PR #2043 完整报告

radixark/miles

Compact NVFP4 BF16 MoE exclusion metadata

合并时间：2026-08-04 05:48

原文链接：http://prhub.com.cn/radixark/miles/pull/2043

执行摘要

- 一句话：NVFP4 转换器压缩 BF16 MoE 排除项，冗余从 3571 降至 499
- 推荐动作：值得快速精读 tools/convert_hf_to_nvfp4.py 的 else 分支，理解“动态 BF16 层专家折叠为容器、同时保留层前缀与容器前缀”的取舍。对维护量化转换管线的同学有参考价值；注意测试当前被 CI 禁用，合并后建议留意后续启用时机。

功能与动机

convert_hf_to_nvfp4.py 已为每个选中 BF16 层序列化紧凑的 model.layers.N. 条目，但仍逐条追加每个被跳过的专家子项；_augment_ignore_list 添加 fused-MoE 父项后未移除子项。Qwen3-30B-A3B 后 8/48 层保持 BF16 时产生 3,072 条冗余（8 层 × 128 专家 × 3 投影）。PR body 明确：重放官方 checkpoint 索引后，排除项从 3,571 降至 499，并且转换器需镜像 MXFP8 的显式 .experts. 处理，让动态 BF16 routed experts 折叠到容器。

实现拆解

1. 修改入口：tools/convert_hf_to_nvfp4.py 的 process_file（权重序列化主循环）。原本对所有 key.endswith(".weight") 的跳过参数直接 append key.replace(".weight", "")。
2. 核心逻辑：改为按三种情况分派。 - 非专家参数：保留精确模块名（如 model.layers.0.input_layernorm）。 - 动态 BF16 层内的专家投影：折叠为 module_name.split(".experts.", 1)[0] + ".experts"，即 model.layers.N.mlp.experts。原因：SGLang 的 ModelOpt FP4 路径用精确 fused-MoE 前缀选择未量化 MoE 实现；同时已有的层前缀model.layers.N.继续保留以防匹配器需要。-其他专家（手动/非动态跳过）：保留精确模块名。 判定动态层依赖 convert_nvfp4 构造的 dynamic_skip_layer_prefixes（首尾 BF16 层集合）。
3. 测试配套：tests/fast-gpu/test_nvfp4_quantizer.py 新增 test_nvfp4_hf_converter_uses_compact_bf16_moe_prefixes，构造 1 层 128 expert × 3 投影 + layernorm 的 BF16 权重，num_layers_at_end_in_bf16=1 在 CPU 运行 convert_nvfp4，断言 config.json 的 quantization_config.ignore 与 hf_quant_config.json 的 quantization.exclude_modules 均等于 ["model.layers.0.", "model.layers.0.input_layernorm", "model.layers.0.mlp.experts"]，并校验输出无 weight_scale 键且 expert 127 保持 BF16。
4. 测试限制：测试文件注册在 fast-gpu 套件且被 disabled="Requires Blackwell/B200 CI runner for NVFP4." 禁用，CI 不会自动执行；作者仅在隔离 CPU 环境手动跑通，并新增

import json。

关键文件：

- tools/convert_hf_to_nvfp4.py（模块 转换工具；类别 source；类型 core-logic；符号 process_file, convert_nvfp4）：核心逻辑改动：BF16 跳过权重的排除项从逐专家投影改为折叠为 fused-MoE 容器，减少数千条冗余配置并保持与 MXFP8 对齐
- tests/fast-gpu/test_nvfp4_quantizer.py（模块 量化测试；类别 test；类型 test-coverage；符号 test_nvfp4_hf_converter_uses_compact_bf16_moe_prefixes）：新增 128 expert 转换回归测试，验证序列化配置紧凑性与 BF16 输出，但当前 CUDA 注册被禁用不会自动执行

关键符号：process_file, convert_nvfp4, test_nvfp4_hf_converter_uses_compact_bf16_moe_prefixes

关键源码片段

tools/convert_hf_to_nvfp4.py

核心逻辑改动：BF16 跳过权重的排除项从逐专家投影改为折叠为 fused-MoE 容器，减少数千条冗余配置并保持与 MXFP8 对齐

```
# process_file 内处理 BF16 跳过权重的分支。
# 每个被跳过的权重仍原样写回输出文件，
# 同时需要把模块名登记到 modules_to_not_convert（即量化排除列表）。
else:
    if key.endswith('.weight'):
        module_name = key[: -len('.weight')] # 去掉 '.weight' 得到模块名
        # 判断该参数是否属于动态选择为 BF16 的首 / 末若干层。
        # dynamic_skip_layer_prefixes 形如 'model.layers.0.'、'model.layers.40.'。
        is_dynamic_bf16 = any(prefix in key for prefix in dynamic_skip_layer_prefixes)
        if '.experts.' not in key:
            # 非专家参数（如 layernorm、attention 投影）保留精确模块名。
            modules_to_not_convert.append(module_name)
        elif is_dynamic_bf16:
            # 动态 BF16 层内的专家投影折叠为单个 fused-MoE 容器条目。
            # SGLang 的 ModelOpt FP4 路径需要用精确的 FusedMoE 前缀
            # 来选中未量化 MoE 实现，因此记录 'model.layers.N.mlp.experts'
            # 而不是 128 个专家的各 3 个投影，避免 8 层产生 3072 条冗余。
            expert_prefix = module_name.split('.experts.', 1)[0] + '.experts'
            modules_to_not_convert.append(expert_prefix)
        else:
            # 手动或非动态跳过的专家仍保留精确模块名。
            modules_to_not_convert.append(module_name)
    q_weights[key] = tensor
```

