PR #25510 完整报告

sgl-project/sglang

[diffusion] tighten selected perf baselines

合并时间：2026-05-17 23:35

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25510

执行摘要

- 一句话：收紧 H100 扩散性能基线并修复数据不一致
- 推荐动作：该 PR 主要是测试基准维护，不涉及核心逻辑改动，但对于管理 CI 性能基线的团队有参考价值。关注点在于如何从 CI 运行提取一致快照并确保数据自洽，避免手动编辑引入错误。对于一般开发者，了解其背景即可，无需深入精读。

功能与动机

PR body 指出：使用来自 #25457 的最新成功 Base CI 运行的一致快照收紧 H100 扩散性能基线；仅更新展示至少 5% 端到端改进且无退化的场景；保持每个场景内部一致（使用同一 CI 请求的全量值替换）。

实现拆解

1. 分析 CI 运行结果：从 #25457 的 CI 日志中提取每个场景的完整性能指标（各级耗时、步耗时、聚合值），确保数据源一致。
2. 替换 H100 基线场景：将 perf_baselines.json 中满足改进条件的场景整体替换为新值，使 denoise_step_ms 与 expected_avg_denoise_ms 等聚合字段计算一致（修复 review 中指出的不匹配问题）。
3. 更新元数据：将 metadata 中的描述日期从 2026-04-15 改为 2026-05-17，反映快照来源时间。
4. 优化 server_args.py 注释：为参数分组添加 # layerwise offload、# offload flags、# quantization 等标题，并将 --use-fsdp-inference 移动到 GPU offload 参数附近（功能不变）。
5. 更新 NPU 基线：在 perf_baselines_npu.json 的 wan2_1_t2v_1.3b_1_npu 场景中，将 denoise_step_ms 的 step 0 从 101.91 修正为 330.0（可能为首次步长），并新增 estimated_full_test_time_s 字段。

关键文件：

- python/sglang/multimodal_gen/test/server/perf_baselines.json（模块 性能基线；类别 test；类型 test-coverage）：核心变更文件，H100 扩散性能基线全面收紧，修复了多个场景的数据不一致问题。
- python/sglang/multimodal_gen/runtime/server_args.py（模块 运行时参数；类别 source；类型 refactor；符号 add_cli_args）：调整了参数组注释和顺序，未改变功能，但提供了

更好的可读性。

- python/sglang/multimodal_gen/test/server/ascend/perf_baselines_npu.json（模块 NPU 性能基线；类别 test；类型 test-coverage）：小幅调整了 wan2_1_t2v_1.3b_1_npu 场景的 denoise_step_ms step 0 并新增 estimated_full_test_time_s 字段。

关键符号：add_cli_args

关键源码片段

python/sglang/multimodal_gen/test/server/perf_baselines.json

核心变更文件，H100 扩散性能基线全面收紧，修复了多个场景的数据不一致问题。

```
// 示例：更新后的 metadata 和 tolerances，以及一个场景的替换
{
    "metadata": {
        "model": "Diffusion Server",
        "hardware": "CI H100 80GB pool",
        "description": "Reference numbers captured from CI H100 runs (2026-05-17).", //
        更新日期
        "last_updated": "2026-05-17"
    },
    "tolerances": { // 容忍度未变化
        "long_term": { "e2e": 0.15, "denoise_stage": 0.1, ... },
        "pr_test": { "e2e": 0.25, ... }
    },
    "scenarios": {
        "flux_image_t2i_2_gpus": { // 示例场景：整体替换
            "stages_ms": {
                "InputValidationStage": 0.08,
                "TextEncodingStage": 38.63, // 较旧值 61.66 下降
                "DenoisingStage": 5000.66, // 较旧值 5299.62 下降
                "DecodingStage": 11.98
            },
            "denoise_step_ms": {
                "0": 73.63, "1": 88.19, ..., "49": 100.09
            },
            "expected_e2e_ms": 5242.81,
            "expected_avg_denoise_ms": 99.84, // 由 50 步真实平均得到
            "expected_median_denoise_ms": 100.74,
            "estimated_full_test_time_s": 125.5
        }
    }
}
```

