PR #33537 完整报告

sgl-project/sglang

Multiple flexibility fixes for DP attention

合并时间：2026-08-05 06:40

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33537

执行摘要

- 一句话：DP 注意力兼容自适应推测解码，修复 MLP 同步与图大小
- 推荐动作：值得精读。PR 展示了如何把内联初始化逻辑提取为可复用 API 的同时保持行为等价，以及如何用单个 helper 统一两个可能产生分歧的图尺寸计算点；对编写自定义 speculative 算法或扩展 DP attention 的工程师有参考价值。

功能与动机

PR body 指出：一些自定义自适应推测解码 worker 在 ForwardBatch.init_new 之外构造 ForwardBatch，而 DP attention 的 MLP 同步元数据只能通过 init_new 内的内联块填充，缺乏可复用入口；这些 worker 还可能产出 hidden_states 尚未物化的 draft 输入，导致无条件的 pad/unpad 崩溃。另外 DP CUDA 图的 batch size 由同步 token 数除以捕获请求宽度获得，但仅对显式枚举的 spec 算法生效，外部注册算法或可变 per-request draft 宽度会得到错误尺寸，而 can_run_graph 却直接使用原始每 rank 请求计数，两个站点可能出现分歧。

实现拆解

1. 在 python/sglang/srt/model_executor/forward_batch_info.py 中将 init_new 内联的 "For MLP sync" 块提取为新的 ForwardBatch.init_mlp_sync_metadata(batch, device) 方法，封装 spec 场景下 spec_scale_global_num_tokens 缩放、CPU/GPU tensor 构建以及 can_run_dp_cuda_graph 赋值；init_new 改为调用该方法，原始路径行为保持不变。
2. 在 forward_batch_info.py 中为 spec_info.hidden_states 的 pad 增加 is not None 守卫（_pad_inputs_to_size），并在 post_forward_mlp_sync_batch 的 target-verify 分支中对 hidden_states 的 unpad 增加同样守卫；同时将 unpad 宽度从 draft_token_num 改为 num_tokens_per_req，使可变 per-request draft 宽度下切片长度正确。
3. 在 python/sglang/srt/model_executor/runner/decode_cuda_graph_runner.py 中新增静态方法 DecodeCudaGraphRunner._max_dp_batch_size(forward_batch)，直接取 original_global_num_tokens_cpu 的最大值作为 DP 图 batch size，缺失时 raise RuntimeError；can_run_graph 与 load_batch 的 replay 路径统一切换到该方法，删除原有按 eagle/standalone/dflash 枚举的宽度除法分支。
4. 测试配套：在 test/registered/unit/model_executor/test_mlp_sync_pad_unpad.py 中新增 3 个用例，分别覆盖 speculative 宽度缩放、无 hidden_states 的 draft pad、以及 _max_dp_batch_size 正常与异常路径；测试以 CPU CI 方式注册（est_time=5, suite=base-a-test-cpu）。

5. 未在本地运行多 GPU DP-attention 端到端与 CUDA 图捕获测试，作者依赖 CI 验证。PR Extra 测试通过 /rerun-test 重跑 4 个相关测试（dp_attention_bcg_kl、mimo_v2_flash、disaggregation_hybrid_attention、deepseek_v4_flash_fp4_b200_cp），全部通过。

关键文件：

- python/sglang/srt/model_executor/forward_batch_info.py（模块 前向批次；类别 source；类型 data-contract；符号 init_mlp_sync_metadata）：核心数据契约变更：提取 init_mlp_sync_metadata 方法供自定义 worker 复用，并对 hidden_states pad/unpad 增加守卫，修正 target-verify unpad 宽度。
- python/sglang/srt/model_executor/runner/decode_cuda_graph_runner.py（模块 解码图；类别 source；类型 data-contract；符号 _max_dp_batch_size）：统一 DP CUDA 图 batch size 计算：新增 _max_dp_batch_size，同时用于 can_run_graph 与 replay 路径，移除算法枚举和宽度除法。
- test/registered/unit/model_executor/test_mlp_sync_pad_unpad.py（模块 MLP 同步；类别 test；类型 test-coverage；符号 test_init_mlp_sync_metadata_scales_speculative_request_width, test_draft_input_without_hidden_states_can_be_padded, test_dp_cuda_graph_batch_size_uses_raw_request_counts）：新增 3 个 CPU 单测覆盖 PR 修复的三个行为点，确保回归防护。

