PR #52706 完整报告

vllm-project/vllm

[Model] Add GraniteSWA and GraniteMoeSWA via existing Granite

合并时间：2026-08-19 11:32

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52706

执行摘要

- 一句话：新增 GraniteSWA/MoeSWA 模型支持
- 推荐动作：值得精读，尤其关注 granite_layer_attn_params 的分层配置解析设计，以及 hmellor 推动的 MoE 权重加载重构；讨论中提到的 per_layer_config API 可作为后续配置演进的参考。

功能与动机

HF 在 5.15.1 发布 GraniteSWA 与 GraniteMoeSWA 后，vLLM 需要注册支持。原 PR #48270 用独立模型类实现，本 PR 改为复用现有实现，避免重复维护。PR body 特别强调 transformers 后端会静默丢弃 sink token、输出错误，只有 vLLM 原生实现才能保证正确性：「transformers backend does NOT handle these models correctly. Sink tokens get dropped silently and so the model gives wrong output」。

实现拆解

1. 统一分层参数解析：在 vllm/model_executor/models/granite.py 新增 granite_layer_attn_params(config, layer_idx)，返回 (sliding_window, rope_theta, has_sink)。普通 Granite 配置不含 layer_types / layer_rope_theta 时回退到 full attention、全局 RoPE、无 sink，保证兼容。
2. 注意力层按层构建：GraniteAttention（granite.py）与 GraniteMoeAttention（granitemoe.py）的 __init__ 均改为接收完整 config，通过 extract_layer_index(prefix) 定位当前层。rope_theta == 0 时不建 rotary_emb（NoPE）；has_sink 时创建 nn.Parameter 并用 sharded_weight_loader 按 TP 切分，最后传给 Attention 的 per_layer_sliding_window 与 sinks。
3. MoE 权重加载重构：granitemoe.py 删除 fused_moe_make_expert_params_mapping 手写映射，改用 hf_to_vllm_mapper + AutoWeightsLoader；granitemoeshared.py 删除自定义 load_weights 中逐专家拆分权重的逻辑，直接复用 GraniteMoeModel.load_weights，该清理由 hmellor 提交并验证 PowerMoE-3b 与 granite-swash-3b-a600m 输出一致。
4. 注册与配套：vllm/model_executor/models/registry.py 把 GraniteSWAForCausalLM 指到 granite.GraniteForCausalLM，GraniteMoeSWAForCausalLM 指到 granitemoeshared.GraniteMoeSharedForCausalLM；tests/models/registry.py 新增 _HfExamplesInfo 并限定 transformers ⩾ 5.15.1；tests/models/language/generation/test_granite.py 新增两个 SWA 检查点的 logprobs 一

致性测试与两个 helper 单元测试；docs/models/supported_models.md 同步登记。

关键文件：

- vllm/model_executor/models/granite.py（模块 模型实现；类别 source；类型 core-logic；符号 granite_layer_attn_params, GraniteAttention）：新增 granite_layer_attn_params 统一解析每层 sliding window / RoPE / sink，并改造 GraniteAttention，是本 PR 的核心设计。
- vllm/model_executor/models/granitemoe.py（模块 模型实现；类别 source；类型 refactor；符号 GraniteMoeAttention, GraniteMoeModel）：GraniteMoeAttention 复用共享 helper，删除手工权重拆分（净删 150 行），是改动量最大的文件。
- vllm/model_executor/models/granitemoeshared.py（模块 模型实现；类别 source；类型 refactor；符号 load_weights）：删除自定义 load_weights 权重拆分逻辑，直接复用 GraniteMoeModel 加载路径，支持 granitemoe_swa 检查点。
- vllm/model_executor/models/registry.py（模块 模型注册；类别 source；类型 configuration）：注册 GraniteSWAForCausalLM / GraniteMoeSWAForCausalLM 到现有实现类，是模型入口。
- tests/models/language/generation/test_granite.py（模块 模型测试；类别 test；类型 test-coverage；符号 test_granite_swa_features_are_off_without_swa_config, test_granite_swa_features_resolve_per_layer）：新增 SWA 检查点一致性测试与 per-layer helper 单元测试，验证默认关闭与新功能解析。
- tests/models/registry.py（模块 测试注册；类别 test；类型 test-coverage）：为新模型补充 _HfExamplesInfo 示例与最低 transformers 版本约束，决定测试矩阵。
- docs/models/supported_models.md（模块 支持文档；类别 docs；类型 documentation）：支持模型列表登记两个新模型，属用户可见文档。

关键符号：granite_layer_attn_params, GraniteAttention.init, GraniteMoeAttention.init, GraniteMoeModel._load_weights, GraniteMoeSharedForCausalLM.load_weights, test_granite_swa_features_are_off_without_swa_config, test_granite_swa_features_resolve_per_layer

