# PR #34070 完整报告

sgl-project/sglang

[CI] Trim redundant nightly test registrations

合并时间：2026-08-08 16:42

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34070

## 执行摘要

- 一句话：裁剪 24 个冗余 nightly 测试注册，每夜节省约 5.5 GPU 小时
- 推荐动作：值得 CI owner 与测试维护者精读，其 "covered elsewhere / cannot run as registered / guard dead code" 的三分法可作为 nightly 测试治理的通用模板。对产品研发而言，需重点关注两处被删测试的替代覆盖：fa_skip_kv_cache piecewise NaN 回归测试与 chunked SGMV LoRA 内核测试；建议在其他可运行套件中补回等价断言，或在对应内核 / 后端发生变更时主动重建回归测试。整体变更方向正确，但删除回归测试时应在 PR 中明确指向替代覆盖的具体用例文件，便于后续审计。

## 功能与动机

PR body 明确提出修剪目标："Nightly registrations that are covered elsewhere, cannot run as registered, or guard code nobody changes. Freed roughly 5.5 GPU-hours per nightly run." 即 nightly 中存在大量重复覆盖、无法按注册拓扑运行、或守卫死代码的测试，造成 GPU 资源浪费。例如 test_eagle_infer_beta_dp_attention_large.py 声明 16 GPU 双节点拓扑却注册在单节点 8-GPU runner 上，两个 gb300 测试携带 disabled="not needed" 从不执行。

## 实现拆解

本 PR 不涉及产品源码，全部改动集中在 test/registered 下的 nightly 测试注册，按 PR body 的分类框架拆解如下：

1. 删除与既有套件重复覆盖的测试：
   - backends/test_qwen3_fp4_trtllm_gen_moe.py（nightly-1-gpu，300s）：PR body 指出 FlashinferTrtllmGenMoeBackendNVFP4Base 在 backends/test_flashinfer_trtllm_gen_moe_backend.py 中以 tp4/ep4 跑同一模型与 MoE 后端。
   - radix_cache/test_cpp_radix_cache.py（60s）与 lora/ 下 4 个测试：test_chunked_sgmv_backend.py（1261 行，最大删除文件）、test_embedding_lora_support.py、test_lora_radix_cache.py、test_lora_tied_lm_head.py。其中 test_chunked_sgmv_backend.py 覆盖 chunked SGMV LoRA 内核的 prefill/decode/verify 三种 batch 模式，并直接对比 Triton shrink/expand 内核与参考实现。

2. 删除无法按注册方式运行的测试：
   - spec/eagle/test_eagle_infer_beta_dp_attention_large.py（nightly-8-gpu-b200，600s）：头部声明 16 GPU 2 节点拓扑，但注册在单节点 8-GPU runner 上，永远无法正确执行。
   - gb300/test_kimi_k25.py、gb300/test_qwen35_nvfp4.py：携带 disabled="not needed"，从不执行。
3. 删除守卫长期不变代码的测试：
   - utils/test_request_logger.py（120s）：断言 --log-requests 输出格式，而 request_logger.py 自 2026-03 起无功能变更。
   - utils/test_scheduler_status_logger.py（120s）：守卫 scheduler status 日志字段。
   - bench_fn/test_bench_serving_functionality.py（300s）：断言请求计数与 header 透传，属单元测试范畴却起服务端。
4. 删除冗余的模型 / 配置覆盖：
   - quant/test_kimi_k25_nvfp4_eagle.py（3000s）、quant/test_kimi_k26_nvfp4_dflash.py（3600s）、backends/test_deepseek_r1_fp8_trtllm_backend.py（3600s）、perf/test_dpsk_v3_fp4_4gpu_perf.py（2000s）、perf/test_gpt_oss_4gpu_perf.py（600s）、models_e2e/test_qwen3_next_models_fp4.py（500s）、disaggregation/test_disaggregation_dwdp_mimo.py（900s）、cuda_graph/piecewise/test_pcg_glm52_fp4.py 与 test_pcg_glm52_fp8_tp8.py（各 900s）、prefill_only/test_fa_skip_kv_cache_piecewise_nan.py（600s）。
5. 保留文件但调整注册方式：
   - lora/test_lora_openai_api.py：纯 mock 单元测试（无 torch、不起服务），已在 per-commit CPU 上运行，nightly-1-gpu 注册被撤销，不再占 GPU runner。
   - utils/test_model_file_verifier.py：撤销 nightly-1-gpu 注册，保持 CPU-only；model_file_verifier.py 自 2026-01-08 未变更。
   - models_e2e/test_dsa_glm52_cache_layer_split.py：迁移到 base-c-test，并将模型路径由本地缓存绝对路径改为公共模型名 nvidia/GLM-5.2-NVFP4（见评论区）。
6. 协作与演进过程：25 个 commit 由 hnyls2002 与 Fridge003 共同完成，期间出现 revert some of the changes、restore kv canary benchmarks 等回退提交，说明 kv canary benchmark 曾先被删除后因缺少断言或价值存疑被恢复，最终保留，体现了对删除边界的反复斟酌。
7. 配套改动：无产品源码、schema、配置文件或部署改动；CI 相关仅剩 test/registered 注册方式的调整，属于纯测试配套变更。

