PR #29479 完整报告

sgl-project/sglang

[AMD] fix dsv4 indexer dtype dispatch on gfx950

合并时间：2026-07-09 17:54

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29479

执行摘要

- 一句话：修复 gfx950 上 FP8 dtype 错误分发
- 推荐动作：本 PR 为平台特异性 bugfix，改动清晰简洁，值得快速合并。建议后续增加单元测试，验证 is_fp8_fnuz() 在不同 GPU 平台上的返回值及对应的 dtype 选择，防止回归。

功能与动机

DeepSeek-V4 的 indexer 和 unified-KV paged-decode kernel 原先使用 is_hip() 对所有 AMD GPU 硬编码 torch.float8_e4m3fnuz，该格式只适用于 CDNA3 (gfx942)。在 gfx950 (MI355) 上需要使用 torch.float8_e4m3fn，否则会导致 FP8 KV cache 精度错误。

实现拆解

1. python/sglang/srt/layers/attention/dsv4/indexer.py：
   - 将导入 is_hip 替换为从 sglang.srt.layers.quantization.fp8_kernel 导入 is_fp8_fnuz。
   - 将顶层的 if is_hip(): ... else: ... 分支替换为单行 FP8_DTYPE = torch.float8_e4m3fnuz if is_fp8_fnuz() else torch.float8_e4m3fn，并移除 FP8_MAX（因其在模块内未被使用）。
2. python/sglang/srt/layers/attention/dsv4/unified_kv_kernels/paged_decode.py：
   - 添加 from sglang.srt.layers.quantization.fp8_kernel import is_fp8_fnuz。
   - 将原硬编码的 _FP8_DTYPE = torch.float8_e4m3fnuz 改为 _FP8_DTYPE = torch.float8_e4m3fnuz if is_fp8_fnuz() else torch.float8_e4m3fn。
   - 更新了存储注释，说明格式差异。

本 PR 无测试配套，也没有配置或部署变更。

关键文件：

- python/sglang/srt/layers/attention/dsv4/indexer.py（模块 注意力；类别 source；类型 dependency-wiring）：核心 fix 文件之一，将硬编码的 is_hip() 条件改为 is_fp8_fnuz()，修正 FP8 数据类型选择逻辑。
- python/sglang/srt/layers/attention/dsv4/unified_kv_kernels/paged_decode.py（模块 注意力；类别 source；类型 dependency-wiring）：另一个核心 fix 文件，修正 unified KV paged-decode kernel 中 FP8 数据类型选择。

关键符号：未识别

关键源码片段

python/sglang/srt/layers/attention/dsv4/indexer.py

核心 fix 文件之一，将硬编码的 is_hip() 条件改为 is_fp8_fnuz()，修正 FP8 数据类型选择逻辑。

```
# 变更后：使用 is_fp8_fnuz() 判断，而非 is_hip()
# 新引入：from sglang.srt.layers.quantization.fp8_kernel import is_fp8_fnuz
# 移除：from sglang.srt.utils import is_hip（该导入保留但不再用于此处）

# 模块级 FP8 数据类型选择
# torch.float8_e4m3fnuz 仅用于 AMD gfx942 (CDNA3)
# gfx950 及其他平台应使用 torch.float8_e4m3fn
FP8_DTYPE = torch.float8_e4m3fnuz if is_fp8_fnuz() else torch.float8_e4m3fn
# FP8_MAX 因未被外部引用而移除（不依赖模块级常量，调用方自行使用 torch.finfo）
```

评论区精华

HaiShaw 在 review 中指出，跨仓库环境下 e4m3fnuz 的 FP8_MAX 应为 224.0 而非 240.0，建议修正。作者 billishyahao 回复确认，并说明重新检查后发现 FP8_MAX 在 indexer.py 中未实际使用，因此将其整体移除，改为直接通过 torch.finfo 推断。该讨论已解决。

- FP8_MAX 值及是否保留 (correctness): 作者检查后认为 FP8_MAX 在 indexer.py 中未使用，将其移除。

风险与影响

- 风险：本 PR 改动极小（仅两文件共 10 行），风险较低。但需注意：
  - 若 is_fp8_fnuz() 在非 AMD 或未正确初始化的环境中表现异常，可能导致 dtype 选择错误。由于该函数来自 sglang.srt.layers.quantization.fp8_kernel，应已覆盖常见场景。
  - 移除 FP8_MAX 不影响模块功能，因其未被引用。
  - 无相应测试覆盖，建议增加针对不同 GPU 类型的 dtype 选择测试。
  - 影响：影响范围仅限于在 gfx950 (MI355) 上运行 DeepSeek-V4 并使用 FP8 KV cache 的场景。修复后，FP8 数据类型从错误的 e4m3fnuz 变为正确的 e4m3fn，从而保证 KV cache 精度。对其他平台（NVIDIA、gfx942）无影响，行为保持不变。
  - 风险标记：缺少测试覆盖

关联脉络

- PR #29421 [Feat][GLM5.2] Add DSA Cache Layer Split under Prefill CP: 关联的 DeepSeek 系列 PR，涉及 KV cache 优化。
- PR #29729 Add opt-in SGLANG_ROPE_CACHE_FP32 to keep RoPE cache in fp32 on non-CUDA: 同为 AMD 相关 dtype 修复 PR。