PR #30097 完整报告

sgl-project/sglang

[MLX] Size the attention KV pool at the compute dtype for quantized models

合并时间：2026-07-07 11:32

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/30097

执行摘要

- 一句话：修正 MLX 量化模型 KV 池 dtype 为计算精度
- 推荐动作：建议合并。该 PR 修复了一个明确的正确性 / 性能 bug，测试覆盖充分，设计清晰（利用 scales 携带计算 dtype）。值得关注的设计决策是：通过 scales 推断 dtype 而非为每个量化格式适配，保持了通用性。

功能与动机

根据 PR body，原 MLX 注意力 KV 池的 dtype 从 k_proj.weight 推断，而 QuantizedLinear 的权重是打包整数，导致 dtype 回退 float32，使池内存容量减半，且前缀命中前向在 float32 运行、非命中在 bf16/fp16 运行，造成精度不一致。此 PR 旨在让池 dtype 跟随计算 dtype（通过 scales 获得），修复正确性并提升性能。

实现拆解

1. 核心 dtype 推断逻辑调整：在 _attention_kv_config_for_layer (model_runner.py) 中，当权重 dtype 不在 _MLX_KV_FLOAT_DTYPES 时，不再直接返回 float32，而是尝试从 sample_attn.k_proj.scales 的 dtype 中获取；若 scales 存在且为 float，则使用 scales.dtype，否则回退到 float32。
2. 未量化模型不受影响：未量化路径保持原逻辑，即直接使用权重 dtype。
3. 新增测试覆盖：新文件 test_mlx_pool_dtype.py 包含 5 个用例：未量化传递权重 dtype、量化 bf16 使用 scales dtype、量化 fp16 使用 scales dtype、量化但 scales 不可用时回退 float32、以及 bytes-per-slot 翻倍的数值验证。
4. 小测试修复：在 test_attention_patching.py 中补充一行 self.assertEqual(scheduler.forward_ct, 1) 以对齐 #29217 引入的 forward_ct 会计变更。

关键文件：

- python/sglang/srt/hardware_backend/mlx/model_runner.py（模块 模型运行器；类别 source；类型 data-contract；符号 _attention_kv_config_for_layer）：核心修复文件：修改 _attention_kv_config_for_layer 方法中的 dtype 推断逻辑，使量化模型 KV 池 dtype 跟随计算精度而非回退 float32。
- test/registered/unit/hardware_backend/mlx/test_mlx_pool_dtype.py（模块 MLX 池测试；类别 test；类型 test-coverage；符号 _tiny_qwen2_model, _runner_for, TestPoolDtypeInference, test_unquantized_model_uses_weight_dtype）：新增测试文件

，覆盖 5 种 dtype 推断场景，保证行为正确性。

- test/registered/unit/hardware_backend/mlx/test_attention_patching.py（模块 测试补丁；类别 test；类型 test-coverage）：小幅度修改：增加 forward_ct 断言以对齐 #29217，保证重叠调度测试的 forward 次数正确。

关键符号：_attention_kv_config_for_layer, _get_attn_config, _compute_pool_size, test_unquantized_model_uses_weight_dtype, test_quantized_model_uses_scales_dtype, test_quantized_fp16_model_uses_scales_dtype, test_quantized_without_usable_scales_falls_back_to_float32, test_pool_bytes_per_slot_halves_for_bf16_quantized_model

关键源码片段

python/sglang/srt/hardware_backend/mlx/model_runner.py

核心修复文件：修改 _attention_kv_config_for_layer 方法中的 dtype 推断逻辑，使量化模型 KV 池 dtype 跟随计算精度而非回退 float32。

```
# model_runner.py — _attention_kv_config_for_layer 的关键片段 (#511-#526)
def _attention_kv_config_for_layer(self, layer_idx: int) -> tuple[int, int, mx.Dtype]:
    # ... 前面的 sliding-window 检查和 n_kv_heads/head_dim 获取 ...
    dtype = mx.float16
    if hasattr(sample_attn, "k_proj") and hasattr(sample_attn.k_proj, "weight"):
        dtype = sample_attn.k_proj.weight.dtype
    if dtype not in _MLX_KV_FLOAT_DTYPES:
        # QuantizedLinear packs weights as integers, but the KV cache
        # stores dequantized projection outputs, which are produced in
        # the compute dtype carried by the quantization scales. Storing
        # at that dtype instead of float32 halves pool bytes per slot
        # and keeps prefix-hit forwards in the same dtype as the no-hit
        # path (a float32 pool promoted every post-hit concat).
        scales = getattr(sample_attn.k_proj, "scales", None)
        if scales is not None and scales.dtype in _MLX_KV_FLOAT_DTYPES:
            dtype = scales.dtype # 使用 scales 携带的浮点 dtype
        else:
            dtype = mx.float32 # 保守回退
    return n_kv_heads, head_dim, dtype
```

