

PR #30613 完整报告

sgl-project/sglang

[AMD] Nightly Test Coverage - Minimax-M3-MXFP8 Accuracy Test

合并时间: 2026-07-26 14:52

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/30613>

执行摘要

- 一句话: AMD MI35x 新增 MiniMax-M3 MXFP8 精度测试
- 推荐动作: 此 PR 主要面向 AMD 平台维护者和 CI 配置参考, 不涉及推理逻辑变更, 无需精读。但测试文件中针对推理模型 chat+thinking 模式的评估方法、环境变量配置等细节值得参考。

功能与动机

关联 issue #31742 已修复, 需要新增 nightly 测试覆盖以确保 MiniMax-M3 在 AMD 平台上的精度不退化。

实现拆解

1. 新增测试文件: 在 test/registered/amd/accuracy/mi35x/ 下创建 test_minimax_m3_tp4_eval_mi35x.py, 定义 ModelConfig 数据类和测试配置 (TP=4, MXFP8, aiter 后端, fp8 KV 缓存), 实现 run_gsm8k_benchmark 函数以 chat+thinking 模式 (推理模型的正确评估方式) 执行 GSM8K 测试, 并编写 TestMiniMaxM3TP4EvalMI35x 单元测试类。
2. 配置 CI 工作流: 分别在 nightly-test-amd.yml (ROCm 7.0) 和 nightly-test-amd-rocm720.yml (ROCm 7.2) 中添加 nightly job nightly-4-gpu-mi35x-minimax-m3, 包含 docker 环境准备、依赖安装、精度测试执行步骤, 并注册到 suite 矩阵。
3. 参数调优历程: 通过多轮 CI 验证调整, 最终确定使用 aiter 后端 + fp8 KV 缓存 + 禁用 radix 缓存的配置, 在 ROCm 7.0 和 7.2 上均达到 0.97+ 精度阈值。

关键文件:

- test/registered/amd/accuracy/mi35x/test_minimax_m3_tp4_eval_mi35x.py (模块 精度测试; 类别 test; 类型 test-coverage; 符号 ModelConfig, post_init, get_display_name, get_answer_value): 新增的测试文件, 定义了模型配置、GSM8K 评估逻辑和测试类, 是 PR 核心。
- .github/workflows/nightly-test-amd-rocm720.yml (模块 CI 配置; 类别 infra; 类型 infrastructure): 在 ROCm 7.2 CI 工作流中添加 nightly job 以执行 MI35x 上的精度测试。
- .github/workflows/nightly-test-amd.yml (模块 CI 配置; 类别 infra; 类型 infrastructure): 在 ROCm 7.0 CI 工作流中添加 nightly job 以执行 MI35x 上的精度测试。

关键符号: ModelConfig, post_init, get_display_name, get_answer_value, run_gsm8k_benchmark, query, TestMiniMaxM3TP4EvalMI35x, setUpClass

关键源码片段

[test/registered/amd/accuracy/mi35x/test_minimax_m3_tp4_eval_mi35x.py](#)

新增的测试文件，定义了模型配置、GSM8K 评估逻辑和测试类，是 PR 核心。

```
# 测试配置文件，定义模型参数和推理环境
from dataclasses import dataclass
from typing import Optional, List

@dataclass
class ModelConfig:
    """Configuration for a model to test."""
    model_path: str
    tp_size: int = 4
    accuracy_threshold: float = 0.93
    other_args: Optional[List[str]] = None
    env_vars: Optional[dict] = None
    timeout: Optional[int] = None
    variant: Optional[str] = None

    def __post_init__(self):
        if self.other_args is None:
            self.other_args = []
        if self.env_vars is None:
            self.env_vars = {}

    def get_display_name(self) -> str:
        if self.variant:
            return f"{self.model_path} ({self.variant})"
        return self.model_path

# 具体测试配置：使用 aiter 注意力后端、fp8 e4m3 KV 缓存、禁用 radix 缓存
MI35X_MINIMAX_M3_TP4_MODELS = [
    ModelConfig(
        model_path="MiniMaxAI/MiniMax-M3-MXFP8",
        tp_size=4,
        accuracy_threshold=0.95,
        timeout=5400,
        variant="TP4+MXFP8+aiterAttn+fp8KV",
        other_args=[
            "--quantization", "mxfp8",
            "--dtype", "bfloat16",
            "--trust-remote-code",
            "--attention-backend", "aiter",
            "--kv-cache-dtype", "fp8_e4m3",
            "--disable-radix-cache",
```

```

        "--chunked-prefill-size", "8192",
        "--mem-fraction-static", "0.80",
        "--watchdog-timeout", "1200",
    ],
    env_vars={
        "SGLANG_USE_AITER": "1",
        # ROCm 7.0 的 rocBLAS 不支持 bf16 输入 /fp32 输出 router GEMM,
        # 强制 fp32 路由路径以获得更精确的专家路由。
        "SGLANG_OPT_USE_BF16_ROUTER_GEMM": "0",
    },
),
]

```

评论区精华

1. 数据泄露风险: bot 评论指出早期版本中 few-shot 示例与评估问题重叠, 建议从 num_shots 偏移开始评估。该问题在后续切换为 chat+thinking 模式后不再相关 (无需 few-shot) 。
 2. 缺乏异常 traceback: bot 建议在 except 块中添加 traceback.print_exc() 以辅助调试。该建议未在最终代码中采纳。
- Few-shot 数据泄露问题 (correctness): 未在最终版本中体现 (但已通过模式切换规避) 。
 - 缺少异常 traceback 打印 (testing): 未采纳 (最终代码中未添加) 。

风险与影响

- 风险:
 1. 测试配置偏差: 若 Aiter 后端或 fp8 KV 缓存行为在未来变更, 可能导致测试假阳性 / 假阴性。当前配置与真实部署场景可能存在差异 (如 radix 缓存禁用) 。
 2. CI 耗时增加: 新增 job 最长期望 90 分钟, 可能延长 nightly CI 总时长。
 3. 硬件依赖: 测试仅在 MI35x (gfx950) 上运行, 其他 AMD GPU 无法覆盖。 - 影响: 对用户无直接影响 (仅 CI 测试)。对系统增加 nightly CI 开销, 但通过 job_filter 机制可选择性执行。对团队增强了 AMD 平台模型的回归防护, 降低精度下降风险。影响程度中等。 - 风险标记: 测试配置偏差风险, CI 执行时间增加, 特定硬件依赖

关联脉络

- 暂无明显关联 PR