PR #52861 完整报告

vllm-project/vllm

[Model][NVIDIA] Route DSA models to the CUDA non-compiled path

合并时间：2026-08-19 15:13

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52861

执行摘要

- 一句话：DSA 模型改为默认走 CUDA 非编译 MRV2 路径
- 推荐动作：值得精读。这是一个典型的默认路径切换 + 平台分派 PR，建议关注三点： 1) vllm/models/deepseek_v32/__init__.py 的 CUDA/ 非 CUDA 导入分派如何与 registry.py 解耦并保持类名契约； 2) vllm/config/vllm.py 中架构集合 + 环境变量自动写入 + CompilationMode.NONE 的配置自动决策模式，可作为其他模型默认执行路径的样板； 3) capability-gated kernel 与 fallback 的硬件兼容策略。也需注意到其测试主要集中在配置层、真实硬件验证依赖 CI 的代价。

功能与动机

PR body 明确指出问题根源：The optimized NVIDIA classes are currently unreachable from the model registry. That leaves DeepSeek V3.2 and GLM-5.2 on the generic runner/graph path and prevents their MTP draft model from using the matching implementation. 即 deepseek_v32 的 NVIDIA 优化实现已存在但从未被注册表引用，导致 DSA 模型只能走通用路径，且 MTP draft 模型无法匹配专属实现。该 PR 是 #49790 的重发版本（原 PR 因分支同步后 commits 清零无法恢复），承接 #48597 跟踪的 NVIDIA DSA 路由项，并与 #51915（opt-in ROCm/MXFP4 正确性路径，不改默认路由）明确区分。

实现拆解

1. 模型注册与平台分派：vllm/model_executor/models/registry.py 将 DeepseekV32ForCausalLM、GlmMoeDsaForCausalLM 的映射从 deepseek_v2 改为 vllm.models.deepseek_v32，并新增 DeepseekV32MTPModel 注册；vllm/models/deepseek_v32/__init__.py 由 is_rocm/is_xpu 分派改为 is_cuda() 优先分派，CUDA 上导出 NVIDIA 实现并以 import alias 提供 GlmMoeDsaForCausalLM，其余平台回退通用 deepseek_v2/deepseek_mtp 实现，显式 AMD 模块保留供 opt-in。
2. 默认执行路径配置：vllm/config/vllm.py 把两个 DSA 架构加入 DEFAULT_V2_MODEL_RUNNER_ARCHITECTURES（ROCm 上排除），新增 DEFAULT_BREAKABLE_CUDAGRAPH_ARCHITECTURES 与 default_breakable_cudagraph_architectures()；新增 _uses_breakable_cudagraph_by_default() 与 _maybe_enable_breakable_cudagraph()：未显式设置环境变量且架构命中时自动写入 VLLM_USE_BREAKABLE_CUDAGRAPH=1，并把 CompilationMode 置为 NONE，保留 =0 的 opt-out。

3. 注意力与 KV cache 兼容：vllm/models/deepseek_v32/attention.py 删除 require_fp8_kv_cache 类属性与初始化断言，改为按 is_quantized_kv_cache(self.kv_cache_dtype) 动态推导 _fp8_query 与 _fp8_kv_needs_view，支持未量化 BF16 与标准 / 压缩 FP8 两种 KV cache 形式。
4. 推测解码配套：vllm/config/speculative.py 的 hf_config_override() 对 deepseek_v32/glm_moe_dsa 的 MTP draft 架构规范为 DeepseekV32MTPModel；移除 deepseek_v32 的 enforce_eager = True（原 FIXME 注释同步删除），MTP 首次进入 CUDA graph 捕获；vllm/v1/spec_decode/llm_base_proposer.py 小幅适配（具体符号未见 diff，依据变更路径推断）。
5. 测试与配置配套：tests/test_config.py 将原 test_rocm_defaults_deepseek_v4_to_mrv1 扩展为 ROCm 双默认断言，并新增 DSA 模型默认 MRV2 + breakable graph、平台默认、MTP 匹配等参数化测试；tests/compile/fusions_e2e/models.py 与 conftest.py 移除 DeepSeek V3.2 编译用例（不再走编译路径）；kernel 融合测试同步适配。CI 覆盖配置导入路径与 SM100 fused norm/RoPE。

