PR #7412 完整报告

PaddlePaddle/FastDeploy

[PD Disaggregation] Enable PD deployment without Router

合并时间：2026-04-15 20:13

原文链接：http://prhub.com.cn/PaddlePaddle/FastDeploy/pull/7412

执行摘要

- 一句话：支持 PD 分离部署无需路由器，放宽配置限制并新增测试验证。
- 推荐动作：该 PR 值得精读，特别是配置松耦合的设计决策（如 init_pd_info 逻辑调整）和测试模拟无路由器部署的方法。建议关注并发处理优化和兼容性权衡，以指导类似部署场景的实现。

功能与动机

根据 PR body，动机是 ' 对齐 RL 使用场景，PD 分离部署不强制依赖 router。' 这意味着为了满足强化学习场景的需求，需要支持无需路由器的 PD 分离部署，以降低部署复杂度和适应特定使用场景。

实现拆解

1. 新增端到端测试文件：在 tests/e2e/test_ernie_03b_pd_wo_router_v1_rdma_tp1.py 中新增测试，模拟无路由器时手动构建 disaggregate_info 并并发向 Prefill 和 Decode 发送请求，验证 RDMA 传输协议。关键符号包括 _build_disaggregate_info 和 _send_pd_request。
2. 重命名配置初始化方法：在 fastdeploy/config.py 中，将 init_cache_info 方法重命名为 init_pd_info，并调整逻辑，使 local 调度器不再强制要求路由器，从而支持无路由器部署。这影响后续调度器初始化。
3. 调整参数校验逻辑：在 fastdeploy/engine/args_utils.py 中，修改 __post_init__ 方法：当 scheduler_name 为 "splitwise" 时抛出错误（可能推动 v0 部署废弃），当 scheduler_name 为 "local" 且 router 为 None 时从错误改为警告，允许运行。
4. 删除废弃脚本：删除 examples/splitwise/start_v0_tp1.sh 脚本，可能因 v0 部署不再支持或整合到新逻辑中。
5. 更新其他调用点：在 fastdeploy/engine/expert_service.py 和 tests/model_executor/test_thinking_budget.py 中，将 init_cache_info 调用更新为 init_pd_info，确保一致性。

关键文件：

- tests/e2e/test_ernie_03b_pd_wo_router_v1_rdma_tp1.py（模块 PD 分离测试；类别 test；类型 test-coverage；符号 _build_disaggregate_info, _send_pd_request, _post_stream, _post）：新增端到端测试，验证无路由器 PD 部署的完整流程，是关键验证

手段。

- fastdeploy/config.py（模块 配置管理；类别 infra；类型 infrastructure；符号 init_cache_info, init_pd_info）：修改配置初始化逻辑，重命名方法并调整 splitwise_version 判断，是支持无路由器部署的核心。
- fastdeploy/engine/args_utils.py（模块 参数校验；类别 infra；类型 infrastructure）：调整参数校验，允许 local 调度器无路由器运行，并禁止 splitwise 调度器以推动 v0 废弃。
- examples/splitwise/start_v0_tp1.sh（模块 示例脚本；类别 other；类型 deletion）：删除废弃的 v0 部署脚本，可能因不再支持或整合到新逻辑中。
- tests/model_executor/test_thinking_budget.py（模块 思考预算测试；类别 test；类型 test-coverage）：更新测试中的方法调用，确保与重命名后的 init_pd_info 一致。
- fastdeploy/engine/expert_service.py（模块 专家服务；类别 infra；类型 infrastructure）：更新专家服务中的方法调用，确保配置初始化正确。

关键符号：init_pd_info, _build_disaggregate_info, _send_pd_request, post_init

关键源码片段

tests/e2e/test_ernie_03b_pd_wo_router_v1_rdma_tp1.py

新增端到端测试，验证无路由器 PD 部署的完整流程，是关键验证手段。

```
def _build_disaggregate_info() -> dict:
    """
    手动构建disaggregate_info，模拟Router的handle_splitwise_request逻辑。
    用于无路由器场景下，提供Prefill和Decode节点的连接信息。
    """
    host_ip = os.getenv("FD_HOST_IP", "127.0.0.1")
    return {
        "prefill_ip": host_ip, # Prefill节点的IP地址
        "decode_ip": host_ip, # Decode节点的IP地址
        "prefill_connector_port": FD_CONNECTOR_PORT, # Prefill连接器端口
        "decode_connector_port": FD_CONNECTOR_PORT + 1, # Decode连接器端口
        "decode_device_ids": ["1"], # Decode设备ID列表
        "decode_rdma_ports": [FD_RDMA_PORT + 1], # Decode RDMA端口列表
        "transfer_protocol": "rdma", # 传输协议设置为RDMA
        "decode_tp_size": 1, # Decode TP大小
    }
```

