 DOI: https://doi.org/10.12349/iser.v7i6.10813

Parallel Optimization of ML-DSA Digital Signatures for Edge Blockchain Scenarios

Jianmei Liu Yaru Wang*

Zhongyuan University of Technology, Zhengzhou, Henan, 451191, China

Abstract

The quantum computing threat renders traditional public-key cryptography insecure, and ML-DSA standardized by NIST becomes the core post-quantum signature scheme. However, its high signing overhead restricts deployment on resource-limited edge blockchain nodes. This paper constructs a Python-PQCrypto hybrid architecture, proposes a dynamic task partitioning parallel optimization method based on multi-process pool to bypass Python GIL and realize true multi-core parallel batch signature computation. Experiments on a 16-core Windows platform show that the throughput of optimized ML-DSA-44 rises from 1225.82 TPS to 6738.42 TPS with a speedup of 5.50×. In blockchain simulation, the system maintains 1937.76 TPS under 5000 transactions, with average signing latency 0.430 ms and verification latency 0.085 ms, and 100% signature success rate. The software parallel optimization scheme effectively relieves the lattice signature computing bottleneck, which can be deployed on existing multi-core edge devices without dedicated cryptographic accelerators.

Keywords

edge blockchain; ML-DSA; post-quantum cryptography; parallel optimization

边缘区块链场景 ML–DSA 数字签名并行优化研究

刘建美 王娅如*

中原工学院，中国 · 河南 郑州 451191

摘 要

量子计算对传统公钥密码体系形成安全威胁，NIST标准化的ML-DSA是主流后量子签名算法，但签名计算开销大，难以部署在算力受限的边缘区块链节点。本文搭建 Python-PQCrypto 混合架构，提出基于进程池的动态任务分片并行优化方案，借助多进程机制规避Python全局解释器锁带来的单核执行限制，实现多核批量签名并行运算。在16核Windows平台开展测试，优化后的ML-DSA-44签名吞吐量由1225.82 TPS提升至6738.42 TPS，加速比达5.50倍。区块链仿真场景下，5000笔交易负载时系统吞吐量稳定1937.76 TPS，平均签名时延0.430 ms、验签时延0.085 ms，在无传输干扰的测试条件下签名与验签校验成功率达100%。该优化方案有效缓解格基签名算力瓶颈，无需专用硬件加速，可直接适配现有多核边缘设备。

关键词

边缘区块链；ML-DSA；后量子密码；并行优化

1 引言

容错量子计算机持续迭代，Shor 算法可快速破解 RSA、椭圆曲线密码等经典公钥体制，现有区块链数字签名面临量子破解风险 [1]。为应对量子安全威胁，NIST 于 2024 年发布 FIPS 204 标准，将 CRYSTALS-Dilithium 标准化为 ML-DSA 格基数字签名算法，依靠模块带错误学习问题实现抗经典与量子攻击，包含 ML-DSA-44/65/87 三组安全参数，适配不同安全等级业务 [2]。

边缘区块链将交易共识、数据签名下沉至网络边缘节点，降低传输时延、减轻骨干网压力，广泛适配物联网、6G 分布式业务 [3]。但边缘网关多为无专用密码加速的通用多核处理器，ML-DSA 签名包含大量多项式矩阵乘、NTT 变换与拒绝采样，单线程串行处理吞吐量仅数百 TPS，无法

【基金项目】河南省科技攻关计划项目（项目编号：242102210189）；河南省自然科学基金青年项目（项目编号：252300423360）；河南省高等学校重点科研项目（项目编号：26A520052）中原工学院科研团队发展项目（项目编号：K2025TD002）；国家留学基金项目（项目编号：202508410108）。

