PR #23269 完整报告

sgl-project/sglang

Support batch size > 1 when enable CP

合并时间：2026-05-28 05:11

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23269

执行摘要

- 一句话：上下文并行支持 batch size > 1
- 推荐动作：值得精读，尤其 ContextParallelMetadata 从单序列到多序列的设计演进，以及 padding 策略的权衡。讨论中的 CPU 开销担忧和未来 Triton 替代方向值得关注。架构师应关注 DSA 路径的遗留 TODO。

功能与动机

PR body 中明确目标："Enable batch size > 1 with context parallel." 此前 CP 的 zigzag 切分和元数据仅支持单请求，严重限制了吞吐。Review 中 kpham-sgl 指出当前 MHA CP 实现中存在两处已知 bug，并强调需要 attn_cp_size == 4 且 bs > 1 的测试覆盖。

实现拆解

1. ContextParallelMetadata 数据结构重构（cp_utils.py）：将 kv_len_prev、kv_len_next 等标量替换为形状 [bs] 的 CUDA tensor（如 kv_len_prev_tensor），新增 cu_seqlens_q_prev_tensor 辅助 FlashAttention，新增 bs 字段记录批大小，total_seq_lens 改为 int。
2. CP 切分判断升级（cp_utils.py 中 can_cp_split）：移除 batch_size == 1 限制，改为逐个检查 extend_seq_lens_cpu 是否满足 len >= cp_size * 2，不满足时返回 False 而非抛异常。
3. Padding 对齐粒度调整（forward_batch_info.py）：对齐基数从 attn_cp_size 改为 attn_cp_size * 2，确保 zigzag 切分负载均衡。
4. 调度策略松绑（schedule_policy.py）：移除 self.prefill_context_parallel_enabled 对 add_one_req 中 can_run_list 长度的限制，允许多请求进入。
5. 注意力后端适配：在 cp_attn_forward_extend 中按 total_q_prev_tokens 切分 q，传入 cu_seqlens_q_prev/next_tensor；在 dsa_indexer.py 中将标量字段替换为 kv_len_prev_list[0] 等列表元素（仍假设 batch=1，预留 TODO）。
6. 调用参数统一：所有模型入口（deepseek_v2.py、deepseek_nextn.py、deepseek_v4.py、qwen3_moe.py 等）将 prepare_context_parallel_metadata 的 extend_lens= 更名为 extend_seqs_len=。
7. 测试配套：删除 test_qwen3_30b.py（原 CP 精度测试），修改 test_mla_cp_fa3_parity.py 适配新字段。

关键文件：

- python/sglang/srt/layers/utils/cp_utils.py（模块 CP 核心；类别 source；类型 core-logic；符号 cp_all_gather_reorganized_into_tensor）：核心变更文件，ContextParallelMetadata 数据类重构支持多 batch，can_cp_split 逻辑调整，cp_attn_forward_extend 适配多序列。
- python/sglang/srt/layers/attention/dsa/dsa_indexer.py（模块 DSA 索引器；类别 source；类型 core-logic）：DSA 注意力路径适配，将标量字段改为从列表中取元素，仍假设 batch=1。
- python/sglang/srt/models/deepseek_v2.py（模块 DeepSeek 模型；类别 source；类型 data-contract）：DeepSeek V2 模型入口，调整 prepare_context_parallel_metadata 参数名为 extend_seqs_len。
- python/sglang/srt/models/deepseek_nextn.py（模块 NextN 模型；类别 source；类型 data-contract）：DeepSeek NextN 模型入口，同样调整参数名。
- python/sglang/srt/managers/schedule_policy.py（模块 调度器；类别 source；类型 core-logic）：调度策略移除 prefill_context_parallel_enabled 限制，允许批处理多请求。
- python/sglang/srt/model_executor/forward_batch_info.py（模块 批处理信息；类别 source；类型 data-contract）：调整 padding 对齐粒度从 attn_cp_size 改为 attn_cp_size * 2，保证 zigzag 负载均衡。
- test/registered/cp/test_qwen3_30b.py（模块 CP 测试；类别 test；类型 deletion；符号 TestQwen330B, setUpClass, tearDownClass, test_gsm8k）：删除原 CP 精度测试文件，可能以其他测试替代。
- test/registered/kernels/test_mla_cp_fa3_parity.py（模块 MLA CP 测试；类别 test；类型 test-coverage）：适配新的 ContextParallelMetadata 字段，构建正确的测试数据。

