PR #27506 完整报告

sgl-project/sglang

Add more testing for chunked prefill

合并时间：2026-06-09 20:19

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27506

执行摘要

- 一句话：为 chunked prefill 添加系统性脚本化测试套件
- 推荐动作：建议调度器维护者精读 test/registered/chunked_prefill/ 中的脚本化测试，理解框架的断言模式；对于其他模块开发者，scripted-runtime 框架可以推广到其他子系统的细粒度测试。该 PR 的测试设计（精确步进 + 状态断言）值得作为项目测试标准。

功能与动机

PR body 指出：'Adds the scripted-runtime test harness (one forward per script yield, precise per-step engine-state assertions over an HTTP-server + scheduler-hook driven engine) and a chunked-prefill test suite, plus kv-canary PP fixtures and the registered scripted-runtime/chunked CI tests.' 这些测试覆盖了先前缺乏细粒度验证的 chunked prefill 调度路径，包括中止、radix 缓存交互、KV 压力场景和 PP 并行等。

实现拆解

1. scripted-runtime 测试框架：在 python/sglang/test/scripted_runtime/ 下构建测试框架，核心包括 ScriptedContext（通过 HTTP 与引擎交互，逐 yield 驱动调度循环）、ScriptedReqHandle（封装请求状态）和 ScriptedTestCase（测试基类）。框架通过 ScriptedSchedulerHook 注入调度器的 on_run_batch 回调，记录每步的 batch 信息（rid、mode、chunks 等），供测试断言。
2. 手动脚本化测试套件：在 test/manual/chunked_prefill/ 中添加 8 个测试模块，涵盖 chunked prefill 的核心场景：test_scripted_abort.py（中止与资源释放）、test_scripted_radix.py（radix 缓存命中与驱逐）、test_scripted_kv_pressure.py（KV 压力与恢复）、test_scripted_multi_req.py（多请求并发）、test_scripted_regression.py（回归场景）、test_scripted_invariants.py（引擎不变量）、test_scripted_chunk_size.py（分片尺寸边界）、test_scripted_lifecycle.py（请求生命周期）等。每个测试类继承 ScriptedTestCase，静态脚本方法通过 yield 驱动引擎一步，并在每一步后对请求状态（kv_pages、chunks_done、status 等）进行精确断言。
3. 端到端测试：在 test/manual/chunked_prefill/ 中添加 test_e2e_*.py，使用完整的 HTTP 服务器 + GSM8K 评估，验证 chunked prefill 在真实推理管道中的准确性和 KV leak 安全性。同时注册了 11 个 GPU 端到端测试到 CI。
4. KV-canary PP 夹具：添加 kv_canary 的 pipeline parallelism 测试夹具，在 PP 配置下验证 SWA 模型和常规模型的 KV 输出完整性。

5. 配套修复与调整：在构建测试过程中发现并修复了若干问题：移除断言 num_examples > available test lines（#27502）、修正 chunks_done 计数逻辑、调整 flush_cache 行为、修复 kv_pages 报告规则等；移除 PP cross-mb in-flight exclusion 死代码。

关键文件：

- test/manual/chunked_prefill/test_scripted_abort.py（模块 中止测试；类别 test；类型 test-coverage；符号 _drain_until_released, TestAbortBasic, test_abort_waiting_chunked_resume, _script_abort_waiting_chunked_resume）：展示了 scripted-runtime 测试的核心模式：每步 yield 驱动调度循环，精确断言 KV/lock 释放状态。测试了中止 chunked 请求的多种边界（刚分片、分片中、零 yield 后等）。
- test/manual/chunked_prefill/test_scripted_special_case.py（模块 特殊场景测试；类别 test；类型 test-coverage；符号 _load_inquirer_pending_for_rid, TestSpecialCaseBasic, test_chunked_in_flight_no_idle, _script_chunked_in_flight_no_idle）：覆盖 chunked prefill 的特殊路径：in-flight 不被 idle、admission 与 chunked 并发、abort 排除 chunked 等。1253 行测试，是最大的手动测试文件。
- test/manual/chunked_prefill/test_scripted_kv_pressure.py（模块 KV 压力测试；类别 test；类型 test-coverage；符号 TestKVPressureBasic, test_lock_refs_tight_concurrent_prefix, _script_lock_refs_tight_concurrent_prefix, test_kv_pressure_with_retract_resume）：测试 KV 压力下的 chunked prefill 行为：lock refs 竞争、retract 恢复、chunked batch 后池状态恢复。验证极端条件下内存是否泄漏。
- test/manual/chunked_prefill/test_scripted_chunk_size.py（模块 分片大小测试；类别 test；类型 test-coverage；符号 _expected_chunks, TestChunkSizeDefault, test_exact_chunk_size, _script_exact_chunk_size）：测试 chunks_done 计数精度和分片边界条件，包括刚好等于 chunk size、超出 1 个 token、N 个 chunks 等。是验证分片逻辑正确性的基础测试。
- test/manual/chunked_prefill/test_scripted_radix.py（模块 Radix 测试；类别 test；类型 test-coverage；符号 TestRadixBasic, test_radix_full_prefix_hit_nine_reqs, _script_radix_full_prefix_hit_nine_reqs, test_radix_hit_full_prefix）：验证 chunked prefill 与 radix 缓存交互：完整前缀命中、部分命中后分片、驱逐后重新分片、resume 路径等。

