PR #27662 完整报告

sgl-project/sglang

[Bench] Add consistent p90/p95/p99 percentiles for all latency metrics

合并时间：2026-06-13 04:44

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27662

执行摘要

- 一句话：统一 benchmark 延迟指标输出 p90/p95/p99 百分位
- 推荐动作：建议合并前修复 *e2e_latencies* 空列表问题，对齐 ttfts 等变量的防护模式——使用 *e2e_latencies or 0* 作为 np.percentile 的输入。该 PR 逻辑简单清晰，合并后能显著提升 benchmark 数据的完整性。

功能与动机

PR body 指出之前不同延迟指标报告的百分位不一致：TTFT 和 TPOT 仅报告 p99，ITL 有 p95/p99 但缺 p90，E2E 有 p90/p99 但缺 p95。统一后所有指标都输出 p90/p95/p99，便于不同场景对比分析。

实现拆解

1. 重构 BenchmarkMetrics dataclass（python/sglang/bench_serving.py）：为每个指标添加缺失的 p90/p95/p99 字段，并按功能分组添加注释（Request counts、Throughput、TTFT、TPOT、ITL、E2E、Concurrency）。
2. 补充百分位计算逻辑（calculate_metrics 函数）：使用 np.percentile 计算并赋值新增字段，复用已有的 or 0 防护避免空列表 crash。
3. 更新终端输出：在 print 区块中添加对应百分位的打印行，输出格式保持与原有一致（字段名左对齐、数值保留两位小数）。
4. 影响输出和 JSON 结果：终端输出和 JSON 结果文件（若使用 --save-result）会自动包含新字段，无需额外更改。

关键文件：

- python/sglang/bench_serving.py（模块 基准测试；类别 source；类型 core-logic）：唯一变更文件，新增所有百分位字段与计算逻辑，并重构 dataclass 结构。

关键符号：*calculate_metrics*, post_init? (BenchmarkMetrics 无自定义 init，仅 dataclass 自动生成)

关键源码片段

python/sglang/bench_serving.py

唯一变更文件，新增所有百分位字段与计算逻辑，并重构 dataclass 结构。

```
# python/sglang/bench_serving.py - BenchmarkMetrics dataclass 改动片段 @dataclass
classBenchmarkMetrics:     # Request counts and token totals     completed: int
total_input: int     total_input_text: int     total_input_vision: int     total_output: int
total_output_retokenized: int     # Throughput (req/s and tok/s)     request_throughput:
float     input_throughput: float     output_throughput: float
output_throughput_retokenized: float     total_throughput: float
total_throughput_retokenized: float     # TTFT - Time to First Token (ms)
mean_ttft_ms: float     median_ttft_ms: float     std_ttft_ms: float     p90_ttft_ms: float
# 新增     p95_ttft_ms: float # 新增     p99_ttft_ms: float     # TPOT - Time per Output
Token, excluding the first token (ms)     mean_tpot_ms: float     median_tpot_ms: float
  std_tpot_ms: float     p90_tpot_ms: float # 新增     p95_tpot_ms: float # 新增
p99_tpot_ms: float     # ITL - Inter-Token Latency (ms)     mean_itl_ms: float
median_itl_ms: float     std_itl_ms: float     p90_itl_ms: float # 新增     p95_itl_ms: float
   p99_itl_ms: float     max_itl_ms: float     # E2E - End-to-End request latency (ms)
mean_e2e_latency_ms: float     median_e2e_latency_ms: float     std_e2e_latency_ms:
float     p90_e2e_latency_ms: float     p95_e2e_latency_ms: float # 新增
p99_e2e_latency_ms: float     # Concurrency and peak metrics     concurrency: float
max_output_tokens_per_s: float = 0.0     max_concurrent_requests: int = 0 #
calculate_metrics 函数中的百分位计算改动（部分） # 注意：ttfts/tpots/itls 已有 `or 0` 防
护，e2e_latencies 未添加 p90_ttft_ms = np.percentile(ttfts or 0, 90) * 1000 p95_ttft_ms
= np.percentile(ttfts or 0, 95) * 1000 p90_tpot_ms = np.percentile(tpots or 0, 90) *
1000 p95_tpot_ms = np.percentile(tpots or 0, 95) * 1000 p90_itl_ms =
np.percentile(itls or 0, 90) * 1000 p95_e2e_latency_ms = np.percentile(e2e_latencies,
95) * 1000 # 此处缺少 `or 0`，若列表为空会抛出 ValueError
```

评论区精华

Gemini Code Assist 机器人指出：当所有请求失败（completed == 0）时，e2e_latencies 为空列表，np.percentile 会抛出 ValueError 导致脚本崩溃。建议对 e2e_latencies 也添加 or 0 防护（类似 ttfts 等已有处理）。该评论未得到回复或 resolved，属于已提出的未解决风险。

- e2e_latencies 空列表导致 np.percentile crash (correctness): 未得到回复或修复，属于已提出的未解决问题。

风险与影响

- 风险：主要风险是 e2e_latencies 空列表场景：当所有请求失败时，新加的 p95_e2e_latency_ms 计算会触发 ValueError，导致 benchmark 脚本在报告结果前崩溃。其他百分位（p90/p95/p99）已通过 or 0 防护（如 ttfts or 0），但 E2E 延迟的 mean/median/std 原有逻辑也使用 e2e_latencies 而无 or 0，只是 np.mean/np.median/np.std 遇到空列表返回 nan 不会 crash。新引入的 np.percentile 则直接抛出异常。
- 影响：影响范围限于 bench_serving.py 一个文件，无外部依赖变更。用户可立即获得更完整的延迟分布信息，提升 benchmark 可观测性。向后兼容：原有字段保持不变，新增字段

不影响已有脚本解析结果。但未修复的 *e2e_latencies* 空列表 *crash* 可能影响全失败场景下的结果输出。

- 风险标记：缺少边缘情况处理，测试覆盖不足

关联脉络

- 暂无明显关联 PR