延迟减少51.4%，P99延迟减少了49.4%

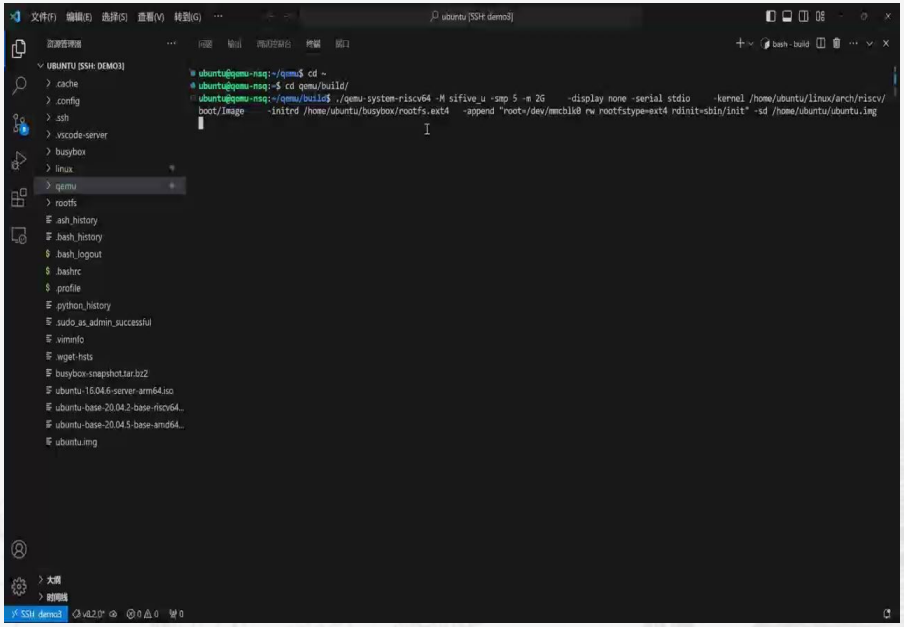
四、SPI Flash仿真器设计与实现

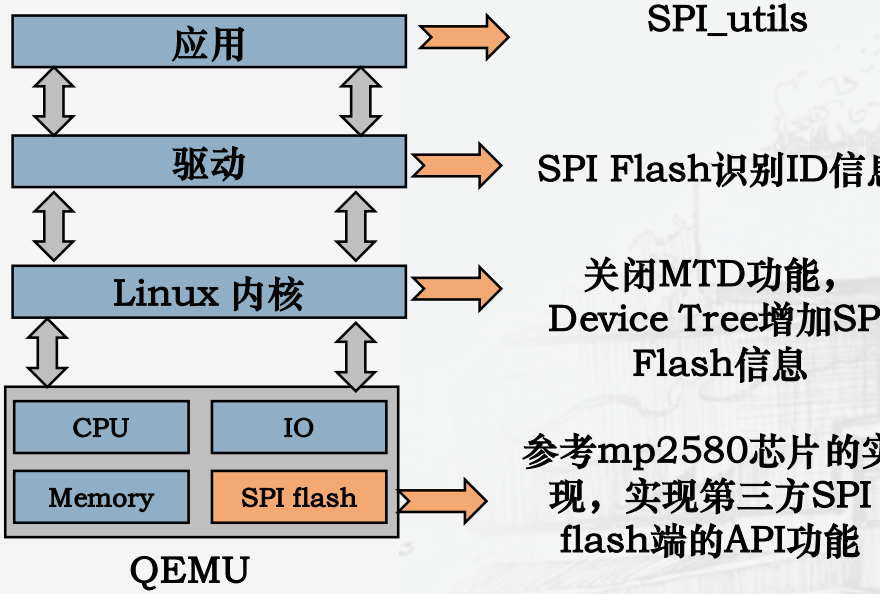
针对的问题：

- 嵌入式软件开发，测试依赖硬件设备交付；
- 全物理、半物理环境测试手段单一；
- 缺少型号软件测试与系统测试工作平台；

解决方案：

- 全数字的系统级仿真框架；
- 基于状态机模型的模拟器架构；





SPI_utils

SPI Flash识别ID信息

关闭MTD功能，Device Tree增加SPI Flash信息

参考mp2580芯片的实现，实现第三方SPI flash端的API功能

Q&A

欢迎各位专家批评指正！