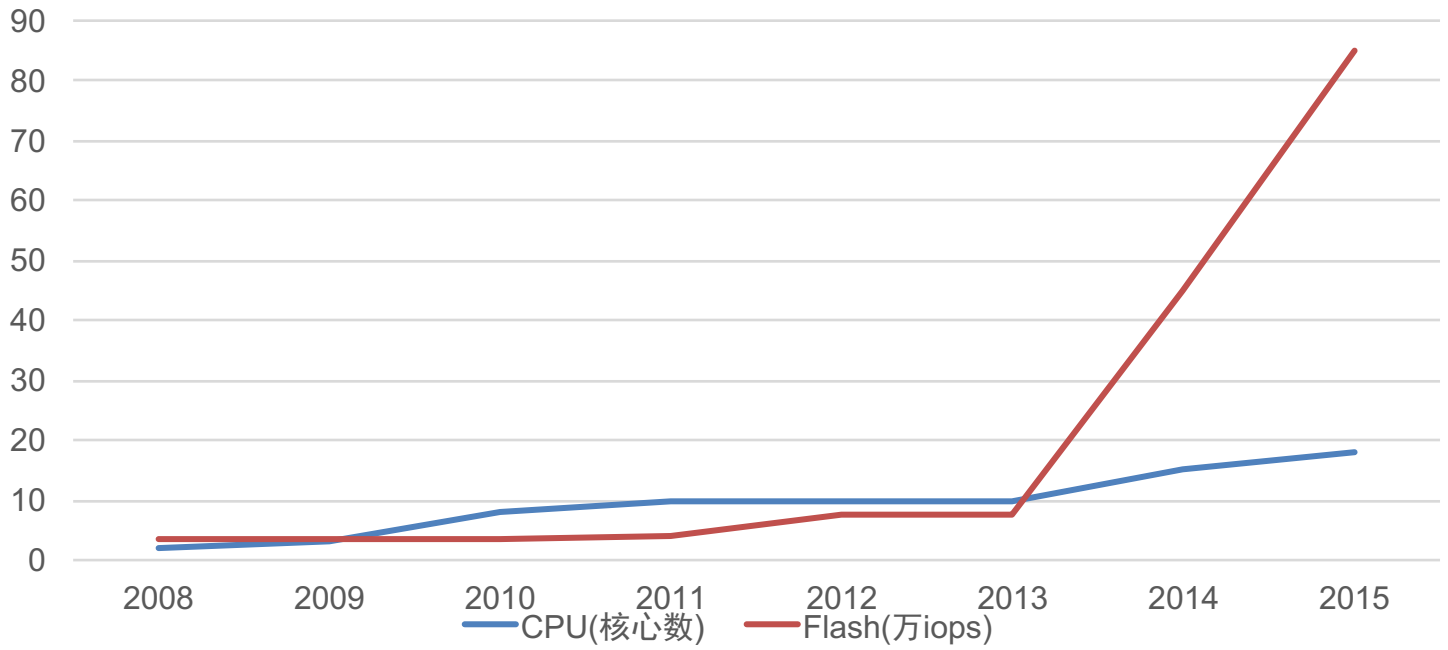
裸金属架构
软件定义存储

华云网际(fusionstack.cn)联合创始人 王劲凯

议程

- 高性能存储技术的演进
- 裸金属架构
- 数据布局

CPU和Flash发展趋势：新时代已经到来



阵列时代的高性能：高速缓存技术

- 特点
  - 磁盘性能低
  - 数据量相对小
  - 有热点
- 主要方法：缓存加速IO
  - 写缓存(NVRAM/BBU)
  - 读缓存(readahead/cache

