PR #26780 完整报告

sgl-project/sglang

[PD] Optimistic prefill

合并时间：2026-06-02 16:16

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26780

执行摘要

- 一句话：PD 分解中乐观预填充，减少 TTFT
- 推荐动作：该 PR 值得精读，特别是重叠调度和状态管理的设计。建议关注 metadata buffer 分配策略和重试回退路径。对于使用 PD 分解的团队，建议评估此优化并配置合适的重试次数。

功能与动机

在 PD 分解中，预填充当前需要等待 bootstrap 完成才能开始计算，该等待直接增加 TTFT，包括调度器间到达时间偏差、解码侧分配延迟和 bootstrap 握手开销。本 PR 通过乐观预填充允许预填充提前开始，如果 bootstrap 在预填充完成前就绪则正常进行，否则重试或回退。（引自 PR body）

实现拆解

1. 核心重试逻辑（prefill.py）：新增 should_force_retry（基于 SHA-256 哈希采样）和 maybe_release_metadata_buffer 函数，管理重试触发和元数据缓冲区释放。
2. 元数据缓冲区管理（prefill.py）：将 add 拆分为 create_sender（仅创建发送器）、ensure_metadata_buffer（分配缓冲区索引）、finalize_bootstrap（bootstrap 完成后初始化发送器），支持先创建后分配。
3. 调度器适配（schedule_policy.py、scheduler.py）：在调度 chunked prefill 前轮询 bootstrap 状态，若未就绪则释放 KV 缓存并将请求重新排到等待队列头部（保留元数据缓冲区）。
4. 请求时间统计重构（req_time_stats.py）：移除 alloc_wait_duration，新增 prefill_retry_count 字段和 reset_prefill_retry_time 方法，重试时重置相关时间戳。
5. 缓存命中率修正（metrics_reporter.py）：增加 reprocessed_log_input_tokens 和 reprocessed_log_hit_tokens，从缓存命中率计算中扣除重试导致的重复 token。
6. 配置与校验（server_args.py）：添加 --optimistic-prefill-retries 参数，并禁用不兼容特性（pipeline parallelism、HiCache、Mamba radix cache）。
7. 测试覆盖：新增 test_disaggregation_optimistic_prefill.py（202 行），包含强制重试采样工具、重试计数器断言、GSM8K 准确率测试、logprob 测试和故障注入测试。

关键文件：

- python/sglang/srt/disaggregation/prefill.py（模块 预填充；类别 source；类型 core-logic；符号 should_force_retry, maybe_release_metadata_buffer, create_sender, ensure_metadata_buffer）：核心变更文件，实现乐观预填充的主要逻辑：强制重试检测、元数据缓冲区管理、bootstrap 最终化等。
- test/registered/disaggregation/test_disaggregation_optimistic_prefill.py（模块 测试；类别 test；类型 test-coverage；符号 rid_that_forces_retry, OptimisticPrefillRetryCounterMixin, assert_retry_counter_increases, TestOptimisticPrefill）：新增的集成测试文件，覆盖重试采样、准确率、logprob、故障注入等场景。
- python/sglang/srt/disaggregation/utils.py（模块 PD 工具；类别 source；类型 refactor；符号 _get_failure_prob, _poll_with_failure_injection, is_aborted）：重构故障注入函数，支持运行时配置失败概率，新增 is_aborted 辅助函数。
- python/sglang/srt/observability/req_time_stats.py（模块 可观测性；类别 source；类型 core-logic；符号 new_from_obj, reset_prefill_retry_time）：更新请求时间统计，添加重试相关字段和方法，移除 alloc_wait_duration。
- python/sglang/srt/server_args.py（模块 配置；类别 source；类型 configuration）：新增 --optimistic-prefill-retries 参数及其兼容性校验逻辑。
- python/sglang/srt/managers/scheduler_components/metrics_reporter.py（模块 指标；类别 source；类型 core-logic）：调整缓存命中率计算，扣除重试导致的重复统计，增加 reprocessed_log_input_tokens 等字段。

关键符号：should_force_retry, maybe_release_metadata_buffer, ensure_metadata_buffer, finalize_bootstrap, reset_prefill_retry_time, _poll_with_failure_injection, is_aborted, advance_logprob_pt

