PR #25093 完整报告

sgl-project/sglang

[AMD] Enable AITER custom all-gather on ROCm

合并时间：2026-06-03 06:57

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25093

执行摘要

- 一句话：在 ROCm 上集成 AITER 自定义 all-gather，加速 TP 通信
- 推荐动作：值得精读。该 PR 展示了在大型项目中安全集成第三方加速库的范例：环境变量开关、完备的 fallback、CUDA 图各阶段一致性处理、以及配套的 benchmark 和 CI 测试。_all_gather_into_tensor 中的条件编排和状态分支设计可供参考。

功能与动机

PR body 指出，SGLang 在 ROCm 上现有 TP all-gather 使用 RCCL 路径，但 AITER 提供的自定义 all-gather 在 DeepSeek-R1-MXFP4-Preview 服务的 logits 形状下更快且输出 bit 一致。因此希望新增一条 HIP 门控的 AITER 路径，类似已有的 AITER custom all-reduce 集成方式。

实现拆解

1. 环境变量注册：在 python/sglang/srt/environ.py 的 Envs 类中新增 SGLANG_USE_AITER_AG = EnvBool(True)，默认启用。
2. 核心逻辑改造：在 python/sglang/srt/distributed/parallel_state.py 的 GroupCoordinator._all_gather_into_tensor 方法开头插入 AITER 分支。条件检查：ROCm 平台、env 启用、ca_comm 存在且拥有 should_custom_ag 等方法、输入输出连续、dtype 为 float32/16/bfloat16、should_custom_ag 返回 True。当 CUDA 图正在捕获时调用 all_gather_reg，piecewise 图调用 all_gather_unreg，普通 warmup 阶段则 output.zero_() 并返回（避免混合使用 NCCL 和 AITER 导致图不一致）。不满足条件时 fallback 到原有 pynccl 或 torch.distributed.all_gather_into_tensor。
3. 辅助方法：新增 _has_aiter_custom_all_gather 检查 ca_comm 具备必要方法且未禁用确定性 collectives；_deterministic_collectives_enabled 判断确定性推理配置。
4. Benchmark 与测试：新增 benchmark/kernels/all_gather/benchmark_aiter.py，支持多 shape/dtype/dim 扫描，并在计时前进行正确性检查（torch.equal 对比 RCCL 与 AITER 输出）。新增 test/registered/ops/test_aiter_allgather_amd.py，作为 CI 测试调用 benchmark 脚本在 TP=2,4,8 下验证多种 dtype 的正确性。
5. 配套修复：根据 review 意见重构了 CUDA 图捕获 / 预热路径，避免 warmup 时走 NCCL 而捕获时走 AITER 的不一致问题；同时添加了 dtype guard 防止非 float 类型进入 AITER（AITER 仅支持 float32/16/bfloat16）。

关键文件：

- python/sglang/srt/distributed/parallel_state.py（模块 分布式；类别 source；类型 core-logic；符号 _has_aiter_custom_all_gather, _deterministic_collectives_enabled）：核心改动：在 _all_gather_into_tensor 中插入 AITER 自定义 all-gather 分支，并新增 _has_aiter_custom_all_gather 等辅助方法。包含 CUDA 图状态处理逻辑。
- benchmark/kernels/all_gather/benchmark_aiter.py（模块 基准测试；类别 source；类型 dependency-wiring；符号 parse_shape_list, parse_dtype_list, parse_dim_list, parse_args）：新增 benchmark 脚本，支持多 shape/dtype/dim 扫描，在计时前验证 AITER 与 RCCL 的正确性，是性能数据和 CI 测试的基础。
- test/registered/ops/test_aiter_allgather_amd.py（模块 测试；类别 test；类型 test-coverage；符号 TestAiterAllGatherAmd, _gpu_count, test_aiter_allgather_matches_rccl）：CI 测试：在 TP=2/4/8 上运行 benchmark 脚本的 --correctness-only 模式，覆盖 10 种 dtype 和两种形状，确保 AITER 路径正确性。
- python/sglang/srt/environ.py（模块 配置；类别 source；类型 core-logic）：注册 SGLANG_USE_AITER_AG 环境变量，默认 True，使能 AITER all-gather 开关。

关键符号：_all_gather_into_tensor, _has_aiter_custom_all_gather, _deterministic_collectives_enabled, reshape_logical, raw_allgather_shape, test_aiter_allgather_matches_rccl

