PR #25886 完整报告

sgl-project/sglang

[Test] Add fwd_occupancy sanity kit

合并时间：2026-05-20 18:34

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25886

执行摘要

- 一句话：新增 GPU 前向占用测试套件并重构 hellaswag 测试
- 推荐动作：值得精读，尤其 FwdOccupancyMixin 通过 Prometheus gauge 实现系统级断言的模式，以及 HellaswagMixin 的提取手法。团队可在后续 PR 中考虑采纳 review 建议（min、Event.wait、锁）进一步提升稳定性。

功能与动机

单批次解码是 CPU 开销主导的场景，overlap scheduler 和 cuda graph 回归容易在批处理吞吐量下降前先在此暴露。通过持续监控sglang:fwd_occupancygauge，可以在回归早期捕获。PR body 指出："Single-batch is where CPU overhead dominates -- overlap scheduler / cuda graph regressions surface here before batched throughput moves."

实现拆解

1. 创建 FwdOccupancyMixin：在 python/sglang/test/kits/fwd_occupancy_kit.py 中定义，包含可配置阈值、采样间隔、预热和测量参数。
2. 前提条件检查：方法 _assert_metrics_device_timer_enabled 确保服务器已启用 --enable-metrics 和 SGLANG_ENABLE_METRICS_DEVICE_TIMER=1，否则直接报错。
3. 预热阶段：_fwd_occupancy_warmup 发送一个短请求（max_new_tokens=64）填充 cuda graph 并让设备定时器度过首个 NaN 窗口。
4. 测量阶段：_fwd_occupancy_measure 在后台线程发起长请求（max_new_tokens=2048），主线程定时刮取 /metrics 收集非 NaN 样本；最终 test_fwd_occupancy 计算样本中位数，断言超过阈值；若服务器开启 spec decoding，还检查 avg_spec_accept_length 是否低于 spec_accept_length_threshold（默认 1.8）。
5. 提取 HellaswagMixin：将原 test_basic_sanity.py 中的 test_accuracy_floor 逻辑独立为 python/sglang/test/kits/hellaswag_kit.py 中的可复用 mixin。
6. 新增 EAGLE3 测试：test/registered/core/test_basic_sanity_eagle3.py 启动 EAGLE3 投机解码服务器，继承 FwdOccupancyMixin、HellaswagMixin 等，CI 注册时间 200 秒。
7. 改造基础测试：test_basic_sanity.py 导入并继承 FwdOccupancyMixin 和 HellaswagMixin，删除内联 test_accuracy_floor，添加 SGLANG_ENABLE_METRICS_DEVICE_TIMER=1 环境变量，阈值设为 97%。

8. 微调测试程序：python/sglang/test/test_programs.py 中移除不稳定的 latency 断言（assert np.abs(latency_gen - latency) < 1），改为只打印耗时供人工审查。

关键文件：

- python/sglang/test/kits/fwd_occupancy_kit.py（模块 占用测试套件；类别 test；类型 test-coverage；符号 FwdOccupancyMixin, _scrape_fwd_occupancy, _assert_metrics_device_timer_enabled, _fwd_occupancy_fire）：核心新增文件，定义 FwdOccupancyMixin，实现 occupancy 探针的采集、预热、测量和断言逻辑
- test/registered/core/test_basic_sanity_eagle3.py（模块 EAGLE3 测试；类别 test；类型 test-coverage；符号 TestBasicSanityEagle3, setUpClass, tearDownClass）：新增 EAGLE3 配置下的基础测试，集成包含 FwdOccupancyMixin 在内的所有混入
- python/sglang/test/kits/hellaswag_kit.py（模块 准确率测试套件；类别 test；类型 test-coverage；符号 HellaswagMixin, test_accuracy_floor）：提取 HellaswagMixin，复用 hellaswag accuracy 测试
- test/registered/core/test_basic_sanity.py（模块 基础测试；类别 test；类型 test-coverage；符号 TestBasicSanity, setUpClass, tearDownClass, fwd_occupancy_threshold）：主测试文件，集成 FwdOccupancyMixin 和 HellaswagMixin，删除内联测试
- python/sglang/test/test_programs.py（模块 测试程序；类别 test；类型 test-coverage；符号 test_hellaswag_select）：微调 test_hellaswag_select，移除不可靠的 latency 断言

关键符号：FwdOccupancyMixin, _scrape_fwd_occupancy, _fwd_occupancy_measure, test_fwd_occupancy, HellaswagMixin.test_accuracy_floor, TestBasicSanityEagle3.setUpClass, TestBasicSanity.setUpClass

关键源码片段

python/sglang/test/kits/fwd_occupancy_kit.py

核心新增文件，定义 FwdOccupancyMixin，实现 occupancy 探针的采集、预热、测量和断言逻辑

```
# 从 /metrics 刮取 sglang:fwd_occupancy 值，返回跨 label 的最大非 NaN 值
# 若刮取失败或全为 NaN 则返回 None
import re
import requests

_FWD_OCCUPANCY_RE = re.compile(
    r"^sglang:fwd_occupancy(?:\{[^}]*\})?\s+(\S+)", re.MULTILINE
)

def _scrape_fwd_occupancy(self):
    try:
        resp = requests.get(
            self.base_url + "/metrics", timeout=10
        )
    except requests.RequestException:
```

```
        return None
    if resp.status_code != 200:
        return None
    vals = []
    for raw in _FWD_OCCUPANCY_RE.findall(resp.text):
        try:
            v = float(raw)
        except ValueError:
            continue
        # NaN filter: gauge resets to NaN on window boundary, skip them
        if v == v: # NaN 不与自身相等
            vals.append(v)
    return max(vals) if vals else None
```

