PR #26635 完整报告

sgl-project/sglang

Improve registration in cpu_graph_runner

合并时间：2026-06-09 09:35

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26635

执行摘要

- 一句话：改进 CPU 图编译自定义 op 注册与显式 fallback 加速
- 推荐动作：值得精读设计：通过显式 make_fallback 规避 Inductor 诊断慢路径的思路可复用于其他算子编译场景。缺少测试配套，建议后续补充 fallback 注册的单元测试。

功能与动机

When a custom op does not have an explicit lowering registered, Inductor can create an implicit fallback, but that path first builds diagnostic messages for the missing lowering. For large CPU compile graphs, formatting those diagnostics may be extremely slow.

实现拆解

1. 在 cpu_graph_runner.py 中新增 _CPU_COMPILE_FAKE_OPS 集合及 register_cpu_compile_fake 装饰器，统一记录所有注册了 fake 实现的 op 名。
2. 新增 register_inductor_fallback_ops() 函数，在 set_torch_compile_config() 末尾调用，遍历已注册的 fake ops 并显式调用 make_fallback(op, warn=False) 避免 Inductor 的隐式慢路径。
3. 将原分散在 model_runner.py 和 cpu_worker.py 中的 shm_allgather fake 实现迁移至 cpu_graph_runner.py 的 register_fake_ops 中，并接受 tp_size 参数。
4. 补充缺失的 apply_rotary_pos_emb_cpu fake 实现。
5. 将 register_fake_ops 中所有 @torch.library.register_fake 改为 @register_cpu_compile_fake，确保 op 名被自动收集。

关键文件：

- python/sglang/srt/model_executor/cpu_graph_runner.py（模块 图编译；类别 source；类型 data-contract；符号 register_fake_ops, register_cpu_compile_fake, register_inductor_fallback_ops, _）：核心重构文件，新增注册管理函数 register_cpu_compile_fake 和 register_inductor_fallback_ops，并迁移注册入口到 set_torch_compile_config。
- python/sglang/srt/model_executor/model_runner.py（模块 模型加载；类别 source；类型 data-contract；符号 _）：移除重复的 shm_allgather fake 注册，集中到 cpu_graph_runner。

- python/sglang/multimodal_gen/runtime/managers/cpu_worker.py（模块 多模态 Worker；类别 source；类型 core-logic；符号 _）：移除重复的 shm_allgather fake 注册，统一到 cpu_graph_runner。

关键符号：register_cpu_compile_fake, register_inductor_fallback_ops, register_fake_ops

关键源码片段

python/sglang/srt/model_executor/cpu_graph_runner.py

核心重构文件，新增注册管理函数 register_cpu_compile_fake 和 register_inductor_fallback_ops，并迁移注册入口到 set_torch_compile_config。

```
# _CPU_COMPILE_FAKE_OPS 用于收集所有注册了 fake 实现的 op 名
_CPU_COMPILE_FAKE_OPS: set[str] = set()


def register_cpu_compile_fake(op_name: str):
    """装饰器：注册 fake 实现并将 op_name 加入集合"""
    _CPU_COMPILE_FAKE_OPS.add(op_name)
    return torch.library.register_fake(f"sgl_kernel::{op_name}")


def register_inductor_fallback_ops():
    """遍历已注册 fake 的 ops，显式调用 make_fallback 避免诊断慢路径"""
    from torch._inductor.lowering import lowerings, make_fallback

    sgl_kernel_ops = torch.ops.sgl_kernel
    for op_name in sorted(_CPU_COMPILE_FAKE_OPS):
        try:
            op = getattr(getattr(sgl_kernel_ops, op_name), "default")
        except AttributeError:
            continue
        if op not in lowerings:
            make_fallback(op, warn=False)


def register_fake_ops(tp_size: int):
    """注册所有 CPU custom ops 的 fake 实现"""
    # 无返回值的 ops
    none_return_ops = [
        "shm_allreduce",
        "bmm_cpu",
        "fused_add_rmsnorm_cpu",
        "decode_attention_cpu",
        "extend_attention_cpu",
        "gemma_fused_add_rmsnorm_cpu",
        "layernorm_cpu",
        "fused_add_layernorm_cpu",
    ]
```

```
for op in none_return_ops:
    @register_cpu_compile_fake(op)
    def _(*args, **kwargs):
        return

# 返回与输入相同 shape 的 ops
for op in [
    "rmsnorm_cpu",
    "l2norm_cpu",
    "fused_experts_cpu",
    "fused_rmsnorm_gated_cpu",
    "shared_expert_cpu",
    "causal_conv1d_update_cpu",
    "causal_conv1d_fwd_cpu",
    "gemma_rmsnorm_cpu",
    "gemma3_rmsnorm_cpu",
    "gemma4_rmsnorm_cpu",
]:
    @register_cpu_compile_fake(op)
    def _(input, *args, **kwargs):
        return torch.empty_like(input)

# 新增：shm_allgather 需要根据 tp_size 拼接
@register_cpu_compile_fake("shm_allgather")
def _(data, dim):
    return torch.cat([data] * tp_size, dim=dim)

# ... 其他 ops 类似使用 @register_cpu_compile_fake
```

评论区精华

review 中无实质性技术讨论，仅合并者 mingfeima 要求 rebase 以适配 torch 2.12 升级，随后批准合并。

- Rebase 请求 (other): 作者已 rebase，PR 通过。

风险与影响

- 风险：风险较低。新机制通过集中管理确保所有 fake op 都被注册 fallback，避免遗漏；shm_allgather 依赖 tp_size，已通过参数化在 register_fake_ops(tp_size) 中统一处理。缺少单元测试覆盖新增的 fallback 注册路径。
- 影响：影响 CPU 后端用户的 torch.compile 体验，编译时间显著缩短。对 GPU 后端无影响。变更集中在核心编译路径，但逻辑正交。
- 风险标记：CPU 编译路径变更，缺少测试覆盖

关联脉络

- 暂无明显关联 PR