关键文件：

- test/registered/lora/test_chunked_sgmv_backend.py（模块 LoRA 内核；类别 test；类型 deletion；符号 reset_kernel_cache, BatchComposition, BatchMode, TestChunkedSGMV）：本次删除的最大文件（1261 行），覆盖 chunked SGMV LoRA 内核在 prefill/decode/verify 三种 batch 模式下的 shrink/expand 正确性，直接对比 Triton 内核与参考实现；删除后 LoRA 内核级验证依赖其他测试覆盖。

- test/registered/utils/test_request_logger.py（模块 请求日志；类别 test；类型 deletion；符号 BaseTestRequestLogger, setUpClass, tearDownClass, _verify_logs）：守卫 --log-requests 输出格式的 e2e 测试，每次启动真实 SGLang 服务约 120s；PR body 论证 request_logger.py 自 2026-03 起无功能变更，删除可显著节省 1-GPU 套件时间。
- test/registered/lora/test_embedding_lora_support.py（模块 嵌入 LoRA；类别 test；类型 deletion；符号 TestEmbeddingLoraSupport, test_engine_encode_validates_enable_lora, test_embedding_lora_fields, TestEmbeddingLoraHFComparison）：验证 embedding 请求结构中 LoRA 字段的归一化、批量展开与 HF/SGLang 相似度对比；被归为 covered elsewhere 删除，但其中 EmbeddingReqInput 的 lora_id/lora_path 归一化断言是结构级校验。
- test/registered/lora/test_lora_tied_lm_head.py（模块 LoRA 绑定；类别 test；类型 deletion；符号 create_lora_adapter_with_lm_head, TestLoRATiedLMHead, setUpClass, tearDownClass）：专门验证 tie_word_embeddings=True 模型在 lm_head 上应用 LoRA 时 SGLang 与 HF+PEFT 的 logprob 一致性，是 tied lm_head 场景少有的针对性测试。
- test/registered/bench_fn/test_bench_serving_functionality.py（模块 压测功能；类别 test；类型 deletion；符号 TestBenchServingFunctionality, test_gsp_multi_turn, _verify_multi_turn_logs, TestBenchServingCustomHeaders）：bench serving 功能测试，断言多轮 GSP 请求计数与自定义 header 透传；PR body 认为该类断言属单元测试范畴，不值得起服务端验证。
- test/registered/prefill_only/test_fa_skip_kv_cache_piecewise_nan.py（模块 预填充回归；类别 test；类型 deletion；符号 _short_prompts, _embed, TestFaSkipKvCachePiecewiseNoNaN, setUp）：删除的测试中回归价值最高：守卫 #21971 fa_skip_kv_cache 嵌入快速路径在 piecewise CUDA graph 下短输入 embedding 全 NaN 的 bug；其注释明确该场景只在 SM80-90 上复现。删除后此类边界问题失去针对性防护。
- test/registered/utils/test_scheduler_status_logger.py（模块 调度日志；类别 test；类型 deletion；符号 TestSchedulerStatusLogger, setUpClass, test_scheduler_status_dump, _find_log_events）：断言 scheduler.status 日志事件的 timestamp/rank/running_rids/queued_rids 字段，守卫调度器状态日志输出。
- test/registered/disaggregation/test_disaggregation_dwdp_mimo.py（模块 解耦服务；类别 test；类型 deletion；符号 TestDisaggregationDWDPMiMo, setUpClass, start_prefill, start_decode）：覆盖 PD 分离下 DWDP prefill（4 GPU）+ DP-attention decode（4 GPU）组合的 GSM8K 评测；因冗余模型 / 配置覆盖被删，但该组合（dwdp + mimo + fa4）在夜间验证中较独特。
- test/registered/spec/eagle/test_eagle_infer_beta_dp_attention_large.py（模块 投机解码；类别 test；类型 deletion；符号 test_gsm8k, TestEagleDPAttnServerLarge, setUpClass, tearDownClass）：声明 16 GPU 双节点拓扑的 EAGLE DP-attention 大模型测试却注册在单节点 8-GPU runner 上，属于注册拓扑不匹配的典型案例，删除合理。
- test/registered/backends/test_deepseek_r1_fp8_trtllm_backend.py（模块 DeepSeek 后端；类别 test；类型 deletion；符号 TestDeepseekR1Fp8Flashinfer, setUpClass, tearDownClass, test_gsm8k）：DeepSeek-V3 fp8 + trtllm_mla + flashinfer_trtllm 的

