

PR #6520 完整报告

verl-project/verl

[ci] chore: npu ci use cann9.0.0

合并时间: 2026-06-01 10:10

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6520>

执行摘要

- 一句话: NPU CANN 升级至 9.0.0 并适配 transformers 5.3.0 MoE
- 推荐动作: 建议关注 CI 镜像切换后的首次运行日志和测试通过率; 模型转换脚本的专家处理分支可提取为公用工具函数, 提升可维护性; 日志目录问题虽未影响合并, 建议后续修复。

功能与动机

PR body 明确说明 'update npu's cann to 9.0.0 version', 主要目的是升级 Ascend NPU 的基础软件栈。同时, 从提交历史可见需要适配 transformers 5.3.0 引入的 API 破坏性变更 (Qwen3MoE 的 experts 格式变化), 因此一并解决兼容性问题。

实现拆解

1. CI 镜像版本更新: 修改 .github/workflows/ 下所有 Ascend 相关工作流 (共 12 个文件), 将 Docker 镜像标签从 verl-8.5.0-* 改为 verl-9.0.0-*, 并同步调整了 reward_model_vllm_ascend.yml 中 set -e 与 bash 参数。
2. 模型转换脚本适配 transformers 5.3.0: 在 scripts/converter_hf_to_mcore.py 的 convert_checkpoint_from_transformers_to_megatron 函数中, 针对 MoE 层重构了专家权重搬运逻辑。旧代码直接枚举 hf_layer.mlp.experts (ModuleList), 新代码先判断 hf_experts 是否拥有 gate_up_proj 属性 (transformers 5.x 的 Qwen3MoE 使用融合的三维张量 [num_experts, 2*intermediate_size, hidden]), 分别处理新旧格式, 并在不支持时抛出 TypeError。
3. 测试脚本补充: 在 tests/special_npu/nightly_ci_ascend/run_grpo_qwen3_8b_mindspeed_llm_npu.sh 中增加 actor_rollout_ref.rollout.calculate_log_probs=True 配置, 用于 log probability 计算; 同时修改了 run_ppo_qwen3-8b_fsdp_npu.sh 的输出日志路径 (引入潜在在目录不存在风险, review 已指出)。
4. Unit test 适配: 在 tests/utils/test_megatron_bshd_preprocess.py 中增加 import verl.utils.device 和 monkeypatch.setattr(device_module, "is_npu_available", False), 确保在非 NPU 环境下 mock is_npu_available 为 False, 以绕过新版本中可能的环境检测。
5. 文档和安装脚本更新: docs/ascend_tutorial/get_start/install_guidance.rst 更新了安装指引中的路径说明, scripts/install_*_npu.sh 也同步更新了镜像版本和参数。

关键文件:

- scripts/converter_hf_to_mcore.py (模块 模型转换; 类别 source; 类型 core-logic; 符号 convert_checkpoint_from_transformers_to_megatron) : 核心变更: 重构 MoE 专家权重拷贝逻辑, 支持 transformers 5.3.0 的 Qwen3MoE 格式, 同时向下兼容旧版本。
- tests/utils/test_megatron_bshd_preprocess.py (模块 回归测试; 类别 test; 类型 test-coverage; 符号 _load_mcore_util_with_stubbed_megatron) : 单元测试增加 NPU 环境检测 mock, 避免测试因 is_npu_available 而跳过关键路径。
- .github/workflows/model_ascend.yml (模块 CI 流水线; 类别 infra; 类型 infrastructure) : Model CI 工作流, Docker 镜像版本升级为 CANN 9.0.0, 用于模型验证测试。
- tests/special_npu/nightly_ci_ascend/run_grpo_qwen3_8b_mindspeedllm_npu.sh (模块 CI 测试; 类别 test; 类型 test-coverage) : NPU 夜间 CI 测试脚本, 新增 calculate_log_probs 配置项以启用 log probability 计算。
- .github/workflows/e2e_ascend.yml (模块 CI 流水线; 类别 infra; 类型 infrastructure) : E2E 测试 CI 工作流, Docker 镜像版本升级为 CANN 9.0.0, 并调整了部分参数。
- .github/workflows/nightly_ascend.yml (模块 CI 流水线; 类别 infra; 类型 infrastructure) : 夜间 CI 工作流, Docker 镜像版本升级为 CANN 9.0.0, 涉及多个 job 的容器镜像更新。

关键符号: convert_checkpoint_from_transformers_to_megatron, safe_copy, _load_mcore_util_with_stubbed_megatron, _check_topk_preprocess

