PR #29103 完整报告

sgl-project/sglang

[AMD] Feat/dsv4 aiter reduce scatter decode

合并时间：2026-06-26 04:45

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29103

执行摘要

- 一句话：DSV4 decode 用 reduce_scatter 替代 all_reduce
- 推荐动作：该 PR 值得精读，特别是以下设计点：1）平台条件默认值的实现（_default_hip + _resolve_default），避免了模块加载时引入 torch 依赖；2）在 reduce_scatter_tensor 中插入自定义通信尝试的扩展模式；3）gatherv 与 reduce_scatter 的互斥条件设计。建议后续补充集成测试和 CUDA 图场景的专项测试。

功能与动机

在 DeepSeek-V4 DP-attention TP-MoE decode 路径（MAX_LEN padding，无 EP，tp_size == attn_dp_size）中，MoE combine 之前使用 MoE 内部 post-experts all_reduce（aiter cross_device_reduce_2stage）后跟 dp_scatter。All_reduce 移动约 2 倍于 reduce_scatter 的字节。本 PR 将其替换为等分 reduce_scatter，使用 aiter 自定义（ROCm）或 RCCL 内核，以降低通信开销。

实现拆解

1. environ.py：添加环境变量 SGLANG_DP_USE_REDUCE_SCATTER，默认值通过懒检测函数 _default_hip 动态确定（ROCm/HIP 返回 True，否则 False）；并在 EnvField 中增加 _resolve_default 方法支持可调用默认值，避免模块加载时导入 torch。
2. parallel_state.py：在 GroupCoordinator.reduce_scatter_tensor 中新增尝试 aiter 自定义 reduce_scatter 的分支 _maybe_aiter_reduce_scatter，条件包括平台（HIP）、env 开启、aiter 可用、输入输出连续且形状满足等分要求；当 CUDA 图捕获时正确处理已注册 / 未注册缓冲区；失败时回退到 RCCL。
3. dp_attention.py：修改 is_dp_gatherv_active()，增加 not _DpGatheredBufferWrapper.is_dp_max_padding() 条件，确保 gatherv 对仅在 SUM_LEN 模式下启用，与 MAX_LEN 下的等分 reduce_scatter 互斥。
4. deepseek_v4.py：在 forward 中引入 _use_reduce_scatter 变量，当满足 MAX_LEN decode、tp==dp、无 EP 且 gatherv 不活跃时启用；将原 _use_gatherv_pair 重命名为 _use_reduce_scatterv；在 MoE 调用时传递 use_reduce_scatter=True 跳过内部 all_reduce，并在 combine 阶段调用 dp_reduce_scatter_tensor（会路由到 parallel_state 的新逻辑）。

关键文件：

- python/sglang/srt/distributed/parallel_state.py（模块 分布式通信；类别 source；类型 core-logic；符号 _has_aiter_custom_reduce_scatter, _maybe_aiter_reduce_scatter）：核心通信方法 reduce_scatter_tensor 添加 aiter 自定义 reduce_scatter 路径，新增 _maybe_aiter_reduce_scatter 和 _has_aiter_custom_reduce_scatter 方法，是本次变更的枢纽。
- python/sglang/srt/environ.py（模块 环境变量与配置；类别 source；类型 dependency-wiring；符号 _default_hip, _resolve_default）：新增 SGLANG_DP_USE_REDUCE_SCATTER 环境变量及其平台条件默认值，引入 callable default 支持，为后续平台特性提供通用机制。
- python/sglang/srt/models/deepseek_v4.py（模块 模型实现；类别 source；类型 data-contract）：模型前向逻辑中引入 _use_reduce_scatter 控制 decode 路径的 combine 策略，修改 MoE 调用参数和 combine 分支，是实际收益发生的位置。
- python/sglang/srt/layers/dp_attention.py（模块 DP 注意力层；类别 source；类型 core-logic）：is_dp_gatherv_active() 增加 MAX_LEN 排除条件，使 gatherv 对仅用于 SUM_LEN 模式，确保与 reduce_scatter 路径互斥。

关键符号：_has_aiter_custom_reduce_scatter, _maybe_aiter_reduce_scatter, _default_hip, _resolve_default, is_dp_gatherv_active

