字同态加密研究

徐一杰

June 17, 2024

Abstract

同态加密（HE）是一种隐私保护计算技术，但高昂的计算开销限制了其应用。我们在 GPU 上实现并优化了三种逐字 HE 方案：BGV、BFV 和 CKKS，显著降低了计算和内存开销。我们还引入了统一框架，优化预计算和内存管理，对三种方案进行基准测试，提供了有效的性能比较和参考。

1 介绍

同态加密（HE）是一种允许在不知密钥的情况下对加密数据进行计算的密码系统，使非交互式安全计算成为可能，并减少通信开销。HE 在隐私保护应用中具有广泛前景，如安全神经网络推理、私有集合操作和私有决策树评估。

2009 年，Gentry 引入自举（bootstrapping）概念，将部分同态加密（SHE）提升为全同态加密（FHE），支持任意数量的运算。目前，FHE 方案大多基于带有错误的环运算（RLWE）问题，分为位同态和字同态。字同态方案（BGV、BFV、CKKS）更高效，支持批处理，将多个明文打包成一个密文进行计算。

尽管 HE 受到广泛关注，但其性能尚不足以满足实际需求。使用 GPU 或 FPGA 进行硬件加速能够缓解这一问题。GPU 加速在机器学习等领域已取得显著成效，但在 HE 方面的研究仍较少，现有工作多局限于单一方案或有限参数大小。

在此背景下，我们介绍了 Phantom，一个优化 BGV、BFV 和 CKKS 三种高级字同态方案的高性能 GPU 库。Phantom 在性能上超越了现有工作。我们的主要贡献包括：

1. 理论优化：引入多项式乘法的数论加权变换等优化，减少数据访问和 IO 延迟，优化密钥交换操作，降低计算复杂度。

2. 统一和优化的 GPU 实现：开发通用算术表达式，确保 BGV、BFV 和 CKKS 方案在单个框架内的兼容性，并提出多种 GPU 优化方法。

3. 综合基准测试：支持对三种方案在各种参数集下的全面基准测试，证明了方法的效率和可扩展性。

Phantom 显著提升了 HE 性能，为隐私保护应用的实际部署提供了强有力的支持。

2 预备知识

我们总结了 GPU 的计算模型和架构。GPU 内存分为两种类型。第一个是读写内存，按访问速度由慢到快包括全局内存 (GMEM)、共享内存 (SMEM) 和寄存器文件 (RF)。第二种是只读内存，包括常量内存和纹理内存，这两种内存都可以缓存。

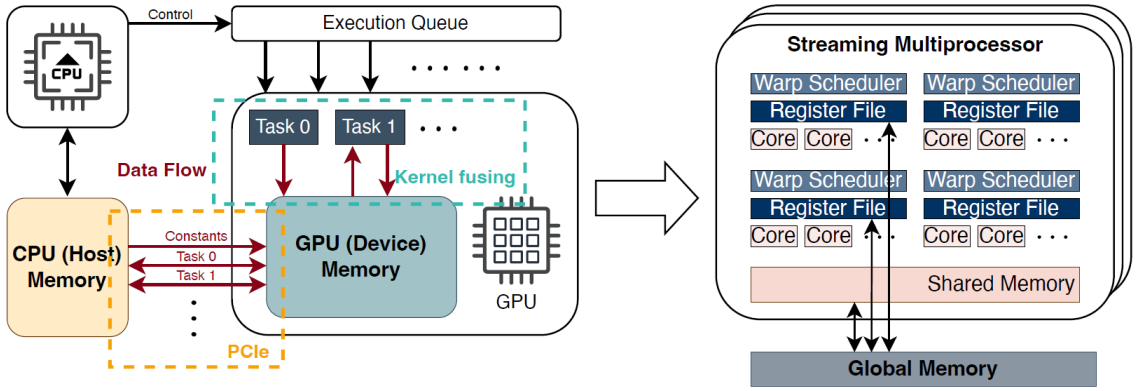

Figure 1: CPU-GPU 计算模型.