tests/fast-gpu/test_nvfp4_quantizer.py

新增 128 expert 转换回归测试，验证序列化配置紧凑性与 BF16 输出，但当前 CUDA 注册被禁用不会自动执行

```
def test_nvfp4_hf_converter_uses_compact_bf16_moe_prefixes(tmp_path):
    # 构造单层模型：128 个专家 × 3 个投影 + 1 个 layernorm，全部为 BF16。
    model_dir = tmp_path / 'model'
    save_dir = tmp_path / 'converted'
    model_dir.mkdir()
    (model_dir / 'config.json').write_text('{"num_hidden_layers": 1}')

    weights = {
        f'model.layers.0.mlp.experts.{expert_idx}.{projection}.weight': torch.ones(
            (1, NVFP4_GROUP_SIZE), dtype=torch.bfloat16
        )
        for expert_idx in range(128)
        for projection in ('gate_proj', 'up_proj', 'down_proj')
    }
    weights['model.layers.0.input_layernorm.weight'] = torch.ones(
        NVFP4_GROUP_SIZE, dtype=torch.bfloat16
    )
    safetensors.torch.save_file(
        weights, model_dir / 'model.safetensors', metadata={'format': 'pt'}
    )

    # 末 1 层保持 BF16，转换器在 CPU 上直接跳过量化该层。
    convert_nvfp4(
        str(model_dir), str(save_dir), device='cpu', num_layers_at_end_in_bf16=1
    )

    # 序列化后的两个量化配置都必须只含紧凑排除条目：
    # 层前缀、精确的非专家名、精确的 fused-MoE 容器名。
    expected_ignore = [
        'model.layers.0.',
        'model.layers.0.input_layernorm',
        'model.layers.0.mlp.experts',
    ]
    config = json.loads((save_dir / 'config.json').read_text())
    assert config['quantization_config']['ignore'] == expected_ignore

    hf_quant_config = json.loads((save_dir / 'hf_quant_config.json').read_text())
    assert hf_quant_config['quantization']['exclude_modules'] == expected_ignore

    # 输出权重全部保持 BF16，不产生任何 weight_scale 量化产物。
    with safetensors.safe_open(
        save_dir / 'model.safetensors', framework='pt', device='cpu'
    ) as f:
        assert all('weight_scale' not in key for key in f.keys())
        assert (
            f.get_tensor('model.layers.0.mlp.experts.127.down_proj.weight').dtype
            == torch.bfloat16
        )
```

评论区精华

本 PR 无实质性 review 讨论：review_comments_count = 0，唯一的 2 条评论来自 gemini-code-assist[bot] 的停用通告，与变更无关；审核人 yueming-yuan 直接 APPROVED 且未留评语。

- 暂无高价值评论线程

风险与影响

- 风险：
  - CI 覆盖缺失：tests/fast-gpu/test_nvfp4_quantizer.py 通过 register_cuda_ci(disabled=...) 禁用，新增回归不会自动运行，后续重构若破坏此行为需人工启用才能发现。
  - 排除列表格式依赖：折叠后依赖 SGLang ModelOpt FP4 路径对 model.layers.N.mlp.experts 精确前缀的识别；若未来 ModelOpt 或 SGLang 行为变化，需要同步调整。PR 明确保留该容器以维持兼容，风险可控。
  - 手动与动态跳过重叠：若某层同时出现在 *extra_high_precision_layers_hf* 与 dynamic_skip_layer_prefixes，专家会走折叠分支而非精确分支；由于折叠后仍覆盖全部 *expert*，实际效果一致，但手动意图在配置中不再可见。
  - 影响范围：仅离线转换工具的配置输出，不触及训练 / rollout 运行时路径。
- 影响：
  - 用户 / 系统：使用 *convert_hf_to_nvfp4.py* 转换大 MoE 模型（如 Qwen3-30B-A3B）时，生成的 *config.json* 与 *hf_quant_config.json* 中排除列表从数千条降至数百条，配置文件体积与后续处理开销显著下降。
  - 团队：与 MXFP8 转换器的排除规则对齐，降低维护两套逻辑的认知负担；新增回归测试为转换输出格式提供保障（但当前未进 CI）。
  - 影响范围：仅离线转换工具的输出元数据，不改变量化数值路径。
  - 风险标记：测试未进入 CI，排除列表格式变更，依赖 ModelOpt 精确前缀

关联脉络

- PR #2014 fix: quantize non-interleaved DSA indexer wk: 同属量化转换 / 排除列表领域，调整 quantizer_fp8.py 中 indexer wk 的量化条件
- PR #1928 [fix] DSA indexer on Blackwell: send the DSA indexer wk unquantized: 同属 Blackwell 量化输出排除逻辑，处理 indexer wk 不量化问题