

PR #51652 完整报告

vllm-project/vllm

[CI/Build] Use file rendezvous for local distributed tests

合并时间: 2026-08-12 10:48

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/51652>

执行摘要

- 一句话: 分布式单节点测试改用 file rendezvous, 消除端口竞争
- 推荐动作: 建议快速阅读: 作为并行 CI 消除 TCP 端口竞态的轻量范例, `get_file_store_init_method` 的用法和范围限定值得借鉴。不涉及生产代码, 无需深入精读。

功能与动机

PR body 指出 `get_open_port()` 会在释放探测 socket 之后、子进程真实初始化进程组之前留出一个时间窗, 并行 CI 下其他进程可以在该间隙抢占端口, 使原本正确的测试以 `EADDRINUSE` 失败。作者将其定位为 #51275 中确认的 test-only follow-up, 目标是移除探测再绑定窗口, 同时明确把 `AITER/custom-all-reduce`、`HTTP server` 端口、`StatelessProcessGroup` 与多节点启动器排除在范围外。

实现拆解

1. 统一 rendezvous 生成入口

两个测试模块均从 `vllm.utils.network_utils` 引入 `get_file_store_init_method`, 在父进程生成一次唯一的 `file:// init method`。FileStore 只依赖本地文件系统协调, 不需要监听 TCP 端口, 因此从根上消除端口竞态。

2. DCP A2A 启动器改造 (tests/distributed/test_dcp_a2a.py)

`_distributed_run` 用 `DISTRIBUTED_INIT_METHOD` 替换 `MASTER_ADDR/MASTER_PORT`; `worker_distributed_packed_a2a_worker` 显式读取 `WORLD_SIZE`, 并在 `dist.init_process_group` 中同时传 `init_method`、`rank`、`world_size`, 覆盖 `cpu:gloo,cuda:nccl` 与 `nccl` 两个分支。

3. Async-TP spawn 测试改造 (tests/compile/passes/distributed/test_async_tp.py)

`test_async_tp_pass_replace` 将 `master_port` 替换为 `distributed_init_method` 并随 `torch.multiprocessing.spawn` 传给 worker; worker 中删除 `MASTER_ADDR/MASTER_PORT`, 改用 `init_distributed_environment` 的显式 `world_size/rank/distributed_init_method/local_rank` 参数完成握手。

4. 唯一性契约测试 (tests/utils/_test_network_utils.py)

新增 `test_get_file_store_init_method_is_unique`, 断言同一进程内连续两次调用返回两个不同的 `file://` 方法, 保护并行测试实例不会因共享同一个 `FileStore` 文件而互相干扰。

5. 验证情况

作者在 CPU 环境跑通网络工具单测 21 passed, 三个文件的 pre-commit 与 mypy (3.10/3.12) 均通过, 并在 macOS ARM 与 Linux x86_64 上验证两进程 Gloo+FileStore 冒烟; GPU 场景交给 Buildkite CI 覆盖。

关键文件:

- `tests/distributed/test_dcp_a2a.py` (模块 分布式测试; 类别 test; 类型 test-coverage; 符号 `_distributed_run`, `_distributed_packed_a2a_worker`) : 核心改动文件: DCP A2A 测试启动器改为唯一 `file://` 握手, 消除探测端口竞态。
- `tests/compile/passes/distributed/test_async_tp.py` (模块 编译测试; 类别 test; 类型 test-coverage; 符号 `test_async_tp_pass_replace`, `async_tp_pass_on_test_model`) : Async-TP spawn 测试同步改为 `FileStore` 握手, 并改为显式调用 `init_distributed_environment`, 是本次改造的另一半核心。
- `tests/utils/_test_network_utils.py` (模块 网络工具; 类别 test; 类型 test-coverage; 符号 `test_get_file_store_init_method_is_unique`) : 新增 `get_file_store_init_method` 唯一性契约测试, 防止并行测试共用同一 `FileStore` 文件。

关键符号: `get_file_store_init_method`, `_distributed_run`, `_distributed_packed_a2a_worker`, `async_tp_pass_on_test_model`, `test_get_file_store_init_method_is_unique`

关键源码片段

`tests/distributed/test_dcp_a2a.py`

核心改动文件: DCP A2A 测试启动器改为唯一 `file://` 握手, 消除探测端口竞态。

```
def _distributed_run(fn, world_size: int, extra_env: dict[str, str]) -> None:
    # 用唯一的 file:// rendezvous 取代 get_open_port() 的“探测后绑定”窗口,
    # 避免并行 CI 中端口在探测释放与 worker 真正初始化之间被其他进程抢占。
    distributed_init_method = get_file_store_init_method()
    processes: list[mp.Process] = []
    for rank in range(world_size):
        env = {
            "RANK": str(rank),
            "LOCAL_RANK": str(rank),
            "WORLD_SIZE": str(world_size),
            "LOCAL_WORLD_SIZE": str(world_size),
            # FileStore 只要求本机文件系统路径, 不需要对外可达端口
            "DISTRIBUTED_INIT_METHOD": distributed_init_method,
            **extra_env,
        }
        process = mp.Process(target=fn, args=(env,))
        processes.append(process)
        process.start()
```

```

for process in processes:
    process.join(timeout=120)

for process in processes:
    if process.is_alive():
        process.kill()
        process.join()
    assert process.exitcode == 0