分布式时代的高性能：并行IO

- 特点
  - 磁盘性能低
  - 数据量大
  - 热点不明显
- 主要方法：并行IO聚合磁盘处理能力
  - 数据拆分
  - 并行IO

Flash时代的高性能：裸金属架构

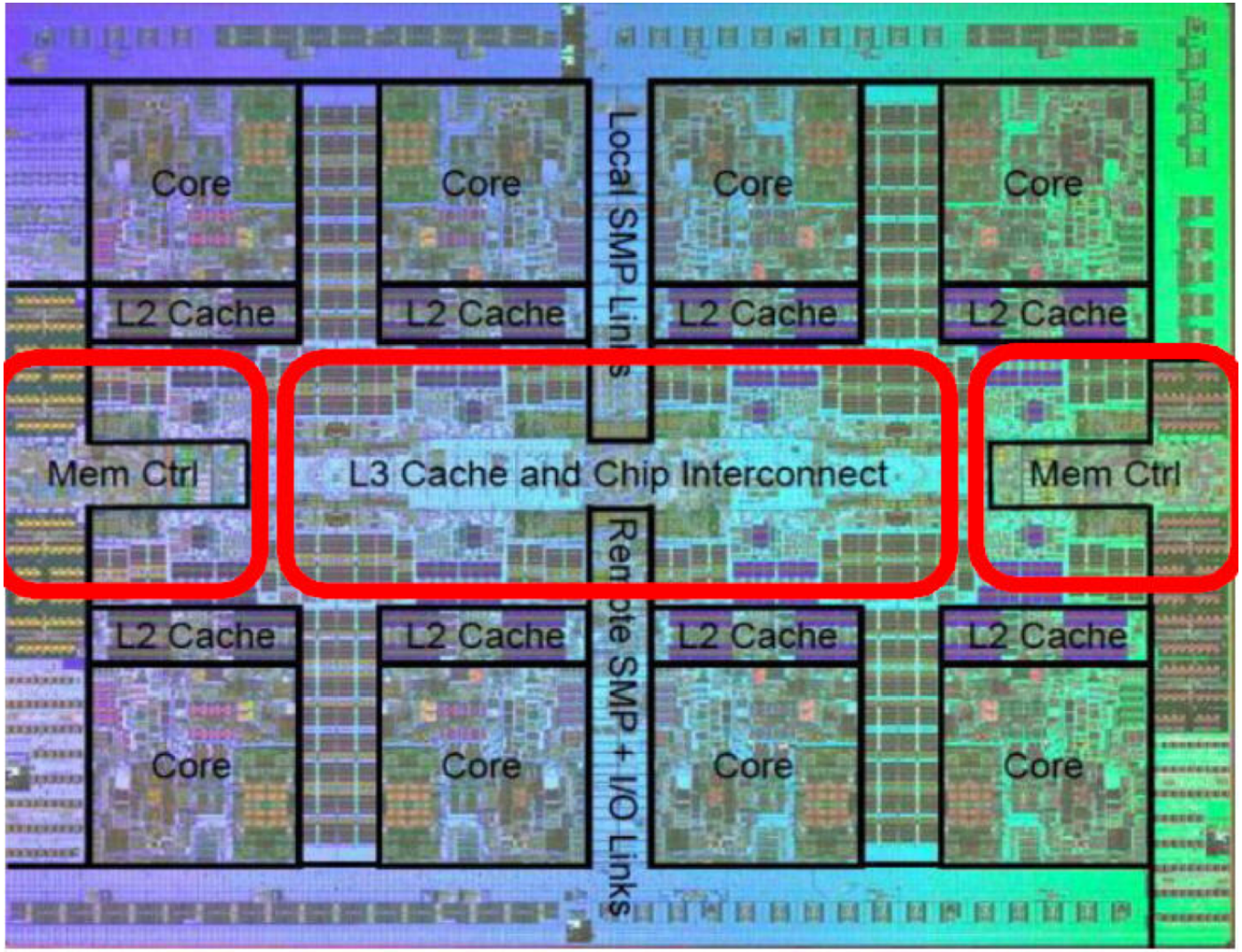
- 特点
  - Flash性能高，延迟低
  - 存储软件落后于Flash，成为存储系统瓶颈
- 主要方法： OS-bypass
  - 完全绕过操作系统编程
  - 基于裸金属重新实现一套完全为存储定制的软件堆栈

