PR #7297 完整报告

verl-project/verl

[megatron] fix: make DeepSeek-V4 context parallelism actually runnable

合并时间：2026-08-10 23:48

原文链接：http://prhub.com.cn/verl-project/verl/pull/7297

执行摘要

- 一句话：补完 DeepSeek-V4 CP 布局贯通，CP 训练真正可跑
- 推荐动作：值得精读。核心设计思想是“布局描述必须与行的产生方式保持一致”——cp_layout 贯穿 pack、attention kernel 与 router-replay 的重放路径，避免“能跑但数值错位”的隐患；inspect.signature 兼容探测对跨版本 Megatron-core 也是可借鉴的模式。同时建议阅读评论区关于 cudnn-frontend 版本导致 NaN 的排查，这为 DeepSeek-V4 用户提供了直接可用的踩坑指南。

功能与动机

PR body 明确说明：#7221 添加了 DeepSeek-V4 所需的 contiguous CP 行布局并接入 MegatronEngine，但开启 context_parallel_size > 1 仍然失败，因为选定的布局从未传给 attention kernels 和 router-replay 张量，示例脚本也未设置 Megatron-Core 强制的 transformer-config 标志。缺失三处分别导致模型构建时的 ValueError（DSv4 Hybrid with CP requires cp_partition_mode='contiguous'）、每次注意力前向的 ValueError（DSv4 THD CP requires a contiguous CP partition.）和 AssertionError（requires a sequence_packing_scheduler for THD inputs）。

实现拆解

1. 扩展 PackedSeqParams 数据契约（verl/models/mcore/util.py）：preprocess_thd_engine 在构造 PackedSeqParams 时把 cp_layout 写入 cp_partition_mode，让注意力 kernel 知道行是如何被切分的；为避免破坏旧版 Megatron-core（该字段不存在），用 inspect.signature(PackedSeqParams.__init__) 做运行期探测，不支持且 cp_layout != "zigzag"、cp_size > 1 时显式抛错，保证老版本不会静默走错布局。
2. 统一 router-replay 的 CP 布局（verl/utils/megatron/router_replay_utils.py）：新增 _context_parallel_layout() 辅助函数，按 experimental_attention_variant == "dsv4_hybrid" 判定返回 contiguous，否则返回 zigzag；merge_router_topk_indices 与 set_router_replay_data 两个调用点均把 cp_layout 传入 preprocess_thd_engine / postprocess_thd_engine（含 replay_mask 的预处理），避免重放路由因布局不一致而错挂 token 或形状不匹配。
3. 补齐示例脚本 CP 参数（examples/grpo_trainer/run_deepseek_v4_flash_megatron.sh）：新增 CP_ARGS 数组，仅当 ACTOR_CP > 1 时追加 cp_partition_mode=contiguous、

sequence_packing_scheduler=dp_balanced、max_seqlen_per_dp_cp_rank 三个参数，每项都注释了它要规避的 Megatron-Core 报错；ACTOR_CP=1 时数组展开为空，默认路径与之前完全一致。

4. 测试与验证配套：未新增测试文件，沿用 #7221 的 CPU 单测（tests/utils/test_megatron_bshd_preprocess.py、tests/workers/test_megatron_value_head_cp_layout_on_cpu.py，15 passed）；作者在 40 卡 GB200（PP=5、EP=8、TP=1）上完成 CP=1 与 CP=2 的 E2E 对照，给出 rollout_probs 差异与 Pearson 相关性、ppo_kl 等指标，并诚实声明未覆盖 CP>2 与梯度数值对比。

