PR #52114 完整报告

vllm-project/vllm

[Model] [Quantization] Add Ling hybrid MXFP4 routed experts support

合并时间：2026-08-13 21:46

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52114

执行摘要

- 一句话：Ling 混合量化 MXFP4 路由专家加载支持
- 推荐动作：该 PR 值得快速浏览，重点查看 _configure_ling_fp8_quant_config 的元数据解析和 _LING_MXFP4_WEIGHTS_MAPPER 正则设计。改动轻量且局限于模型加载路径，影响面可控。建议后续补充针对混量 checkpoint 的加载回归测试，并考虑在 WeightsMapper 注释中说明命名约定的来源。

功能与动机

PR body 说明：Ling checkpoints 使用混合量化（block FP8 用于 dense 与共享专家投影，MXFP4 用于路由专家）。本次变更读取 Ling 特定量化元数据，并将路由专家 scale 名称重映射为 Mxfp4MoEMethod 期望的约定，适用于主模型与 MTP 模型。

实现拆解

1. 扩展配置解析：修改 vllm/model_executor/models/bailing_moe_v3.py 中的 _configure_ling_fp8_quant_config，新增 config: PretrainedConfig 参数。函数在确认是 block FP8 配置后，从 HF 的 quantization_config 中读取 scale_fmt（若为 ue8m0 则设置 quant_config.is_scale_e8m0）和 routed_experts_quant_method（若存在且不是 mxfp4 则抛出 ValueError，否则设置 quant_config.store_dtype = "mxfp4"）。
2. 新增权重名映射：定义 _LING_MXFP4_WEIGHTS_MAPPER（基于 WeightsMapper 的正则表，将 weight_scale_inv 后缀的 _inv 去掉）和 _maybe_remap_ling_mxfp4_weight_names（仅当 Fp8Config.store_dtype == "mxfp4" 时应用映射，否则原样透传）。
3. 主模型接入：在 BailingMoeV3ForCausalLM.load_weights 中，hf_to_vllm_mapper.apply 之后调用 _maybe_remap_ling_mxfp4_weight_names，确保 MXFP4 专家 scale 能被 Mxfp4MoEMethod 正确加载。
4. MTP 模型同步：在 bailing_moe_v3_mtp.py 中导入同一函数，BailingMoeV3MTPModel.__init__ 调用 _configure_ling_fp8_quant_config 时传入 draft config，load_weights 中同样应用名称重映射，保持主模型与 MTP 草稿模型的量化行为一致。
5. 测试配套：本次变更未新增或修改测试文件，依赖现有 CI 和量化路径回归。

关键文件：

- vllm/model_executor/models/bailing_moe_v3.py（模块 量化适配；类别 source；类型 data-contract；符号 _maybe_remap_ling_mxfp4_weight_names, _configure_ling_fp8_quant_config, _LING_MXFP4_WEIGHTS_MAPPER）：主模型加载路径，新增 Ling 混量元数据解析和 MXFP4 权重名重映射，是本次改动的核心。
- vllm/model_executor/models/bailing_moe_v3_mtp.py（模块 MTP 模型；类别 source；类型 data-contract；符号 BailingMoeV3MTPModel.load_weights, BailingMoeV3MTPModel.init）：MTP 模型同步接入混量配置与权重名重映射，保证主模型和草稿模型加载行为一致。

关键符号：_configure_ling_fp8_quant_config, _maybe_remap_ling_mxfp4_weight_names, _LING_MXFP4_WEIGHTS_MAPPER

