PR #2173 完整报告

THUDM/slime

[docker] Update SGLang patch for PD R3 routed experts

合并时间：2026-07-03 13:40

原文链接：http://prhub.com.cn/THUDM/slime/pull/2173

执行摘要

- 一句话：更新 SGLang patch 并增加 routed experts 校验
- 推荐动作：建议精读此 PR，尤其是 slime/utils/types.py 中新增的形状校验逻辑和 sglang.patch 中 routed_experts_start_len 的设计。该变更为后续 PD 场景下正确使用路由专家数据奠定了基础，值得关注其跨 PR 演进。

功能与动机

PR body 明确指出：'Sync the SGLang patch with PD prefill/decode routed expert handling and validate merged routed_experts shapes in rollout samples.' 目的是适配 SGLang 上游 R3 路由专家分离功能，确保 rollout 数据层的正确性。

实现拆解

1. 更新 SGLang 补丁（docker/patch/latest/sglang.patch）：在 SGLang 调度器中增加 routed_experts_start_len 字段，用于区分 prefill 和 decode 阶段的专家路由信息；在 prefill 完成后的传输成功路径中调用 _maybe_collect_routed_experts 收集 decode 侧所需的专家信息；同时新增 release_memory_occupation/resume_memory_occupation 方法和 bootstrap/transfer 超时机制。
2. 增加校验逻辑（slime/utils/types.py）：在 Sample.append_response_tokens 中，当解码出 routed_experts 元数据时，先计算期望的元素总数（token 行数 × num_layers × moe_router_topk），然后与实际元素数量比对，若不匹配则抛出 ValueError，避免后续 reshape 产生隐秘错误。
3. 补充单元测试（tests/test_rollout_metrics.py）：新增 test_append_response_tokens_ignores_split_pd_routed_experts，验证当元数据中存在 pd_prefill_routed_experts 和 pd_decode_routed_experts 时，rollout_routed_experts 应为 None（split 模式不应合并）；新增 test_append_response_tokens_rejects_mismatched_routed_experts_shape，验证元素数量不匹配时正确抛出异常。
4. 更新基础设施：docker/Dockerfile 更换了 sgl-router wheel 的下载 hash；docker/version.txt 版本号从 nightly-dev-20260701a 升级为 nightly-dev-20260703a。

关键文件：

- slime/utils/types.py（模块 数据模型；类别 source；类型 core-logic）：核心变更文件，在 Sample.append_response_tokens 中增加了 routed_experts 形状校验，防止数据不匹配

导致的运行时错误。

- tests/test_rollout_metrics.py（模块 测试；类别 test；类型 test-coverage；符号 test_append_response_tokens_ignores_split_pd_routed_experts, test_append_response_tokens_rejects_mismatched_routed_experts_shape）：新增两个单元测试，覆盖 split PD 忽略和形状不匹配场景，保障校验逻辑正确。
- docker/patch/latest/sglang.patch（模块 SGLang 补丁；类别 infra；类型 patch；符号 setstate）：SGLang 补丁更新是功能核心，实现 PD 下路由专家信息的分离传输和收集机制。
- docker/Dockerfile（模块 Docker；类别 infra；类型 infrastructure）：基础设施变更，更新 sgl-router wheel 版本哈希，确保与 patch 兼容。
- docker/version.txt（模块 Docker；类别 docs；类型 documentation）：版本号同步更新，反映构建日期变更。

关键符号：_apply_meta_info, test_append_response_tokens_ignores_split_pd_routed_experts, test_append_response_tokens_rejects_mismatched_routed_experts_shape

关键源码片段

slime/utils/types.py

核心变更文件，在 Sample.append_response_tokens 中增加了 routed_experts 形状校验，防止数据不匹配导致的运行时错误。

```
# 在 Sample.append_response_tokens 方法中处理 routed_experts 元数据
routed_experts = decode_int32_meta_array(meta_info, "routed_experts")
if routed_experts is not None:
    if args is None:
        raise ValueError("args is required to decode routed experts metadata.")
    # 计算期望的 token 行数（除 prompt 后生成的部分）
    expected_rows = len(self.tokens) - 1
    expected_numel = expected_rows * args.num_layers * args.moe_router_topk
    # 校验元素数量，防止 reshape 导致数据错位
    if routed_experts.numel() != expected_numel:
        raise ValueError(
            "SGLang routed_experts element count does not match sample tokens: "
            f"got={routed_experts.numel()}, expected={expected_numel} "
            f"(tokens={len(self.tokens)}, num_layers={args.num_layers}, "
            f"moe_router_topk={args.moe_router_topk})."
        )
    self.rollout_routed_experts = routed_experts.reshape(
        expected_rows,
        args.num_layers,
        args.moe_router_topk,
    )
```