关键源码片段

vllm/model_executor/models/granite.py

新增 granite_layer_attn_params 统一解析每层 sliding window / RoPE / sink，并改造 GraniteAttention，是本 PR 的核心设计。

```
# vllm/model_executor/models/granite.py
def granite_layer_attn_params(
    config: PretrainedConfig, layer_idx: int
) -> tuple[int | None, float, bool]:
    """解析某一层的 sliding window、RoPE base 与 sink 使用情况。

    普通 Granite 配置没有 SWA 字段，回退为 full attention、全局 RoPE base、
    无 sink。HF SWA 检查点使用 sinks 但没有专门标志，因此在存在 `layer_types`
    时假定启用，并允许 `attention_sinks` 字段覆盖该判断。
```

```
    返回:
        滑动窗口大小（`None` 表示 full attention）、RoPE base theta（`0` 表示
        NoPE）、该层是否使用 attention sink。
    """
    layer_types = getattr(config, "layer_types", None)
    sliding_window = (
        config.sliding_window
        if layer_types is not None and layer_types[layer_idx] == "sliding_attention"
        else None
    )

    layer_rope_theta = getattr(config, "layer_rope_theta", None)
    rope_theta = (
        layer_rope_theta[layer_idx]
        if layer_rope_theta is not None
        else config.rope_parameters["rope_theta"]
    )

    # HF 的 SWA 检查点没有显式 sink 标志：只要配置里出现了 `layer_types`
    # 就假定需要 sink，`attention_sinks` 字段可以显式覆盖这个默认值。
    has_sink = getattr(config, "attention_sinks", layer_types is not None)
    return sliding_window, rope_theta, has_sink
```

vllm/model_executor/models/granitemoe.py

GraniteMoeAttention 复用共享 helper，删除手工权重拆分（净删 150 行），是改动量最大的文件。

```
# vllm/model_executor/models/granitemoe.py（GraniteMoeAttention.__init__ 关键部分）
from .granite import granite_layer_attn_params
from .utils import extract_layer_index

# ... 在完成 qkv_proj / o_proj 构建后 ...
sliding_window, rope_theta, has_sink = granite_layer_attn_params(
    config, extract_layer_index(prefix)
)

# rope_theta 为 0 表示该层 NoPE，直接跳过 rotary_emb 构建；
# 否则用当前层的 theta 覆盖全局 rope_parameters。
self.use_rope = rope_theta != 0
if self.use_rope:
    self.rotary_emb = get_rope(
        self.head_dim,
        max_position=max_position,
        rope_parameters={**config.rope_parameters, "rope_theta": rope_theta},
        is_neox_style=True,
    )

# 每个注意力头一个可学习 sink 参数，按 TP 维度切分加载。
if has_sink:
```

```
        self.sinks = nn.Parameter(torch.empty(self.num_heads), requires_grad=False)
        set_weight_attrs(self.sinks, {"weight_loader": sharded_weight_loader(0)})
    else:
        self.sinks = None

    # 原生 vLLM Attention 同时接收 per_layer_sliding_window 与 sinks。
    self.attn = Attention(
        self.num_heads,
        self.head_dim,
        self.scaling,
        num_kv_heads=self.num_kv_heads,
        cache_config=cache_config,
        quant_config=quant_config,
        per_layer_sliding_window=sliding_window,
        prefix=f"{prefix}.attn",
        sinks=self.sinks,
    )
```

评论区精华

核心讨论集中在两个点：一是 MoE 权重加载方式——hmellor 指出 RoutedExperts 可以直接加载融合权重，无需在加载前手工拆分，并提交清理代码；二是每层 RoPE 参数的配置表示——hmellor 建议嵌套 rope_parameters，daviswer 以每层 RoPE 与 attention 类型相互独立为由选择独立列表，hmellor 最终认可并提示未来可参考 Nvidia 的 per_layer_config API。

- MoE 权重加载：拆分 vs 直接加载 (design): 删除手工拆分逻辑，统一走 hf_to_vllm_mapper + AutoWeightsLoader，并用 PowerMoE-3b 与 granite-swash-3b-a600m 验证输出一致。
- per-layer RoPE 参数表示：独立列表 vs 嵌套 rope_parameters (design): 维持独立列表方案，并记录 per_layer_config API 作为后续演进方向。

风险与影响

- 风险：
  - 依赖 transformers ≥ 5.15.1 才能正确解析 SWA 配置，旧版本下测试会跳过，但用户直接加载模型时依赖 HF config 解析，低版本可能读不出 layer_types / layer_rope_theta。
  - MoE 权重加载从手写拆分改为 hf_to_vllm_mapper 映射，回归面较大；hmellor 已验证两个检查点，但旧版 HF MoE 层命名仍需覆盖。
  - 新功能（sink / per-layer sliding window）依赖 Attention 层参数支持，需关注各后端覆盖范围。
  - transformers 后端本身对这类模型输出错误（sink 静默丢弃），用户若用 transformers 后端对比可能得到错误结果，属上游问题。
  - 新功能默认关闭，现有 Granite 用户无行为变更，整体风险可控。
  - 影响：影响范围集中在 Granite 系列模型代码：granite.py、granitemoe.py、granitemoeshared.py 被重构，净删约 200 行；模型注册表与测试注册表同步扩展；新

增两个受支持模型，对现有 Granite / GraniteMoeShared 用户无行为变化。团队后续新增同类模型变体时可参考此「复用现有实现 + 共享解析器」模式。

- 风险标记：依赖 transformers 5.15.1，transformers 后端 sink 静默丢弃，权重加载重构回归风险，新功能默认关闭

关联脉络

- PR #48270 [Model] Add GraniteSWA and GraniteMoeSWA: 本 PR 明确指出 Supersedes and closes #48270，是替代实现，且旧 PR 的 review 讨论被迁移到本 PR。