裸金属架构

性能杀手

- intterupt
- automatic
- cache miss
- TLB miss
- NUMA

Core as a Compute

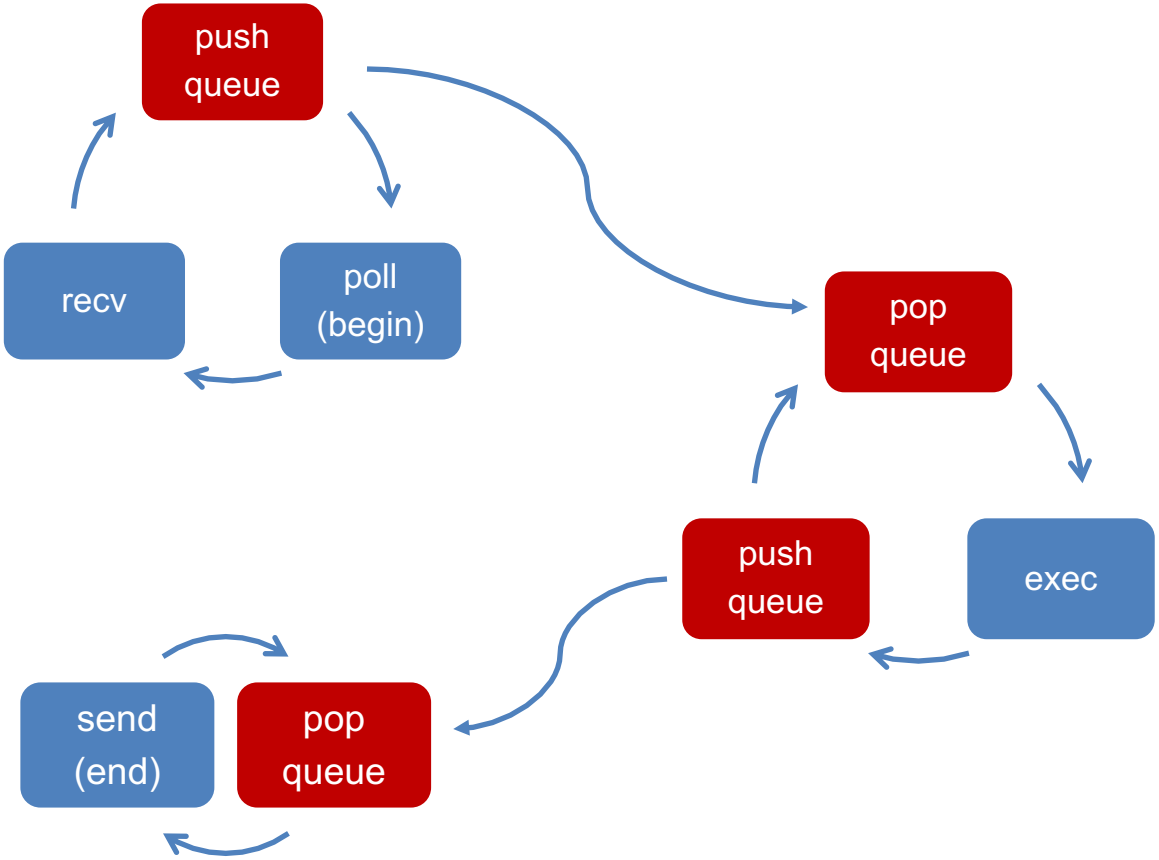


传统模型和裸金属架构对比

- 传统模型
  - 编程模型：producer-consumer
    - 任务调度：preemptive
    - 事件处理：event
    - 多核同步：lock/lock-free
  - 硬件访问：syscall
    - Network：tcp/udp
    - Flash/HDD：vfs
    - Mem：free/malloc
- 裸金属架构
  - 编程模型：run-to-completion
    - 任务调度：cooperative
    - 事件处理：polling
    - 多核同步： session based hash
  - 硬件访问：stack-bypass
    - Network：RDMA/DPDK
    - Flash：SPDK
    - Mem：hugepage

