PR #35061 完整报告

sgl-project/sglang

[Fix] Select custom all-reduce v2 by topology capability

合并时间：2026-08-19 01:30

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35061

执行摘要

- 一句话：修复 TP16 自定义 all-reduce v2 被多节点策略误禁的回归
- 推荐动作：值得精读。核心看点是能力来源单一化：world size 支持范围由架构配置表 keys 派生，避免硬编码 2..16 与真实调优表不一致；以及"跨节点能力 = NVLink clique ∩ VMM allocator"的判定设计，直接对应 MNNVL fabric 的物理传输前提。建议合入后观察 TP16 MNNVL 与 TP8 MNNVL 的性能监控，并补一个公开 benchmark；同时留意 CI Extra 失败。

功能与动机

32541 引入的 TP2-8 专用多节点策略在能力检测前对 TP16 禁用了 SGLANG_OPT_USE_CUSTOM_ALL_REDUCE_V2，迫使本可合资格运行的配置回退。PR body 明确指出："Before that PR, custom all-reduce v2 was enabled by default and eligible TP16 groups reached the topology and allocator capability check"。因此本 PR 移除多余策略，让 SGLANG_OPT_USE_CUSTOM_ALL_REDUCE_V2 恢复为唯一 enable / opt-out 控制点，并由拓扑、VMM allocator 能力和架构调优的 world size 决定跨节点 v2 选择。

实现拆解

1. 收敛启用开关：删除 python/sglang/srt/server_args.py 中的 ServerArgs._handle_custom_all_reduce_v2_multinode 方法及其在 _handle_environment_variables 中的调用，并同步删除 python/sglang/srt/environ.py 中 SGLANG_ENABLE_CUSTOM_ALL_REDUCE_V2_MULTINODE 的定义。理由：多节点是否可用应由运行时拓扑与分配器能力决定，而不是由启动参数策略在能力检测前先斩断；这样 tp_size <= 8 这类近似条件不再误伤 TP16。
2. 配置表重构：python/sglang/srt/distributed/device_communicators/configs/custom_all_reduce_v2.py 中，将按 world size 单次构建配置的 _sm90_config / _sm100_config 重构为按 num_sm 缓存的 _sm90_configs(num_sm) / _sm100_configs(num_sm)，一次生成该

架构下全部受支持 world size 的 AllReduceConfig；新增 _get_all_reduce_configs() 聚合入口、get_all_reduce_config(world_size) 查表、get_supported_world_sizes() 从配置表 keys 推导支持范围。这样 SM90 仅支持 2..8，SM100 才包含 16，TP16 可用性与架构直接绑定。

3. 能力检测改写：python/sglang/srt/distributed/device_communicators/custom_all_reduce_v2.py 的 can_use_custom_all_reduce_v2 不再读取 multinode 环境变量。节点内继续走 can_use_custom_all_reduce_with_nvlink；跨节点则要求 is_one_nvlink_clique(group, device) 与 _is_vmm_backed_allocator(device) 同时成立，因为 graph zero-copy 依赖 IPC handle 仅限节点内，跨节点 eager 路径依赖 FABRIC / POSIX-fd VMM handle。
4. 派发顺序调整：python/sglang/srt/distributed/parallel_state.py 中让合资格的自定义 all-reduce 输入先于 symmetric-memory PyNccl 运行，较大消息仍回退；配套测试 test_custom_allreduce_precedes_symmetric_memory_pynccl 验证 ca_comm 被选中时不会调用 pynccl_comm.all_reduce。
5. 测试迁移与配套：删除 test/registered/unit/server_args/test_mnnvl_auto_inference.py（4 个 server-args 策略用例）；新增 test/registered/unit/distributed/test_custom_all_reduce_v2_capability.py（拓扑能力矩阵，覆盖跨节点 fabric + VMM 组合）；扩充 test/registered/unit/distributed/test_parallel_state.py。本地相关测试 10 个全部通过；无模型数学变更，未提供公开速度基准。