关键源码片段

python/sglang/srt/distributed/parallel_state.py

核心改动：在 _all_gather_into_tensor 中插入 AITER 自定义 all-gather 分支，并新增 _has_aiter_custom_all_gather 等辅助方法。包含 CUDA 图状态处理逻辑。

```
def _all_gather_into_tensor(self, output: torch.Tensor, input: torch.Tensor):
    # AITER custom all-gather (ROCm). Set SGLANG_USE_AITER_AG=0 to disable.
    # AITER's should_custom_ag handles shape/layout validation:
    # 16B alignment, weak-contiguous, supported topology, and per-rank
    # size <= max_size/(world*2).
    ca_comm = self.ca_comm
    if (
        is_hip()
        and envs.SGLANG_USE_AITER_AG.get()
        and self._has_aiter_custom_all_gather()
        and input.is_contiguous()
        and output.is_contiguous()
        # AITER only supports float types; int tensors fall through to NCCL
        and input.dtype in (torch.float32, torch.float16, torch.bfloat16)
        and ca_comm.should_custom_ag(input)
    ):
        if getattr(ca_comm, "_IS_CAPTURING", False):
            if torch.cuda.is_current_stream_capturing():
                # Actual capture: register buffer for graph replay
                ca_comm.all_gather_reg(input, out=output, dim=0)
            elif is_in_piecewise_cuda_graph():
                # Piecewise graph: use unregistered version
```

```
                ca_comm.all_gather_unreg(input, out=output, dim=0)
            else:
                # Warmup: zero out output to keep graph shape, avoid NCCL
                output.zero_()
            return
        else:
            # Eager mode: unregistered variant
            ca_comm.all_gather_unreg(input, out=output, dim=0)
            return

    # Fallback: pynccl or torch.distributed
    pynccl_comm = self.pynccl_comm
    if pynccl_comm is not None and (
        not pynccl_comm.disabled or self.is_symmetric_memory_enabled()
    ):
        ... # existing path
    else:
        torch.distributed.all_gather_into_tensor(
            output, input, group=self.device_group
        )


def _has_aiter_custom_all_gather(self) -> bool:
    if self._deterministic_collectives_enabled():
        return False
    ca_comm = self.ca_comm
    return (
        ca_comm is not None
        and not getattr(ca_comm, "disabled", True)
        and hasattr(ca_comm, "should_custom_ag")
        and hasattr(ca_comm, "all_gather_reg")
        and hasattr(ca_comm, "all_gather_unreg")
    )

@staticmethod
def _deterministic_collectives_enabled() -> bool:
    if envs.SGLANG_USE_1STAGE_ALLREDUCE.is_set():
        return envs.SGLANG_USE_1STAGE_ALLREDUCE.get()
    return envs.SGLANG_ENABLE_DETERMINISTIC_INFERENCE.get()
```

评论区精华

- gemini-code-assist[bot] 指出 CUDA 图 warmup 阶段如果 _IS_CAPTURING 为 True 但 torch.cuda.is_current_stream_capturing() 为 False，代码会 fallthrough 到 NCCL，导致 warmup 用 NCCL 而实际捕获用 AITER，可能引发问题。hubertlu-tw 确认并重构代码，最终通过 output.zero_()+return 避免在 warmup 时调用任何 collective。
- hubertlu-tw 在 dtype guard 上注释：当前仅支持浮点类型，未来 AITER kernel 扩展后可放宽限制。

- CUDA 图 warmup 阶段可能的不一致问题 (correctness): hubertlu-tw 同意并重构：在 warmup 分支中用 output.zero_()+return，避免使用任何 collective。
- dtype 限制与未来扩展 (design): 保留 dtype guard，文档化限制。

风险与影响

- 风险：
  - 平台限制：仅 AMD ROCm/HIP 平台生效，其他平台不受影响。
  - 第三方依赖：需要 AITER 库及 ca_comm 提供 should_custom_ag、all_gather_reg、all_gather_unreg 等方法；若 AITER 不可用或通信器类型不支持，自动回退到 RCCL。
  - dtype 限制：AITER 仅支持 float32/16/bfloat16，int 类型会触发回退。PR 已在条件中显式检查 dtype，避免误入 AITER 导致崩溃（review 中已发现并修复该问题）。
  - CUDA 图一致性：最终代码清晰区分捕获、piecewise 图、warmup 三种状态，避免混合使用不同 collective 导致图无效。
  - 回归风险：仅新增分支，现有路径完全保留；但如果 should_custom_ag 在非预期形状返回 True 可能导致性能损失而非错误。
- 影响：
  - 用户影响：AMD ROCm 用户默认获得 all-gather 性能提升（典型加速 1.13-1.71x），可通过设置 SGLANG_USE_AITER_AG=0 禁用。
  - 系统影响：无外部接口变更，内部通信路径优化。
  - 团队影响：测试和基准脚本可作为后续其他 collective 优化的模板。
  - 风险标记：仅 AMD 平台，依赖 AITER 第三方库，CUDA 图一致性已修复，dtype 限制需关注

关联脉络

- 暂无明显关联 PR