PR #25857 完整报告

sgl-project/sglang

[codex] Reland Wan2.2 ModelOpt CI checkpoints

合并时间：2026-05-20 22:15

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25857

执行摘要

- 一句话：迁移 Wan2.2 ModelOpt CI 至 NVIDIA 官方 FP8/NVFP4 检查点
- 推荐动作：建议阅读：该 PR 演示了如何安全地迁移外部依赖并调整内部默认值。值得关注的设计决策是 swap_weight_nibbles 的 fallback 链，以及如何通过 checkpoint_uses_packed_qkv 保持向后兼容。测试修复的根因分析也值得学习。

功能与动机

PR 源自 #25483 的重新提交，旨在使用 NVIDIA 官方 Diffusers FP8/NVFP4 检查点替换旧的 lmsys transformer overrides，减少维护负担并使 Wan2.2 量化路径与官方仓库对齐。根因是 #25483 中默认 swap_weight_nibbles 变更导致 B200 JIT 测试失败，本次修复通过显式指定合成检查点参数。

实现拆解

1. 变更默认 NVFP4 配置（modelopt_quant.py）：将 ModelOptFp4Config 的 swap_weight_nibbles 默认值从 True 改为 False，并在 from_config 方法中调整 fallback 顺序，优先使用 checkpoint_uses_packed_qkv 作为回退键。
2. 更新配置合并逻辑（transformer_load_utils.py）：在 _merge_modelopt_fp4_configs 中调整属性复制顺序，现在先处理 checkpoint_uses_packed_qkv，再处理 swap_weight_nibbles 且默认回退 False。
3. 注册新模型路径（registry.py）：为 Wan2_2_T2V_A14B 注册新的模型路径 nvidia/Wan2.2-T2V-A14B-Diffusers-NVFP4，使得直接通过 --model-path 加载成为可能。
4. 修复 B200 JIT 测试（test_diffusion_nvfp4_scaled_mm.py）：在合成预交换检查点测试中显式传递 swap_weight_nibbles=True，以匹配经过预先交换的权重数据。
5. 更新 CI 测试用例和文档（gpu_cases.py、testcase_configs.py、quantization.mdx）：将 Wan2.2 测试用例从 --transformer-path 改为直接使用 --model-path 指定官方仓库；启用 run_consistency_check；更新环境变量；刷新文档表格以反映新检查点来源。

关键文件：

- python/sglang/multimodal_gen/runtime/layers/quantization/modelopt_quant.py（模块量化层；类别 source；类型 data-contract；符号 ModelOptFp4Config, from_config, process_weights_after_loading）：该文件是量化配置的核心，修改了 ModelOptFp4Config 的 swap_weight_nibbles 默认值（从 True 改为 False），并调整了

from_config 方法的 fallback 逻辑，影响所有 NVFP4 检查点的加载行为。

- python/sglang/multimodal_gen/tools/build_modelopt_nvfp4_transformer.py（模块 构建工具；类别 source；类型 data-contract；符号 build_modelopt_nvfp4_transformer）：构建工具的默认值解析逻辑被简化：移除了对 flux1-nvfp4 预设的特判，统一默认值为 False。影响所有通过该工具构建的混合精度检查点。
- python/sglang/multimodal_gen/runtime/loader/transformer_load_utils.py（模块 加载器；类别 source；类型 core-logic；符号 _merge_modelopt_fp4_configs）：配置合并函数调整了属性复制顺序：先复制 checkpoint_uses_packed_qkv，再处理 swap_weight_nibbles 且默认回退 False，确保 NVIDIA 官方检查点的加载正确性。
- python/sglang/multimodal_gen/registry.py（模块 注册中心；类别 source；类型 core-logic；符号 _register_configs）：注册了新的 NVIDIA 官方 NVFP4 模型路径，使得 Wan2.2 可以直接通过 --model-path 加载，无需指定 transformer-path。
- python/sglang/multimodal_gen/test/server/gpu_cases.py（模块 GPU 测试；类别 test；类型 test-coverage；符号 MODELOPT_WAN22_FP8_MODEL, MODELOPT_WAN22_NVFP4_B200_ENV_VARS, MODELOPT_WAN22_NVFP4_MODEL）：主要测试变更：将 Wan2.2 测试用例从 --transformer-path 改为直接使用 --model-path 加载官方仓库，并启用一致性检查。
- docs_new/docs/sglang-diffusion/quantization.mdx（模块 文档；类别 other；类型 core-logic）：文档更新：反映新的检查点来源矩阵，从 lmsys 仓库改为 NVIDIA 官方仓库，并调整了加载方式说明。

