周报-向嘉豪 (2025 年 2 月 10 日)

摘要：对 SPHINCS+ 签名算法并行化实现策略的深入研究。借助文献 [WDC+25] 的理论探讨与实验数据，我们对 HT 树、FORS 树及 WOTS+ 算法的并行化方案进行了详细剖析，并评估了最大与最小并行化策略在实际运用中的性能表现与局限性。此外，明确了在 GPU 平台上复现相关实验的初步计划，并针对如何优化系统整体性能提出了未来的研究方向。

下周计划： 1) 复现 [WDC+25] 实验；2) 调整 HASH 运算分组数，提升并行效率；3) 将签名算法应用于实际场景中。

1 论文阅读

本周我们深入研读了 [WDC+25]，该论文系统性地探讨了如何并行化实现 SPHINCS+ 签名算法。文中详细阐述了针对 HT 树、FORS 树及 WOTS+ 算法的并行化策略，并依据各组成部分的执行顺序提出了一种分层并行方案，共划分为四个层次。由于各层之间相对独立，该文提出的组合并行策略可根据具体资源情况灵活调整，从而实现更高的并行效率（PE），即 PE= 效率/资源。

具体而言，如图 1 所示，左侧示例展示了最大并行化情形，即将每次 HASH 运算视为独立任务，但由此引入了四次同步操作，导致计算负载不均衡并引发等待现象；而右侧示例则采用最小并行化策略，将所有 HASH 运算合并为两个任务，虽然有效缓解了负载不均问题，但并行度则显著降低。因此，我们计划在这两种策略之间寻求一个合适的 HASH 运算分组数，以实现计算负载与并行度之间的最佳折中，并由此提升 PE。

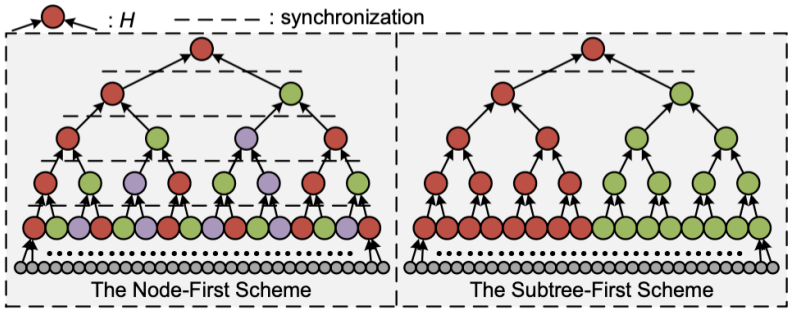

Fig. 2. Two schemes of parallel Merkle tree construction with three threads.

图 1: Merkle Tree 并行化 [WDC+25]

在获得 [WDC+25] 源代码的基础上，我们将着手在 GPU 平台上复现其实验，并在此基础上进行进一步的优化，以提升系统总体性能。同时，考虑将签名算法应用于实际场景中。

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.