tests/test_rollout_metrics.py

新增两个单元测试，覆盖 split PD 忽略和形状不匹配场景，保障校验逻辑正确。

```
# 测试 1：当 SGLang 返回分离的 prefill/decode 路由专家时，rollout_routed_experts 应为 None
@pytest.mark.unit
def test_append_response_tokens_ignores_split_pd_routed_experts():
    sample = Sample(tokens=[101, 102, 103, 104])
    sample.append_response_tokens(
        _make_args(),
        tokens=[],
        trainable=True,
        meta_info={
            "pd_prefill_routed_experts": _b64_int32([0, 1, 2, 3, 4, 5, 6, 7]),
            "pd_decode_routed_experts": _b64_int32([8, 9, 10, 11]),
            "finish_reason": {"type": "stop"},
        },
    )
    assert sample.rollout_routed_experts is None

# 测试 2：元素数量不匹配时抛出 ValueError
@pytest.mark.unit
def test_append_response_tokens_rejects_mismatched_routed_experts_shape():
    sample = Sample(tokens=[101, 102, 103])
    with pytest.raises(ValueError, match="routed_experts element count"):
        sample.append_response_tokens(
            _make_args(),
            tokens=[],
            trainable=True,
            meta_info={
                "routed_experts": _b64_int32([0, 1, 2, 3]), # 期望 2*L*K 但只给了 4 个
                "finish_reason": {"type": "stop"},
            },
        )
```

docker/patch/latest/sglang.patch

SGLang 补丁更新是功能核心，实现 PD 下路由专家信息的分离传输和收集机制。

```
# 在 SGLang Scheduler 的 _add_request_to_queue 方法中，为 decode 模式设置起始长度
if self.disaggregation_mode == DisaggregationMode.DECODE:
    # Prefill 返回 prompt 侧的路由专家；decode 只需要 prompt 之后的行
    req.routed_experts_start_len = len(req.origin_input_ids)

# 在 prefill 传输成功后收集 decode 侧的路由专家
if poll == KVPoll.Success:
    if req.return_routed_experts:
        self.batch_result_processor._maybe_collect_routed_experts(req)
    release_kv_cache(req, self.tree_cache)
```

评论区精华

当前 PR 没有收到 review 评论，但实现暗含一个设计决定：当 SGLang 分别返回 pd_prefill_routed_experts 和 pd_decode_routed_experts 时，slime 选择不合并它们，而是

直接忽略（rollout_routed_experts 置为 None）。这意味着 split PD 模式下的路由专家信息不参与后续训练，只在合并后的 routed_experts 键存在时才使用。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 兼容性风险（slime/utils/types.py）：新增的校验可能对已有 rollout 数据造成破坏——如果 SGLang 返回的 routed_experts 元素数量与期望不符，训练流程将直接崩溃，而之前可能静默通过了错误形状。这可能暴露上游或配置错误。
  2. 补丁稳定性（docker/patch/latest/sglang.patch）：补丁依赖特定的 SGLang commit（对应索引变化），未来 SGLang 更新可能导致补丁冲突。
  3. 依赖变更（docker/Dockerfile）：sgl-router wheel 的 hash 更新，如果新版本存在未暴露的 bug，会影响整个路由器模块。
  4. 覆盖不全：新增的测试只覆盖了 split PD 忽略和形状不匹配两种场景，未测试正常合并的边界情况（如 0 行 token）。 - 影响：影响范围：中等。主要影响使用 PD（prefill/decode 分离部署）且启用 MoE 路由专家（routed_experts）的用户。对非 PD 模式或无专家路由的模型无影响。影响程度：数据一致性增强，但可能暴露此前隐藏的配置错误。团队：需要 sglang 补丁作者确认补丁兼容性。 - 风险标记：校验可能暴露配置错误，补丁依赖上游 SGLang commit，sgl-router 版本变更

关联脉络

- PR #2145 [docker] fix top_p mask speed issue: 同样涉及 docker/patch/latest/sglang-top_p.patch 和 Dockerfile，同一维护者，属于持续的 SGLang 补丁更新系列。
- PR #2169 Merging profiling info into router: 同样修改了 docker/patch/latest/sglang.patch 和 Dockerfile，属于同一模块基础设施演进。