

PR #29661 完整报告

sgl-project/sglang

[AMD] Sgl-data mount opt-in

合并时间: 2026-06-29 22:11

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29661>

执行摘要

- 一句话: AMD CI 缓存挂载改为可选
- 推荐动作: 该 PR 是一个典型的 CI 基础设施优化, 设计清晰, 通过环境变量控制行为具有良好的可扩展性。值得关注的点包括: `ENABLE_CACHE_HOST` 的多值 case 处理设计、重型任务的筛选逻辑、以及 review 中提到的 AITER 缓存未配置问题。建议阅读 `scripts/ci/amd/amd_ci_start_container.sh` 中 case 块的实现, 以及 workflow 中添加环境变量的模式。总体变更简单, 但为 AMD CI 的缓存策略提供了灵活基础。

功能与动机

当前所有 AMD CI 任务都强制挂载 `/sgl-data` PVC, 但并非所有任务都需要如此大的缓存。PR 旨在让 PVC 挂载变为可选, 仅对缓存密集型任务启用, 从而减少对 PVC 的依赖, 提高 CI 基础设施的弹性。PR body 指出: 'make AMD CI `/sgl-data` PVC mounting opt-in via `ENABLE_CACHE_HOST` - keep `/sgl-data` as the container cache path, but use container-local storage by default - enable the PVC mount only for selected heavy AMD CI jobs'。

实现拆解

实现分为以下步骤:

1. 容器启动脚本改造 (`scripts/ci/amd/amd_ci_start_container.sh`): 引入 `ENABLE_CACHE_HOST` 环境变量, 通过 case 语句支持多种取值 (`1/true/yes/on/pvc/persistent`)。当设置为启用时, 检查缓存目录是否存在 (若不存在则报错退出), 并挂载 `-v $CACHE_HOST:/sgl-data`; 当设置为禁用或未设置时, 传递空卷轴, 容器使用本地存储。同时增加帮助信息输出, 提示该环境变量的用途。容器启动后, 通过 `docker exec` 创建 `/sgl-data` 下的四个子目录: `hf-cache/hub`、`pip-cache`、`miopen-cache`、`aiter-kernels`。
2. Nightly 工作流启用: 在 `.github/workflows/nightly-test-amd-rocm720.yml` 和 `.github/workflows/nightly-test-amd.yml` 中, 为重型任务 (DeepSeek-V、Kimi-K2.6、Qwen3、MiniMax-M2.7/2.5 等) 的 `amd_ci_start_container.sh` 步骤添加 `ENABLE_CACHE_HOST: "1"` 环境变量。非重型任务 (如 Grok2、diffusion 等) 保持不变, 默认使用本地缓存。

3. PR workflow部分启用：在 `.github/workflows/pr-test-amd-rocm720.yml` 中，仅对 ROCm 7.2 的两个 DeepSeek-V4 PR 测试任务添加 `ENABLE_CACHE_HOST: "1"`，重用 DeepSeek-V4 的缓存。
4. 测试与验证：通过 `bash -n` 检查脚本语法，Python 解析 YAML 确认每个启用任务的正确性，并验证无 `grok2` 任务错误启用 PVC 挂载。

关键文件：

- `scripts/ci/amd/amd_ci_start_container.sh`（模块 CI 脚本；类别 infra；类型 infrastructure）：核心变更文件，实现了 `ENABLE_CACHE_HOST` 环境变量的解析和 PVC 挂载逻辑的开关控制。
- `.github/workflows/nightly-test-amd-rocm720.yml`（模块 CI 工作流；类别 infra；类型 infrastructure）：为 ROCm 7.2 nightly 的重型任务添加 `ENABLE_CACHE_HOST: "1"`，共计 22 处添加。
- `.github/workflows/nightly-test-amd.yml`（模块 CI 工作流；类别 infra；类型 infrastructure）：为 AMD nightly 的重型任务添加 `ENABLE_CACHE_HOST: "1"`，共计 19 处添加。
- `.github/workflows/pr-test-amd-rocm720.yml`（模块 CI 工作流；类别 infra；类型 infrastructure）：为 ROCm 7.2 PR 的两个 DeepSeek-V4 测试任务添加 `ENABLE_CACHE_HOST: "1"`，共 2 处。

