

PR #53155 完整报告

vllm-project/vllm

[ROCm] Keep GLM-5.2 on MRV1 and disable default breakable cudagraph

合并时间: 2026-09-01 12:17

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/53155>

执行摘要

- 一句话: ROCm 修复 GLM-5.2 误入 MRV2, 默认关 breakable cudagraph
- 推荐动作: 值得精读, 尤其是负责 ROCm / model runner / 配置系统的同学。亮点在于: 一是用 frozenset 常量 + lru_cache 平台查询函数集中管理平台默认行为, 配合 monkeypatch 与 cache_clear() 形成低成本测试模式; 二是 perf 归因方法学 (4 组对照分解 MRV2 与 breakable cudagraph 各自的回退贡献) 值得借鉴, 避免只靠环境变量 workaround 留下约 23% 的性能缺口。同时应记录 mawong-amd 的跟进建议, 在 #53712 合并后复测并考虑是否需要反转部分默认值。

功能与动机

PR#52861 将 DSA 模型默认路由到 V2 model runner (MRV2) 与 breakable CUDA graphs, 但 ROCm 排除集只覆盖了 DeepseekV32ForCausalLM / DeepseekV4ForCausalLM, 漏掉了 GlmMoeDsaForCausalLM。PR body 明确指出: "GLM-5.2 (both FP8 and MXFP4) thus became the only DSA model defaulting to MRV2 + breakable cudagraph on ROCm, which regresses decode latency on gfx950 (MI355X) by ~30–37% TPOT at batch 1 in nightly perf-eval." 通过 4 组对照实验分解出 MRV2 贡献约 3.0 ms、breakable cudagraph 贡献约 1.2 ms 的回退, 因此需要同时改两处才能完全恢复基线。

实现拆解

1. 变更入口: vllm/config/vllm.py 顶部的平台默认配置区, 包含 ROCM_DEFAULT_MRV1_ARCHITECTURES 常量与 default_breakable_cudagraph_architectures() 函数。
2. MRV1 排除集补漏: 把 GlmMoeDsaForCausalLM 加入 ROCM_DEFAULT_MRV1_ARCHITECTURES (原来只有 DeepseekV32 / DeepseekV4 两个架构)。该集合让命中架构在 ROCm 上 use_v2_model_runner 返回 False, 从而留在编译过的 MRV1 路径, 这是修复约 3.0 ms 回退的关键。
3. ROCm breakable 默认全关: default_breakable_cudagraph_architectures() 的 ROCm 分支从 " 排除 Deepseek 两个架构 " 改为直接返回空集 frozenset()。原因是 gfx950 上 breakable cudagraph 对 GLM-5.2 贡献约 1.2 ms 回退, 且当前 ROCm 平台没有任何架构从中受益; VLLM_USE_BREAKABLE_CUDAGRAPH=1 的 opt-in 路径保持不变。
4. 测试同步: tests/test_config.py 更新 test_rocm_keeps_compiled_deepseek_defaults 的 docstring 与断言 (纳入 GlmMoeDsaForCausalLM), 并把

test_dsa_breakable_cudagraph_platform_default 参数化用例中 ("GlmMoeDsaForCausalLM", True, True) 翻转为 ("GlmMoeDsaForCausalLM", True, False)。运行方式为 `pytest tests/test_config.py -k "rocm or dsa or breakable"`。

5. 验证与 CI: 在 gfx950 (MI355X) TP8 上用 vllm bench serve 做对照实验, median TPOT 恢复至 13.46 ms, 与 #52861 之前的好基线一致; 启动日志确认不再出现 Using V2 Model Runner / Breakable CUDA graph enabled; 上游 CI (Buildkite #86417) 与 AMD CI (#12493) 通过。

关键文件:

- vllm/config/vllm.py (模块 配置系统; 类别 source; 类型 core-logic; 符号 ROCM_DEFAULT_MRV1_ARCHITECTURES, default_breakable_cudagraph_architectures): 核心配置逻辑所在: ROCM_DEFAULT_MRV1_ARCHITECTURES 补入 GlmMoeDsaForCausalLM, default_breakable_cudagraph_architectures() 在 ROCm 上改为返回空集。这两处决定了 GLM-5.2 在 ROCm 上回到编译的 MRV1 路径, 并让所有架构默认不启用 breakable cudagraph, 是修复 30~37% TPOT 回退的关键。
- tests/test_config.py (模块 配置测试; 类别 test; 类型 test-coverage; 符号 test_rocm_keeps_compiled_deepseek_defaults, test_dsa_breakable_cudagraph_platform_default): 同步覆盖新默认行为: 断言 GLM-5.2 仍在 ROCM_DEFAULT_MRV1_ARCHITECTURES 且不在 ROCm 的 breakable 默认集合中, 并把参数化用例里 GLM 在 ROCm 上的期望值翻转为 False, 防止后续回归。

