PR #23848 完整报告

sgl-project/sglang

[AMD] Add Kimi-K2.6 in nightly tests for MI30x and MI35x

合并时间：2026-05-05 14:37

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23848

执行摘要

- 一句话：AMD 夜间测试从 K2.5 切换到 K2.6
- 推荐动作：值得关注的是 AMD 团队复用 K2.5 配置的策略，它验证了模型架构兼容性的实用检查。对于 review 中提到的 try-finally 清理模式，建议在后续 PR 中统一修改 accuracy 测试的服务器生命周期管理，以减少 CI 中的不确定失败。此外，可考虑将重复的报表生成函数抽取为公共工具，降低维护成本。

功能与动机

根据 PR 描述，Kimi-K2.6 与其前身 K2.5 共享相同架构，HF 配置差异仅为 eos_token_id（163585 → 163586），因此部署方法可直接复用。替换后确保 AMD 夜间 CI 覆盖最新模型版本，保持与上游同步，并移除已过时的 K2.5 测试调度。

实现拆解

1. 创建 K2.6 测试文件：在 test/registered/amd/accuracy/ 和 test/registered/amd/perf/ 下的 mi30x 与 mi35x 目录各新增一个测试文件，共计 4 个新文件。这些测试复用 K2.5 的 AMD 服务器参数：混合 aiter prefill + triton decode，环境变量 SGLANG_USE_AITER=1 和 SGLANG_ROCM_FUSED_DECODE_MLA=0。
2. 准确率测试：每个测试类（TestKimiK26EvalAMD、TestKimiK26EvalMI35x）启动 8-GPU 服务端，执行 GSM8K few-shot 评估（1319 题），准确率阈值设定为 0.92，结果写入 CI 步骤摘要。MI30x 版本在 setUpClass/tearDownClass 中管理服务进程；MI35x 版本在测试方法内部使用 try...finally 确保清理。
3. 性能测试：使用 NightlyBenchmarkRunner 对批大小 [1, 8, 16, 64] 进行基准测试，生成简化 Markdown 报表（跳过首次 warmup 结果）。MI30x 和 MI35x 的脚本结构一致，仅 GPU 配置名和预期值不同。
4. 工作流文件更新：在 nightly-test-amd.yml 和 nightly-test-amd-rocm720.yml 中，将 K2.5 相关的 job 定义、job_select 选项和 check-all-jobs.needs 条目全部替换为 K2.6。K2.5 测试文件保留但不再被调度。
5. CI 验证：已在分支上成功运行全部准确率测试（MI30x ~52min，MI35x ~27min），准确率均 ≥0.92。性能测试文件尚未接入定期调度，需要手动触发。

关键文件：

- test/registered/amd/accuracy/mi30x/test_kimi_k26_eval_amd.py（模块 准确率测试；类别 test；类型 test-coverage；符号 TestKimiK26EvalAMD, setUpClass, tearDownClass, test_kimi_k26_gsm8k_accuracy）：Kimi-K2.6 MI30x 准确率测试入口，定义服务器启动和 GSM8K 评估流程
- test/registered/amd/accuracy/mi35x/test_kimi_k26_eval_mi35x.py（模块 准确率测试；类别 test；类型 test-coverage；符号 TestKimiK26EvalMI35x, setUpClass, test_kimi_k26_gsm8k_accuracy）：Kimi-K2.6 MI35x 准确率测试，服务器在测试方法内启动并保证清理
- test/registered/amd/perf/mi30x/test_kimi_k26_perf_amd.py（模块 性能测试；类别 test；类型 test-coverage；符号 generate_simple_markdown_report, TestNightlyKimiK26Performance, setUpClass, test_bench_kimi_k26）：Kimi-K2.6 MI30x 性能基准测试，定义批处理吞吐量和 ITL 指标
- test/registered/amd/perf/mi35x/test_kimi_k26_perf_mi35x.py（模块 性能测试；类别 test；类型 test-coverage；符号 generate_simple_markdown_report, TestNightlyKimiK26PerformanceMI35x, setUpClass, test_bench_kimi_k26）：Kimi-K2.6 MI35x 性能基准测试，结构与 MI30x 版本对称，仅 GPU 配置不同
- .github/workflows/nightly-test-amd.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：主工作流将 K2.5 的 job 定义、选择器和依赖全部替换为 K2.6
- .github/workflows/nightly-test-amd-rocm720.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：ROCm 7.2 版工作流同步替换 K2.5 为 K2.6，保持两个运行时的一致性

关键符号：generate_simple_markdown_report, test_bench_kimi_k26, test_kimi_k26_gsm8k_accuracy, setUpClass, tearDownClass