关键文件：

- vllm/config/vllm.py（模块 配置层；类别 source；类型 configuration；符号 DEFAULT_V2_MODEL_RUNNER_ARCHITECTURES, DEFAULT_BREAKABLE_CUDAGRAPH_ARCHITECTURES, default_breakable_cudagraph_architectures, _uses_breakable_cudagraph_by_default）：默认执行路径的核心开关：新增 breakable CUDA graph 架构集与自动启用逻辑，把 DSA 架构纳入 MRV2 默认并定义 ROCm 排除规则。
- vllm/models/deepseek_v32/__init__.py（模块 模型入口；类别 source；类型 data-contract；符号 DeepseekV32ForCausalLM, GlmMoeDsaForCausalLM, DeepseekV32MTP）：平台分派核心：从 is_rocm/is_xpu 分派改为 is_cuda 优先，CUDA 上导出 NVIDIA 实现并 alias GLM-5.2 类，非 CUDA 平台回退通用实现。
- vllm/model_executor/models/registry.py（模块 模型注册表；类别 source；类型 data-contract；符号 DeepseekV32ForCausalLM, GlmMoeDsaForCausalLM, DeepseekV32MTPModel）：注册表契约变更：DSA 主模型与 MTP draft 模型的导入路径指向 vllm.models.deepseek_v32，使优化实现真正可达。
- vllm/config/speculative.py（模块 推测解码；类别 source；类型 core-logic；符号 SpeculativeConfig.hf_config_override, SpeculativeConfig.post_init）：MTP 推测解码配套：draft 架构规范化为 DeepseekV32MTPModel，并移除 deepseek_v32 的 enforce_eager 强制，解除 MTP 的 CUDA graph 限制。
- vllm/models/deepseek_v32/attention.py（模块 注意力层；类别 source；类型 core-logic；符号 DeepseekV32Attention.init）：移除强制 FP8 KV cache 断言，改为按 KV cache dtype 动态选择查询量化与视图变换，使 BF16 未量化 KV cache 也能走 NVIDIA DSA 路径。
- 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_models_default_to_mrv2_and_breakable_cudagraph, test_dsa_breakable_cudagraph_platform_default,

test_dsa_models_select_matching_mtp）：配置层测试覆盖最广：ROCm 保持 compiled默认、DSA 默认 MRV2 + breakable graph、平台默认与 MTP 匹配，是该 PR 行为的主要回归护栏。

- vllm/v1/spec_decode/llm_base_proposer.py（模块 提议模块；类别 source；类型 core-logic）：MTP proposer 的小幅适配（+5/-1），确保 DSA draft 模型走正确的非编译路径；具体 diff 未提供，推断与 torch.compile 或 CUDA graph 开关相关。

关键符号：default_breakable_cudagraph_architectures, default_v2_model_runner_architectures, _uses_breakable_cudagraph_by_default, _maybe_enable_breakable_cudagraph, VllmConfig._is_default_v2_model_runner_model, SpeculativeConfig.hf_config_override, DeepseekV32Attention.init