关键源码片段

python/sglang/srt/distributed/parallel_state.py

核心通信方法 reduce_scatter_tensor 添加 aiter 自定义 reduce_scatter 路径，新增 _maybe_aiter_reduce_scatter 和 _has_aiter_custom_reduce_scatter 方法，是本次变更的枢纽。

```
def reduce_scatter_tensor(self, output: torch.Tensor, input: torch.Tensor):
    if _is_npu:
        self._reduce_scatter_tensor(output, input)
    elif self._maybe_aiter_reduce_scatter(output, input):
        # 尝试 aiter 自定义 reduce_scatter 成功则返回
        return
    else:
        # 回退到通用 RCCL/NCCL reduce_scatter
        reg_reduce_scatter_tensor(output, input, group_name=self.unique_name)


def _maybe_aiter_reduce_scatter(self, output: torch.Tensor, input: torch.Tensor) -> bool:
    # 条件：ROCm/HIP 平台、环境变量 SGLANG_DP_USE_REDUCE_SCATTER 开启、
    # 具有 aiter 自定义 reduce_scatter 能力、输入输出内存连续、
    # 数据类型为 fp32/fp16/bf16
    if not (
        is_hip()
        and envs.SGLANG_DP_USE_REDUCE_SCATTER.get()
        and self._has_aiter_custom_reduce_scatter()
        and input.is_contiguous()
        and output.is_contiguous()
```

```
            and input.dtype in (torch.float32, torch.float16, torch.bfloat16)
        ):
            return False
        ca_comm = self.ca_comm
        # 大小必须通过 should_custom_ar 检查
        if not ca_comm.should_custom_ar(input):
            return False
        # 等分限制：input 行数必须是 output 行数的 world_size 倍
        if input.shape[0] != output.shape[0] * self.world_size:
            return False
        # CUDA 图捕获处理
        if getattr(ca_comm, "_IS_CAPTURING", False):
            if torch.cuda.is_current_stream_capturing():
                ca_comm.reduce_scatter(input, output, registered=True)
            elif is_in_tc_piecewise_cuda_graph():
                ca_comm.reduce_scatter(input, output, registered=False)
            else:
                # 真实 CUDA 图预热阶段：避免不同的 host 集合通信
                output.zero_()
            return True
        ca_comm.reduce_scatter(input, output, registered=False)
        return True
```

python/sglang/srt/environ.py

新增 SGLANG_DP_USE_REDUCE_SCATTER 环境变量及其平台条件默认值，引入 callable default 支持，为后续平台特性提供通用机制。

```
@functools.lru_cache(maxsize=1) def_default_hip() -> bool: """懒检测 ROCm/HIP 平台，避免模块加载时导入 torch。 首次调用时导入 torch 并检查 torch.version.hip； 不可用时返回 False。 """ try: importtorch return torch.version.hip is not None except Exception: return False classEnvField: ... def_resolve_default(self) -> Any: # 支持可调用默认值（如 _default_hip），仅当环境变量未设置时计算 return self.default() if callable(self.default) else self.default defget(self) -> Any: value = os.getenv(self.name) if self._set_to_none: assert value == str(None) return None if value is None: return self._resolve_default() # 改用可调用解析 try: return self.parse(value) except ValueError as e: default = self._resolve_default() warnings.warn( f'Invalid value for{self.name}:{e}, using default "{default}"' ) return default （在 Envs 类中添加：SGLANG_DP_USE_REDUCE_SCATTER = EnvBool(_default_hip)）
```

评论区精华

审核者 HaiShaw 批准了 PR，并要求“请后续添加测试用例”。amd-bot 的 CI 状态评论指出：PR CI 无法执行核心代码路径（需要夜间测试），AMD PR CI 不完整，4 个失败均为预存 / 基础设施问题，本 PR 的功能在 PR CI 中实际上未经测试。合并前已在 ROCm MI300 和 NVIDIA B200 上通过 GSM8K 验证准确率与性能。

- 测试覆盖不足 (testing): 合并前已通过 GSM8K 验证准确率和性能，但测试需在后续 PR 补充。

风险与影响

- 风险：
  1. 平台覆盖风险：SGLANG_DP_USE_REDUCE_SCATTER 默认仅对 ROCm/HIP 开启，CUDA 平台需手动设为 1 才能使用（会回退到 RCCL *reduce_scatter*），但 CUDA 路径的验证仅在 B200 上进行过一次测试，覆盖不足。
  2. 复杂条件回退：*_maybe_aiter_reduce_scatter* 包含多重条件（HIP、*env*、*aiter* 能力、连续性、数据类型、等分形状、CUDA 图状态），任何一条不满足就回退到 RCCL，若回退逻辑有误可能导致 *double-reduce* 或错误结果。
  3. CUDA 图兼容性：代码对 CUDA 图捕获做了三种分支处理（已注册、分步捕获、预热），若判断错误可能触发 host 端集合通信导致图形化失败或性能下降。
  4. 缺少单元测试：该特性没有对应的单元测试，依赖手动验证和夜间测试，回归风险较高。 - 影响：对 AMD 用户：*decode* 阶段通信量减半，预期显著提升吞吐量（特别是 small-batch *decode* 场景）。对 CUDA 用户：默认无影响，可手动开启获得类似收益（但需 NCCL 支持）。对项目团队：需维护 *aiter* 自定义 *reduce_scatter* 内核及其与 RCCL 的桥接逻辑，增加了分布式通信层的复杂度。对 DeepSeek-V4 模型：性能优化，准确性已确认无回归。 - 风险标记：仅 AMD 默认开启，缺少单元测试覆盖，复杂回退逻辑，CUDA 图兼容性

关联脉络

- 暂无明显关联 PR