PR #33615 完整报告

sgl-project/sglang

[Test] Route GEMM backend UTs through real layer modules and weight loaders

合并时间：2026-08-05 11:53

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33615

执行摘要

- 一句话：GEMM 后端单测接入真实层与权重加载器
- 推荐动作：值得精读：它是一份 ' 如何把后端单元测试做成真实语义 ' 的完整样板——真实模块构造 + 真实 weight_loader + 真实前向，只保留 torch 参考为手写，并把可复用夹具与参考编解码抽成共享模块。对计划新增 quant 后端单测的开发者，sglang.test.layer_ut_utils 与 sglang.test.quant_ref_utils 是直接可用的模板；_make_merged_layer 的注释与 shard-scale 折叠说明，也是理解 process_weights_after_loading 行为的上手材料。需要留意：该 PR 依赖 SM100+ 硬件验证，阅读时可结合 #33596 / #33611 一起看，了解测试体系从 *e2e* 到 layer 级再到真实 loader 的三步演进。

功能与动机

PR body 明确定位：'Follow-up to #33596 / #33611: strengthen the backend unit tests' fixtures and compute to the real *e2e* semantics. Only the torch reference (the comparison oracle) stays hand-written.' 此前 #33596 / #33611 引入的 layer-level 单测仍通过 method.create_weights + method.apply 直接驱动量化方法，绕过了生产加载链路：真实 ColumnParallelLinear 基于 quant_config 的方法分发、weight_loader_v2 的 0-dim 标量 scale reshape 分支、MergedColumnParallelLinear 的 per-partition weight_scale_2 / input_scale 汇聚（fused-QKV 回归踩过的坑），以及 FusedMoE.weight_loader 对 gate / up 的排放逻辑。作者在 body 中明确：'gate/up placement becomes the loader's job; the hand-maintained up_first branch is deleted'，即要让单测校验真实加载路径而不是测试自己的手写实现。

实现拆解

1. 线性层单测真实化（NVFP4 + FP8）：test_nvfp4_linear_backends.py 与 test_fp8_blockwise_linear_backends.py 改用共享工具 make_tp1_column_parallel_linear 构造真实 ColumnParallelLinear，量化方法由 quant_config 自动分发（带真实 prefix 与 packed_modules_mapping）；checkpoint 格式权重（含 0-dim 标量 scale）经 load_linear_weights 走 weight_loader_v2；前向从 method.apply(layer, x) 改为 layer(x)，并在 setUpClass 统一 init_single_process_dist（真实层构造需要 TP 分组）。

2. 新增 merged 双分片用例（NVFP4）：_make_merged_layer 用 MergedColumnParallelLinear + packed_modules_mapping={'gate_up_proj': [...]} 构造 fused gate_up_proj，两个分片各自经 weight_loader_v2(shard_id=...) 加载，专门守护 process_weights_after_loading 的 per-partition weight_scale_2 / input_scale 汇聚逻辑；分片强制共享全局 scale，因为 max() 折叠不重量化 block scale，不等 scale 是已知精度隐患（测试注释中明确声明该路径不可测）。
3. MoE 单测真实化：test_nvfp4_moe_backends.py 对每个专家按 w1 / w3 / w2 分片调用真实 FusedMoE.weight_loader(param, loaded, name, shard_id=..., expert_id=...)，gate / up 排放完全交给 loader，删除手写 up_first 分支；topk 改用真实 select_experts + TopKConfig(top_k=TOPK, renormalize=True)；torch 参考保持 checkpoint 语义（w1=gate, w3=up）。
4. 抽取共享测试模块：新增 python/sglang/test/layer_ut_utils.py（init_single_process_dist、make_tp1_column_parallel_linear、load_linear_weights、assert_output_close）与 python/sglang/test/quant_ref_utils.py（convert_swizzled_to_linear、break_fp4_bytes、dequantize_nvfp4_to_dtype、quantize_nvfp4_shard）；参考实现刻意不依赖 sglang.srt，保持 oracle 独立性。
5. 迁移与 bug 修复：test_zaya_cca、test_hpc_ops_moe（端口 29633）、test_gptqmodel_dynamic（backend=nccl）、test_tensor_dump_forward_hook 的 dist 初始化迁到共享夹具；test_fp4_moe、test_cutedsl_moe 的 NVFP4 编解码迁到 quant_ref_utils，同时修复两处拷贝携带的裁剪 bug：convert_swizzled_to_linear 原以 0:k 裁剪 scale 列，非对齐 K 时无法去掉 K-tile padding，统一改为 0:k//block_size。验证：SM100+ 上 NVFP4 5 个、FP8 10 个、MoE 3 个测试全绿，迁移文件重跑通过。