CUDA 内核在 GPU 上由多个线程并行运行，这是最小的执行单元，可以分组成一个线程块。每个线程都有自己的私有 RF，并且 SMEM 由块内的所有线程共享。所有线程都可以访问 GMEM 和只读内存，它们的生命周期最长，涵盖整个计算任务。在执行过程中，线程以每 32 个线程为单位捆绑在一起。

流多处理器 (SM) 保存一个或多个块，SM 中的每个 warp 调度器 (WS) 一次调度并执行一个 warp。在配备 CPU 和 GPU 的异构平台中，一种常见而直接的协同计算机制是，CPU 调度执行队列中的任务，将相应的内核启动给 GPU，并通过 PCIe 传输必要的数据，然后等待 GPU 执行并返回结果。在这种情况下，融合依赖数据的内核可以在一定程度上减少数据传输和内存访问带来的 IO 延迟。为了获得更好的性能，应该防止过度融合，因为如果一个线程块的资源消耗过高，SM 占用率将减少。

3 实现细节与优化

我们展示了实现框架的结构，给出了一个简明的概述。在功能上，它由两个主要部分组成，一个用于预计算，另一个包含 HE 方案的优化实现，可分为三层: 数学/多项式层，RNS 算法层和方案层。

基础层包含 64 位和 128 位整数的底层模操作、伪随机生成器和多项式算术。对于整数运算，我们提供了被很好地优化的常数时间实现，并最大限度地减少了机器指令的数量和寄存器的使用。在此基础上，我们实现了多项式算术，如 NWT 和 FFT，以实现快速和低复杂度的计算。在中间层，我们实现了采样和 RNS 算术模块，其中操作数是 RNS 或双 CRT 表示下的多项式。采样模块由三种多项式系数的采样方法组成，分别是三进制，均匀分布和中心二项分布。

RNS 模块提供了高效的多项式算法，并实现了 BEHZ 型和 HPS 型基转换的常用算法。顶层提供了 BGV、BFV 和 CKKS 方案的高层统一实现。对于所有方案，我们的框架支持以下特性:(1) 同时支持对称和非对称加密;(2) 两个或多个明文和密文的同态加、减、乘;(3) 单个密文的原地同态取幂、取反、旋转。我们的优化主要针对内存效率，以解决 HE 方案的显著内存消耗和数据访问需求。首先，我们优化数据访问模式以消除跨步 GMEM 访问，确保在单个周期内执行 warp 级内存请求以提高效率。其次，我们引入了一种内存池机制，用于高效、安全地处理和传输大数据量。第三，我们的方法包括开发各种核融合技术。这既包括核内融合，它在单个内核内合并算术运算，允许在寄存器中重用临时元素，也包括核间融合，它合并相邻的内核，共享相似的

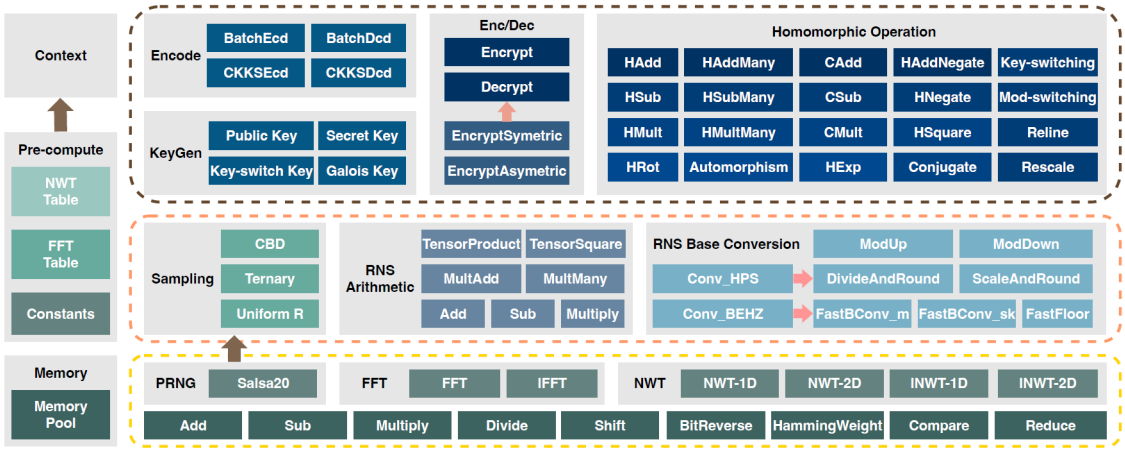

Figure 2: 同态加密结构.