关键符号：can_cp_split, prepare_context_parallel_metadata, cp_attn_forward_extend, cp_all_gather_reorganized_into_tensor, add_one_req, prepare_mlp_sync_batch

关键源码片段

python/sglang/srt/layers/utils/cp_utils.py

核心变更文件，ContextParallelMetadata 数据类重构支持多 batch，can_cp_split 逻辑调整，cp_attn_forward_extend 适配多序列。

```
# ContextParallelMetadata 支持 batch size > 1
@dataclass
class ContextParallelMetadata:
    # Layout lists have length bs * cp_segment_num (= bs * 2 * cp_size).
    split_list: List[int] = None
    zigzag_index: List[int] = None
    cp_reverse_index: List[int] = None
    reverse_split_len: List[int] = None

    # Per-rank-aggregate lists have length cp_size.
    per_rank_actual_token: List[int] = None
```

```
    max_rank_len: List[int] = None

    # Per-sequence FlashAttention tensors (shape [bs] or [bs+1]).
    kv_len_prev_tensor: torch.Tensor = None # [bs] int32 CUDA
    kv_len_next_tensor: torch.Tensor = None # [bs] int32 CUDA
    actual_seq_q_prev_tensor: torch.Tensor = None # [bs] int32 CUDA
    actual_seq_q_next_tensor: torch.Tensor = None # [bs] int32 CUDA
    cu_seqlens_q_prev_tensor: torch.Tensor = None # [bs+1] int32 CUDA
    cu_seqlens_q_next_tensor: torch.Tensor = None # [bs+1] int32 CUDA

    # Per-seq CPU lists (useful for NSA indexer and diagnostics).
    kv_len_prev_list: List[int] = None
    kv_len_next_list: List[int] = None
    actual_seq_q_prev_list: List[int] = None
    actual_seq_q_next_list: List[int] = None

    # Aggregate sum of extend_seq_lens across the batch.
    total_seq_lens: int = 0
    bs: int = 1


def can_cp_split(seq_len: int, cp_size: int, forward_batch):
    # 基础条件：CP 开启、size>1、纯 extend 模式
    from sglang.srt.model_executor.forward_batch_info import ForwardMode
    cur_cp_seq_len = seq_len // (cp_size * 2)
    if not (
        cur_cp_seq_len != 0
        and cp_size > 1
        and forward_batch.forward_mode.is_context_parallel_extend()
        and forward_batch.forward_mode != ForwardMode.MIXED
        and is_prefill_context_parallel_enabled()
    ):
        return False

    # 逐请求检查 extend length 是否足够 zigzag 切分
    extend_lens = getattr(forward_batch, "extend_seq_lens_cpu", None)
    if extend_lens is None:
        return True

    cp_min = cp_size * 2
    for L in extend_lens:
        if L < cp_min:
            # 不满足切分条件的请求 gracefully 回退到非 CP 模式
            return False
    return True
```

python/sglang/srt/layers/attention/dsa/dsa_indexer.py

DSA 注意力路径适配，将标量字段改为从列表中取元素，仍假设 batch=1。

```
# head 版本：从 list 中取第一个元素，仍假设 batch=1 的 DSA 路径
if (
    forward_batch.attn_cp_metadata is not None
    and is_dsa_prefill_cp_in_seq_split()
):
    kv_len_prev = forward_batch.attn_cp_metadata.kv_len_prev_list[0]
    kv_len_next = forward_batch.attn_cp_metadata.kv_len_next_list[0]
    actual_seq_q_prev = forward_batch.attn_cp_metadata.actual_seq_q_prev_list[0]
    actual_seq_q_next = forward_batch.attn_cp_metadata.actual_seq_q_next_list[0]
    # TODO: 支持 multi-batch 后需改为对应 batch 索引
```

