周报

2025 年 3 月 4 日

摘要

- **SHA256 哈希算法的 GPU 并行实现与优化**
- **论文研究动机、贡献和实现部分的修订**

SHA256 线程束级并行实现

线程布局设计

- 线程 0：状态初始化
- 线程 0-15：消息字加载与调度扩展
- 线程 0-7：轮计算中的状态变量管理
- 线程 0：填充和最终输出

优化亮点

- 充分利用线程束内并行性
- 减少线程束分化
- `__shfl_sync()` 实现高效数据共享
- 吞吐量约 120MB/s，6 倍提升，[WDC+25]

论文研究动机更新

强化研究必要性

- **现有实现的局限**
  - 注重最大化吞吐量，忽视单线程执行效率
  - Kim 等人：多次内核启动导致效率低下
  - Wang 等人的 CUSPX：线程利用和资源管理有优化空间
- **关键观察**
  - 底层哈希函数优化不足
  - 未充分考虑线程数量与执行效率间的权衡

论文主要贡献修订

1. **哈希函数级并行方法**
   - 细粒度任务分配减少延迟
   - 显著加速 SPHINCS+ 核心计算原语
2. **自适应线程分配策略**
   - 优化线程数量与内核函数效率间平衡
   - 最小化同步开销
   - 最大化 GPU 计算吞吐量

老师评语

工作量偏小，现在的写作我通篇看了下，离快报差距非常大

- 计划增加工作深度：完善 SHA256 实现细节，补充性能分析与对比实验
- 提高写作质量：重构论文框架，完善技术细节，强化创新点阐述

下周计划

1. 基于优化后的 SHA256 实现，完成 SPHINCS+ 签名方案中 WOTS+ 和 FORS 组件的 GPU 加速实现
2. 设计并实现自适应线程分配策略，针对不同参数集优化计算资源分配

参考文献


Ziheng Wang, Xiaoshe Dong, Heng Chen, Yan Kang, and Qiang Wang.
Cuspx: Efficient gpu implementations of post-quantum signature sphincs⁺.
IEEE Transactions on Computers, 74(1):15–28, 2025.