PR #22985 完整报告

sgl-project/sglang

[AMD] Support eplb for moriep

合并时间：2026-06-11 01:23

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22985

执行摘要

- 一句话：支持 MoRI 专家负载均衡并修复 RCCL P2P 挂起
- 推荐动作：值得精读，尤其是 P2P 分块策略和低延迟钩子集成方式。建议关注环境变量配置的运维参数调整。

功能与动机

根据 PR body，主要目标是支持 EPLB 在 mori 后端下工作，并修复 RCCL 批量 P2P 操作在 EPLB 重平衡时挂起的问题。

实现拆解

1. MoRI dispatch 集成 EPLB（moriep.py）：在 _dispatch_core 和 dispatch_b 中，向 mori dispatch 调用添加 call_local_expert_count=True，获取 GPU 上的 local_expert_count，并通过低延迟钩子 on_deepep_dispatch_low_latency 传递给全局专家分布记录器。
2. RCCL P2P 挂起修复（expert_location_updater.py）：在 _execute_p2p_ops 中，对于 HIP 平台，将 sorted_infos 按 logical_expert_id 分块（默认块大小 32），每块单独调用 batch_isend_irecv 并等待，避免 RCCL 累积过多未完成操作。
3. topk_id 类型修复（expert_location_dispatch.py）：在 _topk_ids_logical_to_physical_static 和 _topk_ids_logical_to_physical_dynamic 中，对 HIP 平台将结果转换为 topk_ids 的 dtype，确保内存格式一致。
4. 其他配合修改（expert_distribution.py、topk.py、aiter.py）：微调导入和调用。
5. 测试：新增 test/manual/ep/test_eplb_mori.py（242 行）覆盖 stat、stat_approx、multi-chunk 三种模式；在 test/registered/amd/test_moriep_small.py 中追加 TestEPLBMoriStat 类，使用 GSM8K 200 题验证准确性。
6. 文档：在 environment_variables.mdx 中添加 SGLANG_EPLB_ROCM_P2P_BATCH_CHUNK_SIZE 说明。

关键文件：

- test/manual/ep/test_eplb_mori.py（模块 测试；类别 test；类型 test-coverage；符号 wait_all_ports_release, TestEPLBMoriStat, setUpClass, tearDownClass）：新增的主测试文件，覆盖了 EPLB + mori 的三种模式（stat、stat_approx、multi-chunk），验证精确度和无挂起。

- test/registered/amd/test_moriep_small.py（模块 测试；类别 test；类型 test-coverage；符号 TestEPLBMoriStat, setUpClass, tearDownClass, test_gsm8k）：已在 CI 中的测试文件，新增了 TestEPLBMoriStat 类，使 EPLB+mori 能在 AMD CI 中被执行。
- python/sglang/srt/eplb/expert_location_updater.py（模块 负载均衡；类别 source；类型 core-logic；符号 _is_hip, SGLANG_EPLB_ROCM_P2P_BATCH_CHUNK_SIZE）：修复 RCCL P2P 挂起的核心文件，在 _execute_p2p_ops 中添加了基于 HIP 的分块逻辑。
- python/sglang/srt/layers/moe/token_dispatcher/moriep.py（模块 MoE 调度；类别 source；类型 dependency-wiring；符号 MoriEPDispatcher._dispatch_core, MoriEPDispatcher.dispatch_b）：EPLB 集成的主入口，将 mori dispatch 的 local_expert_count 接入专家分布记录器。
- python/sglang/srt/eplb/expert_location_dispatch.py（模块 负载均衡；类别 source；类型 bugfix；符号 _topk_ids_logical_to_physical_static, _topk_ids_logical_to_physical_dynamic）：topk_id 逻辑到物理映射的 HIP dtype 修复。
- python/sglang/srt/layers/moe/moe_runner/aiter.py（模块 运行时；类别 source；类型 dependency-wiring；符号 AiterRunner.run）：延迟导入 GateMode 避免非 MXFP4 模型运行时导入错误。
- docs_new/docs/references/environment_variables.mdx（模块 文档；类别 other；类型 documentation）：文档记录新环境变量
- python/sglang/srt/eplb/expert_distribution.py（模块 负载均衡；类别 source；类型 core-logic）：微小调整以支持 mori 的 local_expert_count 记录。
- python/sglang/srt/layers/moe/topk.py（模块 MoE；类别 source；类型 core-logic）：配合 EPLB 的微小修改。

