PR #19143 完整报告

sgl-project/sglang

feat: Support MXFP4 quantized dense models on AMD CDNA2/CDNA3 GPUs

合并时间：2026-04-17 07:51

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/19143

执行摘要

- 一句话：新增 Petit MXFP4 量化方案，支持 AMD CDNA2/CDNA3 GPU 运行 FP4 量化模型。
- 推荐动作：建议精读以了解量化管道集成设计，重点关注 petit_mxfp4.py 中的配置类实现和 petit_utils.py 中的兼容性检查逻辑，这些体现了 AMD 平台扩展和第三方内核集成的权衡。

功能与动机

PR body 中指出：'leverages Petit to efficiently emulate FP4 on AMD CDNA2 / CDNA3 GPUs, enabling SGLang to run FP4-quantized models (e.g., amd--Llama-3.3-70B-Instruct-MXFP4-Preview quantized by ModelOpt) on AMD GPUs that do not natively support FP4.'，旨在扩展 AMD 平台对 FP4 量化模型的支持，解决 AMD GPU 缺乏原生 FP4 能力的问题。

实现拆解

1. 新增量化配置类：创建 python/sglang/srt/layers/quantization/petit_mxfp4.py 和 python/sglang/srt/layers/quantization/petit_nvfp4.py，分别定义 PetitMxfp4Config 和 PetitNvFp4Config 类，实现从配置文件解析、量化方法覆盖等接口，集成到 SGLang 量化管道中。
2. 更新工具函数：修改 python/sglang/srt/layers/quantization/petit_utils.py，添加 _check_petit_mxfp4_supported、verify_petit_mxfp4_supported、is_quark_mxfp4_compatible_config 等函数，支持 MXFP4 和 Quark 格式的验证与权重处理逻辑。
3. 调整导入和配置：更新 python/sglang/srt/layers/quantization/__init__.py 导入 PetitMxfp4Config，并在 python/sglang/srt/configs/model_config.py 中添加 petit_mxfp4 到验证列表和兼容性映射，确保新方案能被识别和使用。
4. 依赖管理：修改 python/pyproject_other.toml，调整 petit_kernel 依赖版本，以支持 Python 3.10 环境（基于讨论中的调整）。
5. 入口点调整：更新 python/sglang/srt/server_args.py，允许 quantization 参数接受 petit_mxfp4 值，作为用户配置入口。

关键文件：

- python/sglang/srt/layers/quantization/petit_mxfp4.py（模块 量化层；类别 source；类型 core-logic；符号 PetitMxfp4Config, init, get_name, get_supported_act_dtypes）：新增 MXFP4 量化配置类的核心文件，定义了 PetitMxfp4Config 和 PetitMxfp4LinearMethod，是用户通过 quantization="petit_mxfp4" 启用新功能的关键入口。
- python/sglang/srt/layers/quantization/petit_utils.py（模块 量化层；类别 source；类型 core-logic；符号 _check_petit_mxfp4_supported, verify_petit_mxfp4_supported, _is_quark_mxfp4_layer_quant_config, is_quark_mxfp4_compatible_config）：关键工具函数文件，新增 MXFP4 支持检查、Quark 兼容性验证和权重处理逻辑，是量化管道运行的基础。
- python/sglang/srt/layers/quantization/petit_nvfp4.py（模块 量化层；类别 source；类型 core-logic；符号 PetitNvFp4Config, init, get_name, get_supported_act_dtypes）：新增 NVFP4 量化配置类文件，从原 petit.py 中分离，专注 NVIDIA GPU 支持，保持代码模块化。
- python/sglang/srt/layers/quantization/petit.py（模块 量化层；类别 source；类型 refactor；符号 PetitNvFp4Config, PetitNvFp4LinearMethod）：从功能实现文件重构为向后兼容的导入填充，确保现有代码对 PetitNvFp4Config 的引用不中断。
- python/sglang/srt/configs/model_config.py（模块 配置验证；类别 source；类型 data-contract）：更新模型配置验证逻辑，添加 petit_mxfp4 到允许的量化方案列表和兼容性映射，确保系统能识别新选项。

关键符号：PetitMxfp4Config.init, PetitMxfp4Config.get_name, PetitMxfp4Config.from_config, PetitMxfp4Config.override_quantization_method, _check_petit_mxfp4_supported, verify_petit_mxfp4_supported, is_quark_mxfp4_compatible_config

