

PR #24346 完整报告

sgl-project/sglang

[UnifiedRadixTree]: Add KL accuracy CI for UnifiedTree with HiCache

合并时间: 2026-05-04 20:18

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/24346>

执行摘要

- 一句话: 为 UnifiedRadixTree 添加 HiCache KL 准确性 CI 测试
- 推荐动作: 值得精读, 展示了如何安全地恢复被注释的测试、调整日志层级以及增强测试基础设施。尤其是 `_flush_cache` 的 `timeout` 参数化改进值得在测试工具中推广。

功能与动机

此前 HiCache 与 CI 的 CUDA 13 环境存在兼容性问题, 测试被注释 (见源码 TODO 注释)。现在兼容问题已修复, 需要重新启用该 CI 测试, 确保 UnifiedRadixTree 结合 HiCache 的 KL 准确性在 CI 中持续验证。

实现拆解

1. 重新启用 HiCache 测试类: 在 `test_unified_radix_cache_kl.py` 中取消 `TestUnifiedMambaRadixCacheWithHiCache` 的注释, 并为其 `setUpClass` 添加 `--enable-hierarchical-cache`、`--hcache-ratio 1.5`、`--hcache-write-policy write_through` 等参数, 同时设置 `SGLANG_ENABLE_UNIFIED_RADIX_TREE=1` 环境变量。
2. 调整现有 SWA 测试配置: 将 `TestUnifiedSWARadixCache` 的 `gsm8k_threshold` 和 `mmlu_threshold` 从 0.75 降低至 0.7, 并将环境变量 `SGLANG_ENABLE_UNIFIED_RADIX_TREE` 从 "0" 改为 "1", 使 SWA 测试也运行在 UnifiedRadixTree 模式下。
3. 降级冗余日志: 在 `unified_radix_cache.py` 中移除 `load_back` 函数中一条 `info` 日志, 并将 `init_load_back` 函数中的一条 `info` 日志改为 `debug` 级别, 减少 CI 日志噪音。
4. 增强 `_flush_cache` 可靠性: 在 `kl_test_utils.py` 中为 `_flush_cache` 添加可配置超时参数 (默认 30 秒) 和 `raise_for_status` 异常检查, 避免测试因 HTTP 超时静默失败。
5. 更新 CI 时间预算: 将 `register_cuda_ci` 的 `est_time` 从 632 秒提升至 760 秒, 为新增的 HiCache 测试预留充足时间。

关键文件:

- `test/registered/radix_cache/test_unified_radix_cache_kl.py` (模块测试; 类别 `test`; 类型 `test-coverage`; 符号 `TestUnifiedMambaRadixCacheWithHiCache`, `setUpClass`, `tearDownClass`): 核心测试文件, 取消了被注释的 HiCache 测试类并重新启用, 调整了 SWA 测试的阈值和环境变量, 是 PR 的主要变更载体。

- python/sglang/srt/mem_cache/unified_radix_cache.py (模块 缓存层; 类别 source; 类型 core-logic; 符号 load_back, init_load_back) : 核心缓存模块, 移除了两条 info 日志并降低另一条日志级别, 减少 CI 日志噪音, 属于细小但稳固的改进。
- python/sglang/test/kl_test_utils.py (模块 测试工具; 类别 test; 类型 test-coverage; 符号 _flush_cache) : 测试工具函数, 增强了 _flush_cache 的超时和异常处理, 提升测试稳定性。

关键符号: _flush_cache, TestUnifiedMambaRadixCacheWithHiCache.setUpClass, load_back, init_load_back

