PR #48507 完整报告

vllm-project/vllm

[Bug][Quantization] Fix humming is_layer_skipped for compressed-tensors "re:" ignore entries

合并时间：2026-07-16 22:47

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/48507

执行摘要

- 一句话：修复 humming 量化跳过层的正则匹配 bug
- 推荐动作：建议合入。这是一个明确的 bug 修复，有充分的测试覆盖，并已获得批准。值得学习的是处理类似配置格式时的向后兼容性设计。

功能与动机

压缩张量检查点使用正则模式（如 re:vision_tower.*）来标记忽略层，但 HummingConfig.is_layer_skipped 使用子串匹配，导致这些层未被跳过而被错误量化。这导致实际权重加载（回退到未量化）与虚拟权重加载（保持量化）之间的差异，破坏了字节级权重共享。

实现拆解

1. 修改 is_layer_skipped 方法：在 vllm/model_executor/layers/quantization/humming.py 中，将原本的 any(substring in prefix ...) 循环替换为逐个条目的判断逻辑：如果条目以 re: 开头，则使用 re.match 进行正则匹配；否则保持原有的子串匹配。
2. 保留向后兼容性：非 re: 条目（如 bitsandbytes 的 modules_to_not_convert）保持原有的子串匹配行为。
3. 新增回归测试文件：创建 tests/quantization/test_humming_ignore.py，包含参数化测试 test_is_layer_skipped_regex_ignore，使用 Kimi-K2.6 的真实忽略列表验证正则匹配的正确性，以及 test_plain_substring_entries_still_work 验证传统子串匹配仍然有效。

关键文件：

- vllm/model_executor/layers/quantization/humming.py（模块 量化；类别 source；类型 core-logic；符号 is_layer_skipped）：核心修复：修改 is_layer_skipped 方法，支持正则匹配 re: 前缀的忽略层条目。
- tests/quantization/test_humming_ignore.py（模块 量化测试；类别 test；类型 test-coverage；符号 test_is_layer_skipped_regex_ignore, test_plain_substring_entries_still_work）：新增回归测试，验证正则和子串匹配场景。

关键符号：is_layer_skipped

关键源码片段

vllm/model_executor/layers/quantization/humming.py

核心修复：修改 is_layer_skipped 方法，支持正则匹配 re: 前缀的忽略层条目。

```
# vllm/model_executor/layers/quantization/humming.py

def is_layer_skipped(self, config: dict[str, Any], prefix: str):
    keys = ["ignored_layers", "ignore", "modules_to_not_convert"]
    ignored_layers = self.get_from_keys_or(config, keys, []) or []
    if hasattr(self, "hf_to_vllm_mapper"):
        ignored_layers = self.hf_to_vllm_mapper.apply_list(ignored_layers)

    for module_name in ignored_layers:
        # compressed-tensors 风格的条目可能以 "re:" 前缀声明正则表达式
        # （例如 "re:vision_tower.*"）。这些必须正则匹配；
        # 纯子串匹配永远无法匹配字面量 "re:..." 字符串，
        # 导致被忽略的层被静默量化。非 "re:" 条目保持现有的子串行为
        # （例如 bitsandbytes 的 modules_to_not_convert）。
        if module_name.startswith("re:"):
            if re.match(module_name[3:], prefix):
                return True
        elif module_name in prefix:
            return True
    if "lm_head" in prefix:
        return True

    for regex in config.get("dynamic", {}):
        if regex[:1] != "-":
            continue
        if re.match(regex[2:], prefix):
            return True

    return False
```

tests/quantization/test_humming_ignore.py

新增回归测试，验证正则和子串匹配场景。

```
# tests/quantization/test_humming_ignore.py
"""Tests for humming's is_layer_skipped handling of compressed-tensors
regex ("re:") ignore entries.

Regression test: compressed-tensors checkpoints (e.g. Kimi-K2.6) list ignored
layers as regex patterns prefixed with "re:" (e.g. "re:vision_tower.*").
humming previously substring-matched these literals, so ignored layers were
never skipped and got (incorrectly) quantized -- which then diverged between
the real weight-load path (falls back to unquantized) and the dummy-load path
(stays quantized), breaking any consumer that byte-compares the two.
"""

import pytest
```

```
from vllm.model_executor.layers.quantization.humming import HummingConfig

# Kimi-K2.6 压缩张量量化配置中的真实忽略列表
K26_IGNORE = [
    "re:.*self_attn.*",
    "re:.*shared_experts.*",
    r"re:.*mlp\.(gate|up|gate_up|down)_proj.*",
    "re:.*lm_head.*",
    "re:vision_tower.*",
    "re:mm_projector.*",
]

@pytest.mark.parametrize(
    "prefix,expected",
    [
        # 被忽略的层（必须跳过 -> 保持未量化）
        ("vision_tower.encoder.blocks.0.mlp.fc0", True),
        ("vision_tower.encoder.blocks.16.wo", True),
        ("mm_projector.linear_1", True),
        ("language_model.model.layers.0.self_attn.o_proj", True),
        ("language_model.model.layers.0.mlp.gate_up_proj", True),
        ("language_model.model.layers.0.mlp.down_proj", True),
        ("language_model.model.layers.3.mlp.shared_experts.gate_proj", True),
        ("model.lm_head", True),
        # 路由专家不会被忽略 -> 必须量化
        ("language_model.model.layers.3.mlp.experts.383.up_proj", False),
        ("language_model.model.layers.3.mlp.experts.0.down_proj", False),
    ],
)
def test_is_layer_skipped_regex_ignore(prefix, expected):
    cfg = HummingConfig(full_config={"ignore": K26_IGNORE})
    assert cfg.is_layer_skipped({"ignore": K26_IGNORE}, prefix) is expected

def test_plain_substring_entries_still_work():
    # bitsandbytes 风格的 modules_to_not_convert 使用纯子串
    cfg = HummingConfig()
    cfg_dict = {"modules_to_not_convert": ["lm_head", "vision"]}
    assert cfg.is_layer_skipped(cfg_dict, "model.vision.encoder.fc") is True
    assert cfg.is_layer_skipped(cfg_dict, "model.layers.0.mlp.up_proj") is False
```

评论区精华

审查者 ErenAta16 交叉验证了 re.match 的选择与参考实现 compressed_tensors/utils.py::_is_equal_or_regex_match 一致，确认了相同的前缀剥离和 re.match 语义。作者 AndyDai-nv 确认 CI 失败与 PR 无关，审查者查看了日志后也确认了这一点。

- re.match 语义与参考实现一致性 (correctness): 确认实现与压缩张量参考实现一致。
- CI 失败是否与 PR 相关 (testing): CI 失败不是 PR 引起的。

风险与影响

- 风险：风险极低：修改仅涉及为 *re:* 前缀条目添加正则匹配分支，非 *re:* 条目的行为完全不变。所有现有功能（bitsandbytes、动态忽略等）不受影响。测试覆盖了真实世界的 Kimi-K2.6 忽略列表和普通子串场景。
- 影响：直接影响使用压缩张量量化（以 *re:* 前缀表示忽略层）的模型，如 Kimi-K2.6。修复后，这些模型在虚拟权重加载路径中不再错误量化忽略层，确保了权重共享 / 传输的正确性。影响范围限于 humming 量化方法。
- 风险标记：暂无

关联脉络

- 暂无明显关联 PR