8-GPU 长时（3600s）后端测试，被归为冗余模型 / 配置覆盖删除。

- test/registered/models_e2e/test_dsa_glm52_cache_layer_split.py（模块 解耦缓存；类别 test；类型 configuration；符号 TestGLM52DSACacheLayerSplit）：未删除但发生注册迁移与路径修复：从 nightly 迁到 base-c-test，并把模型路径改为公共模型名 nvidia/GLM-5.2-NVFP4；是唯一 review 评论的落点。
- test/registered/cuda_graph/piecewise/test_pcg_glm52_fp4.py（模块 CUDA 图；类别 test；类型 deletion；符号 TestPCGGlm52Fp4, setUpClass, tearDownClass, test_gsm8k）：GLM-5.2 fp4 piecewise CUDA graph nightly 测试（4-GPU，900s）被删，与 test_pcg_glm52_fp8_tp8.py 同属 piecewise 覆盖削减。

关键符号：reset_kernel_cache, _compare_shrink_outputs, generate_sequence_lengths, create_lora_weights, BaseTestRequestLogger.setUpClass, BaseTestRequestLogger._verify_logs, BaseTestRequestLogger._wait_until_verified, BaseTestRequestLogger.test_logging, BaseTestRequestLogger.test_openai_chat_logging, TestEmbeddingLoraHFComparison.get_hf_embedding_with_lora, TestEmbeddingLoraHFComparison.get_sglang_embedding_with_lora, create_lora_adapter_with_lm_head, TestLoRATiedLMHead.test_tied_lm_head_lora_hf_sgl_logprob_match, TestBenchServingFunctionality.test_gsp_multi_turn, TestBenchServingFunctionality._verify_multi_turn_logs, TestFaSkipKvCachePiecewiseNoNaN.test_no_nan_with_piecewise, TestFaSkipKvCachePiecewiseNoNaN.test_matches_non_piecewise, TestSchedulerStatusLogger.test_scheduler_status_dump, TestDisaggregationDWDPMiMo.start_prefill, TestDisaggregationDWDPMiMo.start_decode, TestEagleDPAttnServerLarge.test_a_gsm8k, TestDeepseekR1Fp8Flashinfer.setUpClass