关键源码片段

vllm/model_executor/models/bailing_moe_v3.py

主模型加载路径，新增 Ling 混量元数据解析和 MXFP4 权重名重映射，是本次改动的核心。

```
# Ling 采用混合量化：dense/ 共享 expert 用 block FP8，routed experts 用 MXFP4。
# 该函数把 HF 侧的量化元数据翻译成 vLLM 的 Fp8Config 字段。
def _configure_ling_fp8_quant_config(
    quant_config: QuantizationConfig | None,
    config: PretrainedConfig,
) -> None:
    # 只有在 block FP8 配置下才需要解析 Ling 的附加元数据
    if not _is_block_fp8_config(quant_config):
        return

    quant_config.ignored_layers_match_mode = "suffix"
    hf_quant_config = getattr(config, "quantization_config", None)
    if not isinstance(hf_quant_config, dict):
        return

    # Ling 的 scale_fmt 为 ue8m0 时，FP8 scale 采用 E8M0 格式
    quant_config.is_scale_e8m0 = ( # type: ignore[attr-defined]
        hf_quant_config.get("scale_fmt") == "ue8m0"
    )

    routed_quant_method = hf_quant_config.get("routed_experts_quant_method")
    if routed_quant_method is None:
        return
    if routed_quant_method != "mxfp4":
        raise ValueError(
            f"Unsupported routed experts quantization: {routed_quant_method!r}"
        )

    # 告知 Fp8Config 路由专家的存储格式为 MXFP4
    quant_config.store_dtype = "mxfp4"

# Ling checkpoint 中 routed expert 的 scale 参数名带有 "_inv" 后缀，
```

```
# 而 Mxfp4MoEMethod 期望的是不带后缀的普通 scale 名，这里用正则去掉后缀。
_LING_MXFP4_WEIGHTS_MAPPER = WeightsMapper(
    orig_to_new_regex={
        re.compile(
            r"(\.mlp\.experts\.\d+\."
            r"(?:gate_projlup_projldown_proj)\.weight_scale)_inv$"
        ): r"\1"
    }
)


def _maybe_remap_ling_mxfp4_weight_names(
    weights: Iterable[tuple[str, torch.Tensor]],
    quant_config: QuantizationConfig | None,
) -> Iterable[tuple[str, torch.Tensor]]:
    # 仅当 store_dtype 被标记为 mxfp4 时才重映射，避免影响纯 FP8 checkpoint
    if isinstance(quant_config, Fp8Config) and quant_config.store_dtype == "mxfp4":
        return _LING_MXFP4_WEIGHTS_MAPPER.apply(weights)
    return weights
```

评论区精华

该 PR 无实质性 review 评论。claude[bot] 提示因 fork 提交自动审查被禁用，维护者可评论 @claude review 触发一次性审查。合入者 jeejeelee 直接批准，CI 触发后通过。

- fork PR 自动审查未启用 (other): 未触发额外审查，合入者 jeejeelee 直接批准。

风险与影响

- 风险：
  1. 缺少测试覆盖：新增逻辑涉及量化元数据解析和权重名重映射，但没有对应单元测试。若 Ling checkpoint 的权重命名或量化元数据字段发生变化，回归难以被及时发现。
  2. 正则映射与命名约定耦合：_LING_MXFP4_WEIGHTS_MAPPER 的正则只匹配 gate_projlup_projldown_proj 的 weight_scale_inv。若 Ling 后续版本或其他模型的 scale 命名出现变体（如 input_scale_inv），映射会静默失效，导致权重加载错误或缺失。
  3. 依赖 HF 量化元数据字段：scale_fmt 与 routed_experts_quant_method 均为非标准字段，若上游 checkpoint 未携带或字段名变化，is_scale_e8m0 和 store_dtype 可能未被正确设置，影响量化精度。
  4. MTP config 来源：BailingMoeV3MTPModel 通过 _get_draft_hf_config 获取 config，若未配置 speculative decoding 会回退到主模型 config，重复设置幂等但逻辑路径需留意。 - 影响：对用户：启用 Ling 混合量化 checkpoint 时，路由专家权重将按 MXFP4 格式加载，dense 与共享专家仍为 block FP8，降低显存占用并保证加载正确性；对纯 FP8 checkpoint 无行为变化（映射函数直接透传）。对系统：只影响模型加载阶段，不涉及运行时 kernel 或推理路径。对团队：提供了一种处理混合量化权重名差异的可复用模式，后续其他模型可参考 WeightsMapper 正则重映射方式。 - 风险标记：缺少测试覆

盖，依赖 HF 量化元数据字段，正则映射耦合权重命名约定

关联脉络

- 暂无明显关联 PR