# PR #34191 完整报告

sgl-project/sglang

[PD] Skip speculative verify scratch on prefill servers (saves num_draft_tokens x mamba pool per rank)

合并时间：2026-08-10 11:56

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34191

## 执行摘要

- 一句话：PD prefill 服务器跳过 spec verify 相关分配
- 推荐动作：值得精读，尤其是理解复用 draft-head 先例实现显存优化的设计思路。建议后续补充针对 PD prefill 模式的单元测试，覆盖 pool 分配、graph 捕获和 warmup 逻辑。

## 功能与动机

在 PD 分离的 prefill 服务器上，speculative decoding 的 TARGET_VERIFY 永远不会执行，但现有实现仍会分配 verify-only 的 mamba 状态快照（intermediate_ssm_state_cache）并捕获 CUDA graph，导致显存浪费甚至 OOM。PR body 指出：在 256-slot 池下，每 rank 浪费约 24 GB，且 prefill CUDA graph 捕获时可能因瞬时大峰值而 OOM。

## 实现拆解

1. pool 分配：在 kv_cache_configurator.py::_build_hybrid_req_pool 中，当 disaggregation_mode == "prefill" 时，将 speculative_num_draft_tokens 设为 None，复用 draft-head 先例，使 pool 跳过 SpeculativeState 缓冲区的分配。
2. CUDA graph 捕获：在 cuda_graph_setup.py::capture_decode_graph 中，增加早退条件，当是 PD prefill target worker 时直接返回 no_capture，跳过 target-verify graph 的捕获。
3. warmup 前向：在 base_runner.py::_dummy_run 中，对 PD prefill target worker 不提升为 TARGET_VERIFY 模式，而是使用普通 DECODE 进行 warmup，避免因 pool 缺少 SpeculativeState 而触发断言。
4. 测试配套：无新增测试，依赖现有 CI 覆盖。

关键文件：

- python/sglang/srt/model_executor/runner/base_runner.py（模块 执行器；类别 source；类型 data-contract）：控制 warmup forward 模式，防止 PD prefill target worker 触发 TARGET_VERIFY 导致断言失败。
- python/sglang/srt/mem_cache/kv_cache_configurator.py（模块 缓存配置；类别 source；类型 core-logic）：核心逻辑：在 PD prefill 模式下跳过 SpeculativeState 缓冲区分配，节省数 GB 显存。

- python/sglang/srt/model_executor/model_runner_components/cuda_graph_setup.py（模块 图捕获；类别 source；类型 data-contract）：跳过 target-verify CUDA graph 捕获，避免无用的 graph 内存占用。

关键符号：_dummy_run, _build_hybrid_req_pool, capture_decode_graph

## 关键源码片段

### python/sglang/srt/model_executor/runner/base_runner.py

控制 warmup forward 模式，防止 PD prefill target worker 触发 TARGET_VERIFY 导致断言失败。

```
# python/sglang/srt/model_executor/runner/base_runner.py
def _dummy_run(self, ...):
    ...
    num_tokens_per_req = 1
    # PD prefill target worker 的 pool 没有 SpeculativeState,
    # 所以 TARGET_VERIFY dummy forward 会触发 linear-attn 后端的
    # pool-type 断言。改为普通 DECODE 进行 warmup。
    _is_pd_prefill_target = (
        mr.server_args.disaggregation_mode == "prefill" and not mr.is_draft_worker
    )
    if mr.spec_algorithm.is_speculative() and not _is_pd_prefill_target:
        if mr.is_draft_worker:
            assert (
                mr.spec_algorithm.supports_target_verify_for_draft()
            ), "This should not happen"
        capture_forward_mode = ForwardMode.TARGET_VERIFY
        num_tokens_per_req = mr.decode_num_tokens_per_req()
    ...
```

### python/sglang/srt/mem_cache/kv_cache_configurator.py

核心逻辑：在 PD prefill 模式下跳过 SpeculativeState 缓冲区分配，节省数 GB 显存。

```
# python/sglang/srt/mem_cache/kv_cache_configurator.py
def _build_hybrid_req_pool(self, *, max_num_reqs, extra_max_context_len):
    ...
    # PD prefill 服务器从不执行 TARGET_VERIFY，跳过 verify-only 的
    # per-draft-token 状态快照（见 draft-head 先例：None => 池跳过
    # SpeculativeState）。
    speculative_num_draft_tokens=(
        None
        if get_disagg().disaggregation_mode == "prefill"
        else max_speculative_num_draft_tokens()
    ),
    ...
```

### python/sglang/srt/model_executor/model_runner_components/cuda_graph_setup.py

跳过 target-verify CUDA graph 捕获，避免无用的 graph 内存占用。

```
# python/sglang/srt/model_executor/model_runner_components/cuda_graph_setup.py
def capture_decode_graph(*, model_runner):
    ...
    # PD prefill 服务器从不重放 target-verify graph，且其池构建时
    # 不含 spec-verify 所需的 scratch。直接返回 no_capture。
    if (
        model_runner.spec_algorithm.is_speculative()
        and not model_runner.is_draft_worker
        and model_runner.server_args.disaggregation_mode == "prefill"
    ):
        return no_capture
    ...
```

# 评论区精华

无 review 评论。

- 暂无高价值评论线程

# 风险与影响

- 风险：风险较低，但需注意：改动会影响所有 hybrid linear-attention 模型在 PD prefill 模式下的初始化路径，若未来新增模型或功能依赖 TARGET_VERIFY 或 SpeculativeState，可能被跳过。此外，未增加单元测试，回归风险依赖 CI 覆盖。
- 影响：对 PD prefill 服务器显著减少显存占用（每 rank 最多节省约 24 GB），避免 OOM，提升部署密度。对 decode 服务器、聚合部署和 draft worker 无影响。团队需要关注未来涉及 spec-verify 的功能变更。
- 风险标记：缺少测试覆盖

# 关联脉络

- 暂无明显关联 PR