【作者简介】刘建美（1988–），男，中国河南延津人，博士，从事量子计算、区块链安全研究。

【通讯作者】王娅如（1993–），女，中国河南商丘人，博士，从事格密码的量子安全性研究。

满足区块链高并发交易需求。

现有 ML-DSA 优化分为硬件与单核软件两类：FPGA、ASIC 加速方案吞吐性能优异，但硬件改造成本高、无法复用现有边缘设备；ARM 汇编、模约简优化仅提升单核运算效率，然而未挖掘多核并行潜力 [4]。Python 凭借丰富的密码与仿真类库，是区块链边缘场景原型仿真与快速方案验证的主流开发语言。目前 ML-DSA 算法的批量签名并行优化研究大多以高性能原生代码实现为目标，较少考虑 Python 环境全局解释器锁（GIL）、进程调度、数据序列化等特有约束，面向 Python 的 ML-DSA 批量签名并行优化策略与系统性性能评测工作较为匮乏。

针对上述问题，本文设计 Python-PQCrypto 混合分层架构，基于多进程池实现动态任务分片并行签名，规避 GIL 约束；搭建区块链交易仿真器完成多负载测试，验证方案吞吐、时延与稳定性，为边缘区块链后量子签名落地提供轻量化软件优化方案。

2 系统架构与并行优化方案

2.1 三层混合软件架构

整体架构分为应用层、密码中间件层、并行调度层，兼顾开发便捷性与多核运算性能，具体如下。

应用层：基于 Python 实现区块链交易逻辑，聚合批量签名请求、汇总并行计算结果，维护交易请求队列，向下分发签名任务。

密码中间件层：PQCrypto 0.1.16 密码库以 C 语言实现标准化 ML-DSA 三组安全参数，底层内置 NTT 多项式乘法、快速模约简等算法优化。本文基于 ctypes 机制设计并实现 Python 与 C 结构体的低开销数据交互层，支撑上层批量签名并行策略的验证与测试。

并行调度层：调用 Python multiprocessing 模块创建持久化进程池，每个进程拥有独立解释器与内存空间，绕过 GIL 实现多核真实并行。考虑 Python 多进程机制的平台差异性，并行调度模块根据不同操作系统选择不同的进程创建方式：Windows 平台使用 spawn 模式，类 Unix 系统采用 fork，以此保障批量签名并行程序的跨平台可运行性。

2.2 动态任务分片并行调度策略

单笔交易签名无数据依赖，天然具备批量并行条件，调度流程分为四阶段：

任务聚合：设置固定窗口批量聚合签名请求，单批默认 100 笔，平衡调度开销与并行收益；

动态分片：输入 N 条签名请求、K 个工作进程，按向上取整均分任务，依据实时 CPU 负载动态调整分片粒度，分片算法见算法 1；

并行分发：分片任务提交至持久进程池，借助进程池 map 批量映射接口分发至各工作进程独立运算，运行阶段无需进程间同步与进程间通信开销；

结果合并：收集各子进程签名结果，按原始交易顺序拼接输出，保证区块链交易时序一致性。

算法 1 动态任务分片与并行签名

Algorithm 1 动态任务分片与并行签名

```
Input: N 条签名请求、含 K 个工作进程的进程池 P
Output: 有序 N 条数字签名
function PartitionTasks(requests, K)
    n_per_worker = ceil(len(requests) / K)
    chunks = []
    for i = 0 to len(requests) step n_per_worker do
      chunks.append(requests[i : i + n_per_worker])
    end for
    return chunks
    end function

function ParallelSign(requests, pool)
    chunks = PartitionTasks(requests, pool.size)
    results = pool.map(ml_dsa_sign_batch, chunks)
    signatures = Concatenate(results)
    return signatures
end function
```

进程池在系统启动时一次性初始化，全程复用，消除重复创建销毁进程的高额开销，适配边缘长时间连续交易处理场景。

2.3 实验平台与评价指标

2.3.1 硬件与软件环境

测试硬件：Intel i9-12900K 处理器，16 个物理核心（8 性能核 +8 能效核）、24 逻辑处理器，配备 32 GB DDR4 内存。

系　统：Windows11 23H2；Python3.10.13；PQCrypto 0.1.16。底层密码库基于 MSVC2022 编译，编译参数开启 /O2 优化与 AVX2 指令集支持。