关键文件：

- verl/utils/megatron/router_replay_utils.py（模块 路由重放；类别 source；类型 core-logic；符号 _context_parallel_layout）：核心修复点：新增 _context_parallel_layout 统一 CP 布局判定，并在 merge_router_topk_indices 与 set_router_replay_data 两个入口把 cp_layout 传入 THD 预处理 / 后处理，保证重放路由张量与模型 token 行切分完全一致。
- verl/models/mcore/util.py（模块 预处理；类别 source；类型 data-contract；符号 preprocess_thd_engine）：THD 预处理数据契约变更：preprocess_thd_engine 把 cp_layout 写入 PackedSeqParams.cp_partition_mode，并用 inspect.signature 兼容新旧 Megatron-core，是整个修复的根基。
- examples/grpo_trainer/run_deepseek_v4_flash_megatron.sh（模块 示例脚本；类别 other；类型 configuration；符号 CP_ARGS）：入口层配置：新增 CP_ARGS 数组，ACTOR_CP > 1 时追加 Megatron-Core 强制的三个 transformer-config 参数，是用户侧真正跑通 CP 的最后一环。

关键符号：preprocess_thd_engine, _context_parallel_layout, merge_router_topk_indices, set_router_replay_data

关键源码片段

verl/utils/megatron/router_replay_utils.py

核心修复点：新增 _context_parallel_layout 统一 CP 布局判定，并在 merge_router_topk_indices 与 set_router_replay_data 两个入口把 cp_layout 传入 THD 预处理 / 后处理，保证重放路由张量与模型 token 行切分完全一致。

```
# 统一 CP 布局判定：返回 THD packing 使用的行布局，必须与 MegatronEngine 保持一致。
# Router-replay 张量必须与模型自身 token 行的切分方式完全一致，否则重放的路由会
# 挂到错误的 token 上（或直接形状不匹配，因为 zigzag 布局会让对齐长度翻倍）。
def _context_parallel_layout(tf_config) -> str:
    if getattr(tf_config, "experimental_attention_variant", None) == "dsv4_hybrid":
        return "contiguous"
    return "zigzag"


def merge_router_topk_indices(attention_mask, input_ids, mini_layer_topk_idx_list, tf_config, vp_
rank=None):
```

```
with torch.no_grad():
    # ... 收集各 router 的 recorded_topk_idx 并跨 sequence-parallel rank 聚合 ...

    cp_layout = _context_parallel_layout(tf_config)

    if input_ids.is_nested:
        # THD（nested tensor）路径：先按 cp_layout 切行打包，再按同一布局解包。
        # 若这里仍用默认 zigzag，模型行是 contiguous 而重放张量是 zigzag，
        # 路由索引就会错位到别的 token 上。
        batch_size = input_ids.shape[0]
        _, packed_seq_params, _ = preprocess_thd_engine(
            input_ids,
            pre_process=True,
            use_fp8_padding=use_fp8_padding,
            min_local_rows=min_local_rows,
            cp_layout=cp_layout,
        )
        layers_topk_idx = postprocess_thd_engine(
            layers_topk_idx,
            packed_seq_params,
            input_ids,
            batch_size,
            post_process=True,
            cp_layout=cp_layout,
        )
    else:
        # 非 nested 路径走 preprocess_packed_seqs / postprocess_packed_seqs，
        # 该路径本身不感知 CP 布局，行为与之前保持一致。
        batch_size, seq_len = attention_mask.shape[:2]
        _, packed_seq_params = preprocess_packed_seqs(
            input_ids, attention_mask, pre_process=True, use_fp8_padding=use_fp8_padding
        )
        layers_topk_idx = postprocess_packed_seqs(
            layers_topk_idx, packed_seq_params, attention_mask, batch_size, seq_len, post_
            process=True
        )
    mini_layer_topk_idx_list.append(layers_topk_idx.cpu())
```

