

PR #1952 完整报告

THUDM/slime

disable param backup

合并时间: 2026-05-27 12:32

原文链接: <http://prhub.com.cn/THUDM/slime/pull/1952>

执行摘要

- 一句话: 禁用参数 CPU 备份以节省主机内存
- 推荐动作: 该 PR 值得性能优化工程师和训练系统开发者关注, 尤其是如何通过禁用不必要的 CPU 备份来节省内存的设计决策, 以及对 Megatron 底层补丁的实现技巧。

功能与动机

减少训练过程中参数和梯度 CPU 备份带来的大量主机内存开销, 特别是在 offload 训练和大型模型场景下, 此 PR 通过禁用不必要的备份来优化内存使用。这是持续内存优化系列的一部分。

实现拆解

1. 在 Megatron 层增加参数支持: 在 `docker/patch/latest/megatron.patch` 中, 向 `DistributedDataParallel` 和 `_ParamAndGradBuffer` 新增 `disable_param_buffers_cpu_backup` 和 `disable_grad_buffers_cpu_backup` 参数, 并在 `nccl_ub` 为假时, 利用 `torch_memory_saver.region` 创建无 CPU 备份的内存分配上下文。
2. 在配置层调整参数: 在 `slime/utils/arguments.py` 中移除 `--disable-weights-backuper` 参数及其相关逻辑; 在 `slime_validate_args` 中, 当 `offload_train` 时自动设置 `disable_param_buffers_cpu_backup=True`; 在 `_apply_megatron_role_overrides` 中, 对 `critic` 角色默认禁用此选项 (除非显式覆盖)。
3. 在 actor 初始化中去掉了 `translate_gpu_to_cpu` 参数和根据 `enable_weights_backuper` 决定 `single_tag` 的逻辑, 简化 `TensorBackuper` 创建。同时, 在 `wake_up` 方法中添加 `self._switch_model("actor")` 调用, 确保在恢复训练时正确切换模型状态。
4. 在 `slime/ray/placement_group.py` 中, 对 `critic` 模型的参数进行额外处理: 当未通过 `megatron_config_path` 指定配置时, 强制设置 `disable_param_buffers_cpu_backup=False`, 避免继承 actor 的设置导致错误。

关键文件:

- `docker/patch/latest/megatron.patch` (模块 分布式层; 类别 source; 类型 core-logic; 符号 `DistributedDataParallel`, `_ParamAndGradBuffer`, `_make_no_backup_context`): 核心变更, 在 Megatron 分布式训练底层添加禁用参数和梯度 CPU 备份的支持, 利用 `torch_memory_saver` 实现无备份内存分配。
- `slime/ray/placement_group.py` (模块 调度层; 类别 source; 类型 configuration-setting; 符号 `create_training_models`): 确保 `critic` 进程不使用 actor 的 CPU 备份禁用设置,

防止异常。

- slime/utils/arguments.py (模块 配置管理; 类别 source; 类型 core-logic; 符号 add_train_arguments, _apply_megatron_role_overrides, slime_validate_args) : 移除旧的 --disable-weights-backuper 参数, 并添加新参数默认行为。
- slime/backends/megatron_utils/actor.py (模块 训练后端; 类别 source; 类型 core-logic; 符号 init, wake_up, train) : 简化 TensorBackuper 创建, 移除废弃参数, 将模型切换逻辑内联化。

关键符号: init, wake_up, train, create_training_models, add_train_arguments, _apply_megatron_role_overrides, slime_validate_args, DistributedDataParallel.init, _ParamAndGradBuffer.init

关键源码片段

[docker/patch/latest/megatron.patch](#)

核心变更, 在 Megatron 分布式训练底层添加禁用参数和梯度 CPU 备份的支持, 利用 torch_memory_saver 实现无备份内存分配。

```
# Inside _ParamAndGradBuffer.__init__, after setting up basic attributes:
def _make_no_backup_context(tag, disable, flag_name="disable_grad_buffers_cpu_backup"):
    """Return a context manager that disables CPU backup for memory allocation."""
    if disable:
        try:
            from torch_memory_saver import torch_memory_saver
        except ImportError as e:
            raise ImportError(
                f"{flag_name}=True requires torch_memory_saver. "
                "Install with: pip install torch-memory-saver"
            ) from e
        return partial(
            torch_memory_saver.region,
            tag=tag,
            enable_cpu_backup=False,
        )
    return nullcontext

# Create context for gradient buffer
# (parameter buffer context follows similar pattern)
grad_mem_alloc_context = _make_no_backup_context(
    "grad_buffer", disable_grad_buffers_cpu_backup
)
```

[slime/backends/megatron_utils/actor.py](#)

简化 TensorBackuper 创建, 移除废弃参数, 将模型切换逻辑内联化。

```
# In actor's init method:
self.weights_backuper = TensorBackuper.create(
    source_getter=lambda: named_params_and_buffers(
```

```

        self.args,
        self.model,
        convert_to_global_name=args.megatron_to_hf_mode == "raw",
    ),
    single_tag=None, # Previously depended on enable_weights_backupper; now always None
)
self._active_model_tag: str | None = "actor"
self.weights_backupper.backup("actor")

# In wake_up method:
@timer
def wake_up(self) -> None:
    assert self.args.offload_train
    print_memory("before wake_up model")
    torch_memory_saver.resume()
    clear_memory()
    reload_process_groups()
    if self.role == "actor":
        self._switch_model("actor") # New: restore actor model after sleep
    print_memory("after wake_up model")

```

评论区精华

该 PR 没有 review 讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
 - 兼容性风险：移除 `--disable-weights-backuper` 参数，旧有脚本若使用该参数将会报错，需更新配置。
 - 功能冲突：`disable_param_buffers_cpu_backup` 与 `nccl_ub=True` 不兼容，代码中已通过 `assert` 保护，但用户需注意避免同时设置。
 - critic 隔离风险：若 critic 的 `disable_param_buffers_cpu_backup` 从 actor 继承 `True`，可能导致训练异常，代码专门在 `placement_group.py` 和 `arguments.py` 中处理了此情况。
 - 测试覆盖：没有新增专门的测试用例，回归风险主要依赖已有测试。
- 影响：
 - 用户影响：需要移除命令行中的 `--disable-weights-backuper`，但功能由新标志自动接管，预期用户无感知（除了参数移除带来的错误提示）。主机内存使用将显著降低，有利于部署更大模型。
 - 系统影响：减少了 CPU 内存占用，可能提升系统整体稳定性，减少 OOM 风险。
 - 团队影响：与之前的内存优化 PR 形成系列，体现了持续的性能调优方向。

- 风险标记：旧参数移除可能导致用户配置兼容性问题，nccl_ub 下不能禁用备份，需确保 critic 默认启用备份，无新增测试覆盖

关联脉络

- PR #1932 save host memory: 同为主机内存优化系列改进，调整配置项赋值顺序以节省主机内存。
- PR #1924 Reduce host memory with upgraded tms: 同为主机内存优化系列，升级 torch_memory_saver 减少内存占用。