关键符号: default_breakable_cudagraph_architectures,
test_rocm_keeps_compiled_deepseek_defaults,
test_dsa_breakable_cudagraph_platform_default

关键源码片段

vllm/config/vllm.py

核心配置逻辑所在: ROCM_DEFAULT_MRV1_ARCHITECTURES 补入 GlmMoeDsaForCausalLM, default_breakable_cudagraph_architectures() 在 ROCm 上改为返回空集。这两处决定了 GLM-5.2 在 ROCm 上回到编译的 MRV1 路径, 并让所有架构默认不启用 breakable cudagraph, 是修复 30~37% TPOT 回退的关键。

```
# vllm/config/vllm.py —— ROCm 平台 DSA 模型默认运行路径配置 (PR#53155 之后)
```

```
# TODO(rocm): 这些模型在 AMD GPU 上要么不被 MRV2 支持, 要么在 MRV2 下更慢。
```

```
# 命中该集合的架构在 ROCm 上会留在编译过的 MRV1 路径。
```

```
ROCM_DEFAULT_MRV1_ARCHITECTURES = frozenset(  
    {  
        "DeepseekV32ForCausalLM",  
        "DeepseekV4ForCausalLM",  
        "GlmMoeDsaForCausalLM", # PR#53155 补上漏掉的 GLM-5.2, 避免其默认走 MRV2  
    }  
)
```

```
# 各平台默认启用 breakable CUDA graph 的架构名单 (PR#52861 引入 DSA 路由)。
```

```

DEFAULT_BREAKABLE_CUDAGRAPH_ARCHITECTURES = frozenset(
    {
        "DeepseekV32MTPModel",
        "DeepseekV32ForCausalLM",
        "DeepseekV4ForCausalLM",
        "DeepSeekV4MTPModel",
        "Dots3NoteForCausalLM",
        "Dots3NoteMTPModel",
        "GlmMoeDsaForCausalLM",
        "HYV4ForCausalLM",
        "HYV4MTPModel",
        "InklingForCausalLM",
        "InklingForConditionalGeneration",
        "KimiK3ForConditionalGeneration",
        "KimiK3MTPModel",
        "KimiLinearForCausalLM",
        "MiniMaxM3SparseForCausalLM",
        "MiniMaxM3SparseForConditionalGeneration",
    }
)

```

```

@lru_cache
def default_breakable_cudagraph_architectures() -> frozenset[str]:
    """返回当前平台默认启用 breakable CUDA graph 的架构集合。"""
    from vllm.platforms import current_platform

    if current_platform.is_rocm():
        # gfx950 实测 breakable cudagraph 贡献约 1.2 ms / 请求 的回归,
        # 因此 ROCm 默认不启用任何架构; 用户仍可用
        # VLLM_USE_BREAKABLE_CUDAGRAPH=1 强制开启。
        return frozenset()
    return DEFAULT_BREAKABLE_CUDAGRAPH_ARCHITECTURES

```

tests/test_config.py

同步覆盖新默认行为: 断言 GLM-5.2 仍在 `ROCM_DEFAULT_MRVL_ARCHITECTURES` 且不在 ROCm 的 breakable 默认集合中, 并把参数化用例里 GLM 在 ROCm 上的期望值翻转为 False, 防止后续回归。

```

@pytest.mark.parametrize(
    ("architecture", "is_rocm", "expected"),
    [
        ("DeepseekV32ForCausalLM", False, True),
        ("DeepseekV32ForCausalLM", True, False),
        ("GlmMoeDsaForCausalLM", False, True),
        # PR#53155 翻转: ROCm 上 GLM-5.2 默认不再启用 breakable cudagraph
        ("GlmMoeDsaForCausalLM", True, False),
    ],
)

```

```

def test_dsa_breakable_cudagraph_platform_default(
    monkeypatch, architecture, is_rocm, expected
):
    from vllm.config.vllm import default_breakable_cudagraph_architectures
    from vllm.platforms import current_platform

    monkeypatch.delenv("VLLM_USE_BREAKABLE_CUDAGRAPH", raising=False)
    monkeypatch.setattr(current_platform, "is_rocm", lambda: is_rocm)
    # 平台查询函数带 lru_cache, 须清缓存才能让 monkeypatch 的平台判断生效
    default_breakable_cudagraph_architectures.cache_clear()
    config = SimpleNamespace(
        model_config=SimpleNamespace(architectures=[architecture]),
        compilation_config=CompilationConfig(),
    )
    config._uses_breakable_cudagraph_by_default = lambda: (
        VllmConfig._uses_breakable_cudagraph_by_default(config)
    )

    try:
        assert VllmConfig._maybe_enable_breakable_cudagraph(config) is expected
        if expected:
            assert config.compilation_config.mode == CompilationMode.NONE
    finally:
        os.environ.pop("VLLM_USE_BREAKABLE_CUDAGRAPH", None)
        default_breakable_cudagraph_architectures.cache_clear()

