

PR #6230 完整报告

verl-project/verl

[rollout] fix: trtllm rollout docker image and a few scripts

合并时间: 2026-05-07 10:12

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6230>

执行摘要

- 一句话: 修复 TRTLLM rollout 的 Docker 镜像和 CI 脚本兼容性
- 推荐动作: 该 PR 修复了紧急的兼容性问题, 值得合并。但建议合并后:
 1. 验证 Docker 构建是否成功, 确认 cupy 和 ray 版本可用;
 2. 检查 batch_wait 参数是否在 TRTLLM 服务端实现, 若不生效则考虑移除或提交配套实现;
 3. 定期清理 mbridge 临时补丁。

功能与动机

根据 PR 描述, 需要固定 Ray 版本以匹配 TRTLLM 1.3.0rc13 的兼容性要求; 同时恢复在 PR #6126 重构示例时被丢掉的 trtllm 专用参数 (batch_wait_timeout_iters、batch_wait_max_tokens_ratio), 以确保 GRPO 脚本在 TRTLLM 后端下能正常工作。

实现拆解

1. Docker 构建调整: 在 `docker/Dockerfile.stable.trtllm` 中新增 `cupy-cuda12x==14.0.1` 依赖, 并在安装 verl 后强制重新安装 `ray[default]==2.54.1` 以与 TRTLLM 1.3.0rc13 兼容。
2. CI 镜像与 workflow 修复: 更新 `e2e_ppo_grpo_trainer_trtllm.yml` 中的镜像标签为 `trtllm1.3.0rc13-20250504`, 并在两个核心测试步骤中添加 `mbridge` 临时补丁安装 (`pip install --force-reinstall --no-deps --no-build-isolation git+https://github.com/ISEEKYAN/mbridge.git`) 以规避 CI 镜像中 Python 3.12 移出 `imp` 模块的问题。
3. 恢复 VLM E2E 测试: 将原先被注释的 GEO3K VLM 测试重新启用, 并添加了 `train_batch_size=8`、`max_response_length=64` 等若干参数调优, 使测试能在 TRTLLM 后端下运行。
4. 补充 TRTLLM 专用脚本参数: 在 `examples/grpo_trainer/run_qwen3_30b_a3b_megatron.sh` 的 `EXTRA` 数组中为 TRTLLM 模式新增 `batch_wait_timeout_iters=32` 和 `batch_wait_max_tokens_ratio=0.5` 配置。
5. 文档更新: 在 `examples/grpo_trainer/README.md` 中将 Qwen3-235B-A22B 模型在 TRTLLM 支持列从空白标记为 ✓。

关键文件:

- [.github/workflows/e2e_grpo_trainer_trtllm.yml](#) (模块 持续集成; 类别 infra; 类型 infrastructure) : 核心 CI 工作流, 更新镜像版本、添加 mbridge 补丁、恢复 VLM 测试
- [verl/workers/rollout/trtllm_rollout/trtllm_worker_extension.py](#) (模块 工作节点; 类别 source; 类型 core-logic) : 修复 serialization approved_imports 遗漏 bfloat16, 导致权重传输失败
- [docker/Dockerfile.stable.trtllm](#) (模块 部署镜像; 类别 infra; 类型 infrastructure) : 固定 Ray 版本并添加 cupy 依赖, 确保与 TRTLLM 1.3.0rc13 兼容
- [examples/grpo_trainer/run_qwen3_30b_a3b_megatron.sh](#) (模块 训练示例; 类别 other ; 类型 configuration) : 恢复被 PR #6126 移除的 TRTLLM 专用参数
- [examples/grpo_trainer/README.md](#) (模块 文档; 类别 docs; 类型 documentation) : 更新 TRTLLM 支持标记

关键符号: 未识别

关键源码片段

[verl/workers/rollout/trtllm_rollout/trtllm_worker_extension.py](#)

