# PR #2265 完整报告

radixark/miles

[AMD] Enable Qwen3 FSDP hybrid-shard CI on ROCm

合并时间：2026-08-12 09:04

原文链接：http://prhub.com.cn/radixark/miles/pull/2265

## 执行摘要

- 一句话：修复 ROCm 上 FSDP 混合分片并重新启用 CI
- 推荐动作：值得快速精读，特别是 build_fsdp_meshes 的分支写法。亮点在于按能力而非版本选择 API，避免对 ROCm 的 PyTorch 版本硬编码；但它依赖 hasattr 探测私有接口，长期看可在注释中写明 _unflatten 的等价语义（二维 mesh 的 rank 映射），或考虑抽象一个 mesh factory 统一两条路径。建议在后续 FSDP 重构中保留双方等价性的回归测试。

## 功能与动机

PR body 明确指出：ROCm's PyTorch version does not provide the private DeviceMesh._unflatten method used to construct the hybrid-shard mesh。此前该用例因失败在 ROCm CI 上被禁用，且 FSDP 后端移出 experimental、删除 context parallelism 时把已应用的 fallback 一并移除；本次在保持 NVIDIA 路径不变的前提下，用能力检测恢复 ROCm 支持并重新打开 CI 回归。

## 实现拆解

1. 修改 build_fsdp_meshes（miles/backends/fsdp_utils/parallel.py）：将无条件调用 dp_mesh._unflatten 改为 hasattr 能力探测分支，缺失时用 init_device_mesh 按相同的 mesh_shape 与 mesh_dim_names 构造二维网格，保持 hybrid-shard 的复制维 / 分片维语义等价，且不依赖私有 API。
2. 保持 create_fsdp_parallel_state 的消费方式不变：仍从 build_fsdp_meshes 取 dp 与 fsdp 两个 mesh，后续 FSDP2 shard 逻辑无感知。
3. 测试配套：tests/e2e/fsdp/test_qwen3_4B_fsdp_hybrid_shard_r2s2.py 删除 register_rocm_ci 中的 disabled="Disable due to failure"，用例重新注册到 stage-c-4-gpu-mi300x，恢复 ROCm CI 覆盖。
4. 验证：提交者在 4× AMD MI355 GPU（PyTorch 2.9.1+ROCm）上跑通 e2e 与 fast-gpu 的 r1s4/r2s2/r4s1 全部组合；注意 PR 验证硬件是 MI355，而 CI 套件名是 mi300x，属于不同代际的 ROCm 环境。
5. 无配置文件、schema 或部署改动。

关键文件：

- miles/backends/fsdp_utils/parallel.py（模块 网格构建；类别 source；类型 core-logic；符号 build_fsdp_meshes）：核心兼容性修复：在 build_fsdp_meshes 中基于能力检测回退

构造 hybrid-shard mesh，直接决定 ROCm 上 FSDP 能否运行。

- tests/e2e/fsdp/test_qwen3_4B_fsdp_hybrid_shard_r2s2.py（模块 端到端测试；类别 test；类型 test-coverage；符号 register_rocm_ci）：移除 ROCm 禁用标记，重新启用 MI300X 上的 FSDP hybrid-shard e2e 回归，防止兼容性回归。

关键符号：build_fsdp_meshes

## 关键源码片段

### miles/backends/fsdp_utils/parallel.py

核心兼容性修复：在 build_fsdp_meshes 中基于能力检测回退构造 hybrid-shard mesh，直接决定 ROCm 上 FSDP 能否运行。

```
def build_fsdp_meshes(
    device_type: str,
    world_size: int,
    dp_replicate_size: int,
) -> dict[str, DeviceMesh]:
    """Build the data-parallel view and the FSDP2 shard mesh."""
    # 先构造一维数据并行网格，作为 hybrid-shard 的基础视图。
    dp_mesh = init_device_mesh(
        device_type,
        mesh_shape=(world_size,),
        mesh_dim_names=("dp",),
    )
    fsdp_mesh = dp_mesh
    if dp_replicate_size > 1:
        # ROCm 版 PyTorch 没有私有方法 _unflatten，这里用能力检测而非版本判断分流：
        # 存在 _unflatten 时保持 NVIDIA 与新版 PyTorch 的原路径不变；
        # 缺失时退回到公开 API init_device_mesh 构造同样的二维网格。
        if hasattr(dp_mesh, "_unflatten"):
            fsdp_mesh = dp_mesh._unflatten(
                0,
                (dp_replicate_size, world_size // dp_replicate_size),
                ("dp_replicate", "dp_shard"),
            )
        else:
            fsdp_mesh = init_device_mesh(
                device_type,
                mesh_shape=(dp_replicate_size, world_size // dp_replicate_size),
                mesh_dim_names=("dp_replicate", "dp_shard"),
            )

    return {
        "dp": dp_mesh,
        "fsdp": fsdp_mesh,
    }
```