关键源码片段

python/sglang/srt/layers/quantization/petit_mxfp4.py

新增 MXFP4 量化配置类的核心文件，定义了 PetitMxfp4Config 和 PetitMxfp4LinearMethod，是用户通过 quantization="petit_mxfp4" 启用新功能的关键入口。

```
class PetitMxfp4Config(QuantizationConfig):
    """Config class for Petit MXFP4 linear inference on ROCm."""

    def __init__(
        self,
        is_checkpoint_mxfp4_serialized: bool = False,
        group_size: int = 32,
        exclude_modules: Optional[List[str]] = None,
    ) -> None:
        self.is_checkpoint_mxfp4_serialized = is_checkpoint_mxfp4_serialized
        self.group_size = group_size
        self.exclude_modules = exclude_modules or []
        if is_checkpoint_mxfp4_serialized:
            logger.warning(
                "Detected mxfp4 checkpoint for petit kernel path. "
```

```
            "This format is experimental and subject to change."
        )

    @classmethod
    def get_name(cls) -> str:
        return "petit_mxfp4" # 返回量化方案名称，用于用户配置

    @classmethod
    def get_supported_act_dtypes(cls) -> List[torch.dtype]:
        # Petit MXFP4 内核当前仅支持 BF16 激活在 ROCm 上
        return [torch.bfloat16]

    @classmethod
    def from_config(cls, config: Dict[str, Any]) -> "PetitMxfp4Config":
        quant_section = config.get("quantization", config)
        quant_method = (
            quant_section.get("quant_algo")
            or quant_section.get("quant_method")
            or config.get("quant_method")
            or ""
        )
        group_size = quant_section.get("group_size", 32)
        verify_petit_mxfp4_supported(quant_method, group_size, quant_config=config) #
        验证配置兼容性
        exclude_modules = quant_section.get("exclude_modules", [])
        quant_method_lower = str(quant_method).lower()
        is_checkpoint_mxfp4_serialized = "mxfp4" in quant_method_lower or (
            quant_method_lower == "quark" and is_quark_mxfp4_compatible_config(config)
        ) # 检测是否为 MXFP4 或 Quark-MXFP4 序列化格式
        return cls(
            is_checkpoint_mxfp4_serialized=is_checkpoint_mxfp4_serialized,
            group_size=group_size,
            exclude_modules=exclude_modules,
        )
```

评论区精华

- 代码重复问题：gemini-code-assist[bot] 指出 is_layer_excluded 方法在 petit.py 和 petit_mxfp4.py 中重复，作者 fengli1702 回应提取共享 helper 函数解决。
- 自动选择逻辑风险：chatgpt-codex-connector[bot] 警告 PetitMxfp4Config.override_quantization_method 可能自动切换所有 MXFP4 模型，破坏现有 MXFP4/MoE 流，作者修改为仅当用户显式设置 quantization="petit_mxfp4" 时才启用。
- 兼容性检查：BowenBao 询问是否支持 quant_method="quark" 的模型，作者更新 petit_utils.py 添加 Quark-MXFP4 检测，并优化配置验证逻辑，使用 all(...) 代替 any(...) 确保混合精度安全。
- 代码重复与设计优化 (design)：已通过重构提取共享函数，减少代码重复，提升可维护性。

- 自动选择逻辑的安全性 (correctness): 已调整逻辑，避免无意中干扰现有流，确保向后兼容。
- Quark 模型兼容性验证 (correctness): 已扩展支持 Quark-MXFP4 格式，并加强验证逻辑，提升兼容性安全性。

风险与影响

- 风险：- 依赖风险：新增 petit_kernel 依赖（版本 0.0.3），在 Issue 评论中曾因 Python 版本兼容性导致 CI 失败，需确保依赖版本与 Python 环境匹配。
- 兼容性风险：petit_mxfp4 自动选择逻辑可能干扰现有 MXFP4 量化流（如 MoE 路径），已通过讨论调整，但仍需测试覆盖。
- 性能风险：Petit 内核为新集成，可能未经过充分优化或调优，需监控推理吞吐和准确性。
- 配置验证风险：petit_utils.py 中的 is_quark_mxfp4_compatible_config 函数若配置解析错误，可能导致模型加载失败。
- 影响：- 用户影响：AMD GPU 用户现在可以加载和运行 FP4 量化模型（如 amd--Llama-3.3-70B-Instruct-MXFP4-Preview），扩展了硬件支持范围，提升模型部署效率。
- 系统影响：量化管道新增一个选项，增加代码复杂性和维护负担，但通过重构保持清晰度。
- 团队影响：需持续关注 petit_kernel 依赖更新和跨平台测试，确保 AMD CI 稳定。
- 风险标记：依赖版本兼容，自动选择风险，配置验证复杂度

关联脉络

- PR #22952 [AMD] Add SGLANG_MORI_MOE_MAX_INPUT_TOKENS to truncate dispatch before MoE.: 同涉及 AMD 平台功能扩展，关注性能优化和环境变量集成，可对比 AMD 相关改进。
- PR #23045 [AMD] Fix AMD Multimodal Test - skip nvfp4 tests: 涉及 AMD CI 测试和量化模型测试跳过，与本 PR 的量化支持相关，反映跨平台测试挑战。
- PR #22673 [Perf] Precompute gemma_weight to avoid redundant add on every forward：性能优化相关 PR，展示内核层优化模式，可借鉴于 Petit 内核集成后的性能调优。