```

评论区精华

唯一实质性讨论来自 mawong-amd 的 Issue 评论: "Should revisit the perf numbers here after <https://github.com/vllm-project/vllm/pull/53712>, and maybe also let DSV4 go back on MRV2"——即 #53712 合并后需要重新评估本 PR 的 perf 结论, 并可能让 DeepSeek-V4 重新回到 MRV2 路径。该建议未被深入讨论, 作为后续跟进项。claude[bot] 因 fork 来源未自动审查; 维护者 AndreasKaratzas 直接 APPROVED (空 body)。此外 mergify[bot] 曾提示合并冲突, 作者 rebase main 后解决。

- PR#53712 合并后重新评估 ROCm MRV2 性能与 DSV4 归属 (question): 本 PR 先行合并恢复基线; 重新评估留待 #53712 之后跟进。
- fork PR 自动 review 不可用 (other): AndreasKaratzas 直接 APPROVED (空 body), 无实质 review 评论。
- 合并冲突与 rebase (other): 已解决, 最终 head 为 5037b989070a, CI 重新跑通。

风险与影响

- 风险:
 1. ROCm 默认行为全局变更: default_breakable_cudagraph_architectures() 在 ROCm 上直接返回空集, 意味着当前在 ROCm 上性能可能尚可的架构 (如 KimiK3、Dots3Note、MiniMaxM3 等) 也无法默认启用 breakable cudagraph, 属于 "一刀切" 策略, 后续需

按架构逐个回填。

2. 后续结论可能被推翻：mawong-amd 指出 #53712 合并后需重新评估 perf；若 #53712 改善 ROCm 上 MRV2/breakable 性能，本 PR 的部分默认值（尤其 DSV4 归属）可能需要翻转。
3. 测试覆盖限制：作者说明 pytest 未在 perf 容器内实际执行，仅靠 CI 覆盖；tests/test_config.py 的断言依赖 lru_cache 清理 (cache_clear()) 与 monkeypatch 模拟平台，若平台判断逻辑演进需同步维护。
4. MRV1 编译成本：恢复 MRV1 是编译路径，可能带来更长的启动编译时间，换取运行时 TPOT 收益；对强调快速启动的场景有轻微影响。
5. 非 ROCm 无影响：分支条件明确限定 current_platform.is_rocm()，NVIDIA/CPU 等平台不受影响。

• 影响：

1. 用户侧：ROCm gfx950 上运行 GLM-5.2 (FP8 / MXFP4) 的用户，batch 1 decode TPOT 从 17.74 ms 回落至 13.46 ms (约 -24%)，恢复至 #52861 之前的基线水平。
2. 系统侧：ROCm 平台上所有 DSA 架构默认回到 MRV1 路径且不再默认启用 breakable cudagraph；启动日志行为随之变化。
3. 团队侧：AMD 团队需要跟踪 #53712 合并后的 perf 复测，并决定是否让 DeepSeek-V4 / GLM-5.2 重新进入 MRV2；同时为 #41834 (NVIDIA SM12x DeepSeek-V4 enablement) 等后续工作提供 ROCm 对照基线。
4. 非 ROCm：完全不受影响，这是 PR body 明确声明的设计边界。 - 风险标记：ROCm 默认行为全局变更，配置默认值一刀切，依赖后续 PR 重新评估，真机 pytest 未在 PR 内执行

关联脉络

- PR #54646 [Core][MRV2] Freeze gc during V2 CG capture; skip per-descriptor cleanup: 同属 MRV2 / breakable cudagraph 性能优化线 (V2 cudagraph 捕获提速约 6 倍)，其进展可能影响本 PR 后续在 ROCm 上恢复 breakable cudagraph 默认值的评估结论。
- PR #54373 [Bugfix][Spec Decode] Take the DFlash draft's RoPE layout from its own config: 同属 GLM 模型线 (GLM-5.3 的 DFlash 草稿修复)，与 GLM-5.2 在 ROCm 上的回归修复互为对照，反映 GLM 系列在 v1 / 投机解码路径上的持续适配。