关键文件：

- test/registered/unit/layers/quantization/test_nvfp4_linear_backends.py（模块 量化单测；类别 test；类型 test-coverage；符号 _make_quantized_layer, _make_merged_layer, _run_backend, _assert_matches）：NVFP4 线性 GEMM 后端单测的核心改造：真实 ColumnParallelLinear + weight_loader_v2 路径，新增 MergedColumnParallelLinear 双分片 gate_up_proj 用例守护 per-partition scale 汇聚，是本次 PR 最具代表性的变更。
- test/registered/unit/layers/quantization/test_nvfp4_moe_backends.py（模块 混合专家；类别 test；类型 test-coverage；符号 _run_backend, _torch_moe_reference, select_experts, quantize_nvfp4_shard）：MoE 单测真实化：真实 FusedMoE.weight_loader 逐专家加载 w1/w3/w2 分片，topk 走 select_experts，删除手写 up_first 分支，torch 参考改为 checkpoint 语义。
- python/sglang/test/layer_ut_utils.py（模块 测试夹具；类别 test；类型 test-infra；符号 init_single_process_dist, make_tp1_column_parallel_linear, load_linear_weights, assert_output_close）：新增共享测试夹具模块：单进程 dist 初始化、tp=1 ColumnParallelLinear 构造、weight_loader_v2 权重灌入、输出断言，是后续 backend 单测的标准基建。
- python/sglang/test/quant_ref_utils.py（模块 参考实现；类别 test；类型 test-infra；符号 convert_swizzled_to_linear, break_fp4_bytes, dequantize_nvfp4_to_dtype, quantize_nvfp4_shard）：新增共享 NVFP4 参考编解码模块，刻意不依赖 sglang.srt；包

含 convert_swizzled_to_linear 的 K-tile 裁剪 bug 修复，是本次迁移的数值正确性关键。

- test/registered/unit/layers/quantization/test_fp8_blockwise_linear_backends.py（模块 量化单测；类别 test；类型 test-coverage；符号 _make_linear, _check_backend, setUpClass）：FP8 三格式（blockwise / MXFP8 / per-tensor）单测切换到真实 ColumnParallelLinear，0-dim scale 覆盖 weight_loader_v2 标量 reshape 分支。
- test/registered/moe/test_cutedsl_moe.py（模块 混合专家；类别 test；类型 refactor；符号 dequantize_nvfp4_to_dtype）：删除本地 NVFP4 编解码拷贝改从 quant_ref_utils 导入，并移除多余的 device 参数，是共享参考模块的迁移对象之一。
- test/registered/kernels/ops/moe/test_fp4_moe.py（模块 混合专家；类别 test；类型 refactor；符号 dequantize_nvfp4_to_dtype）：删除本地 NVFP4 编解码拷贝改从 quant_ref_utils 导入，其旧 convert_swizzled_to_linear 的 0:k 裁剪 bug 是本次修复的对象。
- test/registered/unit/models/test_zaya_cca.py（模块 模型单测；类别 test；类型 refactor）：dist 初始化迁移到共享 init_single_process_dist，清理 banner 分隔注释，验证共享夹具在 CPU 单测中也适用。
- test/registered/quant/test_gptqmodel_dynamic.py（模块 量化单测；类别 test；类型 refactor）：dist 初始化迁移到共享 init_single_process_dist（backend=nccl），删除 try/except AssertionError 包裹，行为语义保持不变。
- test/registered/moe/test_hpc_ops_moe.py（模块 混合专家；类别 test；类型 refactor）：dist 初始化迁移到共享 init_single_process_dist（master_port=29633），验证共享夹具的参数化端口约定。
- test/registered/debug_utils/test_tensor_dump_forward_hook.py（模块 调试工具；类别 test；类型 refactor）：dist 初始化迁移到共享 init_single_process_dist，是共享夹具覆盖的最小迁移样例。

关键符号：convert_swizzled_to_linear, break_fp4_bytes, dequantize_nvfp4_to_dtype, quantize_nvfp4_shard, init_single_process_dist, make_tp1_column_parallel_linear, load_linear_weights, assert_output_close, _make_merged_layer, _make_quantized_layer

