# 周报

2025 年 3 月 18 日

# 摘要

- 自适应线程分配（**Adaptive Thread**）技术探索
- 论文绪论和自适应线程部分的修改完善

# 自适应线程分配理论模型

## 传统问题

- 通常将线程设为可用最大值
- 忽略函数特性差异
- 资源分配不均衡

## 理论模型

- 自适应线程函数$AT : G \to T$
- 性能模型:
  $T(g_i, t) = \alpha_i + \frac{\beta_i}{t} + \gamma_i \cdot t$
- 最优线程计算: $t_i^* = \sqrt{\frac{\beta_i}{\gamma_i}}$

## 参数含义

- $\alpha_i$: 固定开销
- $\frac{\beta_i}{t}$: 并行加速部分
- $\gamma_i \cdot t$: 线程管理开销
- $t$: 线程数量

## 创新点

- 根据函数特性动态分配
- 平衡并行与管理开销
- 资源最优化利用

# 实验结果

| Block×Thread 配置 | 每操作耗时 (ms) | 相对默认配置性能提升 |
|---|---|---|
| 128×64 | 0.0025 | -13.6% |
| 默认配置 (512×32) [WDC+25] | 0.0022 | 基准 |
| 128×256 | 0.0017 | 22.7% |
| 128×512 | 0.0018 | 18.2% |

**关键发现 (公钥生成):**

- 最优线程配置 (128×256) 提升 22.7% 性能
- 并非最大线程数 (128×512) 才有最优性能
- 验证了理论模型的有效性

# 论文写作进展

学长指导下，我们查阅了 NIST 文献，确认 SPHINCS+ 已于 2024 年 8 月更名为 SLH-DSA 并纳入 FIPS 205 标准。

## 绪论部分修改

- 精简PQC 背景介绍
- 突出SPHINCS+ 与 SLH-DSA 关系
- 重构相关工作部分
- 明确当前 GPU 实现的效率瓶颈:
  - 统一的最大线程分配策略
  - 次优的线程性能
- 强化贡献点表述

## 自适应线程分配部分

- 构建基于函数特性的性能模型
- 提出最优线程数计算公式
- 设计动态实现算法
- 离线分析与运行时调整相结合
- 资源最优利用方案

# 老师评语

## 加快

预计 4 月底，完成初稿。

## 下周计划

1. 完善实验和创新点二写作
2. 设计并实现自动化框架，用于确定各类密码算法操作的最佳线程配置

# 参考文献


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