PR #42509 完整报告

vllm-project/vllm

[ROCm][MLA] FP8 ASM prefill for AITER dense MLA backend on gfx950

合并时间：2026-05-15 23:56

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/42509

执行摘要

- 一句话：FP8 ASM 预填充加速 ROCm gfx950 MLA 预填充
- 推荐动作：值得精读，特别是如何设计自动检测与优雅回退、以及在元数据构建阶段预计算以避免 forward 中同步的技巧，对编写高性能 attention 后端有参考价值。

功能与动机

AITER 密集 MLA 后端默认使用 flash_attn_varlen_func 进行预填充，gfx950 上 AITER 提供了更高效的 FP8 ASM 预填充内核（mla_prefill_ps_asm_fwd + mla_reduce_v1），可显著降低 TTFT 并提高吞吐量。本 PR 自动启用该优化，无需用户调整配置。

实现拆解

1. 自动检测：新增 _fp8_mla_prefill_supported() 函数，使用 on_gfx950() 和导入 mla_prefill_ps_asm_fwd、mla_reduce_v1 来判断是否支持 FP8 ASM 预填充，结果被 LRU 缓存。
2. 元数据扩展：在 AiterMLAMetadata 中添加 10 个 fp8_prefill_* 可选字段，用于存储持久调度（PS）元数据，如 Q/KV 索引、work 信息、reduce 映射等。
3. 缓冲区预分配：AiterMLAMetadataBuilder.__init__ 中调用 _init_fp8_prefill_ps_buffers()，根据 max_num_reqs 和 max_prefill_qlen 通过 get_ps_metadata_info_v1 计算最大规模并预分配设备张量。
4. 元数据构建：在 build() 中，当有预填充批次且 FP8 预填充启用时，调用 _build_fp8_prefill_ps_metadata()，从 common_attn_metadata.query_start_loc_cpu 切片并填充 PS 元数据；同时预计算 num_partial_tiles 并存入 fp8_prefill_num_partial_tiles 以避免 forward 中的同步。
5. 前向分发：AiterMLAImpl.forward_mha 中，当预填充存在且 FP8 启用时，调用 _mla_fp8_prefill_attn(q, k, v, attn_metadata, output)，该函数使用 workspace manager 获取临时缓冲区，执行 mla_prefill_ps_asm_fwd 和 mla_reduce_v1 直接写入 output，否则回退到 flash_attn_varlen_func。
6. 性能优化：审查后消除了 .to("cpu") 和 .item() 同步、不必要的张量分配和复制，并使用 workspace manager 管理中间张量。

关键文件：

- vllm/v1/attention/backends/mla/rocm_aiter_mla.py（模块 注意力后端；类别 source；类型 core-logic；符号 _fp8_mla_prefill_supported, _init_fp8_prefill_ps_buffers, _build_fp8_prefill_ps_metadata, _mla_fp8_prefill_attn）：唯一的变更文件，实现了 FP8 ASM 预填充的全部逻辑，包括自动检测、元数据缓冲区预分配、元数据构建和前向分发。

关键符号：_fp8_mla_prefill_supported, _init_fp8_prefill_ps_buffers, _build_fp8_prefill_ps_metadata, _mla_fp8_prefill_attn, forward_mha