关键符号：_drain_until_released, _expected_chunks, ScriptedTestCase, ScriptedContext.start_req, ScriptedContext.abort, ScriptedContext.pause_generation, ScriptedContext.continue_generation, ScriptedContext.flush_cache, ScriptedContext.engine_stats, ScriptedContext.batch_composition, ScriptedContext.list_active_reqs, ScriptedReqHandle.is_chunking, ScriptedReqHandle.chunks_done, ScriptedReqHandle.kv_pages, ScriptedReqHandle.lock_refs

关键源码片段

test/manual/chunked_prefill/test_scripted_abort.py

展示了 scripted-runtime 测试的核心模式：每步 yield 驱动调度循环，精确断言 KV/lock 释放状态。测试了中止 chunked 请求的多种边界（刚分片、分片中、零 yield 后等）。

```
import unittest

from sglang.test.scripted_runtime.context import ScriptedContext
from sglang.test.scripted_runtime.req_handle import ScriptedReqHandle
from sglang.test.scripted_runtime.test_case import ScriptedTestCase
from sglang.test.scripted_runtime_chunked_helpers import (
    DEFAULT_CHUNK_SIZE, DEFAULT_MAX_STEPS,
    VERY_LONG_PROMPT_LEN, base_engine_kwargs,
    run_until, run_until_finished,
)


def _drain_until_released(t: ScriptedContext, *handles: ScriptedReqHandle):
    # 轮询最多 12 步，直到所有 handle 释放 KV pages 和 lock_refs
    for _ in range(12):
        if all(
            h.kv_pages == 0 and h.lock_refs == 0
            and (h.req is None or h.req.req_pool_idx is None)
            for h in handles
        ):
            return
        yield


class TestAbortBasic(ScriptedTestCase):
    ENGINE_KWARGS = base_engine_kwargs(chunked_prefill_size=DEFAULT_CHUNK_SIZE)

    def test_abort_waiting_chunked_resume(self):
        self.server.execute_script(self._script_abort_waiting_chunked_resume)

    @staticmethod
    def _script_abort_waiting_chunked_resume(t: ScriptedContext):
        # 启动一个足够长的请求，使其进入 chunked prefill 状态
        r = t.start_req(
            prompt_len=VERY_LONG_PROMPT_LEN, max_new_tokens=2, prompt_token=100
        )
        yield from run_until(r, lambda h: h.is_chunking)

        pages_before = r.kv_pages
        assert pages_before > 0, "chunked req 在分片过程中应持有 KV pages"

        # 发送 abort，然后等待资源完全释放
        t.abort(r)
        yield from _drain_until_released(t, r)

        assert r.kv_pages == 0, "abort 必须释放所有 KV pages"
```

```
assert r.req is None or r.req.req_pool_idx is None, "abort 必须释放 req 行"
assert r.lock_refs == 0, "abort 必须释放 lock refs"
```

评论区精华

- assert num_examples 导致大面积 CI 失败：作者在 #27502 中添加的硬断言 num_examples > available test lines 因上游通过切片静默处理而触发多个注册测试失败。经讨论删除该断言，保留切片行为。
- PP cross-mb in-flight exclusion 是否为死代码：作者从另一个分支 cherry-pick 了 _in_flight_other_mb_rids 修复，后在 main 上确认该代码是 dead defensive，因 stateless-scheduler 分支的重写确实引起 PP+chunked KV 损坏但 main 无此问题。最终删除该代码及其测试。
- test_mimo_v2 超时预先存在：作者通过跨分支 rerun 确认 main 上也超时，推断为大型模型下载超时的基础设施波动。后续合并 #27512 后通过。
  - assert num_examples 导致 CI 大面积失败 (correctness): 删除硬断言，保留切片行为。
  - PP cross-mb in-flight exclusion 是否为死代码 (design): 删除 _in_flight_other_mb_rids 代码及其测试。
  - test_mimo_v2 超时是预先存在的 (question): 确认为预先存在的超时，非 PR 引入。

风险与影响

- 风险：测试框架通过 scheduler hook 注入，不改变生产路径，无性能影响。大量新增 CI 测试（约 100 个测试用例）会增加运行时间，估算约 10-15 分钟。框架依赖调度器内部 hook（on_run_batch）和请求对象状态，若调度器接口变化需同步更新测试。测试中使用细粒度断言（如 kv_pages == 0），可能因引擎行为微调而脆弱，但这也是测试的价值所在。整体风险低。
- 影响：对用户无直接影响；对开发者，scripted-runtime 测试框架提供了可复用的测试方法论，可替代部分手动调试，提高 chunked prefill 相关修改的回归拦截能力；对 CI，新增约 100 个测试用例，部分需要多 GPU 资源（PP 测试），可能影响 CI 排队时间。团队应鼓励在调度器修改时运行此套件。
- 风险标记：大量新 CI 测试增加运行时间，依赖调度器内部 hook，测试框架维护成本

关联脉络

- PR #27502 Add mixed-prefix gsm8k eval and its CPU unit test: 本 PR 的基础，引入了混合前缀评估，为 chunked prefill 端到端测试提供评测基础。
- PR #26991 [Scripted Runtime] Extracted chain from scripted_runtime_and_chunked_testing: 包含了 scripted-runtime 框架的早期提取，本 PR 在此基础上进一步发展并合并。
- PR #27512 [Fix] Fix multimodal pad-sentinel clamp for MiMo-V2: 修复了 multimodal 模型的 pad-sentinel，本 PR CI 依赖此修复通过 test_mimo_v2 测试。
- PR #27528 [Fix] Fix FusedMoEWithLoRA hidden_size attribute: 修复了 LoRA 相关属性缺失，本 PR CI 依赖此修复通过 LoRA 测试。