关键源码片段

test/registered/amd/accuracy/mi30x/test_kimi_k26_eval_amd.py

Kimi-K2.6 MI30x 准确率测试入口，定义服务器启动和 GSM8K 评估流程

```
"""AMD Kimi-K2.6 GSM8K Completion Evaluation Test (8-GPU)"""
import os, unittest
from types import SimpleNamespace
import requests
from sglang.srt.utils import kill_process_tree
from sglang.test.ci.ci_register import register_amd_ci
from sglang.test.few_shot_gsm8k import run_eval as run_eval_few_shot_gsm8k
from sglang.test.test_utils import (
    DEFAULT_URL_FOR_TEST, CustomTestCase, is_in_ci,
    popen_launch_server, write_github_step_summary,
)

# 注册到 AMD CI 套件（约 60 min）
register_amd_ci(est_time=3600, suite="nightly-amd-accuracy-8-gpu-kimi-k26", nightly=True)

KIMI_K26_MODEL_PATH = "moonshotai/Kimi-K2.6"
ACCURACY_THRESHOLD = 0.92
```

```
TP_SIZE = 8

class TestKimiK26EvalAMD(CustomTestCase):
    """Kimi-K2.6 GSM8K 准确率测试，MI325 硬件。"""
    @classmethod
    def setUpClass(cls):
        # 启动 8-GPU 服务，使用与 K2.5 相同的混合注意力后端
        cls.model = KIMI_K26_MODEL_PATH
        cls.base_url = DEFAULT_URL_FOR_TEST
        other_args = [
            "--tp", str(TP_SIZE),
            "--decode-attention-backend", "triton",
            "--prefill-attention-backend", "aiter",
            "--trust-remote-code",
            "--model-loader-extra-config", '{"enable_multithread_load": true}',
        ]
        env = os.environ.copy()
        env["SGLANG_USE_AITER"] = "1"
        env["SGLANG_ROCM_FUSED_DECODE_MLA"] = "0"
        cls.process = popen_launch_server(
            cls.model, cls.base_url, timeout=3600,
            other_args=other_args, env=env,
        )

    @classmethod
    def tearDownClass(cls):
        kill_process_tree(cls.process.pid)

    def test_kimi_k26_gsm8k_accuracy(self):
        """执行 GSM8K few-shot 评估并断言准确率。"""
        requests.get(self.base_url + "/flush_cache")
        args = SimpleNamespace(
            num_shots=8, data_path=None, num_questions=1319,
            parallel=1319, max_new_tokens=512,
            host="http://127.0.0.1",
            port=int(self.base_url.split(":")[-1]),
        )
        metrics = run_eval_few_shot_gsm8k(args)
        acc = metrics["accuracy"]

        # CI 环境下输出 Markdown 摘要到 GITHUB_STEP_SUMMARY
        if is_in_ci():
            summary = "### Kimi-K2.6 Model (MI325)

"
            summary += "| Model | TP | Accuracy | Threshold | Status |\n"
            summary += "| ----- | -- | -------- | --------- | ------ |\n"
            status = "✅ PASS" if acc >= ACCURACY_THRESHOLD else "❌ FAIL"
            summary += f"| {KIMI_K26_MODEL_PATH} | {TP_SIZE} | {acc:.3f} | {ACCURACY_
```

```
            THRESHOLD} | {status} |\n"
            write_github_step_summary(summary)

        self.assertGreaterEqual(acc, ACCURACY_THRESHOLD,
            f"Kimi-K2.6 accuracy {acc:.3f} below threshold {ACCURACY_THRESHOLD}")
```

评论区精华

Review 由 gemini-code-assist[bot] 提出三点改进建议：

- 在 test_kimi_k26_eval_amd.py 中，使用 setUpClass/tearDownClass 管理服务器进程不如 try...finally 模式健壮，若 setUp 超时将导致进程泄漏。该问题在 MI35x 版本已采用更安全的模式，但 MI30x 版本未修改。
- 性能测试中的 generate_simple_markdown_report 函数与 ITL 计算逻辑在两个文件中重复，建议合并到 nightly_bench_utils 共享模块。
- MI35x 性能测试文件中硬编码了 /data2/models/huggingface 作为 HF_HOME 默认值，影响可移植性，建议交由 CI 环境变量管理。 另外，HaiShaw 审核后直接批准，未进一步讨论。
- 测试服务器进程清理鲁棒性 (testing): 评论建议采用 try...finally 模式，但 MI30x 版本未被修改，CI 中仍保留泄漏风险。
- 报告生成逻辑代码重复 (design): 建议抽离为共享工具函数，但当前 PR 未做任何复用改进。
- 硬编码路径影响可移植性 (design): 评论建议通过 CI 环境变量管理此类路径，而非硬编码默认值。

风险与影响

- 风险：主要风险均为测试基础设施层面：
  1. 进程清理不健壮：test_kimi_k26_eval_amd.py 使用 setUpClass/tearDownClass，若 setUpClass 中服务器启动超时或失败，tearDownClass 不会执行，可能导致 CI 节点上残留进程。
  2. 可移植性：MI35x 性能测试硬编码了绝对路径 /data2/models/huggingface 作为 HF 缓存目录，在其他环境（如开发者本地或不同 CI 节点）可能因路径不存在而失败。
  3. 代码重复：generate_simple_markdown_report 在两个性能测试文件中完全重复，后续若需修改需同时维护两份，易产生不一致。不影响核心运行时逻辑，但增加测试维护成本。- 影响：影响范围：AMD 夜间测试套件；准确率阈值 0.92 与以前一致，不改变功能。正向影响：CI 覆盖官方最新模型 Kimi-K2.6，K2.5 测试文件保留但不再自动触发，实现平滑过渡。潜在负担：性能测试文件未接入定期调度，需手动触发，可能遗漏性能回归。此外，review 建议的改进未被采纳，未来可能积累技术债。团队影响：AMD CI 维护者需知晓 K2.6 测试的启动参数和清理模式差异。
- 风险标记：硬编码路径影响可移植性，服务器进程清理可能泄漏，报告代码重复，性能测试未接入定时调度

关联脉络

- 暂无明显关联 PR