

欧立瀚

☎ 15360237778 ✉ olh8@foxmail.com



教育经历

厦门大学 985	2023年09月 - 2026年06月
计算机科学与技术 硕士 信息学院	研究方向: 存储系统
华南师范大学 211	2019年09月 - 2023年06月
计算机科学与技术 本科 计算机学院	GPA: 4.2 / 5.0

实习经历

字节跳动 AML-Engine	2025年07月 - 2025年08月
C++ 开发工程师	杭州

从事分布式机器学习的参数服务器研发的相关工作。

- **任务一**: 制作淘汰运维任务监控大盘, 并对长时间任务进行告警。基于大盘发现若干卡点任务、优化任务超时时间设置;
- **任务二**: 定位并修复部分副本淘汰期间上游错误率增加的 BUG, 优化淘汰性能。
 - **定位**: 多线程下文件数据布局被打乱, 导致预取策略失效, 发生读放大 (极端情况读放大可达数十倍), 进而导致 SSD 读流量暴增, CPU 超负载; 淘汰过程读为同步读, 读 IOPS 暴涨前提下淘汰速度被严重拖慢; I/O 任务队列是单一队列+CAS, 冲突严重。
 - **优化**: 通过使用 Thread-Local 的写 Buffer、页对齐消除读放大, 并结合 io-uring 异步批量读、使用 folly::MPMCQueue (内部通过拆分子队列来降低冲突) 提升淘汰性能, 将淘汰由最坏 10 小时以上降低至 2小时。

中国科学院软件所 智能软件研究中心	2024年03月 - 2024年04月
C/C++ 工程师	远程实习

- **介绍**: 实现基于运行时路径跟踪的配置项级别自动化 Linux 内核裁剪框架。
- **内容**: 监测获取 OS 运行不同负载的路径, 从中分离出负载相关路径与非负载相关路径, 并分别根据是否模块代码离线将路径通过两类映射关系、配置项依赖关系生成两类配置项清单。后续可以基于所需操作系统版本与负载类型组合生成最终配置项清单完成内核裁剪。
- **指标提升**: 裁剪约 20% 内核配置项, 实现 19.76% 内核大小缩减, 缩短了 11% 启动时间。

科研经历

基于 DPU 的 LSM 树 Compaction I/O 异步化卸载方案

- **动机**: RocksDB 的 Compaction 任务非常繁重且耗时, 消耗大量 CPU 资源, 阻塞前台服务。
- **设计**: (1) DPU 与 CPU 协同 Compaction 方案: 利用异步化思路将 Compaction 计算任务与 I/O 任务解耦, 并将 I/O 任务卸载到 DPU 上, 由 CPU 和 DPU 并行协同完成 Compaction; (2) 高性能 DPU I/O 框架: 实现零拷贝、低主机 CPU 开销的 DMA 映射环形队列用于 DPU 与 CPU 间通信, 利用 io_uring 异步方案加速 I/O 处理; (3) 零主机开销 SST 文件共享方案: 基于 SST 文件性质, 利用由 DPU 维护的静态分配文件池轻量化实现主机与 DPU 间对 SST 文件的共享一致性。
- **贡献**: 完成架构设计与代码, 论文撰写中 (第一作者)。
- **指标提升**: Compaction 流程缩短 40%, 并节省约 30% 的 CPU 资源。

阿里巴巴达摩院校企合作 - 基于 DPU 的键值存储系统自动调优技术

- **动机**: RocksDB 自动调参方法主要有两种架构: C-S 和 One-Host 架构。C-S 架构调参具有误差与滞后性, One-Host 架构调参带来大量的 CPU 开销。
- **设计**: (1) 利用 DPU 卸载的自动参数调优模块, 实现实时调参同时避免主机 CPU 开销; (2) DPU 部署缓冲区以利用访问路径短优势加速访问与冷热分离, 同时辅助分批训练与数据负载分析。
- **贡献**: 参与架构讨论与部分代码和论文编写。目前论文 ASPLOS 2026 在投 (第二作者)。
- **指标提升**: 读写性能提升约 2.05 倍-3.09 倍, 节省主机端约 35% 的 CPU 资源。

基于持久性内存和高性能固态盘的 LSM 树键值存储系统研究

- **动机**: LevelDB 将 L0 数据无序化和粗粒度压缩的方式带来写停顿和读放大。与 flash 相比, 容量大、速度快且可字节寻址的 Optane 持久性内存可以替代 SSD 实现 L0 细粒度数据管理。
- **设计**: (1) 利用 B+树替换 SSD 中 LevelDB L0 层无序的 SST 文件以优化读性能, 同时通过关键数据与非关键数据分离、叶节点无序化存储、SIMD 指纹批量比较等优化保证 B+ 树在持久性内存上的性能; (2) 以叶节点为粒度进行细粒度压缩和冷热分离, 减少写停顿; (3) 基于指针和标记更新的方式迁移叶节点, 避免迁移阻塞查询。
- **贡献**: 完成代码与论文主要部分的编写 (第一作者)。
- **指标提升**: 整体系统吞吐量提升约 70%; 尾延迟降低约 16 倍。

技能及其他

- **技能**: 熟悉 C++、Python、Linux, 具有良好的英语读写能力 (英语六级: 523)。
- **其他**: 获得过三次学业奖学金, 学习能力强, 有强烈的求知欲, 有良好的团队协作能力。