

PR #26478 完整报告

sgl-project/sglang

[AMD] [CI] Register MI35x GSM8K nightly tests

合并时间: 2026-05-28 10:31

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/26478>

执行摘要

- 一句话: 将 MI35x 上两个模型测试迁移为 nightly 测试
- 推荐动作: 建议关注 nightly 运行结果, 确保测试稳定通过; 可在后续将类似资源密集型测试逐步迁移至 nightly。

功能与动机

原测试注册在 PR CI 中, 但运行时间长、资源占用大, 不适合每次 PR 都执行。通过迁移到 nightly, 可减轻 PR CI 负担, 同时保持对关键模型精度和 HiCache 功能的夜间回归覆盖。

实现拆解

1. 修改 GLM-5.1-MXFP4 测试文件(test_glm51_mxfp4_tp2_gsm8k_mi35x.py): 将注册 suite 从 stage-c-test-large-8-gpu-amd-mi35x 改为 nightly-amd-2-gpu-mi35x-glm51-mxfp4, 添加 nightly=True; 移除 _get_model_path() 和本地路径硬编码, 直接使用模型 ID; 改用 CustomTestCase; 用 run_eval 替代旧的 run_gsm8k_eval; GSM8K 参数调整为全量问题 (num_examples=None) 和高并发 (num_threads=512)。
2. 修改 DeepSeek-R1 HiCache 测试文件(test_deepseek_r1_hicache_mi35x.py): 类似迁移, suite 改为 nightly-amd-8-gpu-mi35x-deepseek-r1-hicache; GSM8K 问题数从 200 扩至全量; 服务器启动超时从 1500s 增至 3600s; 移除手动设置的 HF_HOME 环境变量。
3. 更新 ROCm 7.2 工作流(nightly-test-amd-rocm720.yml): 在作业列表中添加 nightly-2-gpu-mi35x-glm51-mxfp4-rocm720 和 nightly-8-gpu-mi35x-deepseek-r1-hicache-rocm720, 并定义对应 job 配置, 包括 runner 选择、docker 启动、依赖安装和执行命令。
4. 更新标准工作流(nightly-test-amd.yml): 相同改动, 添加 nightly-2-gpu-mi35x-glm51-mxfp4 和 nightly-8-gpu-mi35x-deepseek-r1-hicache 作业。

关键文件:

- test/registered/amd/accuracy/mi35x/test_glm51_mxfp4_tp2_gsm8k_mi35x.py (模块测试; 类别 test; 类型 test-coverage; 符号 _get_model_path, TestGLM51MXFP4TP2GSM8KMI35x, test_gsm8k_accuracy, test_gsm8k) : 核心变更文件: 将 GLM-5.1-MXFP4 精度测试从 PR 级迁移到 nightly, 重写了测试类、评估框架和参数配置。

- `test/registered/amd/test_deepseek_r1_hicache_mi35x.py` (模块 测试; 类别 `test`; 类型 `test-coverage`) : 将 DeepSeek-R1 HiCache 测试迁移为 `nightly`, 扩大了 GSM8K 问题规模并调整超时参数。
- `.github/workflows/nightly-test-amd.yml` (模块 部署脚本; 类别 `infra`; 类型 `infrastructure`) : workflow 文件: 新增 MI35x 专属 `nightly` 作业, 定义运行时资源配置和测试命令。
- `.github/workflows/nightly-test-amd-rocm720.yml` (模块 部署脚本; 类别 `infra`; 类型 `infrastructure`) : ROCm 7.2 workflow, 与标准 workflow 同步添加对应作业。