关键源码片段

scripts/converter_hf_to_mcore.py

核心变更: 重构 MoE 专家权重拷贝逻辑, 支持 transformers 5.3.0 的 Qwen3MoE 格式, 同时向下兼容旧版本。

```
# 适配 Transformers 5.x Qwen3MoE: gate_up_proj + down_proj 为三维张量
# 如果 experts 属性包含 gate_up_proj, 说明是融合格式 (新版本)
if hasattr(hf_experts, "gate_up_proj"):
    for idx in range(num_experts):
        if idx < expert_idx_start or idx >= expert_idx_end:
            continue
        local_expert_idx = idx - expert_idx_start

        # gate_up_proj: [num_experts, 2 * intermediate_size, hidden_size]
        gate_up = hf_experts.gate_up_proj[idx]
        intermediate_size = gate_up.shape[0] // 2
        gate_w = gate_up[:intermediate_size]
        up_w = gate_up[intermediate_size:]

        fc1_weight = torch.cat([gate_w, up_w], dim=0)
        # down_proj: [num_experts, hidden_size, intermediate_size]
        down_w = hf_experts.down_proj[idx]

        numel += safe_copy(fc1_weight, layer.mlp.experts.linear_fc1._parameters[f"weight{local_expert_idx}"])
```

```

numel += safe_copy(down_w, layer.mlp.experts.linear_fc2._parameters[f"weight{local_
expert_idx}"])

# 兼容旧的 transformers / 其他 MoE (ModuleList 格式, 如旧版 Qwen2MoE)
elif hasattr(hf_experts, "__iter__"):
    for idx, hf_expert in enumerate(hf_experts):
        if idx < expert_idx_start or idx >= expert_idx_end:
            continue
        local_expert_idx = idx - expert_idx_start

        fc1_weight = torch.cat([hf_expert.gate_proj.weight, hf_expert.up_proj.weight])
        numel += safe_copy(fc1_weight, layer.mlp.experts.linear_fc1._parameters[f"weight{local_
expert_idx}"])
        numel += safe_copy(
            hf_expert.down_proj.weight, layer.mlp.experts.linear_fc2._parameters[f"weight{local_
expert_idx}"])
    )
else:
    raise TypeError(f"Unsupported experts type: {type(hf_experts)}")

```

tests/utlis/test_megatron_bshd_preprocess.py

单元测试增加 NPU 环境检测 mock, 避免测试因 is_npu_available 而跳过关键路径。

```

def _load_mcore_util_with_stubbed_megatron(monkeypatch, tp_size: int = 4):
    megatron = types.ModuleType("megatron")
    core = types.ModuleType("megatron.core")
    parallel_state = types.ModuleType("megatron.core.parallel_state")
    packed_seq_params = types.ModuleType("megatron.core.packed_seq_params")

    parallel_state.get_context_parallel_world_size = lambda: 1
    parallel_state.get_context_parallel_rank = lambda: 0
    parallel_state.get_tensor_model_parallel_world_size = lambda: tp_size
    packed_seq_params.PackedSeqParams = type("PackedSeqParams", (), {})

    core.parallel_state = parallel_state
    megatron.core = core
    monkeypatch.setitem(sys.modules, "megatron", megatron)
    monkeypatch.setitem(sys.modules, "megatron.core", core)
    monkeypatch.setitem(sys.modules, "megatron.core.parallel_state", parallel_state)
    monkeypatch.setitem(sys.modules, "megatron.core.packed_seq_params", packed_seq_params)

    # 新增: 引入 device 模块用于 mock
    import verl.utlis.device as device_module
    # 确保 is_npu_available 返回 False, 避免测试在 NPU 环境下行为不同
    monkeypatch.setattr(device_module, "is_npu_available", False)

    util_path = Path(__file__).parents[2] / "verl" / "models" / "mcore" / "util.py"
    spec = importlib.util.spec_from_file_location("mcore_util_regression", util_path)
    module = importlib.util.module_from_spec(spec)

```

```
spec.loader.exec_module(module)
return module
```

评论区精华

1. 日志目录风险: gemini-code-assist[bot] 指出 run_ppo_qwen3-8b_fsdp_npu.sh 中对 tee 命令的日志目录可能不存在, 建议先 mkdir -p 并重定向 stderr。该问题在合并时未处理。
 2. Unit test 增强: daikang6 在 review 中自荐在 test_megatron_bshd_preprocess.py 中加入 is_npu_available 的 mock, 最终提交已包含此改动。
 3. 转换逻辑解释: daikang6 在 converter 的 diff hunk 中内联解释了 Qwen3MoE 分支处理的设计, 便于 reviewer 理解。
- 日志目录安全写入风险 (correctness): 未采纳修改, PR 合并时未处理该风险。
 - Unit test 增加 is_npu_available mock (testing): 已采纳, 体现在最终提交中。
 - Converter 中 MoE 适配逻辑解释 (design): 无需额外修改, 解释清楚即可。

风险与影响

- 风险:
 - CANN 版本升级: 9.0.0 可能引入算子行为差异 (如随机数生成、内存管理), 需完整 CI 回归 (已配置 nightly 和 e2e 流水线)。
 - 镜像依赖兼容性: 新镜像中的 transformers 5.3.0 可能与其他库版本冲突, 已有独立 commit 专门处理相关问题。
 - MoE 适配覆盖: 转换脚本仅检测 gate_up_proj 属性, 若将来其他 MoE 模型 (如 DeepSeekMoE) 也更改格式, 可能遗漏; 当前仅确认支持 Qwen3MoE。
 - 日志目录未创建: run_ppo_qwen3-8b_fsdp_npu.sh 中 tee 的目录未预先创建, 可能导致 CI 日志丢失 (已指出但未修复)。
- 影响:
 - 用户和系统: NPU 环境用户需使用 CANN 9.0.0 才能运行新 CI 镜像; 转换脚本改进使 transformers 5.x 用户能从 HuggingFace 正确导出 Qwen3MoE 模型到 Megatron 格式。
 - 团队: CI 维护者需监控新镜像下测试稳定性; 模型团队需持续关注转换脚本的兼容性维护。
 - 影响范围: 涉及 15 个文件、12 个 CI 工作流、1 个核心转换脚本、2 个测试脚本, 影响面较广但可控。
 - 风险标记: CI 镜像依赖风险, 日志路径缺失, Transformers 兼容性

关联脉络

- PR #6506 [megatron, trainer] fix: preserve BSHD top-k distillation shape: 修改了相同的测试文件 tests/utils/test_megatron_bshd_preprocess.py 以及 util 模块, 属于同一功能线的持续修复。
- PR #6374 [megatron] feat: ascend bump into megatron 016: 同样涉及 Ascend NPU 的 CANN 版本升级和相关适配, 是前序基础设施升级。