python/sglang/multimodal_gen/runtime/server_args.py

调整了参数组注释和顺序，未改变功能，但提供了更好的可读性。

```
# 在 add_cli_args 函数中，原位于 --dit-offload-prefetch-size 后的 --use-fsdp-inference 被移动到
offload 标志组末尾前；
```

```
# 新增 # layerwise offload 和 # offload flags 等注释，使参数结构更清晰。
def add_cli_args(parser: FlexibleArgumentParser) -> FlexibleArgumentParser:
    # ... 之前的参数 ...

    # layerwise offload # 新增注释
    parser.add_argument("--dit-cpu-offload", ...)
    parser.add_argument("--dit-layerwise-offload", ...)
    parser.add_argument("--layerwise-offload-components", ...)
    parser.add_argument("--dit-offload-prefetch-size", ...)

    # offload flags # 新分组
    parser.add_argument("--text-encoder-cpu-offload", ...)
    parser.add_argument("--image-encoder-cpu-offload", ...)
    parser.add_argument("--vae-cpu-offload", ...)

    parser.add_argument("--use-fsdp-inference", ...) # 从上方移至此
    parser.add_argument("--pin-cpu-memory", ...)

    # quantization # 新增注释
    parser.add_argument("--quantization", ...)
```

评论区精华

Review 由 gemini-code-assist[bot] 指出两处数据不一致：

- fastwan2_2_ti2v_5b 场景：denoise_step_ms 的三个值平均为 129.70，但 expected_avg_denoise_ms 为 132.32，相差约 2ms。
- ltx_2_3_hq_pipeline 场景：DenoisingStage 耗时 (16162.58) 与 denoise_step_ms 之和 (14941.24) 不匹配；expected_avg_denoise_ms (810.79) 与 18 个步长的实际平均 (830.07) 不一致；expected_e2e_ms (20870.15) 与各阶段之和也偏差较大；LTX2UpsampleStage 值出奇地低。

作者 mickqian 回复：已在 ba85c65f commit 中通过替换为一致的 CI 快照解决。对于 LTX2UpsampleStage=3.23ms，解释为直接取自源 CI 日志。

- fastwan2_2_ti2v_5b 场景 expected_avg_denoise_ms 与步长平均值不一致 (correctness): 作者使用一致的 CI 快照替换场景数据后，平均值与聚合字段匹配（132.31 vs 132.32，四舍五入后一致）。
- ltx_2_3_hq_pipeline 场景多处内部不一致 (correctness): 作者通过替换为一致的 CI 快照解决，确认 LTX2UpsampleStage=3.23ms 直接从日志提取，解释为请求级开销和日志直接值。

风险与影响

- 风险：数据准确性风险：若基线仍存在不一致，可能导致 CI 性能测试误通过或误报警。但作者已使用来自同一 CI 运行的一致快照替换了之前混合的数据，并修复了 review 指出的问题，该风险已降至较低。 server_args.py 的变更仅为注释和参数顺序调整，无功能风险。 NPU 基线只添加了一个字段，风险极低。 总体风险较低。

- 影响：对用户无直接影响；对 CI 测试团队：收紧后的基线能更灵敏地检测回归，但可能也会因更紧的阈值导致更多与性能抖动相关的失败；对开发者：未来性能改进除非超过 5%，否则不会触发基线更新。影响范围仅限于扩散模型在 H100 GPU 上的性能 CI 测试。
- 风险标记：基线数据一致性，CI 稳定性依赖，低功能风险

关联脉络

- PR #25457 [diffusion] add memory-aware component load order: 该 PR 提供了用于收紧基线的 CI 运行源（Base CI run），PR body 中明确引用。