关键源码片段

test/registered/unit/layers/quantization/test_nvfp4_linear_backends.py

NVFP4 线性 GEMM 后端单测的核心改造：真实 ColumnParallelLinear + weight_loader_v2 路径，新增 MergedColumnParallelLinear 双分片 gate_up_proj 用例守护 per-partition scale 汇聚，是本次 PR 最具代表性的变更。

```
def _make_merged_layer(n_half: int, k: int):
    """Two fused output shards (gate_up_proj) loaded per shard; exercises the
    per-partition scale_2 / input_scale gathering that fused-QKV regressions hit."""
    from sglang.srt.layers.linear import MergedColumnParallelLinear

    quant_config = ModelOptFp4Config(
        is_checkpoint_nvfp4_serialized=True,
        group_size=16,
```

```
        use_per_token_activation=False,
        # 带真实模块映射，让量化方法按 fused 导出的语义分发。
        packed_modules_mapping={"gate_up_proj": ["gate_proj", "up_proj"]},
    )
    layer = MergedColumnParallelLinear(
        input_size=k,
        output_sizes=[n_half, n_half],
        bias=False,
        params_dtype=torch.bfloat16,
        quant_config=quant_config,
        prefix="model.layers.0.mlp.gate_up_proj",
        tp_rank=0,
        tp_size=1,
    ).cuda()

    # process_weights_after_loading 会用 max() 折叠分片 scale_2，且不再
    # 重量化 block scale，所以本测试强制两个分片共享同一个全局 scale
    # （modelopt 的 fused 导出本就携带相等的 scale_2）；
    # 分片 scale 不等是已知精度隐患，不属于可测路径。
    shards = [
        torch.randn((n_half, k), device="cuda", dtype=torch.bfloat16) / 10
        for _ in (0, 1)
    ]
    shared_gs = (
        FLOAT8_E4M3_MAX
        * FLOAT4_E2M1_MAX
        / max(w.abs().max().to(torch.float32) for w in shards)
    )
    dequants = []
    for shard_id, w in enumerate(shards):
        w_q, sf_linear, gs, w_dequant = quantize_nvfp4_shard(w, gs=shared_gs)
        # 每个分片独立走 weight_loader_v2 的 shard_id 分支，
        # 守护 per-partition scale 的汇聚逻辑。
        load_linear_weights(
            layer,
            shard_id=shard_id,
            weight=w_q,
            weight_scale=sf_linear,
            weight_scale_2=(1.0 / gs).clone(),
            input_scale=torch.tensor(ACT_SCALE, device="cuda"),
        )
        dequants.append(w_dequant)
    return layer, torch.cat(dequants, dim=0)
```

python/sglang/test/layer_ut_utils.py

新增共享测试夹具模块：单进程 dist 初始化、tp=1 ColumnParallelLinear 构造、weight_loader_v2 权重灌入、输出断言，是后续 backend 单测的标准基建。

```
def init_single_process_dist(master_port: int = 29632, backend: str = "gloo"):
```

```
    """world=1 的 dist + model-parallel 分组；srt 层即使 tp=1 也需要。"""
    os.environ.setdefault("MASTER_ADDR", "127.0.0.1")
    os.environ.setdefault("MASTER_PORT", str(master_port))
    os.environ.setdefault("RANK", "0")
    os.environ.setdefault("WORLD_SIZE", "1")
    os.environ.setdefault("LOCAL_RANK", "0")
    from sglang.srt.distributed.parallel_state import (
        init_distributed_environment,
        initialize_model_parallel,
        model_parallel_is_initialized,
    )

    if not torch.distributed.is_initialized():
        init_distributed_environment(
            world_size=1, rank=0, local_rank=0, backend=backend
        )
    if not model_parallel_is_initialized():
        # 必须用 kwargs 传 backend：位置参数会落进
        # attention_data_parallel_size 槽位，随后在 int // str 上崩溃。
        initialize_model_parallel(
            tensor_model_parallel_size=1,
            expert_model_parallel_size=1,
            pipeline_model_parallel_size=1,
            backend=backend,
        )


def load_linear_weights(layer, shard_id=None, **named_weights):
    """把 checkpoint 格式张量灌进真实 weight_loader_v2。"""
    for name, loaded in named_weights.items():
        if shard_id is None:
            layer.weight_loader_v2(getattr(layer, name), loaded)
        else:
            layer.weight_loader_v2(getattr(layer, name), loaded, shard_id)
```