## 关键源码片段

### test/registered/lora/test_chunked_sgmv_backend.py

本次删除的最大文件（1261 行），覆盖 chunked SGMV LoRA 内核在 prefill/decode/verify 三种 batch 模式下的 shrink/expand 正确性，直接对比 Triton 内核与参考实现；删除后 LoRA 内核级验证依赖其他测试覆盖。

```
# 本 PR (#34070) 删除了这个 1261 行的 LoRA chunked SGMV 内核测试文件。
# 以下是其核心对比逻辑：chunked shrink 内核只保证每个序列
# output[seq_start:seq_end, :rank * num_slices] 区域正确，因此只比较有效列。
def _compare_shrink_outputs(self, chunked_output, reference_output, seq_lengths,
                            lora_assignments, batch_info, num_slices, test_name):
    lora_ranks = batch_info.lora_ranks.cpu().numpy()
    token_offset = 0
    for seq_idx, (lora_idx, seq_len) in enumerate(zip(lora_assignments, seq_lengths)):
        if seq_len == 0:
            continue
        rank = lora_ranks[lora_idx]
        if rank > 0:
            # 只有有效 rank 列有数学保证，padding 部分不参与对比
```

```
            valid_cols = num_slices * rank
            chunked_seq = chunked_output[token_offset: token_offset + seq_len, :valid_cols]
            reference_seq = reference_output[token_offset: token_offset + seq_len, :valid_cols]
            torch.testing.assert_close(
                chunked_seq, reference_seq, rtol=self.RTOL, atol=self.ATOL,
                msg=f"Shrink 失败: {test_name}, sequence {seq_idx} ({lora_idx})",
            )
        token_offset += seq_len
```

## test/registered/utils/test_request_logger.py

守卫 --log-requests 输出格式的 e2e 测试，每次启动真实 SGLang 服务约 120s；PR body 论证 request_logger.py 自 2026-03 起无功能变更，删除可显著节省 1-GPU 套件时间。

```
# 本 PR (#34070) 删除该 e2e 测试。其代价是每次启动真实 SGLang server
# （Qwen3-0.6B, --log-requests）并轮询 stdout 与日志文件，约 120s。
class BaseTestRequestLogger:
    log_requests_format = None

    @classmethod
    def setUpClass(cls):
        cls.stdout = io.StringIO()
        other_args = [
            "--log-requests", "--log-requests-level", "2",
            "--log-requests-format", cls.log_requests_format,
            "--log-requests-target", "stdout", cls.temp_dir,
        ]
        cls.process = popen_launch_server(
            "Qwen/Qwen3-0.6B", DEFAULT_URL_FOR_TEST,
            timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
            other_args=other_args,
            return_stdout_stderr=(cls.stdout, cls.stderr),
        )

    def test_logging(self):
        # 断言 /generate 的 request.received / request.finished 事件里带 routing key
        response = requests.post(
            DEFAULT_URL_FOR_TEST + "/generate",
            json={"text": "Hello", "sampling_params": {"max_new_tokens": 8}},
            headers={"X-SMG-Routing-Key": "test-routing-key-12345"}, timeout=30,
        )
        self.assertEqual(response.status_code, 200)
        self._wait_until_verified(
            self._verify_logs,
            lambda: self.stdout.getvalue() + self.stderr.getvalue(),
            "stdout",
        )
```