评论区精华

- can_cp_split 异常 vs graceful：kpham-sgl 担心抛异常会导致生产崩溃，建议回退；Shunkangz 最初认为应显式暴露，但最终采用逐请求检查并 return False 的 graceful 方案。
- padding 到 2×cp_size：kpham-sgl 提议将 padding 从 cp_size 改为 2×cp_size 以支持 zigzag 负载均衡，被采纳并实现在 forward_batch_info.py。
- cu_seqlens 移至 prepare_context_parallel_metadata：Shunkangz 解释是为了减少 kernel launch 前的 CPU 索引拷贝开销，促进 kernel 尽早发射。
- CPU 元数据开销担忧：Fridge003 建议未来用 Triton kernel 替代 CPU 循环，kpham-sgl 表示 +1，目前未实施但有 TODO。
- DSA indexer 仍假设 batch=1：kpham-sgl 指出 dsa_indexer 中 *_list[0] 只适用于单请求，Shunkangz 确认保持 TODO 以便后续多 batch 支持。
  - can_cp_split 抛异常 vs graceful fallback (design): 采用逐请求检查 extend_seq_len，不满足时 return False，由调度器回退到非 CP 模式。
  - padding 对齐粒度从 cp_size 改为 cp_size*2 (design): 在 forward_batch_info.py 中实施对齐到 attn_cp_size * 2。
  - cu_seqlens 计算移至 prepare_context_parallel_metadata (performance): 接受此改动，将 cu_seqlens 计算提前到元数据准备阶段。
  - CPU 元数据计算开销担忧 (performance): 留下 TODO，后续可考虑 Triton 实现。
  - DSA indexer 仍假设 batch=1 (design): 保留 TODO，DSA 多 batch 支持推迟到后续 PR。

风险与影响

- 风险：
  - 核心路径回归：ContextParallelMetadata 字段大幅变更，所有 CP 相关的注意力、前向、DP padding 逻辑均受影响，需关注 DeepSeek V3/V4、Qwen3 等模型的精度和 crash。
  - CPU 开销增加：prepare_context_parallel_metadata 中对每序列的逐循环计算可能成为 prefill 瓶颈，尤其当 bs 较大时。
  - DSA 路径兼容性：dsa_indexer.py 仍使用 kv_len_prev_list[0]，在真正多 batch 前会限制 DSA CP 的 batch 能力。

    - 测试覆盖缩减：删除了 test_qwen3_30b.py（原 CP 精度测试），若 test_mla_cp_fa3_parity.py 覆盖不足可能漏掉回归。
- 影响：
    - 用户影响：启用 CP 时可获得更高 prefill 吞吐，但需要每个请求的 extend length 足够长（≥2×cp_size），否则回退到非 CP 模式。
    - 系统影响：新增 attn_cp_size * 2 对齐计算，可能增加少量显存 / 计算开销；调度器不再强制 batch=1，可能改变并发行为。
    - 团队影响：后续开发（如 DSA 多 batch、Triton 元数据 kernel）需要在此数据结构基础上继续迭代。
    - 风险标记：核心路径变更，CPU 开销增加，DSA 路径兼容性，测试覆盖缩减

关联脉络

- PR #23292 prepare_context_parallel_metadata 改动（未最终确定）：kpham-sgl 在讨论中引用此 PR 作为参考，涉及 metadata 计算调整。
- PR #25821 nsa 替换为 dsa 的冲突引入：Fridge003 要求解决与此 PR 的冲突，将 'nsa' 替换为 'dsa'。
- PR #26380 [core] WAR barrier for overlap schedule buffer writes: 与本 PR 同属 CP 调度路径优化，后续有交互。