test/registered/core/test_basic_sanity_eagle3.py

新增 EAGLE3 配置下的基础测试，集成包含 FwdOccupancyMixin 在内的所有混入

```
class TestBasicSanityEagle3(
    # 混入多个测试套件
    BasicAPIContractMixin,
    BasicDecodeCorrectnessMixin,
    BasicSchedulerStressMixin,
    FwdOccupancyMixin, # 占用率测试
    HellaswagMixin, # Hellaswag 准确率
    CustomTestCase,
):
    served_model_name = DEFAULT_TARGET_MODEL_EAGLE3
    fwd_occupancy_threshold = 97.0 # 与普通测试保持一致的阈值

    @classmethod
    def setUpClass(cls):
        cls.base_url = DEFAULT_URL_FOR_TEST
        cls.process = popen_launch_server(
            DEFAULT_TARGET_MODEL_EAGLE3,
            cls.base_url,
            timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
            other_args=[
                "--dtype", "float16", # fp16 + triton attn 避免 SM120 数据类型不匹配
                "--attention-backend", "triton",
                "--speculative-algorithm", "EAGLE3",
                "--speculative-draft-model-path", DEFAULT_DRAFT_MODEL_EAGLE3,
                "--speculative-num-steps", "3",
                "--speculative-eagle-topk", "1",
                "--speculative-num-draft-tokens", "4",
                "--cuda-graph-max-bs", "4",
                "--mem-fraction-static", "0.7",
                "--enable-metrics", # 必须开启以暴露 gauge
            ],
            env={"SGLANG_ENABLE_METRICS_DEVICE_TIMER": "1"}, # 启用设备定时器
```

```
        )
```

test/registered/core/test_basic_sanity.py

主测试文件，集成 FwdOccupancyMixin 和 HellaswagMixin，删除内联测试

```
class TestBasicSanity(
    BasicAPIContractMixin,
    BasicDecodeCorrectnessMixin,
    BasicSchedulerStressMixin,
    FwdOccupancyMixin, # 新增混入
    HellaswagMixin, # 提取后的混入
    CustomTestCase,
):
    served_model_name = DEFAULT_MODEL_NAME_FOR_TEST
    # 5090 + Llama-3.1-8B 单批次 decode 测得 ~99 中位数，预留 2pp 余量
    fwd_occupancy_threshold = 97.0

    @classmethod
    def setUpClass(cls):
        cls.base_url = DEFAULT_URL_FOR_TEST
        cls.process = popen_launch_server(
            DEFAULT_MODEL_NAME_FOR_TEST,
            cls.base_url,
            timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
            other_args=[
                "--cuda-graph-max-bs", "4",
                "--mem-fraction-static", "0.7",
                "--enable-metrics",
            ],
            env={"SGLANG_ENABLE_METRICS_DEVICE_TIMER": "1"}, # 与前一版本相比新增
        )
    # 注意：原内联的 test_accuracy_floor 已被删除，由 HellaswagMixin 提供
```

评论区精华

gemini-code-assist[bot] 提出三个改进建议：(1) 在 _scrape_fwd_occupancy 中使用 min 代替 max，以更好暴露 DP 下单 rank 退化；(2) 使用 Event.wait 替代 time.sleep 提高线程响应性；(3) 返回样本副本以避免竞态。当前代码未采纳这些建议（仍使用 max、time.sleep，且无锁保护样本列表），作者在 commit 中特意 ' drop samples_lock '，可能认为锁收益不大。

- 使用 min 代替 max 以检测 DP 下单 rank 退化 (performance): 未被采纳，当前代码仍使用 max。作者可能认为单批次下各 rank 负载均衡，min 可能过于严格。
- 使用 Event.wait 替代 time.sleep 提高线程响应性 (design): 未采纳，当前代码仍使用 time.sleep。
- 返回样本副本以避免竞态条件 (correctness): 未采纳，commit message 提到 'drop samples_lock'，表明有意简化，依赖 join timeout 后的安全。

风险与影响

- 风险：测试依赖 sglang:fwd_occupancy gauge，需要服务器启动 --enable-metrics 并设置 SGLANG_ENABLE_METRICS_DEVICE_TIMER=1，若未正确配置测试将失败。单批次长请求（max_new_tokens=2048）耗时较长（约 20-30 秒），增加 CI 运行时间。中位数阈值（默认 95%，stage-a 调至 97%）可能因 CI 环境波动偶尔告警，需要定期校准。
- 影响：对开发者：Stage-a 测试新增 occupancy 校验，可能暴露之前隐藏的回归（如 overlap scheduler 抖动）。对系统：无运行时影响，仅测试工具。对团队：需要维护阈值与 CI 环境匹配，并可能根据 review 建议改进测试鲁棒性。
- 风险标记：环境依赖（metrics），测试耗时增加，阈值需校准，review 建议未解决

关联脉络

- PR #25862 Address overlap future token map by request-pool index: 该 PR 修改 overlap scheduler 结构，本 PR 的 occupancy 测试正是为检测此类回归而设计。
- PR #25795 Enable breakable CUDA graph for eagle: 引入可中断 CUDA 图，EAGLE3 测试集成于此 PR 中，共享投机解码上下文。