

PR #2208 完整报告

THUDM/slime

Support reloading the default process group

合并时间: 2026-07-15 11:34

原文链接: <http://prhub.com.cn/THUDM/slime/pull/2208>

执行摘要

- 一句话: 支持 NCCL WORLD 进程组的销毁与重新加载
- 推荐动作: 建议深入阅读 `slime/utils/reloadable_process_group.py` 中的实现, 特别是 `_destroy_default_nccl_process_group` 和 `_reload_default_process_group` 的设计模式: 利用 Gloo 组作为同步和过渡、使用 PrefixStore 保证 store 无冲突、通过 generation 字段保证唯一性。这些是分布式系统中重建通信组时的常用技巧。同时关注环境变量逃生口的设计。

功能与动机

支持训练过程中释放 actor 组后重建通信环境, 避免因 WORLD 组损坏需要重启所有进程。PR 2180 引入的 `--release-train` 功能依赖此基础。此外, `slime/backends/megatron_utils/actor.py` 的注释明确指出 'Destroying and recreating WORLD invalidates raw dist.group.WORLD references', 需要通过逃生口控制。

实现拆解

1. 新增 `slime/utils/reloadable_process_group.py`: 定义 `_DefaultProcessGroupState` 数据类型捕获 backend、store、rank、world_size 等信息; 提供 `register_default_process_group` 函数供 Actor 初始化时调用, 将状态存入进程级字典 `default_process_group_states`。
2. 实现销毁与重建: `_destroy_default_nccl_process_group` 通过 Gloo 组同步后调用 `dist.destroy_process_group()`, 然后使用 PrefixStore 创建新命名空间的临时 Gloo WORLD 组; `_reload_default_process_group` 销毁临时 Gloo 组, 恢复原始 backend (如 NCCL) 的 WORLD 组, 并重新初始化规范 Gloo 组。两个函数均通过 `state.generation` 递增确保每次使用唯一的 store 前缀。
3. 修改 `slime/utils/distributed_utils.py`: 新增 `set_gloo_group` 函数允许在 reload 过程中替换全局缓存的 Gloo 组; 修改 `init_gloo_group` 使用 `getattr(dist, 'old_new_group', dist.new_group)` 绕过猴子补丁, 保证规范 Gloo 组不受 `ReloadableProcessGroup` 包装影响。
4. 集成到 Actor 初始化: 在 `slime/backends/megatron_utils/actor.py` 的 `init` 方法中, 于 `monkey_patch_torch_dist()` 和 `super().init()` 之后, 调用 `register_default_process_group`, 受 `SLIME_DESTROY_WORLD_PROCESS_GROUP` 环境变量控制 (默认启用)。

5. 修改 `slime/ray/rollout.py`: 仅调整注释缩进, 无逻辑变更。
6. 添加单元测试 `tests/test_reloadable_process_group_world.py`: 使用 `pytest monkeypatch` 模拟分布式调用, 验证 `register_default_process_group` 正确捕获状态 (`test_register_default_process_group_captures_rendezvous_state`) 、
`destroy_process_groups/reload_process_groups` 的事件顺序正确 (`test_world_and_subgroups_follow_destroy_reload_order`) 、以及未注册 WORLD 时不会误破坏 (`test_unregistered_world_preserves_subgroup_only_behavior`) 。
7. 更新 CI 配置: `.github/workflows/pr-test.yml` 添加新测试到 CPU-only 测试矩阵, 并在依赖安装中增加 `psutil`。

关键文件:

- `slime/utils/reloadable_process_group.py` (模块 通信组管理; 类别 `source`; 类型 `core-logic`; 符号 `_DefaultProcessGroupState`, `register_default_process_group`, `_uses_nccl`, `_new_default_process_group`) : 新增核心模块, 实现了 WORLD 组的注册、销毁、重建以及子组管理。
- `tests/test_reloadable_process_group_world.py` (模块 测试覆盖; 类别 `test`; 类型 `test-coverage`; 符号 `test_register_default_process_group_captures_rendezvous_state`, `test_world_and_subgroups_follow_destroy_reload_order`, `test_unregistered_world_preserves_subgroup_only_behavior`) : 新增单元测试覆盖三种关键场景: 正确捕获状态、destroy/reload 事件顺序、未注册时的安全行为。
- `slime/utils/distributed_utils.py` (模块 分布式工具; 类别 `source`; 类型 `core-logic`; 符号 `set_gloo_group`, `init_gloo_group`) : 添加 `set_gloo_group` 函数, 修改 `init_gloo_group` 使用原始 `new_group`, 防止猴子补丁影响规范 Gloo 组。
- `slime/backends/megatron_utils/actor.py` (模块 Actor; 类别 `source`; 类型 `dependency-wiring`; 符号 `init`) : 在 actor 初始化中注册默认进程组, 并通过环境变量提供开关。
- `slime/ray/rollout.py` (模块 Rollout; 类别 `source`; 类型 `configuration`) : 调整注释缩进, 无逻辑变更。
- `.github/workflows/pr-test.yml` (模块 CI 配置; 类别 `infra`; 类型 `infrastructure`) : 添加新测试到 CPU 测试矩阵, 并安装 `psutil` 依赖。
- `.github/workflows/pr-test.yml.j2` (模块 CI 模板; 类别 `infra`; 类型 `infrastructure`) : 模板文件对应调整。