test/registered/unit/hardware_backend/mlx/test_mlx_pool_dtype.py

新增测试文件，覆盖 5 种 dtype 推断场景，保证行为正确性。

```
# test_mlx_pool_dtype.py — 测试类主体
@unittest.skipUnless(_HAS_MLX, _SKIP_REASON)
class TestPoolDtypeInference(CustomTestCase):
    def test_unquantized_model_uses_weight_dtype(self):
        # 未量化模型应直接使用权重 dtype
        model = _tiny_qwen2_model()
        model.set_dtype(mx.float16)
        _, _, dtype = _runner_for(model)._get_attn_config()
```

```
        self.assertEqual(dtype, mx.float16)

    def test_quantized_model_uses_scales_dtype(self):
        # 量化 bf16 模型应使用 scales 的 bf16 dtype
        model = _tiny_qwen2_model()
        model.set_dtype(mx.bfloat16)
        nn.quantize(model, group_size=64, bits=4)
        attn = model.model.layers[0].self_attn
        self.assertNotIn(attn.k_proj.weight.dtype, {mx.float16, mx.bfloat16})
        self.assertEqual(attn.k_proj.scales.dtype, mx.bfloat16)
        _, _, dtype = _runner_for(model)._get_attn_config()
        self.assertEqual(dtype, mx.bfloat16)

    def test_quantized_fp16_model_uses_scales_dtype(self):
        # 量化 fp16 模型应使用 scales 的 fp16 dtype
        model = _tiny_qwen2_model()
        model.set_dtype(mx.float16)
        nn.quantize(model, group_size=64, bits=4)
        _, _, dtype = _runner_for(model)._get_attn_config()
        self.assertEqual(dtype, mx.float16)

    def test_quantized_without_usable_scales_falls_back_to_float32(self):
        # 当 scales 不是浮点时，应回退到 float32
        model = _tiny_qwen2_model()
        model.set_dtype(mx.bfloat16)
        nn.quantize(model, group_size=64, bits=4)
        for layer in model.model.layers:
            layer.self_attn.k_proj.scales = layer.self_attn.k_proj.scales.astype(
                mx.uint32
            )
        _, _, dtype = _runner_for(model)._get_attn_config()
        self.assertEqual(dtype, mx.float32)

    def test_pool_bytes_per_slot_halves_for_bf16_quantized_model(self):
        # 验证 bf16 量化模型 bytes/slot 是 float32 回退的一半
        model = _tiny_qwen2_model()
        model.set_dtype(mx.bfloat16)
        nn.quantize(model, group_size=64, bits=4)
        n_kv_heads, head_dim, dtype = _runner_for(model)._get_attn_config()
        num_layers = 2
        bytes_per_slot = 2 * num_layers * n_kv_heads * head_dim * dtype.size
        fp32_bytes_per_slot = 2 * num_layers * n_kv_heads * head_dim * 4
        self.assertEqual(bytes_per_slot * 2, fp32_bytes_per_slot)
```

评论区精华

Review 中 yeahdongcn 指出新测试文件缺少 `if __name__== "__main__": unittest.main()` 入口，作者在后续 commit 中修复。此外，yeahdongcn 提到该新增测试尚未自动注册 CI，需通过关联 PR #30121 处理。

- 测试文件缺少 __main__入口 (testing): 已修复，后续 commit 添加了入口。

风险与影响

- 风险：风险较低：核心变更仅在量化模型路径中生效，若 scales 属性缺失或 dtype 非 float，自动回退 float32 保持保守行为。但自定义 RoPE 池 - 散射内核 (SGLANG_MLX_USE_CUSTOM_ROPE) 之前因 dtype 不匹配在 dispatch 时直接失败，现在修复后可运行在 bf16 路径上，可能暴露其他内核级问题。影响范围局限于 MLX 后端。
- 影响：对用户：量化模型 KV 池内存 footprint 减半，自动池大小估算 token 容量增加约 2x；前缀命中与非命中前向精度不再分裂。对系统：自定义 RoPE 内核现在可配合量化模型工作。对团队：新增 5 个单元测试作为回归保护。整体影响有限，仅针对 Apple Silicon 上的 MLX 量化模型。
- 风险标记：量化模型通过 scales 推断 dtype，回退机制完善，自定义 RoPE 内核路径现在可运行于量化模型，需额外验证

关联脉络

- PR #29217 （未提供标题，但关联 forward_ct 会计）：此 PR 中 test_attention_patching.py 的修改对齐了 #29217 引入的 forward_ct 会计变更。
- PR #30121 （未提供标题，但关联 CI 注册）：讨论中提及 #30121 将新增测试注册到 CI 阶段 A 的显式列表，确保新测试在 CI 中运行。