

PR #29084 完整报告

sgl-project/sglang

[AMD][DI][CI] 1/N: MI355X disaggregation nightly benchmark

合并时间: 2026-06-26 10:21

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29084>

执行摘要

- 一句话: AMD MI355X 2 节点 1P1D 分解式夜间基准测试
- 推荐动作: 值得仔细阅读, 特别是其自包含 CI 设计、GSM8K 门控与性能测量的结合方式、以及自动镜像解析策略。该 PR 可作为 AMD 平台后续模型基准测试的模板。

功能与动机

Brings full nightly coverage of all four DeepSeek-V4 (Flash/Pro, FP8/FP4) 2-node 1P1D PD-disaggregation runs to AMD MI355X, with correctness gated before performance so regressions surface immediately. The design is intentionally sglang-native and self-contained. (来自 PR body)

实现拆解

1. 工作流与矩阵生成: 在 `.github/workflows/nightly-amd-mi355x-disagg.yml` 中新增 `nightly-amd-mi355x-disagg` 工作流, 通过 `generate_matrix.py` 从 `nightly-configs.yaml` 读取 4 个配置 (Flash/Pro x FP8/FP4) 生成矩阵, 每个配置独立运行, `fail-fast: false`。
2. 自动镜像解析: 工作流的 `setup job` 通过 Docker Hub API 解析 `lmsysorg/sglang-rocm` 仓库中最新的 `rocm720-mi35x` 标签, 传递给自托管 runner, 确保每次夜间测试使用最新镜像。
3. 启动脚本: `scripts/ci/slurm/launch_mi355x.sh` 是核心启动器, 通过 `salloc` 分配 2 节点 (1P1D), 在 Docker 容器中依次启动: `prefill server`、`decode server`、`sgl-router` 负载均衡器、GSM8K 准确性门控 (8-shot, 阈值 0.91), 最后执行 `bench_serving` 并发扫描 (1→256), 每个并发输出单独 JSON 结果。
4. Recipe 配置: 每个模型组合有独立的 YAML recipe (如 `mi355x-fp4/dsv4pro/1k1k/1p1d.yaml`), 定义资源分配、SGLang 参数 (TP8/EP1/DP1)、运行时镜像、RoCE 设备、并发列表和准确性门控参数。
5. 结果处理与总结: `process_result.py` 通过环境变量 `HW` 区分硬件 (默认 GB200), `summarize.py` 支持动态标题并按硬件显示结果; 所有结果 JSON 通过 workflow 的 `collect-results job` 上传。

关键文件:

- `scripts/ci/slurm/launch_mi355x.sh` (模块 启动脚本; 类别 `infra`; 类型 `infrastructure`; 符号 `emit`): 核心启动脚本, 包含节点分配、Docker 运行、GSM8K 门控、并发扫描等完

整逻辑。

- `.github/workflows/nightly-amd-mi355x-disagg.yml` (模块 CI 工作流; 类别 `infra`; 类型 `infrastructure`) : GitHub Actions 工作流定义, 包含矩阵生成、镜像解析、Slurm 清理等关键步骤。
- `scripts/ci/slurm/recipes/mi355x-fp4/dsv4pro/1k1k/1p1d.yaml` (模块 基准配置; 类别 `infra`; 类型 `infrastructure`) : DeepSeek-V4-Pro FP4 配置 `recipe`, 定义资源、SGLang 参数和基准设置。
- `scripts/ci/slurm/recipes/mi355x-fp8/dsv4pro/1k1k/1p1d.yaml` (模块 基准配置; 类别 `infra`; 类型 `infrastructure`) : DeepSeek-V4-Pro FP8 配置 `recipe`。
- `scripts/ci/slurm/recipes/mi355x-fp4/dsv4flash/1k1k/1p1d.yaml` (模块 基准配置; 类别 `infra`; 类型 `infrastructure`) : DeepSeek-V4-Flash FP4 配置 `recipe`。
- `scripts/ci/slurm/recipes/mi355x-fp8/dsv4flash/1k1k/1p1d.yaml` (模块 基准配置; 类别 `infra`; 类型 `infrastructure`) : DeepSeek-V4-Flash FP8 配置 `recipe`。
- `scripts/ci/slurm/nightly-configs.yaml` (模块 矩阵配置; 类别 `infra`; 类型 `infrastructure`) : 统一矩阵配置文件, 包含 4 个模型条目的 `model_path`、`config_file` 等。
- `scripts/ci/slurm/summarize.py` (模块 结果汇总; 类别 `infra`; 类型 `infrastructure`) : 报告总结脚本, 需要适应多硬件标题。
- `scripts/ci/slurm/process_result.py` (模块 结果处理; 类别 `infra`; 类型 `infrastructure`) : 结果处理脚本, 增加硬件环境变量以支持多平台。
- `scripts/ci/slurm/generate_matrix.py` (模块 矩阵生成; 类别 `infra`; 类型 `infrastructure`) : 生成矩阵脚本, 增加 `model_path` 支持。