关键源码片段

vllm/config/vllm.py

默认执行路径的核心开关：新增 breakable CUDA graph 架构集与自动启用逻辑，把 DSA 架构纳入 MRV2 默认并定义 ROCm 排除规则。

```
# vllm/config/vllm.py
# MRV2 默认架构集合：新增 DeepseekV32ForCausalLM 与 GlmMoeDsaForCausalLM，
# 使 DSA 模型在 NVIDIA 上默认走 V2 model runner
DEFAULT_V2_MODEL_RUNNER_ARCHITECTURES = frozenset({
    "DeepseekV2ForCausalLM",
    "DeepseekV32ForCausalLM", # DeepSeek V3.2，本次新增
    "DeepseekV4ForCausalLM",
    "GlmMoeDsaForCausalLM", # GLM-5.2，本次新增
    "GraniteMoeForCausalLM",
    "InklingForCausalLM",
    "InklingForConditionalGeneration",
    "KimiK3ForConditionalGeneration",
    "LongcatFlashNgramForCausalLM",
    "Qwen2MoeForCausalLM",
})

# 默认启用可断 CUDA 图（FULL + PIECEWISE）的架构集合
DEFAULT_BREAKABLE_CUDAGRAPH_ARCHITECTURES = frozenset({
    "DeepseekV32MTPModel",
    "DeepseekV32ForCausalLM",
    "DeepseekV4ForCausalLM",
    "DeepSeekV4MTPModel",
    "GlmMoeDsaForCausalLM",
    "InklingForCausalLM",
    "InklingForConditionalGeneration",
    "KimiK3ForConditionalGeneration",
    "KimiK3MTPModel",
    "KimiLinearForCausalLM",
    "MiniMaxM3SparseForCausalLM",
    "MiniMaxM3SparseForConditionalGeneration",
```

```
})


@lru_cache
def default_breakable_cudagraph_architectures() -> frozenset[str]:
    """Architectures defaulting to breakable CUDA graphs on this platform."""
    from vllm.platforms import current_platform

    if current_platform.is_rocm():
        # ROCm 保留 DeepSeek V3.2 的编译 MRV1 路径，不自动启用可断图
        return DEFAULT_BREAKABLE_CUDAGRAPH_ARCHITECTURES - {
            "DeepseekV32ForCausalLM",
            "DeepseekV32MTPModel",
        }
    return DEFAULT_BREAKABLE_CUDAGRAPH_ARCHITECTURES


# 以下两个方法位于 VllmConfig 类内部（类体不完整，仅展示核心逻辑）
class VllmConfig:
    def _uses_breakable_cudagraph_by_default(self) -> bool:
        model_config = self.model_config
        if model_config is None:
            return False
        architectures = set(model_config.architectures)
        return bool(architectures & default_breakable_cudagraph_architectures())

    def _maybe_enable_breakable_cudagraph(self) -> bool:
        # 未显式设置环境变量且架构命中默认集合时，自动开启可断 CUDA 图
        if (
            "VLLM_USE_BREAKABLE_CUDAGRAPH" not in os.environ
            and self._uses_breakable_cudagraph_by_default()
        ):
            os.environ["VLLM_USE_BREAKABLE_CUDAGRAPH"] = "1"
            logger.info_once(
                "Auto-enabling VLLM_USE_BREAKABLE_CUDAGRAPH=1. "
                "Set VLLM_USE_BREAKABLE_CUDAGRAPH=0 to opt out."
            )

        from vllm.compilation.breakable_cudagraph import (
            is_breakable_cudagraph_enabled,
        )

        enabled = is_breakable_cudagraph_enabled()
        if enabled:
            # 可断图路径不再需要 torch.compile，编译模式直接置为 NONE
            self.compilation_config.mode = CompilationMode.NONE
        return enabled
```

vllm/models/deepseek_v32/__init__.py

平台分派核心：从 is_rocm/is_xpu 分派改为 is_cuda 优先，CUDA 上导出 NVIDIA 实现并 alias GLM-5.2 类，非 CUDA 平台回退通用实现。

```
# vllm/models/deepseek_v32/__init__.py
# 平台入口：CUDA 使用 NVIDIA 专属 DSA 实现，其余平台回退通用实现
from vllm.platforms import current_platform

if current_platform.is_cuda():
    # GLM-5.2（glm_moe_dsa）复用 CUDA DSA 模块；
    # 能力相关优化 kernel 在模块内部按算力 gate 并自动降级
    from .nvidia.model import DeepseekV32ForCausalLM
    from .nvidia.model import DeepseekV32ForCausalLM as GlmMoeDsaForCausalLM
    from .nvidia.mtp import DeepseekV32MTP
else:
    # ROCm、XPU、CPU 保持通用 deepseek_v2 / deepseek_mtp 实现
    from vllm.model_executor.models.deepseek_mtp import DeepSeekMTP as DeepseekV32MTP
    from vllm.model_executor.models.deepseek_v2 import (
        DeepseekV3ForCausalLM as DeepseekV32ForCausalLM,
    )
    from vllm.model_executor.models.deepseek_v2 import GlmMoeDsaForCausalLM

__all__ = [
    "DeepseekV32ForCausalLM",
    "DeepseekV32MTP",
    "GlmMoeDsaForCausalLM",
]
```

vllm/models/deepseek_v32/attention.py

移除强制 FP8 KV cache 断言，改为按 KV cache dtype 动态选择查询量化与视图变换，使 BF16 未量化 KV cache 也能走 NVIDIA DSA 路径。

```
# vllm/models/deepseek_v32/attention.py
# DeepseekV32Attention.__init__ 中 KV cache 与查询量化选择逻辑：
# 移除强制 FP8 KV cache 断言，未量化 BF16 默认启动也能走该路径
fp8_attention = is_quantized_kv_cache(self.kv_cache_dtype)
# FP8 时优先使用支持量化 query 输入的稀疏 MLA 后端
self._fp8_query = fp8_attention and self.impl.supports_quant_query_input
# fp8_ds_mla（FlashMLA 稀疏）布局无需额外 view 变换
self._fp8_kv_needs_view = fp8_attention and self.kv_cache_dtype != "fp8_ds_mla"
```