关键源码片段

python/sglang/srt/disaggregation/prefill.py

核心变更文件，实现乐观预填充的主要逻辑：强制重试检测、元数据缓冲区管理、bootstrap 最终化等。

```
import hashlib
from sglang.srt.managers.schedule_batch import Req
from sglang.srt.environ import envs

def should_force_retry(req: Req) -> bool:
    # Test hook to force a request into optimistic prefill retry.
    retry_prob = envs.SGLANG_TEST_FORCE_OPTIMISTIC_PREFILL_RETRY_PROB.get()
    # 如果 重试概率 <= 0 或请求已经重试过或被撤销，则不强制重试
    if retry_prob <= 0 or req.time_stats.prefill_retry_count > 0 or req.is_retracted:
        return False
    # 使用 SHA-256 哈希请求 ID 来实现确定性采样
    digest = hashlib.sha256(str(req.rid).encode()).digest()
    return int.from_bytes(digest[:8], 'big') < retry_prob * 2**64

def maybe_release_metadata_buffer(
```

```
    req: Req, allocator: ReqToMetadataIdxAllocator
) -> None:
    # 释放请求的元数据缓冲区索引，如果已分配
    if req.metadata_buffer_index >= 0:
        allocator.free(req.metadata_buffer_index)
        req.metadata_buffer_index = -1
```

评论区精华

Review 中核心讨论点包括：

- Metadata buffer 耗尽死锁风险：gemini-code-assist[bot] 指出 pop_bootstrapped 中使用 break 当缓冲区耗尽时可能导致 head-of-line blocking，建议改为 continue。作者采纳。
- 测试用例静默通过：bot 发现 test_survive_requests 未递增 successes 且未处理 HTTP 500，但作者澄清所有请求预期失败，无需 assertion。ShangmingCai 认可。
- 重试前 KV 缓存释放：ShangmingCai 询问是否需要显式释放 KV 缓存防止泄漏，作者解释 finalize_bootstrap 在此时不应失败，并添加断言。
- 文档补充：ShangmingCai 建议增加文档，作者计划后续默认启用时更新。
  - Metadata buffer 耗尽导致死锁风险 (correctness): 作者接受建议，将 break 改为 continue，避免死锁。
  - 测试用例静默通过问题 (testing): 作者回应所有请求预期失败，无需 assertion。ShangmingCai 随后认可。
  - 重试前 KV 缓存释放确认 (correctness): 作者回复 finalize_bootstrap 不应失败，已添加断言确认并注释说明。
  - 文档补充建议 (documentation): 作者同意计划后续默认启用时更新文档。

风险与影响

- 风险：
  - Metadata buffer 耗尽：尽管已改为 continue 避免死锁，但若所有缓冲区都被占用，后续请求仍需等待，可能影响吞吐。
  - 与高级特性不兼容：乐观预填充与 HiCache、Mamba radix cache、pipeline parallelism 不兼容，虽然在配置校验中禁用，但用户可能期望这些特性同时启用，导致配置混淆。
  - 时间统计变更影响监控：移除 alloc_wait_duration 可能使依赖于该指标的监控 dashboard 失效。
  - 重试逻辑增加状态复杂度：引入 pending_bootstrap、prefill_retry_count 等状态，可能与其他调度特性（如 retract）交互导致意外行为。
- 影响：
  - 对用户：启用后 TTFT 显著降低（bench 显示 P99 降低 50%），但需注意与 HiCache、Mamba 等特性互斥。
  - 对系统：增加了调度重试路径和元数据缓冲区占用量，但通过重试预算限制开销。
  - 对团队：新增核心调度逻辑，需要维护测试稳定性和兼容性矩阵。

- 风险标记：metadata buffer 耗尽风险，与 HiCache/Mamba/PP 不兼容，重试逻辑增加调度复杂度

关联脉络

- PR #26227 [PD]: Support HiCache prefetching and pd-incremental transfer on decode side: 同为 PD 分解路径优化，但乐观预填充当前不兼容 HiCache，需要理解其预取机制以避免冲突。