关键符号：ForwardBatch.init_mlp_sync_metadata, DecodeCudaGraphRunner._max_dp_batch_size

关键源码片段

python/sglang/srt/model_executor/forward_batch_info.py

核心数据契约变更：提取 init_mlp_sync_metadata 方法供自定义 worker 复用，并对 hidden_states pad/unpad 增加守卫，修正 target-verify unpad 宽度。

```
def init_mlp_sync_metadata(
    self, batch: ScheduleBatch, device: Union[str, torch.device]
) -> None:
    """Populate per-rank token counts for DP-attention MLP synchronization."""
    # batch 未携带全局 token 计数时跳过（例如部分自定义 worker 路径）
    if batch.global_num_tokens is None:
        return

    assert batch.global_num_tokens_for_logprob is not None

    # 推测解码场景下按每请求宽度缩放全局 token 数，使所有 DP rank
    # 看到一致的计数；延迟导入 spec_info 避免循环依赖
    if self.spec_info is not None:
        from sglang.srt.speculative.spec_info import spec_scale_global_num_tokens

        global_num_tokens, global_num_tokens_for_logprob = (
            spec_scale_global_num_tokens(
                self.spec_info,
                batch.global_num_tokens,
```

```
                batch.global_num_tokens_for_logprob,
            )
        )
    else:
        global_num_tokens = batch.global_num_tokens
        global_num_tokens_for_logprob = batch.global_num_tokens_for_logprob

    # 保存原始计数与缩放后的计数：GPU 版本用于 MLP 同步通信，
    # CPU 版本用于 DP CUDA 图的尺寸决策
    self.original_global_num_tokens_cpu = batch.global_num_tokens
    self.global_num_tokens_cpu = global_num_tokens
    self.global_num_tokens_gpu = torch.tensor(
        global_num_tokens, dtype=torch.int64
    ).to(device, non_blocking=True)

    self.global_num_tokens_for_logprob_cpu = global_num_tokens_for_logprob
    self.global_num_tokens_for_logprob_gpu = torch.tensor(
        global_num_tokens_for_logprob, dtype=torch.int64
    ).to(device, non_blocking=True)

    # 记录该 batch 是否满足 DP CUDA 图执行条件，供 runner 决策使用
    self.can_run_dp_cuda_graph = batch.can_run_dp_cuda_graph
```

评论区精华

仅作者自审：merrymercy 在 review 中直接 approve。Issue 评论中有 bot 提示 Gemini Code Assist 已停止审查，之后作者通过 /rerun-test 命令重跑 DP attention 相关测试（test_dp_attention_bcg_kl.py、test_mimo_v2_flash.py、test_disaggregation_hybrid_attention.py、test_deepseek_v4_flash_fp4_b200_cp.py），四个测试在 2-gpu-h100、8-gpu-h200、4-gpu-b200 上全部通过。

- 作者自审 approve (other): 合并通过。

风险与影响

- 风险：核心风险集中在 ForwardBatch 初始化路径：init_new 中提取方法后行为应完全等价，但任何属性名或逻辑遗漏都可能导致 DP attention 下 MLP 同步静默出错；hidden_states 增加 is not None 守卫虽避免崩溃，但可能掩盖某些路径未物化 hidden_states 的真实语义；_max_dp_batch_size 在计数缺失时 raise RuntimeError，若现有调用方未填充 original_global_num_tokens_cpu（例如非 DP 图路径），可能导致新的启动失败；target-verify unpad 改用 num_tokens_per_req 后，若某些 spec 实现该字段未正确填充，会改变切片长度。测试仅覆盖 CPU 单测，多 GPU DP attention 端到端与 CUDA 图捕获未在本地验证，依赖 CI。
- 影响：影响使用 DP attention 并启用自适应推测解码的场景，消除了自定义 worker 构建 ForwardBatch 时 MLP 同步元数据不可用和 draft 输入 pad/unpad 崩溃的问题；统一了 can_run_graph 与 replay 的图尺寸计算，使任意 spec 算法（包括外部注册、可变宽度）都能得到正确 DP CUDA 图大小；对默认路径无行为变化。团队需注意 3 个 CPU 单测已注

册到 base-a-test-cpu 套件，CI 会持续覆盖。

- 风险标记：核心路径变更，多 GPU 验证依赖 CI，新增异常路径

关联脉络

- PR #33306 Avoid TRTLLM prefill output copy: 同样修改了 python/sglang/srt/model_executor/forward_batch_info.py，属于 forward batch 数据结构与执行路径的持续演进。