修复 serialization approved_imports 遗漏 bfloat16, 导致权重传输失败

```
# 在 update_weights 方法中使用的 approved_imports 字典
# 控制反序列化权重句柄时允许导入的类型, 防止恶意 pickle
approved_imports = {
    "builtins": [
        "list", "tuple", "str", "int", "float", "bool", "bytes", "dict",
        "NoneType", "type",
    ],
    "torch": [
        "Tensor", "FloatTensor", "DoubleTensor", "HalfTensor", "BFloat16Tensor",
        "IntTensor", "LongTensor", "ShortTensor", "CharTensor", "ByteTensor",
        "BoolTensor", "Size", "dtype", "device",
        "float32", "float16",
        "bfloat16", # 新增: 缺少此条目会导致反序列化 bf16 张量时抛出异常
        "int32", "int64", "int16", "int8", "uint8", "bool",
    ],
    "torch multiprocessing.reductions": [
        "rebuild_cuda_tensor",
        "rebuild_tensor",
    ],
    "torch._utils": [
        "_rebuild_tensor_v2",
    ],
    "torch.storage": [
        "_load_from_bytes",
        "_TypedStorage",
        "UntypedStorage",
        "TypedStorage",
    ],
}
```

```
}  
all_handles = serialization.loads(decoded_data, approved_imports=approved_imports)
```

[docker/Dockerfile.stable.trtllm](#)

固定 Ray 版本并添加 cupy 依赖，确保与 TRTLLM 1.3.0rc13 兼容

```
# 安装 cupy-cuda12x (用于特定 GPU 加速内核) —— 注意: 版本 14.0.1 需确保在私有源上存在  
RUN pip3 install --no-cache-dir cupy-cuda12x==14.0.1 && \  
    pip3 install --no-cache-dir nvtx matplotlib liger_kernel cachetools && \  
    ...  
  
# 在安装 verl 之后固定 Ray 版本以保持与 TRTLLM 1.3.0rc13 的兼容性  
RUN pip install --no-cache-dir "ray[default]==2.54.1"
```

评论区精华

review 中 [gemini-code-assist\[bot\]](#) 提出了三个问题:

- `cupy-cuda12x==14.0.1` 版本可能不存在 (PyPI 最新为 13.3.0)，可能导致 Docker 构建失败;
- `ray[default]==2.54.1` 不是有效发布版本，也会导致构建失败;
- 新增的 `batch_wait_timeout_iters` 和 `batch_wait_max_tokens_ratio` 参数在 `TRTLLMHttpServer` 实现中未被传递到 `SchedulerConfig`，因此实际上不会生效。

这些评论均未得到作者回复或解决，但 PR 最终被项目所有者 [wuxibin89](#) 批准合并。因此这些潜在问题可能已被线下处理或评估为可接受，存在一定风险。

- `cupy/ray` 版本可能无效 (correctness): PR 作者未回复，但 PR 被批准合并，可能版本已验证为内部源或已解决。
- 新增参数在服务端未实现 (design): 未解决参数生效问题，但合并后可能需后续实现。

风险与影响

- 风险：主要风险：
 - Docker 构建可能因 `cupy/ray` 版本无效而失败;
 - 新增的 `batch_wait` 参数在服务端未实现，实际运行时不会生效，可能导致用户错误地认为参数已生效;
 - CI 工作流程中增加了大量参数，可能与其他配置冲突;
 - `mbridge` 补丁是临时方案，后续 CI 镜像更新后需移除。
- 影响：影响范围：
 - 用户：使用 TRTLLM rollout 的用户将获得更新的 Docker 镜像和 CI 测试，但需验证参数是否生效。
 - 系统：Docker 镜像版本固定，避免意外的版本冲突; CI 测试覆盖恢复。
 - 团队：维护者需关注 [gemini](#) 提出的问题是否被实际解决，否则需后续修复。
 - 风险标记：潜在 Docker 构建失败，新增参数可能不生效，临时补丁后续需移除

关联脉络

- PR #6126 [misc] refactor: re-format examples and deprecate old examples: 本 PR 恢复被该 PR 移除的 trtllm 参数
- PR #6215 [ci] chore: bump trtllm to 1.3.0rc13 and verl to v0.7.1: 该 PR 升级 TRTLLM 版本, 本 PR 适配其 Docker 和 CI 配置