关键符号：ExpertLocationUpdater.update, _execute_p2p_ops, MoriEPDispatcher._dispatch_core, MoriEPDispatcher.dispatch_b, _topk_ids_logical_to_physical_static, _topk_ids_logical_to_physical_dynamic, TestEPLBMoriStat.test_gsm8k, wait_all_ports_release

关键源码片段

test/manual/ep/test_eplb_mori.py

新增的主测试文件，覆盖了 EPLB + mori 的三种模式（stat、stat_approx、multi-chunk），验证精确度和无挂起。

```
import os
from types import SimpleNamespace
from sglang.srt.server_args import ZMQ_TCP_PORT_DELTA
from sglang.srt.utils import kill_process_tree
from sglang.srt.utils.network import is_port_available
from sglang.test.few_shot_gsm8k import run_eval as run_eval_few_shot_gsm8k
from sglang.test.test_utils import (
    DEFAULT_DEEPEP_MODEL_NAME_FOR_TEST,
    DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
    DEFAULT_URL_FOR_TEST,
    CustomTestCase,
```

```
    popen_launch_server,
)

# 等待所有 ZMQ 端口释放，避免残留进程冲突
def wait_all_ports_release(base_url, timeout_s=60):
    import time
    port = int(base_url.split(":")[-1])
    offsets = [0, ZMQ_TCP_PORT_DELTA, ZMQ_TCP_PORT_DELTA + 1, ZMQ_TCP_PORT_DELTA +
    2, ZMQ_TCP_PORT_DELTA + 3, ZMQ_TCP_PORT_DELTA + 4]
    for _ in range(timeout_s):
        if all(is_port_available(port + off) for off in offsets):
            return
        time.sleep(1)
    print(f"Warning: some ports still occupied after {timeout_s}s")

mori_env = {
    **os.environ,
    "SGLANG_USE_AITER": "1",
    "SGLANG_MORI_DISPATCH_DTYPE": "bf16",
    "SGLANG_MORI_NUM_MAX_DISPATCH_TOKENS_PER_RANK": "4096",
    "SGLANG_EPLB_ROCM_P2P_BATCH_CHUNK_SIZE": "32",
    "MORI_SHMEM_MODE": "ISOLATION",
}
# ... 测试类和 GSM8K 运行逻辑
class TestEPLBMoriStat(CustomTestCase):
    @classmethod
    def setUpClass(cls):
        cls.model = DEFAULT_DEEPEP_MODEL_NAME_FOR_TEST
        cls.base_url = DEFAULT_URL_FOR_TEST
        other_args = (
            common_args + eplb_args +
            ["--deepep-mode", "normal", "--expert-distribution-recorder-mode", "stat"]
        )
        cls.process = popen_launch_server(cls.model, cls.base_url, timeout=DEFAULT_TIMEOUT_
        FOR_SERVER_LAUNCH * 5,
                                          other_args=other_args, env=mori_env)
    def test_gsm8k(self):
        args = SimpleNamespace(num_shots=5, data_path=None, num_questions=1209, max_new_
        tokens=512,
                               parallel=1209, host="http://127.0.0.1", port=int(self.base_url.split(":")[-1])
                               )
        metrics = run_eval_few_shot_gsm8k(args)
        self.assertGreaterEqual(metrics["accuracy"], 0.9)
```