python/sglang/test/quant_ref_utils.py

新增共享 NVFP4 参考编解码模块，刻意不依赖 sglang.srt；包含 convert_swizzled_to_linear 的 K-tile 裁剪 bug 修复，是本次迁移的数值正确性关键。

```
def convert_swizzled_to_linear(a_sf_swizzled: torch.Tensor, m, k, block_size=16):
    m_tiles = (m + 128 - 1) // 128
    f = block_size * 4
    k_tiles = (k + f - 1) // f
    tmp = torch.reshape(a_sf_swizzled, (1, m_tiles, k_tiles, 32, 4, 4))
    tmp = torch.permute(tmp, (0, 1, 4, 3, 2, 5))
    out = tmp.reshape(m_tiles * 128, k_tiles * f // block_size)
    # 关键修复：裁剪 scale 列数应为 k // block_size 而不是 k。
    # test_fp4_moe / test_cutedsl_moe 的旧拷贝写成 0:k，对非对齐 K
    # 会错误保留 K-tile padding，导致参考去量化出现偏差。
```

```
    return out[0:m, 0 : k // block_size]


def quantize_nvfp4_shard(w: torch.Tensor, gs=None):
    """对单个 checkpoint 分片做 NVFP4 量化；返回 (packed, 线性 sf,
    全局 scale, fp32 去量化参考)。"""
    from flashinfer import fp4_quantize

    n, k = w.shape
    if gs is None:
        gs = FLOAT8_E4M3_MAX * FLOAT4_E2M1_MAX / w.abs().max().to(torch.float32)
    w_q, w_sf_swizzled = fp4_quantize(w, gs)
    sf_linear = convert_swizzled_to_linear(
        w_sf_swizzled.view(torch.float8_e4m3fn), n, k, 16
    )
    w_dequant = dequantize_nvfp4_to_dtype(w_q, w_sf_swizzled, gs, torch.float32)
    return w_q, sf_linear, gs, w_dequant
```

评论区精华

该 PR 无实质 review 评论（review_comments_count = 0），作者自审自合（merged_by = hnyls2002）。唯一的讨论线程是 CI 重跑：作者发出 /rerun-test 指令，指定 8 个测试文件，机器人确认在 4-gpu-b200（5 个测试）、1-gpu-5090（1 个测试）、1-gpu-h100（3 个测试）全部通过。PR body 中记录的关键设计判断：其一，process_weights_after_loading 用 max() 折叠分片 scale 而不重量化 block scale，故分片 scale 不等是已知精度隐患、不是可测路径，merged 用例刻意让分片共享全局 scale（镜像 modelopt fused 导出）；其二，参考编解码刻意与 sglang.srt 解耦（'Deliberately independent of sglang.srt'），避免用被测代码校验被测代码；其三，0-dim scale 被用来锻炼 weight_loader_v2 的标量 reshape 分支，补齐此前单测的加载路径盲区。

- CI 重跑验证 (test): 目标测试全绿后作者自合入（merged_by: hnyls2002）；PR 无 review 评论，无未解决的讨论点。

风险与影响

- 风险：
  1. 无生产代码变更：11 个变更文件全部为测试或测试工具，风险面局限在测试层。
  2. 参考数值变化：统一到 quant_ref_utils 并修复裁剪 bug 后，test_fp4_moe / test_cutedsl_moe 对非对齐 K 的参考去量化结果与旧版不同；作者在 SM100+ 上重跑通过，但其他算力（如 H100 上的 fp8 路径）仍需 CI 确认。
  3. 共享端口：init_single_process_dist 依赖固定 master_port，当前各测试用 29631 / 29632 / 29633 错开；若未来同机并行运行需继续遵守该约定，否则会互相干扰。
  4. 测试与 loader 强耦合：weight_loader_v2 或 process_weights_after_loading 的回归会直接使后端单测失败（这正是目的），但失败定位时需区分 loader 问题与 kernel 问题。
  5. 覆盖盲区：merged 用例显式排除分片 scale 不等场景，该精度隐患依旧无测试覆盖；test_gptqmodel_dynamic 从 try/except 包裹的初始化改为无条件

init_single_process_dist，依赖其幂等性，若并行状态已被其他测试初始化需留意行为。 - 影响：对用户无直接影响（无生产代码变更）。对系统的收益在于回归防护更贴近真实语义：单测现在覆盖量化方法分发、weight_loader_v2 标量 reshape 分支、fused 层 per-partition scale 汇聚与 FusedMoE 的 gate / up 排放，正是 fused-QKV 等历史回归的多发点。对团队而言，layer_ut_utils / quant_ref_utils 成为后续新增量化后端单测的标准基建，净删除约 156 行重复代码（-525 / +369）；每个测试文件只保留自己的 oracle 与 shape 配置，CI 注册（base-b / extra-b / nightly）不变。 - 风险标记：参考数值因裁剪修复变化，共享 dist 端口需错开，测试与 loader 强耦合，不等分片 scale无覆盖，纯测试变更

关联脉络

- PR #33596 [Test] Replace GEMM backend e2e matrices with layer-level unit tests: 本 PR 的直接前身：创建 layer-level 单测骨架，本 PR 将其升级为真实层模块 + weight_loader_v2 路径并抽取共享夹具。
- PR #33611 [Test] Replace NVFP4 MoE runner backend e2e matrix with a layer-level unit test: MoE 侧对应 PR：本 PR 对 test_nvfp4_moe_backends.py 做真实 FusedMoE.weight_loader 化，删除手写 up_first 分支。