关键符号: `test_gsm8k`, `test_gsm8k_accuracy`

关键源码片段

`test/registered/amd/test_deepseek_r1_hicache_mi35x.py`

将 DeepSeek-R1 HiCache 测试迁移为 `nightly`, 扩大了 GSM8K 问题规模并调整超时参数。

```
"""MI35x DeepSeek-R1-0528 FP8 HiCache Nightly Test (8-GPU)
```

```
Regression guard for HiCache with full GSM8K evaluation.
Registry: nightly-amd-8-gpu-mi35x-deepseek-r1-hicache suite
"""
```

```
import os, shutil, tempfile, unittest
from types import SimpleNamespace
```

```
from sglang.srt.utils import kill_process_tree
from sglang.test.ci.ci_register import register_amd_ci
from sglang.test.run_eval import run_eval
from sglang.test.test_utils import (
    DEFAULT_URL_FOR_TEST,
    CustomTestCase,
    is_in_ci,
    popen_launch_server,
    write_github_step_summary,
)
```

```
# 注册 nightly, 超时 5400 秒 (1.5 小时)
register_amd_ci(
    est_time=5400,
    suite="nightly-amd-8-gpu-mi35x-deepseek-r1-hicache",
    nightly=True,
)
```

```
DEEPSEEK_R1_MODEL_PATH = "deepseek-ai/DeepSeek-R1-0528"
SERVER_LAUNCH_TIMEOUT = 3600
GSM8K_ACCURACY_THRESHOLD = 0.93
GSM8K_NUM_EXAMPLES = None # 全量 1319 题
GSM8K_NUM_THREADS = 64
```

```

class TestDeepSeekR1HiCacheMI35x(CustomTestCase):
    @classmethod
    def setUpClass(cls):
        cls.model = DEEPSEEK_R1_MODEL_PATH
        cls.base_url = DEFAULT_URL_FOR_TEST
        cls.l3_storage_dir = tempfile.mkdtemp(prefix="dsr1-hicache-l3-")
        env = {
            **os.environ,
            "SGLANG_USE_AITER": "1",
            "ROCM_QUICK_REDUCE_QUANTIZATION": "NONE",
            "SGLANG_AITER_FP8_PREFILL_ATTN": "0",
            "SAFETENSORS_FAST_GPU": "1",
            "SGLANG_HICACHE_FILE_BACKEND_STORAGE_DIR": cls.l3_storage_dir,
        }
        other_args = [
            "--tp", "8",
            # ... (HiCache specific arguments)
            "--enable-hierarchical-cache",
            "--hicache-storage-backend", "file",
        ]
        cls.process = popen_launch_server(
            cls.model, cls.base_url,
            timeout=SERVER_LAUNCH_TIMEOUT,
            env=env,
            other_args=other_args,
        )

    @classmethod
    def tearDownClass(cls):
        if cls.process:
            kill_process_tree(cls.process.pid)
        if cls.l3_storage_dir:
            shutil.rmtree(cls.l3_storage_dir, ignore_errors=True)

    def test_gsm8k(self):
        args = SimpleNamespace(
            base_url=self.base_url,
            model=self.model,
            eval_name="gsm8k",
            api="completion",
            num_examples=GSM8K_NUM_EXAMPLES,
            num_threads=GSM8K_NUM_THREADS,
            max_tokens=512,
            temperature=0.0,
        )
        metrics = run_eval(args)
        score = metrics["score"]
        if is_in_ci():
            write_github_step_summary(

```

```
    """### DeepSeek-R1-0528 FP8 HiCache GSM8K (MI35x)

    """
    f"| Model | Examples | Max Parallel | Score | Threshold | Latency |\n"
    f"| {self.model} | full | {GSM8K_NUM_THREADS} | {score:.3f} | {GSM8K_ACCURACY_
    THRESHOLD:.2f} | {metrics.get('latency', 0):.1f}s |\n"
    )
    self.assertGreaterEqual(score, GSM8K_ACCURACY_THRESHOLD)
```

评论区精华

无关键讨论。仅有 Gemini Code Assist bot 自动评论，汇总了变更内容，未提出具体问题或争议。

- 暂无高价值评论线程

风险与影响

- 风险：

1. 测试延迟发现问题：这两个测试从 PR CI 中移除，回归问题可能直到第二天 nightly 运行时才被发现，增加调试成本。
2. 资源消耗：GLM-5.1 测试使用 2 GPU，DeepSeek-R1 使用 8 GPU，nightly 运行可能长时间占用 MI35x 资源，影响其他 nightly 任务排期。
3. 参数调整风险：GLM-5.1 测试的并发线程从 1200 降到 512，但问题集改为全量（原 1200），可能导致测试时间大幅增加，超时风险上升。
4. HuggingFace 路径依赖：移除了本地路径 fallback，测试完全依赖在线 Hub 或 HF_HOME 环境变量，若网络或缓存异常可能导致失败。

- 影响：

- 开发者：PR CI 速度提升，不再等待这两个长测试。
- 系统：nightly CI 新增两个长期作业，总运行时间增加约 90-120 分钟。
- 团队：需养成每日检查 nightly 报告的习惯，以捕获可能的回归。
- 测试层级：形成了“PR 级快速验证 + Nightly 全量覆盖”的分层架构，是 CI 优化的良好实践。
- 风险标记：测试降级，资源占用，网络依赖

关联脉络

- 暂无明显关联 PR