2.3.2 评价指标

1）吞吐量 TPS：每秒完成签名交易数，TPS=N/T_{total}，N 为批量交易总数，T_{total} 为总执行时长。 2）加速比：单核串行执行时间 / 多核并行执行时间，衡量多核资源利用率。3）平均时延：单条签名 / 验签平均耗时，$T_{avg}=T_{total}/N$。 4）成功率：签名、验签校验通过交易占比。全部计时采用高精度 time.perf_counter ()，每组实验重复 5 次取中位数，变异系数 CV<5% 视为结果有效。

2.4 区块链交易仿真器

本文搭建边缘区块链交易仿真模型，完整模拟节点交易全流程：密钥生成缓存、256 字节随机交易载荷签名、公钥验签、区块序列化打包。支持 100~5000 可变批量交易、1~16 可调工作进程数；密钥生成耗时单独统计，不计入签名吞吐计算，聚焦核心签名算力瓶颈。

3 实验结果与分析

3.1 ML–DSA 单核基准性能

三组安全参数单核串行基准耗时如表 1 所示，签名运

算开销远高于密钥生成与验签，ML-DSA-44 单次签名 4.22 ms，串行理论吞吐量仅 237 TPS，无法支撑高并发区块链业务；安全等级越高多项式维度越大，ML-DSA-87 签名耗时为 ML-DSA-44 的 2.8 倍。

表 1. ML–DSA 单核基准性能测试结果（单位：ms）

算法	密钥生成 (ms)	签名 (ms)	验签 (ms)
ML-DSA-44	1.85	4.22	1.12
ML-DSA-65	2.94	7.15	1.88
ML-DSA-87	4.56	11.82	3.05

3.2 多核并行吞吐与加速比

以 ML-DSA-44、单批 100 笔为测试用例，不同进程数并行性能如表 2。随着工作进程提升，吞吐量持续上涨，16 进程下吞吐达 6738.42 TPS，相对单核基准加速比 5.50。加速比无线性增长，存在边际收益递减：8 核至 16 核仅提升 1.14 倍，原因为超线程逻辑核资源争抢、进程间序列化通信、内存缓存竞争、操作系统调度开销。实际边缘部署可匹配物理核心数设置进程，降低上下文切换损耗。

表 2. ML–DSA–44 多核并行性能

工作进程数	总耗时 (s)	吞吐量	加速比
1	40.789	1225.82	1.00
2	25.483	1962.07	1.60
4	14.735	3393.30	2.77
8	8.460	5910.40	4.82
16	7.420	6738.42	5.50

3.3 区块链仿真场景验证

16 进程、1000 笔交易仿真测试结果：系统吞吐量 2203.73 TPS，单交易平均签名时延 0.370 ms、验签 0.076 ms，验签速度约为签名 5 倍；全部交易签名、验签校验通过，成功率 100%，证明并行分片机制无逻辑错误，不会造成交易失效、共识中断。

3.4 系统负载可扩展性测试

16 进程下批量交易规模从 100 增至 5000 笔，性能变化如表 3。交易规模扩大 50 倍，吞吐量仅下降 18.6%，具备强负载稳定性；签名时延小幅上升至 0.430 ms，仍维持亚毫秒级，适配实时边缘业务；总处理时间与交易数量近似线性增长，进程池调度开销可被大批量任务分摊。

表 3. 多交易规模可扩展性测试（16 进程）

交易总量	TPS	平均签名 (ms)	平均验签 (ms)	总耗时 (s)
100	2380.62	0.347	0.072	0.042
500	2242.50	0.370	0.075	0.223
1000	2335.99	0.353	0.074	0.428
2000	2109.12	0.395	0.078	0.948
5000	1937.76	0.430	0.085	2.580

