PR #26607 完整报告

sgl-project/sglang

Do not cap DeepSeek V4 PD prefill by SWA pool size

合并时间：2026-06-01 19:29

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26607

执行摘要

- 一句话：解除 DS V4 PD prefill 的 SWA 上限
- 推荐动作：该 PR 改动虽小，但揭示了继承层次导致的容量管理 bug，值得研读。对 PD 分离部署和 SWA pool 设计感兴趣的工程师可以重点关注。

功能与动机

DS V4 PD prefill 此前被 SWA pool size 限制，因为 DeepSeekV4TokenToKVPool 继承自通用 SWA pool，prefill 阶段错误地使用 SWA pool 容量作为 admission 上限，导致长请求被拒。类似问题已在解码阶段通过 PR#24857 修复，但 prefill 仍未修复。

实现拆解

1. 修改 max_total_num_tokens 来源：在 prefill.py 的 PrefillServer.__init__中，将原先从构造参数赋值改为从 self.scheduler.tp_worker.model_runner.max_token_pool_size 获取，该值代表全量 KV pool 容量，不受 SWA pool 限制。
2. 移除 SWA 容量 cap：删除对 is_hybrid_swa 的判断，不再将 max_total_num_tokens 与 swa_max_total_num_tokens 取小，因为 DS V4 在 prefill 时仅为 sliding window 分配 SWA KV，不应对长 prompt 进行 SWA pool 限制。
3. 与解码侧保持一致：该逻辑与 schedule_policy.py 中非 PD 场景的判断对齐，确保 prefill 和 decode 使用相同的容量边界。

关键文件：

- python/sglang/srt/disaggregation/prefill.py（模块 解聚层；类别 source；类型 core-logic；符号 init）：核心逻辑修复，解除 SWA pool 对 prefill 的容量限制

关键符号：init

关键源码片段

python/sglang/srt/disaggregation/prefill.py

核心逻辑修复，解除 SWA pool 对 prefill 的容量限制

```
# 文件：python/sglang/srt/disaggregation/prefill.py
# 类 PrefillServer 的 __init__ 方法

def __init__(
```

```
    self,
    token_to_kv_pool: KVCache,
    draft_token_to_kv_pool: Optional[KVCache],
    req_to_metadata_buffer_idx_allocator: ReqToMetadataIdxAllocator,
    metadata_buffers: MetadataBuffers,
    tp_rank: int,
    tp_size: int,
    gpu_id: int,
    bootstrap_port: int,
    gloo_group: ProcessGroup,
    max_total_num_tokens: int, # 此参数不再使用，保留兼容性
    scheduler: Scheduler,
    pp_rank: int,
    pp_size: int,
    transfer_backend: TransferBackend,
):
    # ... 其他属性赋值 ...
    self.scheduler = scheduler

    # 变更：从 model_runner 获取真实的 max_token_pool_size
    # 原代码使用传入的 max_total_num_tokens 参数，该参数可能被 SWA pool 限制
    self.max_total_num_tokens = (
        self.scheduler.tp_worker.model_runner.max_token_pool_size
    )

    # ... transfer_backend 相关代码 ...
    self.kv_manager = self._init_kv_manager()

    # 删除以下代码：不再用 SWA pool 大小 cap prefill 容量
    # if self.scheduler.tp_worker.is_hybrid_swa:
    # self.max_total_num_tokens = min(
    # self.max_total_num_tokens,
    # self.scheduler.tp_worker.model_runner.swa_max_total_num_tokens,
    # )
```

评论区精华

Review 中 reviewer ispobock 提出了两个核心问题：

- 是否将此修复推广到其他 SWA 模型？ispobock 质疑所有 SWA 模型是否都只传输 sliding window states，并建议引入一个 flag 区分。作者未直接回应，该问题保持开放。
- 对齐 model_runner 的 max_token_pool_size：ispobock 建议直接使用 max_token_pool_size 以与 decode 侧一致，该建议被作者采纳。 此外 yhyang201 给出了 LGTM 认可。
- 是否将此修复扩展到其他 SWA 模型 (design)：作者未明确回应，PR 仅修改了 prefill.py 且未改动基类和 flag，该问题留待后续验证。
- 对齐 model_runner 的 max_token_pool_size (correctness)：作者采纳建议，直接使用 model_runner.max_token_pool_size。

风险与影响

- 风险：本变更仅涉及 PD prefill 场景，主要风险在于其他混合 SWA 模型（如 MiMo、Gemma4）可能仍需要此 cap。如果这些模型在 PD prefill 时也依赖 swa_max_total_num_tokens 作为上限，则可能因容量过高导致 OOM。但由于非 PD 场景已使用类似逻辑，此风险较低。建议在测试中覆盖其他 SWA 模型的 PD 场景。
- 影响：对 DeepSeek V4 用户：修复了 PD prefill 无法处理长上下文的阻断 bug，吞吐提升。对系统：减小了监控指标中 prefill 拒绝率的误报。对其他 SWA 模型：潜在回归风险需进一步验证。总体影响范围局限于 PD 分离部署模式。
- 风险标记：其他 SWA 模型潜在回归，缺少测试覆盖

关联脉络

- PR #24857 Related fix for decode SWA: 此 PR 修复了解码阶段的类似问题，本 PR 是其在 prefill 阶段的对应修复。