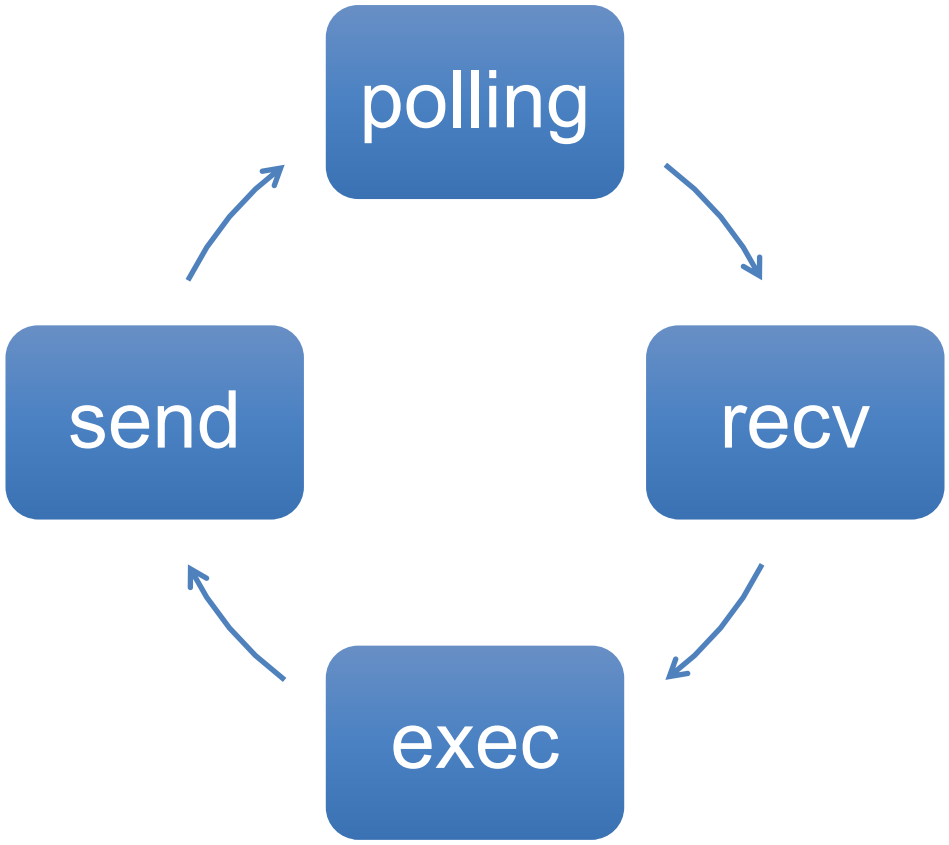
编程模型：producer-consumer

- 线程之间传递内存
  - NUMA问题
  - L2 cache miss
- 依赖锁和原子操作



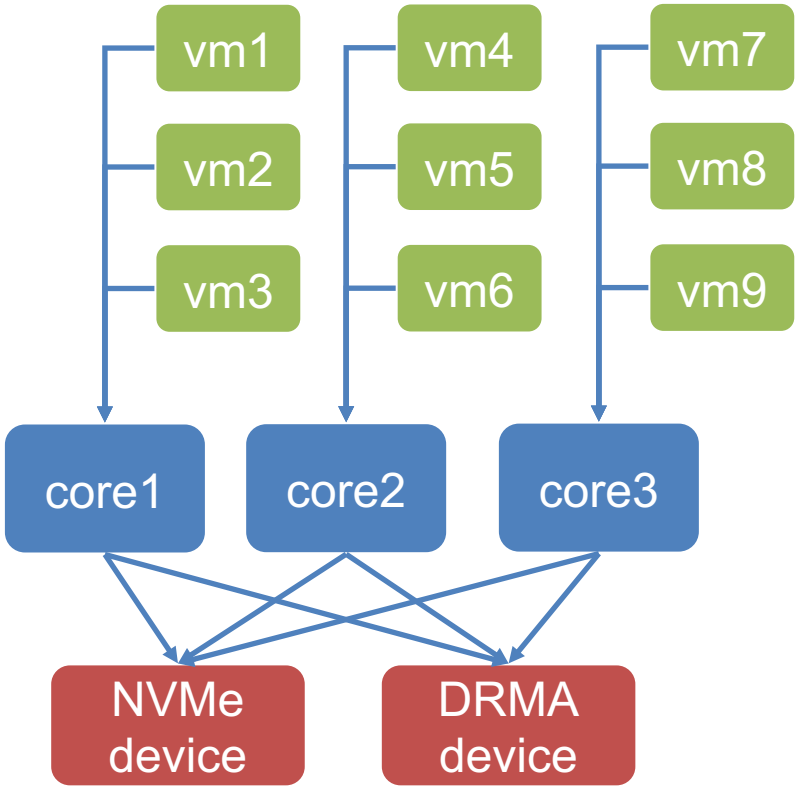
编程模型：**run-to-completion**