关键符号: `emit`

关键源码片段

`scripts/ci/slurm/launch_mi355x.sh`

核心启动脚本, 包含节点分配、Docker 运行、GSM8K 门控、并发扫描等完整逻辑。

```
# Resolve HF cache snapshot 通过 refs/main 指向最新快照
if [[ -f "$MODEL_PATH/refs/main" && -d "$MODEL_PATH/snapshots" ]]; then
    SNAP_HASH="$(cat "$MODEL_PATH/refs/main")"
    RESOLVED="$MODEL_PATH/snapshots/$SNAP_HASH"
    if [[ -d "$RESOLVED" ]]; then
        echo "resolved snapshot: $MODEL_PATH -> $RESOLVED"
        MODEL_PATH="$RESOLVED"
    else
        echo "WARNING: refs/main points to $RESOLVED but directory missing, using as-is" >&2
    fi
fi

# 根据精度设置 FP4 专家标志
# PRECISION=fp4 时启用 SGLANG_DSV4_FP4_EXPERTS, 否则无
if [[ "$PRECISION" == "fp4" ]]; then
    EXPERT_FLAGS="-e SGLANG_DSV4_FP4_EXPERTS=1"
```

```

else
    EXPERT_FLAGS=""
fi

# 通过 GSM8K 正确性门控 (8-shot), 阈值 0.91
# 失败则退出, 不执行后续性能扫描
echo "=== GSM8K Accuracy Gate ==="
python3 -m sglang.test.few_shot_gsm8k \
    --num-shots 8 --num-questions 1319 \
    --threshold 0.91 \
    --host "http://$LB_HOST:$LB_PORT" || exit 1

```

评论区精华

1. 使用上游镜像: functionstackx 建议使用 lmsysorg/sglang-rocm 上游夜间镜像而非私有仓库, HaiShaw 赞同, Lzy17 最终切换并验证。
 2. 负载均衡器选择: functionstackx 询问是否可用正式的 sgl-router 替代自定义 standalone_lb.py, Lzy17 确认替换并删除自定义脚本, 利用镜像中已有的 sglang_router.launch_router 模块。
 3. process_result.py 硬件标识: HaiShaw 起初反对将硬编码 "gb200" 改为环境变量, Lzy17 解释 MI355X 需要不同标识且默认保持不变, HaiShaw 随后撤回异议。
- 使用上游 Docker 镜像 (design): 切换至 lmsysorg/sglang-rocm, 所有 recipe 和工作流镜像引用均已更新。
 - 负载均衡器选择: sgl-router vs standalone_lb.py (design): 采用 python -m sglang_router.launch_router 替代 standalone_lb.py, 提升可维护性并验证生产 LB 路径。
 - process_result.py 硬件标识的默认值 (design): 保留环境变量方案, 默认 HW=gb200, MI355X 工作流显式设置 HW=mi355x。

风险与影响

- 风险:
 1. 镜像兼容性: 自动解析最新镜像可能引入未测试的依赖变更, 导致启动失败或性能偏离; 需确保 recipe 中的回退镜像与解析逻辑一致。
 2. GSM8K 门控误判: 准确性阈值 0.91 为固定值, 若模型微调后准确率自然波动可能导致门控误报, 但当前 GSM8K 基准稳定。
 3. Slurm 资源竞争: 矩阵中 4 个模型独立运行, 若 runner 资源不足可能导致排队超时; fail-fast: false 确保部分失败不影响其他任务。
 4. 共享脚本改动: process_result.py 和 summarize.py 是 GB200 和 MI355X 共享的, 需确保条件分支不影响已有工作流; 经差分测试, GB200 路径输出字节一致。
- 影响:
 1. AMD CI 流程: 新增独立的夜间基准测试工作流, 为 MI355X 提供持续的 DeepSeek-V4 性能与准确性监控, 可早期发现回归。

2. 跨硬件兼容：共享的结果处理脚本做了硬件无关化改造（HW 环境变量），但保持向后兼容；GB200 工作流不受影响。
3. 团队工作量：引入新维护负担（4 个 recipe、启动脚本、镜像解析），但设计高度复用现有矩阵 / 结果层，增加维护成本有限。
4. 可扩展性：启动脚本支持多 P/D 拓扑（通过 recipe 资源字段），后续可增加 wideEP 或多 prefill 节点配置。 - 风险标记：新 CI 工作流，硬件依赖，镜像兼容性，GSM8K 阈值

关联脉络

- 暂无明显关联 PR