## test/registered/prefill_only/test_fa_skip_kv_cache_piecewise_nan.py

删除的测试中回归价值最高：守卫 #21971 fa_skip_kv_cache 嵌入快速路径在 piecewise CUDA graph 下短输入 embedding 全 NaN 的 bug；其注释明确该场景只在 SM80-90 上复现。删除后此类边界问题失去针对性防护。

```
# 本 PR (#34070) 删除了该文件。它守卫 #21971 引入的 fa_skip_kv_cache 嵌入快速路径
# 在 piecewise CUDA graph 下的 NaN 回归：当 prefill 被 padding 到 bucket 边界时，
# flash_attn_varlen_func 的 q 行数大于 cu_seqlens_q[-1]，边界 query 块会污染
# 最后一个真实 token 的输出，而 embedding 模型正是取 LAST token 池化，导致约 40%
# 的短输入返回全 NaN。
import os
import torch

from sglang import Engine # 通过 Engine API 单请求逐个 encode，避免 batch 掩盖边界问题
from sglang.srt.utils import get_device_sm
from sglang.test.ci.ci_register import register_cuda_ci
from sglang.test.test_utils import CustomTestCase

# 该回归路径只在 Ampere/Ada/Hopper (SM 80-90) 上运行，故原注册落在 H100 nightly 池
_FA3_SM_MIN, _FA3_SM_MAX = 80, 90


def _short_prompts():
    """构造 1..149 token 的短输入，保证多个长度落在 bucket 边界 (80/96/112/128...) 之下。"""
    words = ["the", "quick", "brown", "fox", "jumps", "lazy", "dog", "token", "sample"]
    return [" ".join(words[i % len(words)] for i in range(n)) for n in range(1, 150)]


class TestFaSkipKvCachePiecewiseNoNaN(CustomTestCase):
    def setUp(self):
        # 运行时按 CUDA SM 门控：不支持 FA3 的 Blackwell runner 上 skipTest 而非失败
        sm = get_device_sm()
        if not (_FA3_SM_MIN <= sm <= _FA3_SM_MAX):
            self.skipTest(f"fa3 + piecewise embedding 需要 SM {_FA3_SM_MIN}-{_FA3_SM_MAX}")

    def test_no_nan_with_piecewise(self):
        # 开启 fa_skip_kv_cache 的条件：is_embedding + chunked_prefill_size == -1
        # + disable_radix_cache + 非 MLA 模型 + FA3 后端
        engine = Engine(
            model_path=os.environ.get("SGLANG_TEST_EMB_MODEL", "Qwen/Qwen3-Embedding-0.
            6B"),
            is_embedding=True,
            attention_backend="fa3",
            chunked_prefill_size=-1,
            disable_radix_cache=True,
            cuda_graph_backend_prefill="tc_piecewise",
            cuda_graph_max_bs_prefill=32768,
            cuda_graph_tc_compiler="inductor",
        )
        try:
```

```
            embs = [torch.tensor(engine.encode(p)["embedding"]) for p in _short_prompts()]
            nan_idx = [i for i, e in enumerate(embs) if torch.isnan(e).any()]
            self.assertEqual(nan_idx, [], f"{len(nan_idx)} 个短输入 embedding 含 NaN")
        finally:
            engine.shutdown()
```

## 评论区精华

仓库仅有 1 条 review 评论，来自 Fridge003：在 test_dsa_glm52_cache_layer_split.py 的 diff 上建议将模型路径从本地缓存绝对路径 /data/radixark/model-cache/hub/models--nvidia--GLM-5.2-NVFP4/snapshots/... 改为公共模型名 "nvidia/GLM-5.2-NVFP4"。

背景是该测试从 nightly 8-GPU 套件迁移到 base-c-test 后，必须使用所有 runner 均可访问的模型标识才能保证可移植性，否则会像被删的 gb300 测试一样在特定 runner 上无法运行。该建议已被采纳合入。Fridge003 最终给出 APPROVED。

- GLM-5.2 缓存层拆分测试改用公共模型名 (design): 建议已被采纳：最终提交使用 nvidia/GLM-5.2-NVFP4 公共模型 id，文件随注册迁移一起合入。

