周报-向嘉豪 (2025 年 3 月 10 日)

摘要：本周主要工作集中在 XMSS 树结构的并行实现优化及相关论文的写作与重构上。针对 SPHINCS+ 签名方案中的 XMSS 组件，我们实施了节点级并行和 WOTS 级并行两层次的并行技术，并结合动态调度策略进行优化。实验结果表明，在二级并行中加入动态调度带来了轻微的性能提升（8.16x 至 8.21x），而在三级并行中采用动态调度与 GPU 优化技术后，性能显著提高（142.65x 至 159.30x），PKGEN 操作的执行时间从 0.220ms 降至 0.197ms，实现了 10.9% 的性能提升。同时，我们确定论文的理论基础，将动态调度作为性能提升的理论依据，去构建针对不同核函数的最优线程配置映射函数，并设计完整的调度实现机制。

下周计划：1）构建最优线程配置映射函数，强化理论分析部分；2）对剩余组件进行动态调度的实现。

1 XMSS 树并行实现优化

SPHINCS+ 签名方案中的 XMSS（树哈希）组件是影响整体性能的关键因素。本周我们重点优化了 XMSS 树结构的并行实现，主要采用了二个层次的并行技术：**节点级并行 (Node-based parallelism)**：同一层次的节点并行计算, 此为 [WDC+25] 中实现的二级并行。**WOTS 级并行 (WOTS-based parallelism)**：单个节点内的 WOTS 链并行计算, 此为 [WDC+25] 中实现的三级并行。

实验结果表明，这些并行优化技术显著提高了签名性能。以下是在参数配置为 `n = 24, h = 66, d = 22, b = 8, k = 33, w = 16, len = 51` 的 SLH-DSA-SHA-256-192f 算法上的基准测试结果：

操作	执行时间	性能提升
PKGEN 串行实现 [WDC+25]	31.382 ms	基准
PKGEN 二级并行 [WDC+25]	3.844 ms	8.16x
PKGEN 二级并行 + 动态调度	3.822 ms	8.21x
PKGEN 二三级并行 [WDC+25]	0.220 ms	142.65x
PKGEN 二三级并行 + 动态调度 +GPU 优化	0.197 ms	159.30x

表 1: XMSS 树并行实现的性能对比

首先我们将动态调度(在运行的过程中调整运行的线程数，而非固定线程数去并行运行)，而将其放入到二级并行中，性能只从 8.16x 提升到 8.21x，这说明动态调度在二级并行中并没有显著的性能提升。为此我们猜测是由于 WOTS 节点串型实现影响了性能提升。我们将动态调度放入到三级并行中，性能从 142.65x 提升到 159.30x，这说明动态调度在三级并行中有显著的性能提升。同时我们结合了 GPU 优化技术，对 XMSS 树生成签名进行优化，从 0.220 ms 提升到 0.197 ms，有10.9%的性能提升。

2 论文写作

在论文重构过程中，我们发现现有的创新点缺乏理论基础。为此，我们决定将动态调度作为性能提升的理论依据，同时补充实验数据以提高工作的重要性。我们之前的实验表明，在同一 GPU 平台上，对于不同核函数 g，存在最优线程数 t 使性能达到最大。这一发现成为我们动态调度方法的理论基础。

在整个签名过程中，需要对不同核函数进行动态调度。签名过程可表示为一个运行序列 $((g_1, t_1), \ldots, (g_n, t_n))$，我们的目标是使整体性能最优。由于签名算法是确定的，我们已知 $(g_1, \ldots, g_n)$，需要确定 $(t_1, \ldots, t_n)$。因此，我们的核心创新点是构建映射函数$F : G \to T$，它能为每个核函数 g_i 分配最优线程配置 t_i。为实现对签名各组件的动态调度，我们会设计一套完整的调度实现机制。这一实现机制是我们工作的第二个创新点，为动态调度策略提供了实践基础。

参考文献

[WDC+25] 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.