PR #26576 完整报告

sgl-project/sglang

[EPD] feat: encoder DP mode with per-rank subprocess workers

合并时间：2026-06-04 12:37

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26576

执行摘要

- 一句话：编码器服务器新增 per-rank 子进程数据并行模式
- 推荐动作：建议精读。该 PR 引入了复杂的进程间通信和生命周期管理，设计决策（如 ZMQ IPC、worker watchdog、背压控制）值得学习。对于生产部署，建议添加集成测试覆盖故障场景。

功能与动机

当前解耦编码器服务器在同一进程中运行单个 MMEncoder 和 FastAPI 应用，GPU 工作阻塞事件循环，且每个实例只能利用一个 GPU。此 PR 通过添加 per-rank 子进程数据并行模式解决该问题，使得路由进程保持空闲，GPU 工作并行化。

实现拆解

1. 环境变量：environ.py 新增 SGLANG_ENCODER_DP_WORKER_MAX_INFLIGHT（默认 64），控制每个 worker 的最大并发请求数。
2. DPDispatcher 类：encode_server.py 新增路由中心，维护 pending_counts、alive_ranks、_dead_ranks，暴露 dispatch、dispatch_send、broadcast 方法，采用 least-pending + round-robin 选择 rank。
3. Worker 进程启动：_launch_server_dp 函数检测 dp_size > 1，通过 multiprocessing.Process 启动子进程，使用 maybe_reindex_device_id 隔离 GPU。
4. 子进程运行逻辑：run_dp_worker 内含独立 MMEncoder 和 EncoderScheduler，通过 ZMQ PUSH/PULL 与主进程通信，使用 asyncio.Semaphore 限流。
5. 路由适配：修改 /encode、/send、/health_generate、/start_profile、/stop_profile 等端点，在 DP 活跃时委托给 DPDispatcher，正确传递 HTTP 状态码。
6. 健康检查与错误处理：ResultListener 循环监听 worker 结果；watchdog 检测进程退出并标记死 rank；超时机制避免请求无限等待。 注意：本次 PR 没有包含新增的测试文件。

关键文件：

- python/sglang/srt/disaggregation/encode_server.py（模块 编码器；类别 source；类型 core-logic；符号 _launch_server_dp, DPDispatcher, run_dp_worker, _dp_worker_encode_and_send）：实现编码器数据并行模式的核心文件，包含 DPDispatcher、worker 启动、请求路由等全部逻辑

- python/sglang/srt/environ.py（模块 配置项；类别 config；类型 configuration）：添加环境变量 SGLANG_ENCODER_DP_WORKER_MAX_INFLIGHT，控制 DP worker 最大并发请求数

关键符号：DPDispatcher, DPDispatcher.init, DPDispatcher.dispatch, DPDispatcher.dispatch_send, DPDispatcher.broadcast, _launch_server_dp, run_dp_worker, _dp_worker_encode_and_send, _dp_worker_health_encode, _push_embedding_to_prefill

关键源码片段

python/sglang/srt/disaggregation/encode_server.py

实现编码器数据并行模式的核心文件，包含 DPDispatcher、worker 启动、请求路由等全部逻辑

```
async def _push_embedding_to_prefill(enc: MMEncoder, request: dict) -> None:
    """
    将编码后的 embedding 推送到 prefill 阶段。
    对于 mooncake 后端为 no-op（transfer 由独立的 /send 处理）。
    embedding_port=None 会在上游被拒绝，因此此处 ports 必然存在。
    """
    req_id = request["req_id"]
    backend = enc.server_args.encoder_transfer_backend

    # zmq_to_tokenizer：单个 embedding_port，直接 send 后清理
    if backend == "zmq_to_tokenizer":
        await enc.send(
            req_id=req_id,
            prefill_host=request["prefill_host"],
            embedding_port=request["embedding_port"],
        )
        enc.embedding_to_send.pop(req_id, None)
        return

    # zmq_to_scheduler：多个 embedding_port（list），并发 send
    if backend == "zmq_to_scheduler":
        ports = request["embedding_port"]
        assert isinstance(ports, list)
        await asyncio.gather(
            *(
                enc.send(
                    req_id=req_id,
                    prefill_host=request["prefill_host"],
                    embedding_port=p,
                )
                for p in ports
            )
        )
        enc.embedding_to_send.pop(req_id, None)
```

评论区精华

- 僵尸进程风险：gemini-code-assist 指出 _result_listener 在 30 次连续错误后退出会导致永久僵尸状态，建议改为无限重试加退避。作者已修复。
- GPU 隔离方式：ShangmingCai 认为通过覆盖 CUDA_VISIBLE_DEVICES 环境变量危险。作者改用 maybe_reindex_device_id(gpu_id) 上下文管理器，与引擎其他组件一致。
- 健康检查严格性：ShangmingCai 和 ZhengWG 推动健康检查应反映 worker 存活状态。最终实现 all_ranks_alive 检查，任何死 rank 返回 503。
- TP+DP 支持：ZhengWG 询问能否同时使用 TP>1 和 DP。作者解释当前限制，计划后续 PR 支持。
  - _result_listener 退出条件导致僵尸状态 (correctness): 作者移除退出条件，并添加退避重试（commit 可见）。
  - CUDA_VISIBLE_DEVICES 覆盖危险 (design): 作者改用 maybe_reindex_device_id(gpu_id) 上下文管理器，保持父进程设置，与引擎现有模式一致。
  - 健康检查逻辑改进 (correctness): 作者实现 all_ranks_alive 检查，任何死 rank 返回 503。
  - TP+DP 支持 (design): 暂不支持 tp_size>1，作者将作为后续 PR。

风险与影响

- 风险：具体风险：
  - 子进程泄漏：主进程异常退出可能留下孤儿进程，虽注册 atexit 但无法覆盖 SIGKILL。建议配合进程管理器。
  - ZMQ 通信可靠性：worker 崩溃导致请求超时；ResultListener 退避重试减轻瞬态故障。
  - 全局状态一致性：dp_dispatcher 是模块级变量，非线程安全，但当前单线程事件循环风险低。
  - 配置冲突：dp_size > 1 要求 tp_size == 1，不符合时直接报错，用户需明确感知。
  - 影响：对用户：启用 --dp-size > 1 后，编码器吞吐量可随 GPU 数量线性扩展，但对小型模型可能增加延迟。 对系统：增加子进程数量，消耗额外 GPU 内存和 CPU 资源。 对团队：维护风险和复杂性增加，但功能齐全，为后续 TP+DP 打好基础。
- 风险标记：子进程管理风险，ZMQ 通信健壮性，全局状态一致性，配置冲突风险

关联脉络

- 暂无明显关联 PR