关键符号: `_DefaultProcessGroupState`, `register_default_process_group`, `_uses_nccl`, `_new_default_process_group`, `_destroy_default_nccl_process_group`, `_reload_default_process_group`, `set_gloo_group`, `init_gloo_group` (修改), `MegatronTrainRayActor.init` (修改), `destroy_process_groups` (现有), `reload_process_groups` (现有)

关键源码片段

`slime/utils/reloadable_process_group.py`

新增核心模块, 实现了 WORLD 组的注册、销毁、重建以及子组管理。

```

@dataclass
class _DefaultProcessGroupState:
    '''Captures the rendezvous state of the original WORLD group for later restore.'''
    backend: str
    timeout: timedelta
    store: Any # Keep reference to rank-0 TCPStore
    rank: int
    world_size: int
    generation: int = 0
    nccl_world_destroyed: bool = False

def register_default_process_group(timeout: timedelta) -> None:
    '''Snapshot the current WORLD group state.

    Stores a reference to the underlying rendezvous store to keep it alive
    after ``destroy_process_group()``.
    ...

    if not dist.is_initialized():
        raise RuntimeError('Cannot register WORLD before torch.distributed is initialized')
    pid = os.getpid()
    backend = str(dist.get_backend())
    state = _DefaultProcessGroupState(
        backend=backend,
        timeout=timeout,
        store=_get_default_store(), # prevent store garbage collection
        rank=dist.get_rank(),
        world_size=dist.get_world_size(),
    )
    default_process_group_states[pid] = state

def _destroy_default_nccl_process_group() -> None:
    '''Destroy NCCL WORLD group and replace with temporary Gloo WORLD.

    1. Use canonical Gloo group to barrier all ranks.
    2. Call ``dist.destroy_process_group()`` (destroys all sub-groups).
    3. Replace cached Gloo group with ``None``.
    4. Initialize a new Gloo WORLD via ``_new_default_process_group``.
    5. Re-cache the new default group as the canonical Gloo group.
    ...

    state = default_process_group_states.get(os.getpid())
    if state is None or state.nccl_world_destroyed or not _uses_nccl(state.backend):
        return

    # Use the CPU group as an out-of-band synchronization point.
    dist.barrier(group=get_gloo_group())
    dist.destroy_process_group()
    set_gloo_group(None)

```

```
_new_default_process_group(state, backend='gloo')
set_gloo_group(_get_default_group())
state.nccl_world_destroyed = True
```

评论区精华

无 review 讨论，PR 由作者自行审查合并。代码注释中体现了关键设计决策：使用 Gloo 组作为过渡（避免 CUDA 操作），通过 PrefixStore 生成唯一 store 前缀防止 stale key，以及通过环境变量提供逃生口。

- 暂无高价值评论线程

风险与影响

- 风险：
 - 若外部代码缓存了 dist.group.WORLD 引用，销毁后引用将失效，可能导致静默错误或崩溃。代码通过 SLIME_DESTROY_WORLD_PROCESS_GROUP=0 提供关闭能力。
 - 修改 init_gloo_group 使用 old_new_group 绕过猴子补丁，如果未来 PyTorch 或 Monkey Patch 逻辑改变，可能出现不一致。
 - 测试仅使用 monkeypatch 模拟复杂分布式调用，未进行实际多进程集成测试，可能遗漏真实环境下的竞态条件或 NCCL 错误处理问题。
 - 新增依赖 psutil 用于测试（测试文件未明显使用，可能是间接依赖），需确认无冲突。
- 影响：
 - 对用户：需在训练启动时调用 register_default_process_group（已在 actor 初始化中自动调用）。如需禁用可通过环境变量关闭。
 - 对系统：新增 slime/utils/reloadable_process_group.py 核心模块，修改 distributed_utils.py 和 actor.py。影响所有使用 MegatronTrainRayActor 的训练流程。
 - 对团队：提供了可复用的 WORLD 组重建模式，便于后续实现动态资源管理（如 release-train）。
 - 影响程度：中等，主要影响需要动态管理通信组的训练场景。
 - 风险标记：核心路径变更（actor.init），测试仅依赖 monkeypatch，环境变量逃生口

关联脉络

- PR #2180 Add --release-train: PR 2180 引入了训练后释放 actor 组的能力，该 PR 提供了 WORLD 组销毁重建的基础机制，两者共同实现动态资源管理。