关键符号：未识别

关键源码片段

`scripts/ci/amd/amd_ci_start_container.sh`

核心变更文件，实现了 `ENABLE_CACHE_HOST` 环境变量的解析和 PVC 挂载逻辑的开关控制。

```
# 容器启动脚本片段：处理 ENABLE_CACHE_HOST 环境变量
CACHE_HOST=/home/runner/sglang-data
ENABLE_CACHE_HOST="${ENABLE_CACHE_HOST:-0}"

# 支持多种形式的启用 / 禁用取值，实现 opt-in 机制
case "${ENABLE_CACHE_HOST,,}" in # 转为小写后匹配
  1|true|yes|on|pvc|persistent)
    if [[ ! -d "$CACHE_HOST" ]]; then
      echo "Error: ENABLE_CACHE_HOST=1 but ${CACHE_HOST} does not exist." >&2
      exit 1 # 目录不存在时显式退出，避免静默失败
    fi
    CACHE_VOLUME="-v $CACHE_HOST:/sgl-data"
    echo "Mounting persistent CI data: ${CACHE_HOST} -> /sgl-data"
    ;;
  0|false|no|off|)
    CACHE_VOLUME=""
    echo "Not mounting ${CACHE_HOST}; /sgl-data will be container-local."
    ;;
  *)
```

```
echo "Error: unsupported ENABLE_CACHE_HOST='${ENABLE_CACHE_HOST}'" >&2
echo "Use 1/true/pvc/persistent or 0/false/off." >&2
exit 1
;;
esac
```

```
# 容器启动后初始化缓存子目录
docker exec ci_sglang mkdir -p \
  /sgl-data/hf-cache/hub \
  /sgl-data/pip-cache \
  /sgl-data/miopen-cache \
  /sgl-data/aiter-kernels
```

评论区精华

Review 中 [gemini-code-assist\[bot\]](#) 提出两条建议：

-  自动创建缓存目录：当 `ENABLE_CACHE_HOST=1` 但 `/home/runner/sglang-data` 不存在时，建议用 `mkdir -p` 自动创建而非直接报错，以提升用户友好性。该建议未被采纳（当前版本保持显式失败）。
-  AITER kernels 缓存未配置：容器内创建了 `/sgl-data/aiter-kernels`，但未向容器传递 `TRITON_CACHE_DIR` 或类似环境变量，导致该目录不会被 Triton 或 AITER 使用，缓存无法持久化。该问题未在本次 PR 中修复，需后续关注。
- 自动创建缓存目录的健壮性 (correctness): 未采纳，当前版本保持显式失败退出。
- AITER kernels 缓存未配置环境变量 (question): 未在本次 PR 中修复，需后续关注。

风险与影响

- 风险：
 1. 缓存目录不可用风险：若 `ENABLE_CACHE_HOST=1` 但指定目录不存在，脚本会直接退出 (`exit 1`)，可能导致 CI 任务中断。需确保对应节点上目录已预创建。
 2. AITER 缓存未生效：尽管创建了 `/sgl-data/aiter-kernels` 目录，但未配置环境变量指向该路径，编译过的 kernels 仍存于容器本地（如 `~/.triton/cache`），无法通过 PVC 持久化，降低了缓存复用效果。
 3. 配置遗漏风险：新增的 10+ 个 `ENABLE_CACHE_HOST: "1"` 分布在三个 workflow 文件中，未来若增加新任务需手动添加，可能遗漏。
 4. 回归风险：默认行为从挂载 PVC 变为不挂载，会影响依赖 `/sgl-data` 预缓存的任务性能，但任务列表已仔细筛选。
 - 影响：对 AMD CI 基础设施的影响中等：重型任务继续享受缓存加速，轻型任务释放 PVC 资源。对用户无直接影响，因为只涉及 CI 配置。对团队维护成本略有增加，需关注新任务的缓存策略是否匹配。
 - 风险标记：CI 配置变更，缓存策略调整，缺少 `TRITON_CACHE_DIR` 配置

关联脉络

- 暂无明显关联 PR