关键源码片段

test/registered/radix_cache/test_unified_radix_cache_kl.py

核心测试文件, 取消了被注释的 HiCache 测试类并重新启用, 调整了 SWA 测试的阈值和环境变量, 是 PR 的主要变更载体。

```
# test/registered/radix_cache/test_unified_radix_cache_kl.py
# 新启用 HiCache 测试类 (原本被注释) :
class TestUnifiedMambaRadixCacheWithHiCache(UnifiedRadixTreeTestMixin, CustomTestCase):
    """Mamba hybrid + HiCache + UnifiedRadixCache."""
    kl_threshold = 0.003
    prefill_cache_assert = staticmethod(
        make_mamba_prefill_assert(chunk_size=MAMBA_CHUNK_SIZE)
    )
    decode_cache_assert = staticmethod(
        make_mamba_decode_assert(track_interval=MAMBA_TRACK_INTERVAL)
    )

    @classmethod
    def setUpClass(cls):
        cls.model = MAMBA_MODEL
        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=[
                "--tp-size", "4",
                "--chunked-prefill-size", "2048",
                "--mem-fraction-static", "0.85",
                "--mamba-scheduler-strategy", "extra_buffer",
                "--mamba-track-interval", str(MAMBA_TRACK_INTERVAL),
                # 以下为 HiCache 相关参数:
                "--enable-hierarchical-cache",
                "--hcache-ratio", "1.5",
                "--hcache-write-policy", "write_through",
                "--hcache-io-backend", "direct",
                "--hcache-mem-layout", "page_first_direct",
```

```

    ],
    env={"SGLANG_ENABLE_UNIFIED_RADIX_TREE": "1"},
)
cls.input_ids = get_input_ids(cls.model, num_samples=18)

@classmethod
def tearDownClass(cls):
    kill_process_tree(cls.process.pid)

```

python/sglang/srt/mem_cache/unified_radix_cache.py

核心缓存模块，移除了两条 info 日志并降低另一条日志级别，减少 CI 日志噪音，属于细小但稳固的改进。

```

# python/sglang/srt/mem_cache/unified_radix_cache.py
# 变更后：移除 / 降级冗余日志

def load_back(self, last_hit_node, mem_quota, *, req=None):
    # ... 之前的代码 ...
    # 去掉了原先的 logger.info("load_back: kv_tokens=%d...")
    # Load H→D
    aux_xfers = [x for xfers in comp_xfers.values() for x in xfers]
    # ...

def init_load_back(self, params):
    # ...
    if last_node.evicted or params.host_hit_length > 0:
        loading_values = self.load_back(last_node, mem_quota, req=req)
        if loading_values is not None:
            logger.debug( # 原为 logger.info
                "init_load_back success: loaded %d tokens for node %d",
                len(loading_values), last_node.id,
            )
            return loading_values, last_node
    # ...

```

python/sglang/test/kl_test_utils.py

测试工具函数，增强了 _flush_cache 的超时和异常处理，提升测试稳定性。

```

# python/sglang/test/kl_test_utils.py
# 增强后的 _flush_cache 函数:
def _flush_cache(base_url, timeout_s=30):
    """Flush server cache with timeout and error checking."""
    response = requests.post(
        base_url + "/flush_cache",
        params={"timeout": timeout_s},
        timeout=timeout_s + 10, # 额外 10 秒缓冲
    )
    response.raise_for_status() # 非 2xx 直接抛异常

```

评论区精华

该 PR 审核简单，未产生实质性讨论，直接获得批准。

- 暂无高价值评论线程

风险与影响

- 风险：启用新的 4-GPU H100 测试将 CI 预计时长从 632s 增加至 760s，但影响仅限 stage-c-test-4-gpu-h100 流水线。_flush_cache 增强可能暴露隐藏超时问题，但提高了测试可靠性。修改 SWA 测试阈值可能降低精度检测灵敏度，但 0.7 仍属合理范围。
- 影响：直接影响 test_unified_radix_cache_kl.py 测试套件，新增一个耗时约 2 分钟的 KL 精度测试。间接提升 HiCache + UnifiedRadixTree 组合的 CI 覆盖度，降低回归风险。对用户无直接影响，对开发者来说是 CI 质量改进。
- 风险标记：CI 耗时增加，测试稳定性依赖于 HiCache 兼容性

关联脉络

- PR #24257 [PD]: Support incremental transfer for mooncake transfer engine: 同为 HiCache 相关改进，增加了增量 KV 传输功能，本 PR 的 CI 测试可验证其正确性。
- PR #24318 Fix swa chunk req deferred: 同为调度和缓存修复，SWA 测试的阈值调整与此相关。