关键文件：

- python/sglang/srt/distributed/device_communicators/configs/custom_all_reduce_v2.py（模块 通信配置；类别 source；类型 core-logic；符号 _sm90_configs, _sm100_configs, _get_all_reduce_configs, get_all_reduce_config）：核心配置表重构：从单配置函数改为按 num_sm 缓存的配置表，使支持的 world size 随架构自然变化，是本次 TP16 放行的基础。
- python/sglang/srt/distributed/device_communicators/custom_all_reduce_v2.py（模块 通信能力；类别 source；类型 dependency-wiring；符号 can_use_custom_all_reduce_v2, _is_vmm_backed_allocator）：能力检测核心入口：跨节点启用条件改为 NVLink clique + VMM allocator，直接决定 TP16 能否恢复 v2。
- python/sglang/srt/server_args.py（模块 服务参数；类别 source；类型 core-logic；符号 _handle_custom_all_reduce_v2_multinode）：删除 server-args 层多节点策略及调用，恢复单一开关，消除 TP16 被提前禁用的根因。
- python/sglang/srt/environ.py（模块 环境变量；类别 source；类型 configuration）：删除废弃的 SGLANG_ENABLE_CUSTOM_ALL_REDUCE_V2_MULTINODE 环境变量定义，收敛配置面。
- python/sglang/srt/distributed/parallel_state.py（模块 并行状态；类别 source；类型 core-logic）：调整 all_reduce 派发顺序，自定义 all-reduce 先于 symmetric-memory PyNccl，是本次行为变化的一部分。
- test/registered/unit/distributed/test_custom_all_reduce_v2_capability.py（模块 能力测试；类别 test；类型 test-coverage；符号 _patch_group, test_topology_capability, is_one_clique, is_vmm_backed）：新增能力矩阵测试，锁定跨节点 fabric + VMM 的组合

条件，防止能力检测退化为恒真或恒假。

- test/registered/unit/server_args/test_mnnvl_auto_inference.py（模块 参数测试；类别 test；类型 test-deletion；符号 _cleared, ctx, TestCaV2MultinodeAuto, test_fabric_multinode_auto_enables）：删除针对已移除 server-args 策略的 4 个测试用例，测试面随策略移除而收缩。
- test/registered/unit/distributed/test_parallel_state.py（模块 派发测试；类别 test；类型 test-coverage；符号 test_custom_allreduce_precedes_symmetric_memory_pynccl）：新增派发顺序测试，验证自定义 all-reduce 优先于 symmetric-memory PyNccl 的新调度契约。

关键符号：*can_use_custom_all_reduce_v2*, *_is_vmm_backed_allocator*, *get_supported_world_sizes*, *_sm90_configs*, *_sm100_configs*, *_get_all_reduce_configs*, *get_all_reduce_config*

关键源码片段

python/sglang/srt/distributed/device_communicators/custom_all_reduce_v2.py

能力检测核心入口：跨节点启用条件改为 NVLink clique + VMM allocator，直接决定 TP16 能否恢复 v2。

```
# 检查当前缓存分配器是否由 expandable segments 的 VMM 支撑。
# 多节点 zero-copy 依赖 FABRIC / POSIX-fd VMM handle，因此这是跨节点放行的前置条件。
def _is_vmm_backed_allocator(device: torch.device) -> bool:
    probe = torch.empty(1, dtype=torch.uint8, device=device)
    return is_vmm_pointer(probe.data_ptr())


def can_use_custom_all_reduce_v2(
    group: ProcessGroup,
    device: torch.device,
) -> bool:
    # 支持范围不再硬编码 2..16，而从当前架构配置表 keys 推导：
    # SM90 只有 2..8，SM100 才有 16，Hopper TP16 不会被错误放行。
    supported = get_supported_world_sizes()
    if dist.get_world_size(group=group) not in supported:
        return False

    # 跨节点（MNNVL）：节点内 NVLink 检查不适用，真正门槛是
    # single NVLink clique + VMM allocator 同时成立；否则回退旧实现。
    if not all(in_the_same_node_as(group, source_rank=0)):
        return is_one_nvlink_clique(group, device) and _is_vmm_backed_allocator(device)

    # 节点内路径保持原有 NVLink 能力检查
    full_nvlink = can_use_custom_all_reduce_with_nvlink(
        group=group,
        device=device,
```

```
        supported_world_size=list(supported),
        cls_name="CustomAllReduceV2",
    )
    return full_nvlink is True
```

