PR #26970 完整报告

sgl-project/sglang

[perf] Replicate embed_tokens to drop the post-embed all-reduce

合并时间：2026-06-03 07:48

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26970

执行摘要

- 一句话：复制 embed_tokens 消除 TP all-reduce，提升解码性能 1-2%
- 推荐动作：此 PR 是典型的空间换时间设计，代码简洁且注释充分。建议对 DeepSeek 模型优化感兴趣的工程师仔细阅读 get_embedding_tp_kwargs 的实现和文档串，理解其与 DP attention 的交互。审阅人的讨论也值得关注，在实际部署时应根据 TP 规模和模型参数评估收益。

功能与动机

消除 EmbedLookup 后的 all-reduce 通信瓶颈是提升解码吞吐的关键。在 TP 规模较大时，all-reduce 开销显著；复制 embedding 表格用空间换时间，能获得稳定 1-2% 性能提升。该优化主要针对 DeepSeek-V2 系列及其关联的 draft 模型（EAGLE3/NextN），因为它们共享 embed_tokens 权重，必须保持一致的布局。

实现拆解

1. 新增环境变量：在 python/sglang/srt/environ.py 的 Envs 类中添加 SGLANG_ENABLE_EMBED_REPLICATION（默认 False），作为嵌入复制的全局开关。
2. 提取 TP 布局决策逻辑：在 python/sglang/srt/layers/vocab_parallel_embedding.py 中新增 get_embedding_tp_kwargs() 函数。当环境变量为 True 时返回 {"enable_tp": False}（全复制）；否则返回 {"enable_tp": True, "use_attn_tp_group": is_dp_attention_enabled()}，保持原有 TP 分片逻辑并适配 DP attention（DP attention 下规约只在 attention-TP 组内，而不是完整 TP 组）。
3. 改造模型构造入口：在 deepseek_v2.py、deepseek_nextn.py 和 kimi_k25_eagle3.py 中，将 embed_tokens = VocabParallelEmbedding(..., use_attn_tp_group=is_dp_attention_enabled()) 统一替换为 embed_tokens = VocabParallelEmbedding(..., **get_embedding_tp_kwargs())。同时移除不再需要的 is_dp_attention_enabled 导入，精简代码。

关键文件：

- python/sglang/srt/layers/vocab_parallel_embedding.py（模块 嵌入层；类别 source；类型 core-logic；符号 get_embedding_tp_kwargs）：核心变更：新增 get_embedding_tp_kwargs 函数，统一嵌入 TP 布局逻辑，支持通过环境变量切换复制模式

- python/sglang/srt/environ.py（模块 环境变量；类别 source；类型 core-logic；符号 SGLANG_ENABLE_EMBED_REPLICATION）：新增 SGLANG_ENABLE_EMBED_REPLICATION 环境变量，作为嵌入复制的全局开关
- python/sglang/srt/models/deepseek_v2.py（模块 模型实现；类别 source；类型 data-contract）：主模型文件，修改 embed_tokens 构造调用以使用 get_embedding_tp_kwargs，移除旧的 is_dp_attention_enabled 导入
- python/sglang/srt/models/deepseek_nextn.py（模块 模型实现；类别 source；类型 data-contract）：NextN draft 模型，同样的构造调用替换，确保与 target 布局一致
- python/sglang/srt/models/kimi_k25_eagle3.py（模块 模型实现；类别 source；类型 data-contract）：Kimi K2.5 EAGLE3 draft 模型，新增 get_embedding_tp_kwargs 导入和调用

关键符号：get_embedding_tp_kwargs

关键源码片段

python/sglang/srt/layers/vocab_parallel_embedding.py

核心变更：新增 get_embedding_tp_kwargs 函数，统一嵌入 TP 布局逻辑，支持通过环境变量切换复制模式

```
from sglang.srt.environ import envs

def get_embedding_tp_kwargs() -> dict:
    """Vocab-parallel layout kwargs for the *input embedding* of models that
    support embedding replication (the DeepSeek-V2 target family: DeepSeek
    V3.1 / Kimi K2.5, plus their EAGLE3 / NextN drafts).

    EAGLE / NextN share the target's ``embed_tokens.weight`` tensor with the
    draft (``set_embed`` / ``set_embed_and_head``), so the target and every
    draft that shares it MUST use the same vocab-parallel layout -- otherwise
    the draft's masking/index math runs against a tensor with a different
    layout and accept_len silently drops. Route all of them through this one
    helper so they can never drift.
    """
    if envs.SGLANG_ENABLE_EMBED_REPLICATION.get():
        # 复制模式：每个 rank 持有完整 embedding，跳过后续的 all-reduce
        return {"enable_tp": False}
    # 默认 TP 分片模式：按 vocab 维度切分；
    # 若启用 DP attention，则使用 attention-TP group 规约（而非完整 TP group）
    return {"enable_tp": True, "use_attn_tp_group": is_dp_attention_enabled()}
```

python/sglang/srt/environ.py

新增 SGLANG_ENABLE_EMBED_REPLICATION 环境变量，作为嵌入复制的全局开关

```
# Replicate the input embedding across TP ranks instead of sharding it
# along the vocab dimension (saves an all-reduce/all-gather in the embed
# lookup at the cost of replicated embedding weights). Drives both the
# target and every draft that shares its embedding (see
```

```
# get_embedding_tp_kwargs); they must stay in lock-step. Currently only
# applies to the Deepseek-V2 family (Deepseek V3.1, Kimi K2.5) + drafts.
SGLANG_ENABLE_EMBED_REPLICATION = EnvBool(False)
```

评论区精华

审阅人 kpham-sgl 提出是否应暴露更多可调节参数，因为 TP=4 时复制 embedding 浪费内存，可能减少 KV 缓存可用空间，反而损害性能。作者未在讨论中直接回应，但最终实现保持了默认关闭的环境变量形式，提供基本的开关控制。审阅人随后批准了该 PR。

- 控制参数暴露 (design): 作者保持了默认关闭的环境变量形式，提供了基本的开关控制，审阅人最终批准了 PR。

风险与影响

- 风险:
  1. 内存开销风险：每个 TP rank 存储完整 embedding，对于 DeepSeek V3（vocab 128k，hidden 7168），每个 rank 额外约 1.75GB（BF16），可能严重挤压 KV 缓存空间，尤其在低 TP 配置下，性能可能不升反降。
  2. Draft 一致性风险：EAGLE3/NextN draft 共享 target 的 embed_tokens，必须使用相同的 TP 布局；get_embedding_tp_kwargs 确保一致性，但未来新增 draft 模型时必须同步使用该函数。
  3. 默认关闭降低影响：用户需手动启用，生产环境可通过实验确定适用场景。 - 影响：对用户：需要设置环境变量来启用特性；对系统：减少通信延迟但增加内存占用；对团队：改动较小，易于维护，但需注意后续模型集成时的一致性。 - 风险标记：内存开销增加，低 TP 负收益风险，draft 布局同步要求

关联脉络

- 暂无明显关联 PR