- 内存本地化
- 无原子操作
- 批处理



编程模型：**Session based hash**

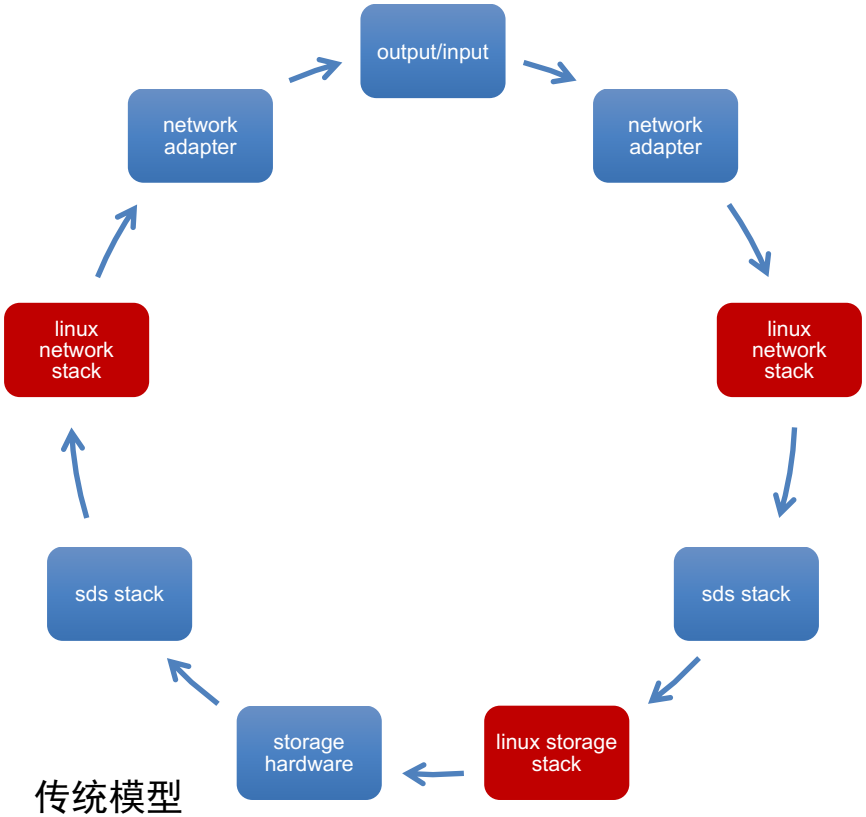
- 相同数据访问映射到相同的core上
  - 私有内存分配器
  - 私有元数据
  - 私有的connection
  - 私有NVMe Session