关键源码片段

vllm/v1/attention/backends/mla/rocm_aiter_mla.py

唯一的变更文件，实现了 FP8 ASM 预填充的全部逻辑，包括自动检测、元数据缓冲区预分配、元数据构建和前向分发。

```
# Auto-detect FP8 ASM prefill support
@functools.lru_cache(maxsize=1)
def _fp8_mla_prefill_supported() -> bool:
    """Check if platform is gfx950 and AITER provides the required kernels."""
    try:
        from vllm.platforms.rocm import on_gfx950
    except Exception:
        return False
    if not on_gfx950():
        return False
    try:
        from aiter import mla_prefill_ps_asm_fwd, mla_reduce_v1
    except Exception:
        return False
    return True

# FP8 prefill PS metadata fields in AiterMLAMetadata
@dataclass
class AiterMLAMetadata(MLACommonMetadata[AiterMLADecodeMetadata]):
    # ... original decode fields ...
    fp8_prefill_qo_indptr: torch.Tensor | None = None
    fp8_prefill_kv_indptr: torch.Tensor | None = None
    fp8_prefill_kv_indices: torch.Tensor | None = None
    fp8_prefill_work_indptr: torch.Tensor | None = None
    fp8_prefill_work_info_set: torch.Tensor | None = None
    fp8_prefill_reduce_indptr: torch.Tensor | None = None
    fp8_prefill_reduce_final_map: torch.Tensor | None = None
    fp8_prefill_reduce_partial_map: torch.Tensor | None = None
    fp8_prefill_max_q_len: int | None = None
    fp8_prefill_num_partial_tiles: int | None = None

# Buffer pre-allocation in builder __init__
class AiterMLAMetadataBuilder(...):
    def __init__(self, ...):
        super().__init__(...)
```

```
        self._fp8_prefill_enabled = _fp8_mla_prefill_supported()
        if self._fp8_prefill_enabled:
            max_prefill_qlen = min(
                vllm_config.model_config.max_model_len,
                vllm_config.scheduler_config.max_num_batched_tokens,
            )
            self._init_fp8_prefill_ps_buffers(
                max_num_reqs, max_prefill_qlen, device
            )
```

评论区精华

- gemini-code-assist: 指出 .to("cpu") 在元数据构建中引入同步，建议改用 common_attn_metadata.query_start_loc_cpu；.item() 在 forward 中导致同步应移至元数据构建；质疑 FP8 cast 是否冗余；指出不必要的分配和 copy_。
- tjtanaa: 要求简化 gqa_ratio 计算、使用 workspace manager 分配临时缓冲区、移除多余注释、让 _mla_fp8_prefill_attn 接受 output 参数直接写入。
- maeehart: 逐一回应并解决所有评论，在 commit 9bf64924 中统一修复：使用 CPU 切片、缓存 num_partial_tiles、保留必要 FP8 cast 并更新注释、消除中间分配、采用 workspace manager 模式。
  - 使用 .to("cpu") 导致 host-device 同步 (performance): maeehart 修改 _build_fp8_prefill_ps_metadata 以接收 common_attn_metadata 并切片 CPU 张量，消除了设备到主机拷贝。
  - 调用 .item() 在 forward 中导致同步 (performance): maeehart 将 num_partial_tiles 的计算移到 _build_fp8_prefill_ps_metadata，结果存入 fp8_prefill_num_partial_tiles forward 直接读取 int。
  - 冗余的 FP8 转换 (correctness): 保留显式 cast，更新注释以澄清必要性，避免未来混淆。
  - 不必要的张量分配和 copy_ 操作 (performance): maeehart 修改函数签名使其接受 out 参数，内部 view 后直接传入 ASM 和 reduce 内核，消除中间分配和复制。
  - 使用 workspace manager 管理临时缓冲区 (design): maeehart 采纳，在 _mla_fp8_prefill_attn 中通过 workspace manager 获取 scratch，并在函数返回前释放。

风险与影响

- 风险：技术风险较低，因为新增路径仅在 gfx950 + AITER 提供对应内核时启用，否则静默回退到原有 flash_attn_varlen_func。主要风险包括：依赖外部 aiter 库版本；预分配缓冲区增加少量显存占用；性能提升有限（TTFT -14.8%，输出吞吐 +2.3%），且 TPOT 略有噪声；测试未覆盖回退路径及多 segment 组合。
- 影响：影响范围限于 gfx950 用户（MI355X），他们将自动获得预填充性能提升，无需配置变更。其他硬件或配置不受影响。系统层面增加少量显存占用用于 PS 元数据缓冲区。团队需要维护新增的自动检测和元数据逻辑，但代码集中在单一文件且带有清晰注释。
- 风险标记：核心路径变更，依赖外部库，仅限 gfx950，缺少测试覆盖

关联脉络

- PR #42604 DeepSeekV4-Pro enable cuda graph full and piecewise mode: 同属 ROCm MLA 注意力后端优化系列，涉及同一硬件平台和模型族。