并行性。后一种技术的目的是在较低延迟内存中维护临时数据，从而减少对耗时的 GMEM 传输的需求。

总的来说，这些增强有助于减少内存请求指令和 IO 延迟，从而显著提高 HE 实现的性能。

4 实验数据

我们给出了实现我们的内存优化策略之前和之后的执行时间。这些结果对应于我们在第 3.4 节中提出的三种核融合技术，对 |N| 14,15,16 进行评估。我们的第一个优化涉及 cudaMemcpy 操作的融合。通过重新组织数据序列，我们简化了数据复制过程，使其能够通过单个 cudaMemcpy 调用而不是多个调用来执行。这项改变几乎使这个过程的效率提高了一倍。在第二次优化中，我们将校正步骤融合到 BGV 方案的 ModDown 操作中。这种集成允许将校正项直接添加到 ModDown 中的余数运算中，有效地利用 RF 并减少对大量 GMEM 负载和存储请求的需要。因此我们观察到三个参数集的加速范围从 1.73 倍 1.90 倍。第三个优化将缩放操作融合到 INWT 内核中。这种方法显著降低了与缩放计算相关的开销，主要是由于降低了内存访问需求。在这里，我们实现了高达 1.48 倍的加速。这些结果证明了我们提出的优化在提高操作的整体性能方面的实质性有效性。

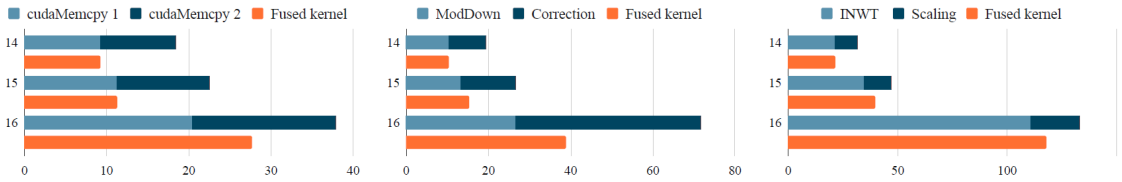

Figure 3: 实验结果.

5 结论

在这项工作中，我们提出了 BGV, BFV 和 CKKS 的优化 GPU 实现，并评估了性能。我们减少了操作的计算和内存开销，并展示了在不同量级参数下实现最优性能的方法。我们开发了一个框架来整合这三种方案的实现，并提供了实现方案的全面基准。

References

[1] H. Yang, S. Shen, W. Dai, L. Zhou, Z. Liu and Y. Zhao, "Phantom: A CUDA-Accelerated Word-Wise Homomorphic Encryption Library," in IEEE Transactions on Dependable and Secure Computing, doi: 10.1109/TDSC.2024.3363900.

[2] A. Brutzkus, R. Gilad-Bachrach, and O. Elisha, "Low latency privacy preserving inference," 2019, pp. 812–821.

[3] H. Chen, K. Laine, and P. Rindal, "Fast private set intersection from homomorphic encryption," in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, 2017, pp. 1243–1255. [Online]. Available: https://doi.org/10.1145/3133956.3134061

[4] W.-j. Lu, J.-j. Zhou, and J. Sakuma, "Non-interactive and output expressive private comparison from homomorphic encryption," in Proceedings of the 2018 on Asia Conference on Computer and Communications Security, AsiaCCS 2018, 2018, pp. 67–74.

[5] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, 2009.