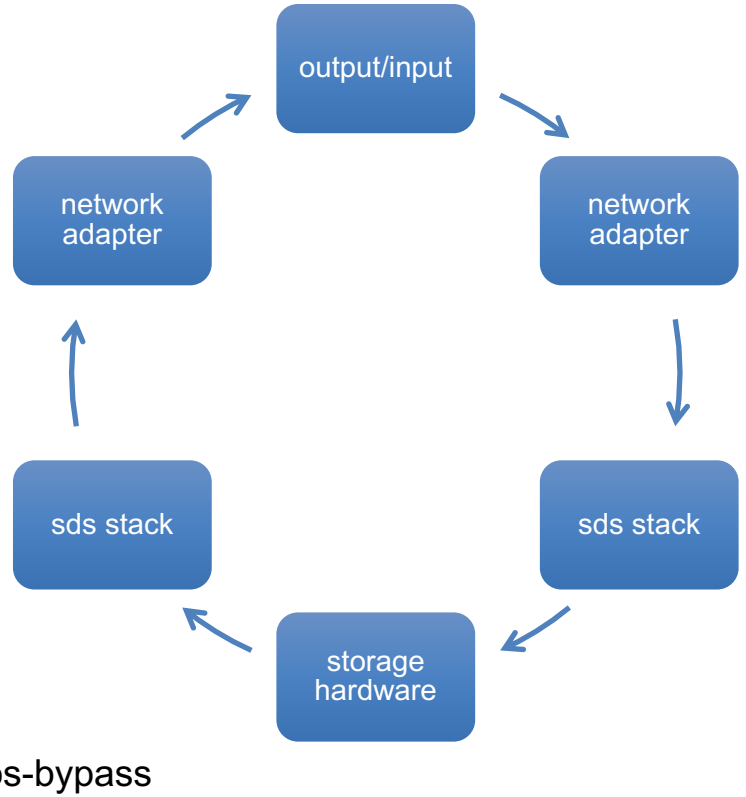


编程模型： **cooperative(coroutine)**

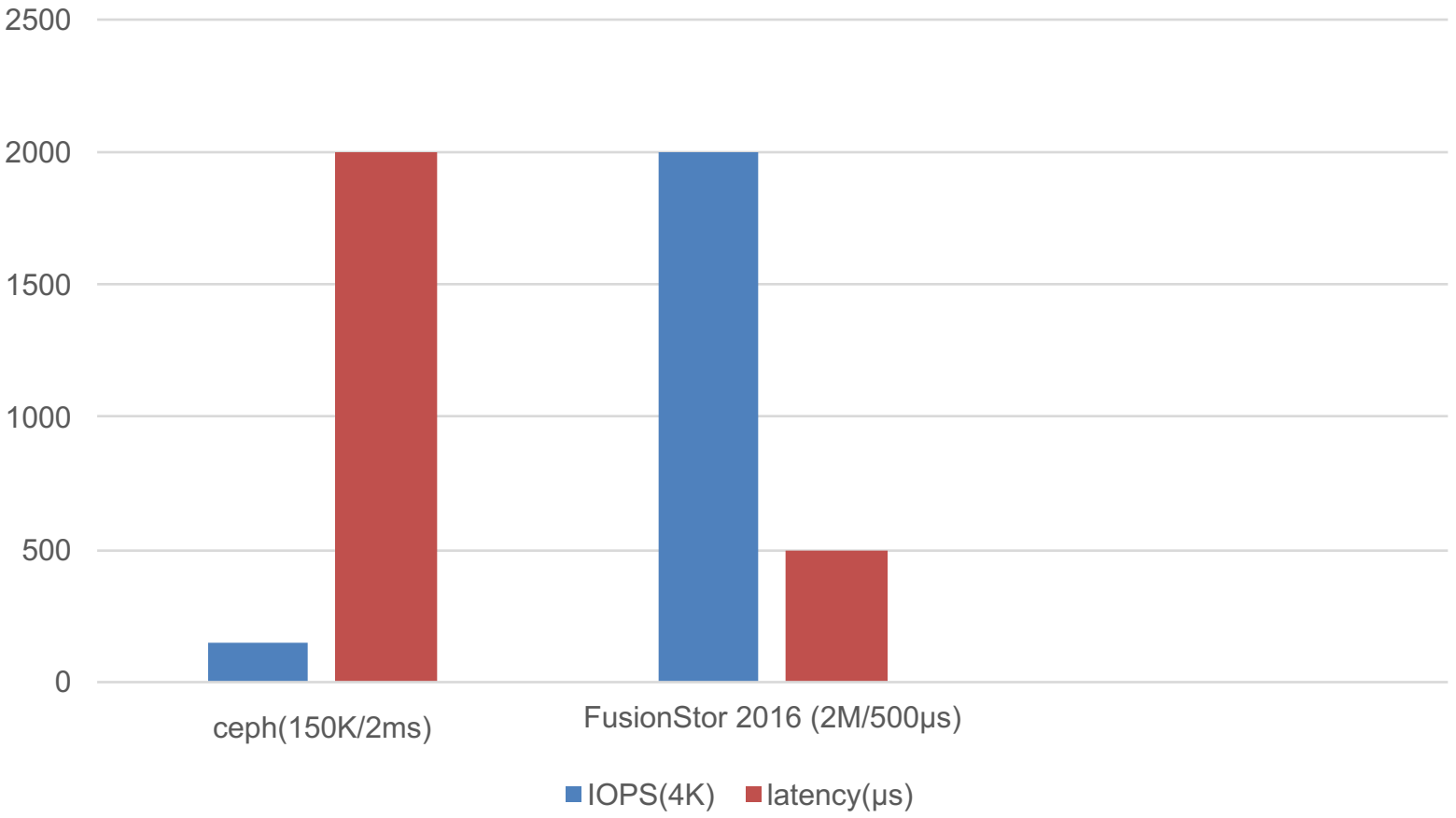
- 简化异步编程复杂度
- coroutine本身并不提升性能
- 调试困难，gdb不可用
- swapcontext开销大(150万次/秒)，需要改进

