

PR #29348 完整报告

sgl-project/sglang

[test] Split perf table out of fwd occupancy kit report

合并时间: 2026-06-26 11:23

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29348>

执行摘要

- 一句话: 拆分 fwd occupancy kit 报告为性能与占用双表
- 推荐动作: 值得阅读以了解如何组织测试报告, 尤其对于需要设计 CI 可读报告的工程师。
设计决策: 将不同语义的指标分离到各自表格, 并在断言前打印, 提高调试友好性。这对后续编写类似测试报告有参考价值。

功能与动机

PR body 指出: 之前报告将 `token tps` 捆绑在 `fwd_occupancy` 表中导致表头内容不匹配; `avg_spec_accept_length` 在 median 断言之后才获取 / 打印, 因此当占用检查失败时该指标不会显示; 非占用数据应放在独立表格中。本 PR 旨在解决这些报告问题, 提高调试可见性。

实现拆解

按以下步骤实现变更:

1. 修改 `_fwd_occupancy_fire` 返回完整 meta_info: 原函数返回 (completion_tokens, wall_time), 现返回 (meta_info, elapsed), 以便后续从中提取 prompt_tokens、decode_throughput 等字段。(文件: fwd_occupancy_kit.py)
2. 新增 `_fwd_occupancy_perf` 聚合性能指标: 该函数从 meta_info 和 elapsed 计算 input_tokens、output_tokens、decode_tps、mean_itl_ms、wall_tps, 返回一个 dict。(文件: fwd_occupancy_kit.py)
3. 修改 `_fwd_occupancy_measure` 返回 perf 字典: 原返回 (samples, token_tps), 现改为返回 (samples, _fwd_occupancy_perf(result)), 其中 result 包含 meta_info 和 elapsed。(文件: fwd_occupancy_kit.py)
4. 调整 test_fwd_occupancy 报告逻辑: 在计算 median 之前, 先通过 /server_info 获取 avg_spec_accept_length (若存在)。然后打印两个表格:
 - perf 表格: 包含 input_tokens、output_tokens、decode_tps、mean_itl_ms、wall_tps, 以及可选的 avg_spec_accept_length。
 - 占用率表格: 包含 samples、median、peak、p10、threshold。两个表格均在 median 断言前打印, 确保断言失败时所有指标仍可见。(文件: fwd_occupancy_kit.py)
5. 守卫空样本集: 当 samples 为空时 (例如 gauge 全为 NaN), median/p10/peak 打印 NaN 占位符, 避免统计函数报错。(文件: fwd_occupancy_kit.py)

仅修改了一个文件，无其他配置或部署配套变更。

关键文件：

- python/sclang/test/kits/fwd_occupancy_kit.py (模块 占用测试；类别 test；类型 test-coverage；符号 _fwd_occupancy_perf, _fwd_occupancy_fire, _fwd_occupancy_measure, test_fwd_occupancy)：唯一变更文件，包含所有报告拆分逻辑：新增 _fwd_occupancy_perf 函数，修改 fire/measure 流程，调整 test 函数打印双表。

关键符号：_fwd_occupancy_perf, _fwd_occupancy_fire, _fwd_occupancy_measure, test_fwd_occupancy

关键源码片段

python/sclang/test/kits/fwd_occupancy_kit.py

唯一变更文件，包含所有报告拆分逻辑：新增 _fwd_occupancy_perf 函数，修改 fire/measure 流程，调整 test 函数打印双表。

```
# _fwd_occupancy_perf: 从单次请求的 result 中提取性能指标
def _fwd_occupancy_perf(self, result):
    """Aggregate per-request perf metrics from the fire result
    (input/output tokens, decode tps, mean inter-token latency,
    wall-clock tps)."""
    meta = result["meta_info"] or {} # 避免空 dict 时 get 返回 None
    elapsed = result["elapsed"]
    out = meta.get("completion_tokens", 0) or 0 # 防御性：若为 None 则取 0
    decode_tps = meta.get("decode_throughput", 0.0) or 0.0
    return {
        "input_tokens": meta.get("prompt_tokens", 0) or 0,
        "output_tokens": out,
        "decode_tps": decode_tps,
        # 平均 token 间隔 (ms): decode_tps 为每秒 token 数，取倒数乘 1000
        "mean_itl_ms": (1000.0 / decode_tps) if decode_tps > 0 else 0.0,
        # 端到端吞吐：总输出 token / 总 wall 时间
        "wall_tps": (out / elapsed) if elapsed > 0 else 0.0,
    }

# test_fwd_occupancy 中获取 avg_accept 并打印两个表格的部分
def test_fwd_occupancy(self):
    ...
    # 获取 avg_spec_accept_length (若使用了推测解码则存在值)
    try:
        info = requests.get(self.base_url + "/server_info", timeout=...).json()
        avg_accept = info.get("avg_spec_accept_length", None)
    except Exception:
        avg_accept = None

    # 构建 perf 表格行，转为列表便于添加可选字段
    rows = [
        ("input tokens", perf["input_tokens"]),
```

```

        ("output tokens", perf["output_tokens"]),
        ("decode tps", perf["decode_tps"]),
        ("mean itl (ms)", perf["mean_itl_ms"]),
        ("wall tps", perf["wall_tps"]),
    ]
    if avg_accept is not None:
        rows.append(("avg spec accept", avg_accept))

    # 在断言之前打印 perf 表格，确保失败时依然可见
    print("\n" + tabulate.tabulate(rows, headers=["perf metric", "value"]))

    # 处理占用率样本：若为空则用 NaN 占位，防止统计函数崩溃
    if samples:
        samples_sorted = sorted(samples)
        median = statistics.median(samples_sorted)
        peak = samples_sorted[-1]
        p10_idx = max(0, len(samples_sorted) // 10)
        p10 = samples_sorted[p10_idx]
    else:
        median = peak = p10 = float("nan")

    # 打印占用率表格
    print(tabulate.tabulate(
        [
            ("samples (n)", len(samples)),
            ("median", median),
            ("peak", peak),
            ("p10", p10),
            ("threshold", self.fwd_occupancy_threshold),
        ],
        headers=["fwd_occupancy", "value"],
    ))

    # 最后断言：median 必须达到阈值
    self.assertGreaterEqual(median, self.fwd_occupancy_threshold, ...)

```

评论区精华

PR 中无 review 评论 (0 comments)，仅作者自己在 CI 操作 (/tag-and-rerun-ci)。无公开讨论记录。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低，变更限定于测试工具的输出格式和内部逻辑。主要风险为：如果外部 CI 脚本解析了旧的表格格式（如 token_tps 在占用表内），格式变更可能导致解析失败。但考虑到该工具用于在 CI 中人工查看报告，且格式是 Markdown 表格，CI 脚本不太可能严格解析。另外，新增的 _fwd_occupancy_perf 依赖 meta_info 中的 decode_throughput 字

段，若服务端未暴露该字段，则相关指标显示为 0。但该字段在 SGLang 中已存在，风险较小。

- 影响：仅影响 fwd_occupancy_kit 测试用例的输出格式和其中间变量。对用户、系统和其他测试无影响。影响范围极小，程度低。
- 风险标记：测试输出格式变更

关联脉络

- PR #29332 [test] Report token tps in fwd occupancy kit and force ignore_eos: 本 PR 是 #29332 的 follow-up，修复合并后的报告问题。