### tests/e2e/fsdp/test_qwen3_4B_fsdp_hybrid_shard_r2s2.py

移除 ROCm 禁用标记，重新启用 MI300X 上的 FSDP hybrid-shard *e2e* 回归，防止兼容性回归。

```
# CUDA 侧注册不变：继续在 H200 上执行该 e2e 用例。
register_cuda_ci(
    est_time=600,
    suite="stage-c-4-gpu-h200",
    labels=["fsdp"],
)
# ROCm 侧重新启用：之前的失败已由 parallel.py 的 _unflatten fallback 修复，
# 因此移除 disabled 标记，让该用例回到 MI300X 套件做持续回归。
register_rocm_ci(
    est_time=600,
    suite="stage-c-4-gpu-mi300x",
    labels=["fsdp", "amd"],
)
```

# 评论区精华

该 PR 没有任何内联 review 评论，仅由维护者 guapisolo APPROVED。有价值的线索来自提交信息：'Reapply the DeviceMesh capability fallback after the FSDP backend moved out of experimental and dropped context parallelism'，说明这不是全新修复，而是把先前存在、在 #2386 删除 context parallelism 时被顺带移除的 fallback 重新应用。设计取舍在于用 hasattr 能力检测而非版本判断，避免对 PyTorch 版本做硬编码分支。

- 暂无高价值评论线程

# 风险与影响

- 风险：miles/backends/fsdp_utils/parallel.py 的 build_fsdp_meshes 是 FSDP 后端所有 hybrid-shard 运行的必经路径，fallback 分支若在网格维度映射上与 _unflatten 有细微差异（如 rank 坐标顺序），可能导致集合通信组错配；目前靠 r1s4/r2s2/r4s1 组合测试覆盖验证，但测试只在 4 卡规模通过。hasattr 探测私有 API 本身较脆弱：一旦未来 PyTorch 在保留 _unflatten 语义不变的前提下改名或删除，fallback 会自动接管全部环境，行为可能偏离预期；不过 fallback 用的是公开稳定的 init_device_mesh。ROCm CI 重新启用存在 flaky 风险：此前因失败禁用，本次验证环境为 MI355，而 CI 跑在 MI300X 上，驱动与库版本组合不同，*e2e* 用例还会下载模型与数据集，网络或驱动问题可能造成偶发失败。另外 *create_fsdp_parallel_state* 中 *device_type* 仍硬编码为 *cuda*，ROCm 的 PyTorch 通常兼容可见性为 *cuda*，但若未来需要显式区分 hip，该处也需同步调整。
- 影响：对用户：ROCm 用户可以正常跑 Qwen3-4B FSDP hybrid-shard（DP_REPLICATE_SIZE>1）训练，不再因私有 API 缺失而中断；NVIDIA 与新版 PyTorch 用户不受影响。对系统：FSDP 后端 mesh 构建路径增加一个等价分支，逻辑可读性略降但风险集中在 4 卡以上组合。对团队：MI300X CI 套件恢复一个 *e2e* 回归用例，后续 FSDP 或 ROCm 改动会被持续监控；同时为 AMD 平台支持积累验证证据。
- 风险标记：私有 API 兼容分支，CI 重开 flaky 风险，mesh 语义依赖测试验证

## 关联脉络

- PR #2386 fix: drop context parallelism from the FSDP backend: 同为目标文件 miles/backends/fsdp_utils/parallel.py 的 FSDP 后端维护，提交信息明确说 fallback 在删除 context parallelism 时被移除，本 PR 重新应用时需同步确认互不冲突。
- PR #2384 fix(fsdp): stop store_true from shadowing bool defaults in FSDPArgs: 同属 FSDP 后端参数与逻辑修复线，说明该后端近期在持续做稳定性收敛，本 PR 是其跨平台延伸。
- PR #2388 fix(fsdp): make --config actually apply, and reject keys it does not know: 同属 FSDP 后端修复，与本 PR 一起体现 FSDP 配置与网格构建的兼容性维护脉络。