关键符号：ModelOptFp4Config.init, ModelOptFp4Config.from_config, ModelOptFp4Config.process_weights_after_loading, _merge_modelopt_fp4_configs, build_modelopt_nvfp4_transformer, _register_configs

关键源码片段

python/sglang/multimodal_gen/runtime/layers/quantization/modelopt_quant.py

该文件是量化配置的核心，修改了 ModelOptFp4Config 的 swap_weight_nibbles 默认值（从 True 改为 False），并调整了 from_config 方法的 fallback 逻辑，影响所有 NVFP4 检查点的加载行为。

```
class ModelOptFp4Config(ModelOptQuantConfig):
    """Config class for NVFP4."""

    def __init__(
        self,
        is_checkpoint_nvfp4_serialized: bool = False,
        group_size: int = None,
        exclude_modules: List[str] = None,
        packed_modules_mapping: Optional[Dict[str, List[str]]] = None,
        checkpoint_uses_packed_qkv: bool = False,
        swap_weight_nibbles: bool = False, # 默认值从 True 改为 False，匹配官方检查点加载顺序
    ) -> None:
```

```
        super().__init__(exclude_modules, packed_modules_mapping)
        self.is_checkpoint_nvfp4_serialized = is_checkpoint_nvfp4_serialized
        if is_checkpoint_nvfp4_serialized:
            logger.warning("Detected nvfp4 checkpoint...")
        self.group_size = group_size
        self.checkpoint_uses_packed_qkv = checkpoint_uses_packed_qkv
        self.swap_weight_nibbles = swap_weight_nibbles

    @classmethod
    def from_config(cls, config: Dict[str, Any]) -> ModelOptFp4Config:
        group_size = None
        exclude_modules = []
        swap_weight_nibbles = False # 默认值从 True 改为 False

        # 扁平格式 (config.json quantization_config)
        quant_method = config.get("quant_algo")
        if quant_method is not None:
            group_size = config.get("group_size")
            if group_size is None:
                config_groups = config.get("config_groups", {})
                if config_groups:
                    first_group = next(iter(config_groups.values()), {})
                    group_size = first_group.get("weights", {}).get("group_size")
            exclude_modules = config.get("ignore", [])
            swap_weight_nibbles = config.get(
                "swap_weight_nibbles",
                config.get("checkpoint_uses_packed_qkv", False), # 新增 fallback 到 checkpoint_uses_
                packed_qkv
            )
        else:
            # 嵌套格式 (hf_quant_config.json)
            try:
                quant_config = cls.get_from_keys(config, ["quantization"])
                quant_method = quant_config["quant_algo"]
                group_size = ModelOptFp4Config.common_group_size(config)
                exclude_modules = quant_config.get("exclude_modules", [])
                swap_weight_nibbles = quant_config.get(
                    "swap_weight_nibbles",
                    config.get(
                        "swap_weight_nibbles",
                        config.get("checkpoint_uses_packed_qkv", False), # 三层 fallback
                    ),
                )
            except (ValueError, KeyError):
                raise ValueError("Cannot find 'quant_algo' in quantization config.")
        # ... 后续省略
```

python/sglang/multimodal_gen/tools/build_modelopt_nvfp4_transformer.py

构建工具的默认值解析逻辑被简化：移除了对 flux1-nvfp4 预设的特判，统一默认值为 False。影响所有通过该工具构建的混合精度检查点。

```
def build_modelopt_nvfp4_transformer(
    *,
    base_transformer_dir: str,
    modelopt_hf_dir: str,
    output_dir: str,
    pattern_preset: str = "none",
    keep_bf16_patterns: Sequence[str] | None = None,
    swap_weight_nibbles: bool | None = None, # 由调用方控制，不再有 preset 依赖
    overwrite: bool = False,
) -> dict[str, int | bool]:
    source_dir = _resolve_transformer_dir(modelopt_hf_dir)
    base_dir = _resolve_transformer_dir(base_transformer_dir)

    patterns = _preset_patterns(pattern_preset)
    if keep_bf16_patterns:
        patterns.extend(keep_bf16_patterns)

    # 移除了 pattern_preset 为 "flux1-nvfp4" 时返回 True 的特判，统一默认 False
    resolved_swap_weight_nibbles = (
        swap_weight_nibbles if swap_weight_nibbles is not None else False
    )
    output_config = _updated_quant_config(
        _load_config(source_dir),
        fallback_patterns=patterns,
        swap_weight_nibbles=resolved_swap_weight_nibbles,
    )
    # ... 后续省略
```