## 风险与影响

- 风险：
  - 回归保护流失（最值得关注）：prefill_only/test_fa_skip_kv_cache_piecewise_nan.py 被删除，而它守卫的是 #21971 引入的 fa_skip_kv_cache embedding 快速路径在 piecewise CUDA graph 下的 NaN 回归（约 40% 短输入 embedding 全 NaN），且文件注释明确该路径只在 Ampere/Ada/Hopper（SM 80-90）上可复现。删除后此类边界 bug 缺少针对性回归防护，其归类为 redundant model/config coverage 也略显牵强，该测试实为唯一针对此场景的回归测试。
  - 内核级覆盖依赖 "covered elsewhere" 假设：test_chunked_sgmv_backend.py（1261 行）直接对比 chunked SGMV shrink/expand Triton 内核与参考实现，覆盖 prefill/decode/verify 三种模式，并引用 chunked_embedding_lora_a、kv_b_lora_absorbed 等符号。若 PR body 中 " 已被 test_flashinfer_trtllm_gen_moe_backend.py 覆盖 " 的断言不完全成立，LoRA 内核级正确性验证会出现空洞。
  - 间接覆盖链脆弱：部分删除依赖 " 其他测试已覆盖 " 的假设，但这些假设未被自动化校验；若承担覆盖的测试日后被其他 PR 删除，会形成双重空洞。
  - 大模型场景夜间覆盖减少：EAGLE DP-attention 16-GPU 大模型测试因拓扑不匹配被删除后，该场景在 nightly 中不再有针对性验证；且同类测试（如 test_eagle_infer_beta_dp_attention_large.py 内部逻辑）本就依赖 GSM8K 准确率与 spec accept length 双重断言。
  - CI 波动风险：大幅削减后可能短期掩盖回归，建议观察 2-4 周 nightly 失败率，确认没有因删除而漏报。
- 影响：
  - 对 CI 基础设施：每夜节省约 5.5 GPU 小时，主要来自 1-GPU（约 1560s+）、4-GPU（约 6400s+）、8-GPU（约 8160s+）三类 nightly 套件；长期看每月可节省约 165

GPU 小时。

- 对回归覆盖：nightly 中 lora（4 个文件）、piecewise CUDA graph（3 个文件）、kimi quant、deepseek trtllm 后端、EAGLE DP-attention、disaggregation DWDP 等方向的夜间验证面显著缩小。
- 对团队：nightly 排队时间缩短、成本下降，测试维护者获得一套可复用的 " 注册是否值得 " 评估框架；产品研发需注意被删测试对应的功能日后变更时要主动补回覆盖。
- 对用户与系统：无任何运行时行为影响。
- 风险标记：测试覆盖缩减，回归保护缺失，CI 基础设施变更，多人多轮修改回退

# 关联脉络

- PR #34100 [Fix] Give the piecewise CUDA graph test stub an hf_config: 同属 piecewise CUDA graph 测试治理线：本 PR 删除了 test_pcg_glm52_fp4.py、test_pcg_glm52_fp8_tp8.py、test_fa_skip_kv_cache_piecewise_nan.py，而 #34100 在修复同一测试域（multimodal piecewise CUDA graph 测试桩）的 CI 失败，两者反映该区域测试正处于收紧与修复并行的状态。
- PR #32785 fix: avoid piecewise prefill graph for trtllm_mla: 涉及 piecewise prefill CUDA graph 的启用条件与回归覆盖，本 PR 删除的 fa_skip_kv_cache piecewise 回归测试正是该功能的守卫测试；后续若 trtllm_mla 等后端调整 piecewise 行为，需关注覆盖是否仍在。
- PR #34017 [Fix] Judge the phase-checker device-assert test by its FAIL line, not the exit code: 同属 nightly CI 稳定性治理：本 PR 削减 nightly 测试规模，#34017 修复测试失败判定逻辑，两者共同指向仓库近期对 nightly suite 成本与可靠性的系统性优化。