3.5 方案对比分析

现有纯软件单核优化方案 ML-DSA-44 吞吐量为 1000~1500 TPS，与本文单核基准 1225.82 TPS 一致；本文并行优化后吞吐接近 FPGA 硬件加速万级 TPS 水平，但无需专用密码硬件。基于阿姆达尔定律，假设 90% 计算任务可并行，16 核理论加速比约 5.93，本文实测 5.50，并行调度通信、内存竞争等额外开销控制在合理范围。本方案可与 NTT 模约简底层优化形成互补：底层优化降低单笔签名运算耗时，多核并行优化批量并发吞吐，二者结合可进一步提升性能 [4]。

4 方案局限性与未来方向

4.1 研究局限

本文实验平台选用高性能 Intel 桌面处理器开展验证，未在市面上主流低算力 ARM 边缘网关完成适配测试，ARM 平台自带的 NEON SIMD 指令集能够进一步加速底层 NTT 运算，后续可补充相关测试。本次实验仅针对 ML-DSA-44 低安全等级展开完整性能验证，ML-DSA-65、ML-DSA-87 两类高安全参数集的并行运算性能缺少批量对比数据。同时系统采用固定规模进程池运行，无法根据设备实时 CPU 占用、温度、功耗动态调整工作进程数量，适配异构边缘设备的灵活性不足。另外全部测试在后台负载纯净的实验室环境完成，没有模拟真实边缘设备多业务并发抢占 CPU 资源的复杂场景，实际落地性能会存在小幅偏差。

4.2 后续研究方向

后续研究将从五个维度展开优化与拓展。第一，完成 ARM 边缘设备专属适配优化，融合 NEON SIMD 指令加速 NTT 多项式运算，在树莓派、NVIDIA Jetson 等物联网边缘网关完成全场景性能测试。第二，设计自适应动态调度机制，依据设备实时 CPU 负载、温度阈值、电池剩余电量动态增减进程数量，适配电池供电、存在热限制的低功耗边缘终端。第三，搭建软硬件混合加速框架，将 ML-DSA 内部多项式乘、NTT 等核心运算卸载至 FPGA 硬件，与本文多进程调度框架结合，进一步提升系统吞吐上限。第四，开展并行场景侧信道安全分析，研究多核并行执行过程中缓存资源争抢带来的信息泄露风险，加固并行签名底层实现。第五，将并行签名模块嵌入完整区块链共识流程，完善混合经典 / 后量子密码体系下的跨链签名互操作机制。

5 结语

面向边缘区块链节点 ML-DSA 签名算力不足的问题，本文构建 Python-PQCrypto 三层混合架构，设计多进程动态分片并行签名优化方案，针对 Python 全局解释器锁（GIL）造成的多核利用低效问题，采用多进程并行架构隔离各签名任务，绕过单进程并行瓶颈，挖掘多核处理器在批量签名场景下的运算潜力。16 核平台测试表明，ML-DSA-44 签名吞吐提升 5.50 倍，区块链仿真高负载下维持稳定亚毫秒时延。该方案属于纯软件轻量化优化，无需新增硬件加速设备，可快速部署至现有边缘区块链网关，有效解决后量子格基签名

在分布式边缘场景的并发算力瓶颈。后续将针对 ARM 低功耗设备、自适应调度、硬件协同加速开展深入研究，完善后量子密码边缘落地全链路优化体系。

参考文献

[1] 陈驰，马建峰。后量子密码标准化进展与区块链适配关键问题 [J]. 通信学报，2024,45 (04):132-148.

[2] National Institute of Standards and Technology. FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA)[S]. Gaithersburg: NIST, 2024. https://doi.org/10.6028/NIST.FIPS.204

[3] 刘建伟，张宗洋。边缘区块链安全体系架构与格基密码部署挑战 [J]. 计算机学报，2024,47 (07):1561-1580.

[4] Longa P, Naehrig M. Speeding up the Number Theoretic Transform for Faster Ideal Lattice-Based Cryptography [J]. Designs, Codes and Cryptography, 2022, 90 (8): 1741-1770. https://doi.org/10.1007/s10623-022-01089-9

[5] 杨波，王瑞锦。边缘计算场景下区块链交易并行处理优化方法 [J]. 信息科学与工程研究，2024,11 (02):38-46.