test/registered/unit/distributed/test_custom_all_reduce_v2_capability.py

新增能力矩阵测试，锁定跨节点 fabric + VMM 的组合条件，防止能力检测退化为恒真或恒假。

```
# 拓扑能力矩阵测试：same_node=False 表示跨节点，此时必须同时满足
# single NVLink clique 与 VMM allocator；same_node=True 时不要求 fabric / VMM。
@pytest.mark.parametrize(
    ("same_node", "has_fabric_clique", "uses_vmm", "expected"),
    [
        (False, True, True, True),
        (False, False, True, False),
        (False, True, False, False),
        (True, None, None, True),
    ],
)
def test_topology_capability(
    monkeypatch, same_node, has_fabric_clique, uses_vmm, expected
):
    # 跨节点用例用 world size 16（覆盖本次修复的 TP16 场景），节点内用 8
    world_size = 8 if same_node else 16
    group, device = _patch_group(
        monkeypatch,
        world_size=world_size,
        same_node=same_node,
    )

    # 节点内分组不应触碰 fabric clique 与 VMM 探测，用 pytest.fail 约束调用面
    def is_one_clique(group, device):
        if same_node:
            pytest.fail("intra-node groups do not need a fabric clique")
        return has_fabric_clique

    def is_vmm_backed(device):
        if same_node:
            pytest.fail("intra-node groups do not need VMM")
        return uses_vmm

    intra_node_capability = Mock(return_value=True)
    monkeypatch.setattr(custom_all_reduce_v2, "is_one_nvlink_clique", is_one_clique)
    monkeypatch.setattr(custom_all_reduce_v2, "_is_vmm_backed_allocator", is_vmm_backed)
    monkeypatch.setattr(
        custom_all_reduce_v2,
        "can_use_custom_all_reduce_with_nvlink",
        intra_node_capability,
    )
```

```
    assert custom_all_reduce_v2.can_use_custom_all_reduce_v2(group, device) is expected
    if same_node:
        intra_node_capability.assert_called_once_with(
            group=group,
            device=device,
            supported_world_size=[world_size],
            cls_name="CustomAllReduceV2",
        )
    else:
        intra_node_capability.assert_not_called()
```

评论区精华

PR 没有任何 review comments，merrymercy 直接 APPROVED，说明改动虽涉及分布式通信但评审路径清晰。Issue 区 nvpohanh 将 @ajit283 加入关注，但未产生公开技术讨论。值得注意的未解事项是 PR Test (Extra) 有一个失败任务（Run #31978608691），公开讨论中未给出解释。

- 评审与 CI 状态 (other): 无遗留技术争议，已合入 main；EXTRA CI 失败需另行排查。

风险与影响

- 风险：
  1. 派发顺序变化：parallel_state.py 让自定义 all-reduce 优先于 symmetric-memory PyNccl，影响所有 TP 规模的默认路径，虽然保留 fallback，但缺少端到端 benchmark，存在性能回归未被量化的可能。
  2. 运行时探测依赖：跨节点放行依赖 is_one_nvlink_clique 与 _is_vmm_backed_allocator 的运行时探测；探测不准可能导致错误回退（性能损失）或错误启用，测试覆盖了 False / False 分支但未覆盖真实硬件组合。
  3. 支持范围推导：get_supported_world_sizes() 从配置表 keys 推导，未来新增架构配置时若漏写某个 world size，会静默跳过 v2。
  4. 测试保护变化：test_mnnvl_auto_inference.py 删除后，server-args 层不再有保护性断言，若后续重新引入类似策略，只能靠 capability 测试间接覆盖。
  5. CI未决问题：EXTRACI存在失败任务且未在讨论中解释，需确认是否为偶发环境问题。 - 影响：用户侧：GB200 / GB300 MNNVL TP16 等「跨节点 single clique + VMM」部署恢复自定义 all-reduce v2 加速；非 fabric 多节点和 Hopper TP16 继续回退到 legacy 路径。运维侧：环境变量契约简化，SGLANG_ENABLE_CUSTOM_ALL_REDUCE_V2_MULTINODE 被移除，依赖它的脚本需同步更新。系统侧：all-reduce 派发优先级变化可能影响与 PyNccl / symmetric-memory 的交互，需要观察 TP8 / TP16 的吞吐与延迟指标。组织侧：测试从启动参数策略迁移到能力矩阵，降低 server-args 维护成本，但需要确认 CI Extra 失败。 - 风险标记：通信核心路径变更，运行时能力探测依赖，缺少公开性能基准，EXTRA CI 失败未解释

关联脉络

- PR #32541 [Kimi] Support kimi-k3: 该 PR 引入的 TP2-8 特定多节点策略在能力检测前禁用了 TP16 的 SGLANG_OPT_USE_CUSTOM_ALL_REDUCE_V2，是本次修复的回归来源；本 PR 删除该策略并改为拓扑能力检测。