python/sglang/multimodal_gen/runtime/loader/transformer_load_utils.py

配置合并函数调整了属性复制顺序：先复制 checkpoint_uses_packed_qkv，再处理 swap_weight_nibbles 且默认回退 False，确保 NVIDIA 官方检查点的加载正确性。

```
def _merge_modelopt_fp4_configs(
    existing_config: Optional[ModelOptFp4Config],
    inferred_config: Optional[ModelOptFp4Config],
) -> Optional[ModelOptFp4Config]:
    """Merge FP4 configs prioritizing inferred exclude list but preserving repo-level knobs."""
    if inferred_config is None:
        return existing_config
    if _get_quant_config_name(inferred_config) != "modelopt_fp4":
        return existing_config or inferred_config
    if existing_config is None:
        return inferred_config
    if _get_quant_config_name(existing_config) != "modelopt_fp4":
        return existing_config

    # ... exclude_modules 合并逻辑省略
```

```
    # 关键变更：先处理 checkpoint_uses_packed_qkv，再处理 swap_weight_nibbles
    inferred_config.checkpoint_uses_packed_qkv = getattr(
        inferred_config, "checkpoint_uses_packed_qkv", False
    ) or getattr(existing_config, "checkpoint_uses_packed_qkv", False)
    inferred_config.swap_weight_nibbles = getattr(
        inferred_config, "swap_weight_nibbles", False # 默认值从 True 改为 False
    ) or getattr(existing_config, "swap_weight_nibbles", False)
    if getattr(inferred_config, "group_size", None) is None:
        inferred_config.group_size = getattr(existing_config, "group_size", None)

    return inferred_config
```

评论区精华

PR 没有直接 review 评论，但 PR body 详细描述了根因：

> "#25483 intentionally changes the ModelOpt FP4 fallback default for swap_weight_nibbles to False so NVIDIA full-repo checkpoints without the field load in runtime order." "The B200 JIT test builds a synthetic checkpoint by pre-swapping nibbles with _swap_fp4_nibbles(weight_fp4), but did not pass the knob explicitly, so after the default flip its expected weight stayed one nibble-swap away from the processed layer weight." 该讨论已通过显式传递参数解决。

- swap_weight_nibbles 默认值变更导致的测试失败及修复 (correctness): 在合成检查点测试中显式传递 swap_weight_nibbles=True 以匹配预交换数据。

风险与影响

- 风险：
  1. 默认值变更影响广泛：swap_weight_nibbles 从 True 变为 False 会影响所有 NVFP4 检查点加载，尤其是未显式设置该字段的配置。虽然已通过 fallback 到 checkpoint_uses_packed_qkv 缓解，但依赖旧默认值的外部工作流可能出错。
  2. 模型路径变更：从 lmsys transformer 改为 NVIDIA 官方仓库，若官方仓库不可用或更新，CI 可能失败。
  3. 多文件一致性：变更涉及 quantization 配置、加载器、注册表、测试配置等 6 个核心文件，任何不一致可能导致运行时错误。
  4. JIT 测试局限：修复仅针对特定合成检查点，若其他测试也依赖旧默认值可能未被覆盖。
- 影响：影响范围包括：
  - 用户：使用 Wan2.2 NVFP4 检查点的用户现在可以指定官方仓库 ID 直接加载（例如 --model-path nvidia/Wan2.2-T2V-A14B-Diffusers-NVFP4），无需单独准备 transformer 路径。但需注意默认加载行为变化。
  - 系统：NVFP4 加载路径统一，减少维护分支。
  - CI：B200 CI 现在使用官方 NVFP4 检查点，并启用一致性检查；Wan2.2 FP8/NVFP4 测试用例简化，不再依赖 lmsys 中间仓库。
  - 文档：quantization.mdx 更新了支持矩阵，移除了 lmsys transformer 引用。

- 风险标记：默认值变更，依赖官方检查点可用性，多模块协调变更，JIT 测试脆弱性

关联脉络

- PR #25483 [diffusion] Move Wan2.2 ModelOpt CI/docs from old lmsys overrides to NVIDIA full Diffusers FP8/NVFP4 checkpoints: 本 PR 是 25483 的重新提交（reland），修复了其中默认值变更导致的测试失败问题