PR #35293 完整报告

sgl-project/sglang

test: switch the Inkling-Small NVFP4 deterministic suite to DSPARK

合并时间：2026-08-20 21:54

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35293

执行摘要

- 一句话：NVFP4 确定性测试切到 DSPARK，补齐 sconv 与 radix 缓存覆盖
- 推荐动作：值得精读。虽然只是测试文件，但它把三个系统性问题串起来了：EAGLE 禁用 prefill CUDA graph 导致的覆盖盲区、DSPARK 整块提交与 radix cache 分歧点 /checkpoint 对齐的微妙交互、以及 178 GB B200 与 284 GB GB300 之间的显存余量差异。对维护 radix cache 位精确性或 speculative decoding 相关测试的工程师，特别是理解 " 为什么某些测试永远跑不到目标代码 " 这一类问题，具有很高参考价值；建议顺带读 #34043 和 #28386 的改动来补齐上下文。

功能与动机

PR body 明确指出：TestUnifiedHybridMTPBitExact 是在 radix cache 位精确覆盖路线图（#34899）下新增的 spec 侧测试，但它无法触达目标逻辑——_is_mamba_track_enabled() 是 PrefillCudaGraphRunner 的方法，而 EAGLE 目标会整体禁用 prefill CUDA graph（#28386），runner 根本不会被构造。DSPARK 和 DFlash 不是 is_eagle()，保留 prefill graph，且正是 #34043 实际修复的算法（该 PR 测得 DFlash 在 SWEBench 子集上 pass rate 从 55% 跌到 35% 后修复回归到与无 spec 一致）。因此把确定性套件切换到 DSPARK 是覆盖 specdec 状态损坏回归的必要路径。

实现拆解

1. 定位覆盖盲区：分析 _is_mamba_track_enabled() 的 gate 位置，确认它挂在 PrefillCudaGraphRunner 上；EAGLE 目标直接禁用 prefill CUDA graph（#28386），导致 test_unified_radix_cache_kl_hybrid_bitexact.py 中的 MTP 类永远执行不到 mamba/sconv checkpoint 保存逻辑。
2. 切换确定性套件到 DSPARK：在 test/registered/models_e2e/test_inkling_small_nvfp4.py 中，将 TestInklingSmallNvfp4Deterministic 重命名为 TestInklingSmallNvfp4DsparkDeterministic，新增 _DSPARK_DRAFT_PATH 环境变量（默认 RadixArk/Inkling-Small-DSpark），并在 server 启动参数中追加 --speculative-algorithm DSPARK、--speculative-draft-model-path、--speculative-draft-attention-backend fa4，同时保留 modelopt_fp4 和 flashinfer_trtllm fp4 GEMM/MoE 路径，FP4 回归覆盖不丢失。
3. 调整资源与时间预算：--mem-fraction-static 从 0.85 降到 0.80——首次 CI 在 B200（178 GB）上中途 OOM，draft 权重加 speculative CUDA graph（private pools 974.89

MiB）把剩余显存压到 0.40 GB；GB300 单卡 284 GB 有 60% 余量所以测不出来。0.80 下 KV pool 仍报告约 6.2M tokens，对该文件约 130k 的用量绰绰有余。est_time 从1700 提升到 2000，覆盖 draft 加载和 graph capture 的开销。

4. 放宽 decode-cache 命中率门槛：min_cache_hit_ratio 从 0.99 降到 0.90。原因是 DSPARK 最后一次 verify 会整块提交，请求可能超出 max_new_tokens 最多 chain_len-1 个 token（实测 0–7，block_size=7）；这些 token 进入 KV pool 和 radix insert，却不包含在返回的 output_ids 中，后续轮次在 prompt + returned 处与缓存序列分歧。当唯一的 mamba checkpoint（floor(committed/interval)*interval）落在分歧点之后，整个 decode 区域不可复用，B200 上约 1/32 的 prompt 会天然 miss。
5. 验证方法（revert-then-red）：分别用 bf16 和 NVFP4 checkpoint 在 GB300 上回退 #34043 验证——bf16 prefill_cache_hit 读 8.05e-02，NVFP4 读 1.10e-01 且 decode 侧出现 Too few decode cache hits: 31/32；无 spec 对照组全部为 0，证明信号是 speculative-state 特有。修复在位时整文件 5 passed（1093 s）。