```

```

def _distributed_packed_a2a_worker(env: dict[str, str]) -> None:
    update_environment_variables(env)
    local_rank = int(env["LOCAL_RANK"])
    world_size = int(env["WORLD_SIZE"]) # 显式读取, 与显式 init_method 配套
    torch.accelerator.set_device_index(local_rank)

    if envs.VLLM_DISTRIBUTED_USE_SPLIT_GROUP:
        dist.init_process_group(
            backend="cpu:gloo,cuda:nccl",
            device_id=torch.device(f"cuda:{local_rank}"),
            init_method=env["DISTRIBUTED_INIT_METHOD"],
            rank=local_rank,
            world_size=world_size,
        )
    else:
        dist.init_process_group(
            backend="nccl",
            init_method=env["DISTRIBUTED_INIT_METHOD"],
            rank=local_rank,
            world_size=world_size,
        )

```

tests/compile/passes/distributed/test_async_tp.py

Async-TP spawn 测试同步改为 FileStore 握手, 并改为显式调用 `init_distributed_environment`, 是本次改造的另一半核心。

```

def async_tp_pass_on_test_model(
    local_rank: int,
    world_size: int,
    test_model_cls: torch.nn.Module,
    batch_size: int,
    seq_len: int,
    hidden_size: int,
    dtype: torch.dtype,
    dynamic: bool,
    distributed_init_method: str, # 父进程通过 get_file_store_init_method() 生成
):
    set_random_seed(0)
    device = torch.device(f"{DEVICE_TYPE}:{local_rank}")

```

```

torch.accelerator.set_device_index(device)
torch.set_default_device(device)
torch.set_default_dtype(dtype)

update_environment_variables(
    {
        "RANK": str(local_rank),
        "LOCAL_RANK": str(local_rank),
        "WORLD_SIZE": str(world_size),
        # 注意: MASTER_ADDR / MASTER_PORT 不再设置, 改用显式 init_method
    }
)

# 显式传参绕过环境变量中的 TCP 探测结果, 握手完全由文件 Store 承担
init_distributed_environment(
    world_size=world_size,
    rank=local_rank,
    distributed_init_method=distributed_init_method,
    local_rank=local_rank,
)

```

tests/utils/_test_network_utils.py

新增 `get_file_store_init_method` 唯一性契约测试, 防止并行测试共用同一 `FileStore` 文件。

```

def test_get_file_store_init_method_is_unique():
    # 同一进程内连续两次调用必须返回不同的 file:// 路径,
    # 否则并行测试实例会争用同一个 FileStore 文件。
    init_methods = {get_file_store_init_method() for _ in range(2)}

    assert len(init_methods) == 2
    assert all(method.startswith("file://") for method in init_methods)

```

评论区精华

本 PR 没有实质性的代码评审争论, 主要流程事件如下:

- claude[bot] 提示 PR 来自 fork, 自动评审被禁用, 需维护者手动触发。
- 提交者 yu-xin-c rebase 后请求 aoshen02 确认范围, 说明 GPU 用例只能交由上游 CI。
- aoshen02 两次触发 /ci run, CI 未报告问题。
- ZJY0516 直接批准, 未留补充意见。
- 评审流程与范围确认 (question): aoshen02 触发 CI 两次且无反对; ZJY0516 直接批准, 无补充意见。

风险与影响

- 风险: ### 功能范围收敛 仅覆盖父进程 spawn 的单节点测试; 若未来扩展为多节点或需要外部可达的 rendezvous, `file://` 将不适用, 需重新引入 `TCPStore`。

平台差异

FileStore 依赖本地文件系统与文件锁语义；作者只在 macOS ARM 与 Linux x86_64 做了 Gloo 冒烟验证，Windows 等平台行为未覆盖。

环境变量依赖变化

test_async_tp.py 删除 MASTER_ADDR/MASTER_PORT 后，若 init_distributed_environment 后续实现重新依赖这些环境变量，可能出现隐性回归。

唯一性依赖

get_file_store_init_method 的唯一性契约刚被新测试锁定，若实现演变为复用固定路径，并行测试将互相干扰。

- 影响：影响面严格限定在 3 个测试文件与 CI 稳定性，对用户无运行时影响。直接收益是消除 Buildkite 并行 CI 下 EADDRINUSE 类偶发失败，降低分布式测试假阴性；对开发者而言，提供了一个无端口分布式握手的参考模式。
- 风险标记：竞态修复，单节点范围，缺少 GPU 实测，FileStore 平台差异

关联脉络

- PR #51275（原文未提供标题）：PR body 明确表示本 PR 是 #51275 中确认的 test-only follow-up，作用范围由 #51275 讨论确定。
- PR #50999（原文未提供标题）：PR body 提到生产执行器变更为 #50999，并说明其不修改这些测试启动器，用于排除影响范围。
- PR #51635（原文未提供标题）：PR body 提到它是 #50999 的 ROCm 兼容性 follow-up，同属测试基础设施影响范围核查。