fastdeploy/config.py

修改配置初始化逻辑，重命名方法并调整splitwise_version判断，是支持无路由器部署的核心。

```
def init_pd_info(self):
    """
    初始化PD部署信息，支持无路由器场景。

    根据调度器名称确定splitwise版本：v0用于splitwise或dp调度器，v1用于local调度器（路由器可选）。
    """
```

```
    # TODO: 分组splitwise参数
    # PD分离部署有两种方法:
    # 1. v0: 使用splitwise_scheduler或dp_scheduler
    # 2. v1: 使用local_scheduler + router（可选）
    self.splitwise_version = None
    if self.scheduler_config.name in ("splitwise", "dp"):
        self.splitwise_version = "v0" # v0版本，依赖splitwise调度器
    elif self.scheduler_config.name == "local":
        self.splitwise_version = "v1" # v1版本，local调度器支持无路由器
```

fastdeploy/engine/args_utils.py

调整参数校验，允许 local 调度器无路由器运行，并禁止 splitwise 调度器以推动 v0 废弃。

```
if self.splitwise_role != "mixed":
    if self.scheduler_name == "splitwise":
        # 禁止使用splitwise调度器，推动v0部署废弃
        raise ValueError(
            "Setting scheduler_name as splitwise is not supported in pd deployment, "
            "please use router as scheduler."
        )
    if self.scheduler_name == "local" and self.router is None:
        # 允许local调度器无路由器运行，改为警告提示
        console_logger.warning(
            f"Running {self.splitwise_role} role with {self.scheduler_name} "
            f"scheduler without --router. Router registration and request routing will be disabled."
        )
```

评论区精华

Review 中主要讨论点：1. 兼容性风险：Copilot 指出 args_utils.py 中禁止 scheduler_name=="splitwise" 可能影响 v0 部署兼容性，建议明确处理或更新文档。2. 方法重命名遗漏调用：Copilot 警告 init_cache_info 重命名为 init_pd_info 后，其他调用点（如 expert_service.py）需同步更新，否则可能导致运行时错误。3. 测试并发逻辑缺陷：Copilot 指出测试中 _send_pd_request 使用 ThreadPoolExecutor 可能阻塞，未真正实现非阻塞 fan-out 行为，建议优化。4. 测试覆盖不足：fastdeploy-bot 建议补充其他传输协议（如 IPC）的测试用例，确保无路由器场景下所有协议正常工作。

- 兼容性风险：禁止 splitwise 调度器 (design): 未在 PR 中明确解决，可能需后续评估或提供迁移指引。
- 方法重命名遗漏调用 (correctness): PR 中已更新 expert_service.py 和测试文件，但可能仍有遗漏调用点，风险部分缓解。
- 测试并发逻辑缺陷 (correctness): 未在 PR 中修改，测试逻辑可能存在阻塞问题，影响准确性。
- 测试覆盖不足 (testing): 仅测试 RDMA 协议，其他协议未覆盖，测试覆盖不全面。

风险与影响

- 风险：技术风险包括：1. 方法重命名风险：init_cache_info 改为 init_pd_info 后，如果代码库中仍有未更新的调用点，会导致 AttributeError，影响运行时稳定性。2. 并发逻辑缺陷：测试中 _send_pd_request 的并发实现可能阻塞，未模拟 Router 的 fan-out 行为，可能导致测试不准确或资源泄露。3. 兼容性风险：禁止 scheduler_name=="splitwise" 可能破坏现有 v0 部署，需评估影响并提供迁移路径。4. 测试覆盖不足：仅测试 RDMA 协议，其他传输协议未覆盖，可能隐藏无路由器场景下的潜在问题。
- 影响：对用户的影响：PD 分离部署现在可以不依赖路由器运行，降低了部署复杂度，尤其适用于 RL 等特定场景，提高了部署灵活性。对系统的影响：调度器和配置逻辑更松耦合，但需确保无路由器时的正确性和性能，可能增加配置复杂性。对团队的影响：需要更新相关文档和示例，并注意向后兼容性，团队在维护时需关注方法重命名和测试覆盖。
- 风险标记：方法重命名遗漏调用，测试并发阻塞，兼容性风险，测试覆盖不足

关联脉络

- PR #7364 [BugFix][PD Disaggregation][KVCache] Fix low cache hit rate in PD split (disaggregation) scenario: 同属 PD 分离部署场景，涉及调度器和缓存管理，与本 PR 的部署配置相关。
- PR #7407 [BugFix][Scheduler]Fix FD_DISABLE_CHUNKED_PREFILL max_num_batched_tokens limit: 涉及调度器配置调整，与本 PR 的参数校验和调度器逻辑修改有间接关联。