stack-bypass：IO路径对比





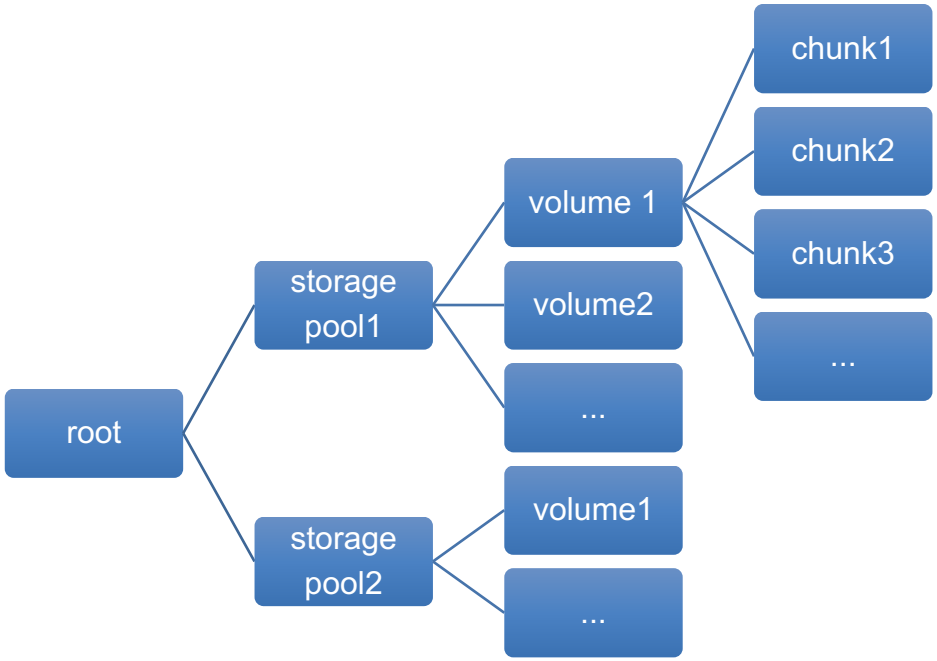
性能对比


2500
2000
1500
1000
500
0
ceph(150K/2ms)
FusionStor 2016 (2M/500μs)
IOPS(4K)
latency(μs)