verl/models/mcore/util.py

THD 预处理数据契约变更：preprocess_thd_engine 把 cp_layout 写入 PackedSeqParams.cp_partition_mode，并用 inspect.signature 兼容新旧 Megatron-core，是整个修复的根基。

```
# 兼容探测：老版本 Megatron-core 的 PackedSeqParams 没有 cp_partition_mode 字段，
# 只支持 zigzag 布局。用 inspect.signature 检查 __init__ 参数（而不是 dataclass fields），
# 这样接受 **kwargs 的 duck-typed 测试桩也能被识别为支持该字段。
_PACKED_SEQ_PARAMS_INIT_PARAMS = inspect.signature(PackedSeqParams.__init__).
parameters
_PACKED_SEQ_PARAMS_HAS_CP_PARTITION_MODE = "cp_partition_mode" in _PACKED_SEQ_
```

```
    PARAMS_INIT_PARAMS or any(
        p.kind is inspect.Parameter.VAR_KEYWORD for p in _PACKED_SEQ_PARAMS_INIT_PARAMS.
        values()
    )

    # ... 前面已按 cp_layout 把 token 行 shard 到各 CP rank ...

    if _PACKED_SEQ_PARAMS_HAS_CP_PARTITION_MODE:
        # 把实际使用的布局写进 PackedSeqParams，告诉注意力 kernel 行是怎么切的。
        # 若仍用默认值 "zigzag"，DSv4 注意力会在每次 forward 抛出
        # "DSv4 THD CP requires a contiguous CP partition."
        extra_packed_args["cp_partition_mode"] = cp_layout
    elif cp_layout != "zigzag" and cp_size > 1:
        # 老版本 Megatron-core 无法表达 contiguous 布局，直接报错而不是静默用错布局。
        raise ValueError(
            f"cp_layout='{cp_layout}' requires PackedSeqParams.cp_partition_mode, which this "
            "Megatron-core version does not provide. Upgrade Megatron-core or use the zigzag
            layout."
        )

    packed_seq_params = PackedSeqParams(
        qkv_format="thd",
        cu_seqlens_q=cu_seqlens_padded,
        max_seqlen_q=max_seqlen_in_batch,
        cu_seqlens_kv=cu_seqlens_padded,
        max_seqlen_kv=max_seqlen_in_batch,
        cu_seqlens_q_padded=cu_seqlens_padded,
        cu_seqlens_kv_padded=cu_seqlens_padded,
        **extra_packed_args,
    )
```

examples/grpo_trainer/run_deepseek_v4_flash_megatron.sh

入口层配置：新增 CP_ARGS 数组，ACTOR_CP > 1 时追加 Megatron-Core 强制的三个 transformer-config 参数，是用户侧真正跑通 CP 的最后一环。

```
    # 上下文并行需要三个额外的 transformer-config 设置，每个都由 Megatron-Core 强制检查，
    # 缺失时会在模型构建或第一次注意力前向直接中止。只在 CP > 1 时追加。
    #
    # cp_partition_mode=contiguous：DSv4 注意力要求每个 CP rank 持有 packed THD buffer 的
    # 一段连续区间；Megatron-Core 默认 zigzag，会报
    # "DSv4 Hybrid with CP requires cp_partition_mode='contiguous'."
    # sequence_packing_scheduler=dp_balanced：缺失时断言
    # "DSv4 Hybrid with CP requires a sequence_packing_scheduler for THD inputs."
    # 该参数需要 Transformer Engine >= 2.9。
    # max_seqlen_per_dp_cp_rank：按文档取 max sequence length / cp_size，
    # 驱动子样本如何分配到每个 DPxCP rank。
    CP_ARGS=()
    if [ "${ACTOR_CP}" -gt 1 ]; then
        CP_ARGS=(
```

```
        ++actor_rollout_ref.actor.megatron.override_transformer_config.cp_partition_mode=
        contiguous
        ++actor_rollout_ref.actor.megatron.override_transformer_config.sequence_packing_
        scheduler=dp_balanced
        ++actor_rollout_ref.actor.megatron.override_transformer_config.max_seqlen_per_dp_cp_
        rank=$(((MAX_PROMPT_LENGTH + MAX_RESPONSE_LENGTH) / ACTOR_CP))
    )
fi

# 追加到启动命令的 ACTOR 之后；ACTOR_CP=1 时数组为空，默认路径保持不变。
python3 -m verl.trainer.main_ppo \
    "${DATA[@]}" \
    "${MODEL[@]}" \
    "${ACTOR[@]}" \
    "${CP_ARGS[@]}" \
    "${ROLLOUT[@]}" \
    "${REWARD[@]}" \
    "${TRAINER[@]}" \
```