关键文件：

- test/registered/models_e2e/test_inkling_small_nvfp4.py（模块 确定性测试；类别 test；类型 test-coverage；符号 TestInklingSmallNvfp4Deterministic, TestInklingSmallNvfp4DsparkDeterministic）：唯一变更文件。将 Inkling-Small NVFP4 确定性套件从无推测模式切换到 DSPARK 驱动 decode，使 #34043 修复的 sconv/mamba checkpoint 保存路径可被测试触达；同时调整显存预算、CI 时长和 decode-cache 命中率门槛以稳定通过 B200/GB300 验证。

关键符号：TestInklingSmallNvfp4DsparkDeterministic.setUpClass, TestInklingSmallNvfp4DsparkDeterministic.test_input_output_logprobs_match_prefill_cache_hit, TestInklingSmallNvfp4DsparkDeterministic.test_input_output_logprobs_match_decode_cache_hit

关键源码片段

test/registered/models_e2e/test_inkling_small_nvfp4.py

唯一变更文件。将 Inkling-Small NVFP4 确定性套件从无推测模式切换到 DSPARK 驱动 decode，使 #34043 修复的 sconv/mamba checkpoint 保存路径可被测试触达；同时调整显存预算、CI 时长和 decode-cache 命中率门槛以稳定通过 B200/GB300 验证。

```
# DSPARK draft 权重路径允许通过环境变量覆盖，CI 中默认使用 RadixArk 版本。
_DSPARK_DRAFT_PATH = os.environ.get(
    "INKLING_SMALL_DSPARK_DRAFT_PATH", "RadixArk/Inkling-Small-DSpark"
)


# CI 预估时长从 1700 提高到 2000：多出的开销来自 draft 加载与 speculative
# CUDA graph 捕获。
register_cuda_ci(est_time=2000, stage="extra-b", runner_config="4-gpu-b200")


class TestInklingSmallNvfp4DsparkDeterministic(CustomTestCase):
    """为什么必须用 DSPARK 驱动 decode：
```

```
1. mamba/sconv checkpoint 保存逻辑挂在 PrefillCudaGraphRunner 上，
   EAGLE 目标会整体禁用 prefill CUDA 图（见 #28386），导致
   test_unified_radix_cache_kl_hybrid_bitexact.py 中的 MTP 类永远
   无法触达该路径。
2. DSPARK / DFlash 会保留 prefill 图，且正是 #34043 实际修复的算法。
3. 回退 #34043 后本类在 prefill_cache_hit 上读出 1.10e-01，
   无推测加速时为精确 0，证明这是 spec-state 特有的回归信号。
"""

@classmethod
def setUpClass(cls):
    cls.model = _MODEL_PATH
    cls.base_url = DEFAULT_URL_FOR_TEST
    cls.process = popen_launch_server(
        cls.model,
        cls.base_url,
        timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
        other_args=[
            # 以下省略与基础类相同的公共参数（tp=4、modelopt_fp4、
            # fa4、page-size 128 等），只列出本类的关键差异。
            # --mem-fraction-static 从 0.85 降到 0.80：
            # draft 权重加 speculative CUDA graph 在 178 GB 的
            # B200 上于 0.85 时中途 OOM；GB300 单卡 284 GB 多出
            # 60% 余量，所以只在 GB300 上验证无法发现问题。
            "--mem-fraction-static", "0.80",
            "--mamba-track-interval", str(KL_TRACK_INTERVAL),
            "--enable-deterministic-inference",
            # DSPARK 驱动 decode，使 sconv/mamba 保存逻辑真正被执行。
            "--speculative-algorithm", "DSPARK",
            "--speculative-draft-model-path", _DSPARK_DRAFT_PATH,
            "--speculative-draft-attention-backend", "fa4",
        ],
        env={**os.environ, "SGLANG_ENABLE_UNIFIED_RADIX_TREE": "1"},
    )

def test_input_output_logprobs_match_decode_cache_hit(self):
    # 门槛从 0.99 放宽到 0.90：DSPARK 最后一次验证会整块提交，使请求
    # 超出 max_new_tokens 最多 block_size-1 个 token；这些 token 写进
    # 了 KV 池与 radix 树，却没有出现在返回的 output_ids 中，导致后续
    # 轮次在 prompt + returned 处发生分歧。当唯一的 mamba checkpoint
    # （floor(committed / interval) * interval）落在分歧点之后时，整个
    # decode 区域不可复用，约 1/32 的 prompt 会天然 miss。回退 #34043
    # 的另一症状表现为 prefill_cache_hit 上升 1.10e-01，仍远超 1e-9
    # 的下限，因此检测能力没有被削弱。
    self._run(
        assert_logprobs_match_decode_cache_hit,
        min_cache_hit_ratio=0.90,
    )
```