python/sglang/srt/eplb/expert_location_updater.py

修复 RCCL P2P 挂起的核心文件，在 _execute_p2p_ops 中添加了基于 HIP 的分块逻辑。

```
# 导入新增
from sglang.srt.utils import get_bool_env_var, get_int_env_var, is_hip
```

```
_log_INPUT = get_bool_env_var("SGLANG_EXPERT_LOCATION_UPDATER_LOG_INPUT")
_is_hip = is_hip()

class ExpertLocationUpdater:
    # ... 其他方法
    def _execute_p2p_ops(p2p_op_infos):
        sorted_infos = sorted(p2p_op_infos, key=lambda info: info[0])
        p2p_ops = [op for _, ops in sorted_infos for op in ops]
        if len(p2p_ops) == 0:
            return

        if _is_hip:
            # ROCm 平台：将 P2P 操作按 expert_id 分块，避免 RCCL 积累过多未完成操作导致挂起
            batch_chunk_size = get_int_env_var("SGLANG_EPLB_ROCM_P2P_BATCH_CHUNK_SIZE",
            32)
            ops_by_expert = {eid: ops for eid, ops in sorted_infos}
            for start in range(0, num_physical_experts, batch_chunk_size):
                batch_ops = []
                for eid in range(start, min(start + batch_chunk_size, num_physical_experts)):
                    if eid in ops_by_expert:
                        batch_ops.extend(ops_by_expert[eid])
                if batch_ops:
                    reqs = torch.distributed.batch_isend_irecv(batch_ops)
                    for req in reqs:
                        req.wait()
        else:
            # CUDA 路径保持不变
            reqs = torch.distributed.batch_isend_irecv(p2p_ops)
            for req in reqs:
                req.wait()
```

python/sglang/srt/layers/moe/token_dispatcher/moriep.py

EPLB 集成的主入口，将 mori dispatch 的 local_expert_count 接入专家分布记录器。

```
# 导入专家分布记录器
from sglang.srt.eplb.expert_distribution import get_global_expert_distribution_recorder

class MoriEPDispatcher:
    def _dispatch_core(self, ...):
        # 同步路径
        ( ... ) = dispatch_fn(
            hidden_states, topk_weights, scale, topk_ids,
            call_local_expert_count=True, # 新增参数，让 mori 返回 local_expert_count
        )
        # ...
        # 使用低延迟钩子记录 local_expert_count（GPU tensor，而非 CPU list）
        get_global_expert_distribution_recorder().on_deepep_dispatch_low_latency(
            self.mori_op.local_expert_count
        )
```

```
        # ...
    def dispatch_b(self, ...):
        # 异步低延迟路径
        self.mori_op.dispatch_recv(call_local_expert_count=True)
        get_global_expert_distribution_recorder().on_deepep_dispatch_low_latency(
            self.mori_op.local_expert_count
        )
```

评论区精华

Review 中 HaiShaw 提出两个关键意见：

- 变更应仅针对 ROCm 平台，避免影响 CUDA 路径。作者通过添加 _is_hip 条件实现。
- 环境变量名应体现 ROCm 限定，从 SGLANG_EPLB_P2P_BATCH_CHUNK_SIZE 改为 SGLANG_EPLB_ROCM_P2P_BATCH_CHUNK_SIZE。
- ROCm-specific 变更建议 (design): 作者添加 _is_hip 条件包裹相关代码，确保 CUDA 路径不受影响。
- 环境变量命名 (design): 作者采纳并更新了代码和文档。

风险与影响

- 风险：主要风险：
  - 新环境变量默认值（32）是否适用于所有规模和负载？若块大小不匹配 expert 数量，可能导致 P2P 操作配对失败。代码要求所有 rank 使用相同边界，但未强制同步配置。
  - MoRI dispatch 的 call_local_expert_count 参数需要 mori 库版本支持，若版本不匹配可能运行时报错。
  - topk_id dtype 转换仅在 HIP 路径执行，假设 topk_ids 原始 dtype 为 int32，若其他后端使用 int64 可能截断。
  - 测试仅在 AMD CI 运行，未覆盖大规模长稳场景。
  - 影响：影响范围：仅限于 AMD (ROCm) 平台且使用 mori 后端的 MoE 模型。用户可通过 --enable-eplb 和 --moe-a2a-backend mori 启用。预期改善专家负载均衡效果，提升整体吞吐。团队需维护新环境变量和测试。
  - 风险标记：环境变量默认值需验证，mori 库版本依赖，HIP dtype 转换假设，测试仅在 AMD CI 运行

关联脉络

- 暂无明显关联 PR