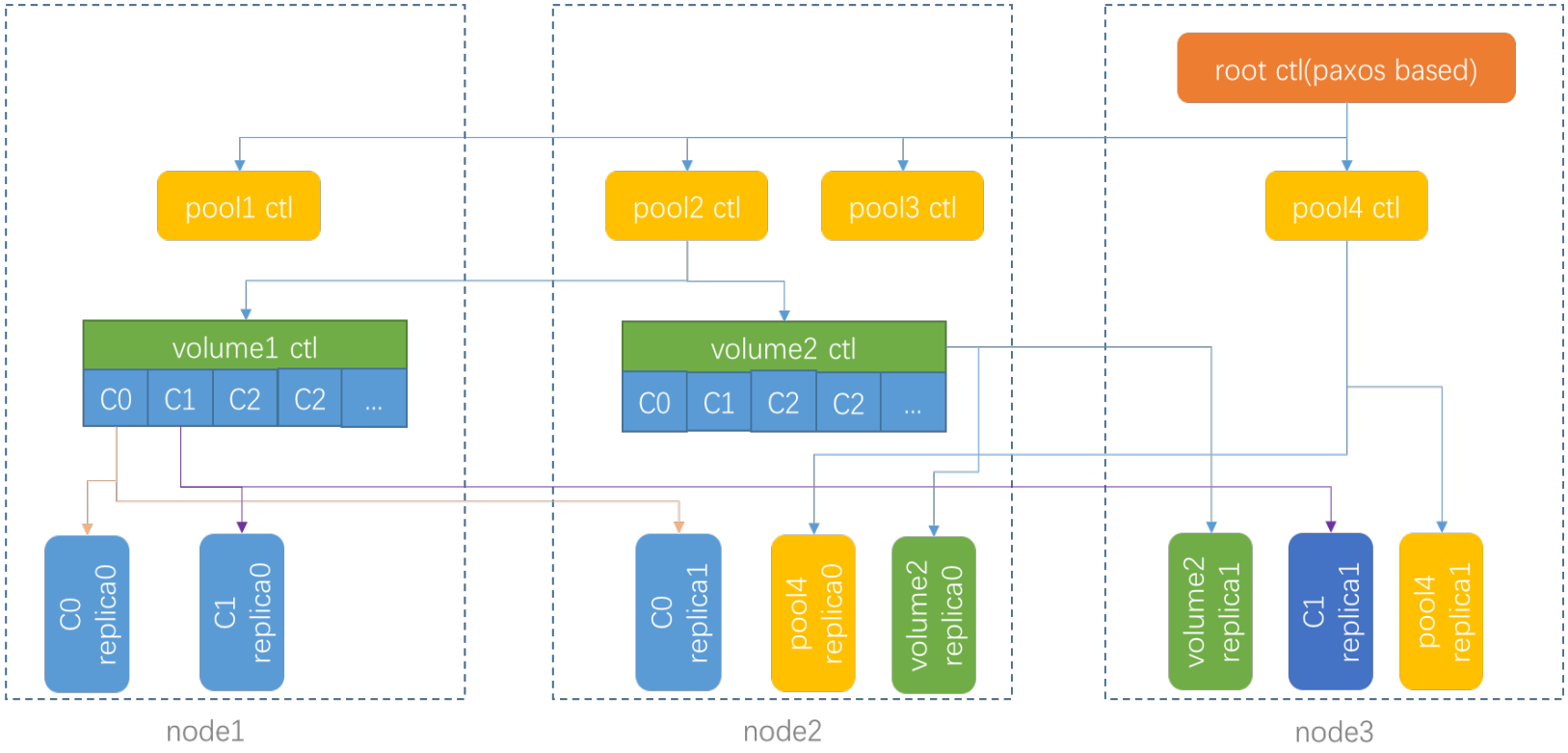
数据布局

逻辑布局

- metadata based
- 1M chunk



物理布局


root ctl(paxos based)
pool1 ctl
pool2 ctl
pool3 ctl
pool4 ctl
volume1 ctl
C0
C1
C2
C2
...
volume2 ctl
C0
C1
C2
C2
...
C0 replica0
C1 replica0
C0 replica1
pool4 replica0
volume2 replica0
volume2 replica1
C1 replica1
pool4 replica1
node1
node2
node3

Thank you