评论区精华

本 PR 几乎没有人工 review 讨论：ispobock 直接 APPROVED（空 body）。唯一一条 review 评论来自 ChatGPT codex bot，标记为 P1：

> Register the GB300 test in a defined CI suite —— When any registered-suite CI job invokes test/run_suite.py, this registration resolves to extra-b-test-4-gpu-gb300, but that suite is absent from both the CUDA suite allowlist and . github/workflows/pr-test-extra.yml ...

值得注意的是，该评论引用的文件 test/registered/radix_cache/unified_radix_tree/test_unified_radix_cache_kl_dspark_bitexact.py 并不在最终 PR 的 diff 中（本 PR 只改了一个文件），comment 关联的 commit hash（269f917247）也不在提交列表里，属于机器人评审产生的噪音或过期评论。真正有信息量的讨论都写在 PR body 和类 docstring 里：作者用 20+ 次 /rerun-test 反复暴露 B200 端 OOM 和 decode-cache 命中波动，最终以三项参数调整稳定闭环，这个过程本身就是最有价值的 " 讨论 "。

- GB300 测试需要 CI 套件注册（Codex 机器人 P1）(testing): 该评论锚定的文件（test_unified_radix_cache_kl_dspark_bitexact.py）不在本 PR 最终 diff 中，review commit hash 也与提交列表不符，属于噪音评论；ispobock 直接批准合并，未见人工跟进。

风险与影响

- 风险：
  1. decode 侧症状检测力被削弱：回退 #34043 时的第二个症状是 Too few decode cache hits: 31/32（约 96.9%），但 0.90 的新门槛不会再触断这个断言；该回退只能靠 prefill_cache_hit 的 1.10e-01 拦截。如果未来出现只影响 decode 区复用、不影响 prefill 命中数的回归，这套测试可能漏报。
  2. 非 spec 臂覆盖移除：NVFP4 确定性类不再跑无推测对照组，作者把非 spec 的 cache-restore 逻辑覆盖归给 test_unified_radix_cache_kl_hybrid_bitexact.py 的压缩 checkpoint；该文件的维护状态决定了这个盲区是否成立。
  3. 显存余量收紧：--mem-fraction-static 0.80 在 B200 上仍偏紧，任何增加 graph 捕获内存或 draft 模型内存的改动都可能让该测试重新 OOM；GB300 的 284 GB 无法暴露 B200 上的资源压力。
  4. 测试自归属难度提升：作者在 docstring 中明确写到，256-expert MoE 下仅去掉 --enable-deterministic-inference 就能读到 5.0e-02 量级的信号，与 bug 信号（1.10e-01）同量级，标红时需要手动去推测加速重跑来区分状态复用 bug 与 batch 方差。- 影响：用户 / 产品：无影响，本 PR 不包含任何生产代码变更。系统 /CI：test/registered/models_e2e/test_inkling_small_nvfp4.py 的确定性类改由 DSPARK 驱动，CI 预估时长从 1700 s 增至 2000 s，占用 4-gpu-b200 和 4-gpu-gb300 槽位；B200 端历史重跑记录显示该套件此前波动较大，经参数调整后稳定通过。团队：为 #34043（sconv state 内存损坏修复）补上了真正可触达的 spec-side 位精确回归防线，同时保留了对 modelopt_fp4 权重量化和 FP4 GEMM/MoE 路径的守护；该文件也成为 radix cache + speculative decoding 交互语义的一份重要参考实现。

- 风险标记：测试覆盖迁移，decode 侧回归可能漏报，显存余量收紧，CI 时长增加，无生产代码变更

关联脉络

- PR #34043 [srt] Fix sconv state memory corruption on specdec: 本 PR 的核心目的就是为 #34043 的修复建立可触达的回归防线：DSPARK/DFlash 推测池在 radix cache 下损坏 sconv 状态的问题，正是本测试要守护的对象，PR 中用 revert-then-red 验证了其检测力。
- PR #28386 Disable prefill CUDA graph for EAGLE target: PR body 和测试 docstring 均引用 #28386：EAGLE 目标禁用 prefill CUDA graph 导致 PrefillCudaGraphRunner 不构造，是本 PR 将测试切换到 DSPARK 的直接原因。