评论区精华

Review 状态出现一次反复：HollowMan6 先 APPROVED，随后因 CI 运行疑似挂起而 DISMISSED，最后在兼容性修复提交后再次 APPROVED 并合并，挂起的具体根因未在评论中进一步追踪。Issue 区关于 NaN 的讨论价值较高：HaochenYuan 报告在 GB200 上 dsa_kernel_fusion + selective/moe recompute 出现 NaN，HollowMan6 表示在 H100 上用 full recompute + fusion 未复现；HaochenYuan 最终定位为容器内 nvidia-cudnn-frontend 版本早于 cudnn-frontend#427，导致 cudnn.csa 缺失、Megatron-Core 静默回退 eager CSA compressor，与本文无关；sophiayyya 在 H100 上复现 kl NaN，按同样的依赖安装方式（apache-tvm-ffi==0.1.11 + nvidia-cudnn-frontend==1.27.0）后消失。

- CI 挂起导致 review 状态反复 (testing): 最终通过并合并，挂起的具体根因未在评论中明确追踪，可能与 CPU 单测环境或 Megatron-core 版本差异有关。
- DeepSeek-V4 训练 NaN 与 cudnn-frontend 版本 (question): 根因是镜像依赖版本而非本 PR；通过 pip install --no-deps apache-tvm-ffi==0.1.11 与 pip install --force-reinstall --no-deps nvidia-cudnn-frontend==1.27.0 解决。

风险与影响

- 风险：一是 verl/models/mcore/util.py 的兼容探测依赖 inspect.signature(PackedSeqParams.__init__)，若未来 Megatron-core 改变字段定义方式（如 slots 化或改名），探测逻辑可能误判，但当前 dataclass 的 __init__ 签名稳定，低风险。二是 verl/utils/megatron/router_replay_utils.py 的 _context_parallel_layout 与 MegatronEngine._get_context_parallel_layout 必须手动保持同步，docstring 已提示，但缺少自动断言，未来任一方向新增 CP 布局时可能静默错挂路由。三是验证范围有限：CP>2 未测、backward 的梯度数值未在 CP=1/CP=2 间对比（作者自述 grad_norm 为 0，backward 主要验证了不产生 NaN/inf），存在数值隐患被掩盖的可能。四是 review 中出现过一次 CI hang，虽最终通过，但根因未澄清。五是 preprocess_thd_engine 是

Megatron THD 公共路径，改动影响所有走该函数的使用方，不过默认 zigzag 路径行为不变。

- 影响：对用户：DeepSeek-V4 开启 context_parallel_size > 1 从直接崩溃变为可用，示例脚本自动追加所需参数，ACTOR_CP=1 路径字节级不变，无 API 变更。对系统：THD 预处理的数据契约扩展，其他模型默认 zigzag 布局不受影响，但任何新模型若引入新的 CP 布局都需同步 _context_parallel_layout 与 MegatronEngine._get_context_parallel_layout。对团队：提供了可复用的 CP 数值一致性验证范式（CP=1 对照 CP=2 的 rollout 对数概率相关性），并沉淀了 DSv4 训练 NaN 的容器依赖根因排查结论，降低了后续排障成本。
- 风险标记：核心路径变更（THD 数据契约），依赖 Megatron-core 版本能力，布局规则需双点手动同步，CP>2 与梯度数值未验证，review 中曾出现 CI hang

关联脉络

- PR #7221 Add contiguous CP layout for DeepSeek-V4: PR body 明确说明本 PR 是其 follow-up：#7221 引入 contiguous CP 行布局并接入 MegatronEngine，本 PR 补完布局向 attention kernels 与 router-replay 的传递，以及示例脚本的 transformer-config 参数。
- PR #6473 HollowMan6 提到的 DeepSeek-V4 相关实验 PR: 讨论中 HollowMan6 表示在 H100 上实验 dsa kernel fusion 时正基于该 PR，同属 DeepSeek-V4 训练链路；NaN 排查结论也服务于该链路。
- PR #7407 [megatron,veomni] feat: use torch.int16 for routed_experts: 同改 verl/utils/megatron/router_replay_utils.py 与 verl/models/mcore/util.py，将路由索引与 router-replay 机制进一步演进（int16 支持超 255 专家），与本 PR 同属 Megatron MoE 路由重放体系。