

CCF 2025 HIP 编程竞赛项目报告

付嘉晨, 黄俊雄

2025 年 9 月 11 日

1 项目概述

本项目实现了 CCF 2025 HIP 编程竞赛的三个 GPU 加速算法挑战：**前缀和**、**Softmax** 和**全对最短路径 (APSP)**。所有算法均基于 HIP/ROCm 实现，针对 AMD GPU 架构优化，采用单 GPU 执行策略。项目在算法实现、性能优化、输出加速等方面取得了显著成果。

2 算法实现

2.1 前缀和 (Prefix Sum)

算法特点. 1) 实现分块扫描三阶段算法：块内扫描 → 块和扫描 → 偏移累加；2) 支持 Hillis-Steele 和 Blelloch 两种扫描实现，可编译时切换；3) 使用共享内存优化块内计算，支持非 2 次幂长度数组。

关键优化. 1) BLOCK_SIZE=256, 适配 AMD GPU 架构；2) 递归扫描处理大规模数据，层数为 $O(\log_B N)$ ；3) 高性能输出：32MB 缓冲区 + std::to_chars, 显著减少 I/O 开销。

2.2 Softmax

算法特点. 1) 实现数值稳定的 softmax: $y_i = \frac{\exp(x_i - \max)}{\sum(\exp(x_j - \max))}$ ；2) 两阶段归约：先求全局最大值，再计算指数和并归一化；3) 严格遵循数值稳定性要求，避免溢出/下溢。

关键优化. 1) 4 个专用 GPU 内核：分块最大值、全局最大值归约、指数和计算、归一化；2) 分母保护：S + 1e-12 确保满足容差要求；3) FAST_OUTPUT 优化：输出性能提升 3-6 倍，特别适用于大规模数据。

2.3 全对最短路径 (APSP)

算法特点. 1) 基于分块 Floyd-Warshall 算法，三阶段执行：(a) Phase 1: 处理 pivot 块 (k,k), (b) Phase 2: 更新第 k 行/列块, (c) Phase 3: 更新其余所有块；2) 使用共享内存进行块内 tile 复用，减少全局内存访问。

关键优化. 1) **内存优化**：页锁定内存 + 异步传输，H2D 传输提升 35 倍 (355ms→9ms)；2) **GPU 计算优化**：HIP 事件精确同步替代设备同步，性能提升 19-26%；3) **编译优化**：现代编译标志 (--offload-arch)、循环展开、分支预测；4) **输出优化**：32MB 缓冲区，输出速度从 100MB/s 提升至 540MB/s。

3 技术亮点

3.1 渐进式性能优化策略

以 APSP 为例，采用系统性的优化方法：1) **阶段 1**：页锁定内存优化，D2H 性能提升 82%；2) **阶段 2**：异步传输 + 流并行，H2D 性能提升 97%；3) **阶段 3**：GPU 计算核心优化，整体性能提升 19-26%。

3.2 统一的高效输出框架

三个项目均实现 FAST_OUTPUT 优化: **1)** 大缓冲区 (32MB) 减少系统调用; **2)** `std::to_chars` 替代格式化输出; **3)** 关闭 `iostream/stdio` 同步, 启用全缓冲模式; **4)** 保持字节级输出兼容性。

3.3 工程化实践

1) 错误处理: 完整的 HIP API 错误检查和资源管理; **2) 可配置性:** 编译时开关控制优化策略 (FAST_OUTPUT、算法选择); **3) 测试验证:** 完整的测试套件, 自动化校验正确性; **4) 性能监控:** 详细的分阶段计时, 支持性能分析。

4 性能结果

4.1 定量性能数据 (基于 MI100/gfx908)

表 1: 基于 MI100/gfx908 的性能数据

| 项目 | 关键优化 | 性能提升 | 代表性指标 |
|------------|--------------------|-------------|----------------|
| Prefix Sum | Blelloch 扫描 + 快速输出 | 显著减少 I/O 开销 | 支持 1 亿级元素 |
| Softmax | 数值稳定 + 快速输出 | 输出加速 3-6 倍 | 满足 $1e-6$ 绝对容差 |
| APSP | 综合优化 | 传输加速 35 倍 | |

4.2 关键性能突破

1) 数据传输: 从 388ms 降至 15ms (APSP); **2) 输出效率:** 从 100MB/s 提升至 540MB/s; **3) GPU 利用率:** 事件驱动同步显著提升并行度。

5 技术创新

1) 混合同步策略: HIP 事件 + 流实现精确依赖管理, 避免过度同步; **2) 自适应内存管理:** 页锁定内存 + 异步传输的最佳实践组合; **3) 编译器协同优化:** 现代 HIP 编译标志 + 手工循环优化相结合; **4) 跨项目复用模式:** 高效输出框架在三个算法间的成功移植。