评论区精华

该 PR 来自 fork，claude[bot] 自动评审被禁用（This pull request is from a fork — automated review is disabled. A repository maintainer can comment @claude review to run a one-time review.），因此没有人工 review 评论。核心技术讨论实际沉淀在 commit 演进与维护者 CI 评论中：WoosukKwon 在 CI 结果里给出 NVFP4 TP4 dummy weights 下 concurrency 1 至 1024 的吞吐提升为 +16.42% 至 +8.29%，且每个并发档位候选 mean TPOT 均更低，同时附 GSM8K 94.768% 与 MTP 接受率 79.1% 数据；commit 历史显示早期

强制 FP8 KV cache 的设计在 Keep existing DSA FP8 cache requirement 与 Simplify DSA query and cache form selection 两轮中被放松为动态选择，属于已解决的设计权衡。

- Fork PR 自动评审被禁用 (other): 未产生人工 review 评论，代码质量由 CI 与维护者合入把关。
- NVFP4 TP4 性能对比与 MTP 验证 (performance): 路由切换生效且无明显精度回退，MTP=3 在 GB200 上带来显著加速。

风险与影响

- 风险：
  - 默认行为变更：所有 NVIDIA GPU 上 DeepSeek V3.2/GLM-5.2 从 generic compiled 路径切到 deepseek_v32 + MRV2 + breakable graph；PR body 声明 pre-SM100 GPU 与 DeepSeek V3.2 实机未做 benchmark，capability gate 与 kernel fallback 只能靠配置测试覆盖。
  - 全局环境变量写入：vllm/config/vllm.py 的 _maybe_enable_breakable_cudagraph() 会写入 os.environ["VLLM_USE_BREAKABLE_CUDAGRAPH"]，在多模型同进程或测试环境中可能产生跨模型副作用。
  - 编译覆盖下降：tests/compile/fusions_e2e 删除 DeepSeek V3.2 编译用例，若未来回退 MRV1 路径将缺少回归护栏。
  - MTP eager 强制移除：DeepSeek V3.2 MTP 首次进入 CUDA graph 捕获，图捕获失败风险主要依赖配置测试与 CI 把关。
  - KV cache 断言放宽：attention.py 不再强制 FP8，未量化 BF16 下若 backend 不支持量化 query 会自动走 _fp8_query=False 分支，可能掩盖性能或精度特征变化。
- 影响：
  - 用户侧：NVIDIA 用户默认获得 DSA 专属实现与 MTP 加速，DeepSeek V3.2 默认 KV cache 行为从强制 FP8 变为跟随指定 dtype；ROCm/XPU/CPU 用户无感知（保留通用路径）。
  - 系统侧：registry.py、vllm/config/vllm.py、speculative.py 三处数据契约同步变更；DEFAULT_BREAKABLE_CUDAGRAPH_ARCHITECTURES 同时覆盖 Inkling、KimiK3、MiniMaxM3 等架构，自动启用逻辑对它们的默认行为同样生效，存在连锁影响面。
  - 性能：GLM-5.2 在 4x GB200 上 MTP=3 输出吞吐提升约 2.34 倍（含 TTFT），NVFP4 TP4 下并发 1 至 1024 提升 8% 至 16%。
  - 团队：16 个 commit 呈现参与者早期实现 + 维护者与 Codex 多轮简化的协作模式，测试集中在配置层，真实模型验证依赖大型 GPU CI。
  - 风险标记：核心路由变更，跨模块配置契约，环境变量全局写入，pre-SM100 硬件无实机验证，编译测试用例移除

关联脉络

- PR #52929 Add NemotronH_Omni_Reasoning_V3 as a supported Nemotron architecture: 同改 vllm/config/speculative.py 与

vllm/model_executor/models/registry.py，同属 MTP 推测解码 + 新模型注册演进线。

- PR #52948 [Model] Support bidirectional (encoder-only) attention for DeepSeek e…: 同改 DeepSeek 模型系与 registry 相关文件，同属 DeepSeek 架构能力扩展。
- PR #52690 [Bugfix] Restore model info caching for package backends: 同改 vllm/model_executor/models/registry.py，涉及注册表加载路径与缓存行为，与本 PR 的注册契约变更存在交互。