PR #1949 完整报告

THUDM/slime

[docker] fix sglang pd prefill abort request

合并时间：2026-05-26 21:56

原文链接：http://prhub.com.cn/THUDM/slime/pull/1949

执行摘要

- 一句话：修复 sglang PD prefill 超时 abort 与 KV 同步失败处理
- 推荐动作：该 PR 是关键的健壮性修复，建议合并。对于生产环境，建议根据实际网络状况调整超时配置。

功能与动机

在分解式（disaggregated）推理架构中，当预填充阶段出现超时或 KV 传输失败时，系统应能优雅地 abort 请求，避免继续处理无效请求导致状态不一致。此 PR 旨在增强系统的健壮性。

实现拆解

1. 在 prefill.py 的 PrefillBootstrapQueue.process_disagg_prefill_inflight_queue 中：
   - 引入 bootstrap_timeout 环境变量（默认 600s），在轮询请求状态时检测超时，超时则构造错误消息并调用 prepare_abort 和 stream_output，将请求从队列中移除。
   - 在 PrefillBootstrapQueue 中新增 release_memory_occupation 和 resume_memory_occupation 方法，用于在暂停引擎时释放和重新注册 KV 缓存。
   - 在 prefill 侧的 fill_batch 方法中，在遍历请求时提前检查 req.finished()，若已 abort 则立即释放 KV 缓存并跳过后续处理。
2. 在 prefill.py 的 KV 传输逻辑中：
   - 在发送 extra state chunk 后检查返回值，若 ret != 0 则记录失败并同步状态至 decode 端。
3. 在 sglang.patch 中：
   - 对应更新 sglang 源码中的 _pause_engine 函数，确保在 pause/resume 过程中能正确处理 abort 请求。
4. 版本号更新：
   - docker/version.txt 从 nightly-dev-20260525a 更新为 nightly-dev-20260526a。

关键文件：

- docker/patch/latest/sglang.patch（模块 sglang；类别 source；类型 bugfix；符号 _pause_engine）：该 patch 包含对 sglang 源码的多个关键修改：超时 abort、KV 传输失败处理、_pause_engine 增强，是本次 PR 的核心变更。

- docker/version.txt（模块 Docker；类别 infra；类型 configuration）：Docker 版本标识更新，配套 patch 变更。

关键符号：_pause_engine, process_disagg_prefill_inflight_queue, fill_batch, release_memory_occupation, resume_memory_occupation

关键源码片段

docker/patch/latest/sglang.patch

该 patch 包含对 sglang 源码的多个关键修改：超时 abort、KV 传输失败处理、_pause_engine 增强，是本次 PR 的核心变更。

```
defprocess_disagg_prefill_inflight_queue(self, rids_to_check=None):     # 设置 bootstrap 超时时间，可通过环境变量 SGLANG_DISAGGREGATION_TRANSFER_TIMEOUT 配置，默认 600 秒     bootstrap_timeout = float(
os.environ.get("SGLANG_DISAGGREGATION_TRANSFER_TIMEOUT", "600")     )     now = time.perf_counter()     for i, (req, poll) in enumerate(zip(self.queue, polls)):         if rids_to_check is not None and req.rid not in rids_to_check:             continue         if poll == KVPoll.WaitingForInput:             # 检测超时，若当前时间与启动时间之差超过超时值，则触发 abort             if now - req.disagg_kv_sender.start_time > bootstrap_timeout:                 error_message = (                     f"Prefill bootstrap timeout (>{bootstrap_timeout}s) "                     f"for request rank={self.tp_rank}"             f"{req.rid=}{req.bootstrap_room=}"                 )
logger.error(error_message)                 prepare_abort(req, error_message, status_code=HTTPStatus.GATEWAY_TIMEOUT)
self.scheduler.stream_output([req], req.return_logprob)
indices_to_remove.add(i)                 failed_reqs.append(req)                 if self.scheduler.enable_metrics:
self.scheduler.metrics_collector.increment_bootstrap_failed_reqs()             continue     elif poll == KVPoll.Failed:             # 处理 bootstrap 失败的情况             ...  （注：以上为简化示意，完整代码见 patch 文件）
```

评论区精华

无 review 评论。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 超时时间硬编码风险：SGLANG_DISAGGREGATION_TRANSFER_TIMEOUT 默认 600s，若网络较慢可能误 abort。
  2. 中断循环风险：在 process_disagg_prefill_inflight_queue 中，超时 abort 后 continue 跳过后续处理，但可能遗漏其他状态的请求处理流程。
  3. 状态覆盖风险：已 abort 的请求仍可能被后续逻辑误判为成功。

- 影响:
  - 用户影响：增强预填充阶段异常处理的可靠性，减少因超时或传输失败导致的用户无响应问题。
  - 系统稳定性：改善分解式架构在异常条件下的稳定性。
  - 维护影响：patch 文件变更与上游 sglang 版本同步，需在每次更新 sglang 时验证。
  - 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- 暂无明显关联 PR