PR #15528 完整报告

sgl-project/sglang

[CI] dynamic load-balanced partitioning for diffusion CI

合并时间：2026-04-12 13:02

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/15528

PR 15528 分析报告

执行摘要

本 PR 通过引入动态负载均衡分区机制，优化了 sglang 仓库中扩散模型 CI 测试的平衡性和总运行时间，主要改动涉及 CI 工作流、测试运行脚本和新增工具脚本，旨在解决静态分区导致的时长不均问题，提升开发效率。

功能与动机

为什么做：当前扩散 CI 测试使用静态分区，导致部分分片远长于其他分片，延长了 CI 总周转时间。PR body 中明确指出，此变更旨在实现运行时感知的分区，使扩散 CI 任务更均衡，减少等待时间。

实现拆解

做了什么：

- CI 工作流层：修改 .github/workflows/pr-test-multimodal-gen.yml，添加 compute-diffusion-partitions 任务，动态计算分区矩阵，并更新 multimodal-gen-test-1-gpu 和 multimodal-gen-test-2-gpu 使用动态分区数量。
- 测试运行层：重构 python/sglang/multimodal_gen/test/run_suite.py，替换固定轮询分区为 LPT（Longest Processing Time）算法，支持参数化用例和独立文件分区，并生成执行报告。
- 工具脚本层：新增 scripts/ci/utils/diffusion/ 下的脚本：
  - diffusion_case_parser.py：使用 AST 解析提取测试用例信息。
  - compute_diffusion_partitions.py：基于估算时间计算动态分区。
  - verify_diffusion_coverage.py：验证测试覆盖率。
- 数据层：更新 python/sglang/multimodal_gen/test/server/perf_baselines.json，添加 estimated_full_test_time_s 字段，为分区提供时间估算。

评论区精华

讨论了什么：

- 代码重复：gemini-code-assist[bot] 指出 compute_partitions.py 和 run_suite.py 中存在常量重复，Prozac614 已修复。

- 最大分区限制：mickqian 建议设置硬限制，Prozac614 回应已在脚本中添加 max-partitions 参数。
- case 列表维护：mickqian 询问是否需要维护相同 GPU 数的 case 列表，Prozac614 认为无需维护，计划简化。
- 服务器启动时间估算：mickqian 提到 estimated_full_test_time_s 应包括服务器启动时间，Prozac614 计划硬编码估算，但尚未完全解决。

风险与影响

风险：

- 动态分区算法（LPT）可能计算错误，导致分区不均或 CI 失败。
- AST 解析脚本对代码结构敏感，变更可能破坏解析逻辑。
- 覆盖率验证可能遗漏边缘情况，如独立文件执行失败。
- 估算时间不准确（如未计入服务器启动时间）可能影响分区效果。

影响：

- 用户无直接影响。
- 系统：优化 CI 效率，减少总运行时间，但需维护新增估算字段。
- 团队：提升开发体验，但增加对动态分区逻辑的维护负担。

关联脉络

与历史 PR 的关系：

- 与 PR #21960（Component Accuracy PR）冲突需整合，表明 CI 功能的持续演进和跨团队协作。
- 近期历史 PR 如 #22609（更新 B200 测试时间）和 #22602（优化 CI 依赖下载）显示仓库对 CI 效率的持续关注，本 PR 是这一趋势的一部分，专注于扩散测试的负载均衡。