

PR #39171 完整报告

sgl-project/sglang

[DeepSeek-V4.1] Bump FlashMLA to the fork's rebase head (v4.1 kernels)

合并时间: 2026-09-14 06:43

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39171>

执行摘要

- 一句话: FlashMLA 升版接入 V4.1 内核并补 B300 原生编译
- 推荐动作: 值得精读。本 PR 是外部内核 fork 依赖升级的典型样本, 重点关注三处: 一是 pin 升级伴随上游目录重构时的适配模式 (路径改写 + 手工前向声明 + shim 编译单元); 二是 sm_103a 编译目标的量化取舍, commit message 里给出了完整的回归与收益数据; 三是测试探针 prompt 设计的统计思维——先确认被测对象在 base 模型上有固定答案, 再谈度量。对需要维护 sgl-kernel 或处理类似第三方程内核集成的工程师, 这些决策记录比代码本身更有参考价值。

功能与动机

PR body 只有一句说明: "It's just #38942 rebased on main branch", 动机直接继承自 #38942。首条提交说明给出核心动机: 新 pin 3e18517 "merges DeepSeek's v4.1 kernel drop into the SGLang fork", 即把 DeepSeek 官方提交的 v4.1 MLA 内核合入 SGLang 维护的 FlashMLA fork, 本 PR 负责把它接到 sgl-kernel 的 AOT 构建里。第二条提交解释了一个关键取舍: B300 是 sm_103, 仅编译 sm_100f 虽然兼容可跑, 但 ptxas 面向原生架构生成的 SASS 更优, 启用 sm_103a 后 B300 decode 吞吐提升约 3-8%, 否则 pin 升级本身在 bs=1 和 bs=32 下会分别回归约 4% 和 9%。最后一条提交纠正了回归测试探针的缺陷: 裸指令 prompt 在 base 检查点上没有固定答案, DeepSeek-V4-Flash-0731 会用语言不定的 JSON 记录继续, 探针测的是续写语言而不是输出健康度。

实现拆解

本 PR 的变更入口是 AOT 构建脚本 python/sglang/kernels/aot/cmake/flashmla.cmake, 其余改动围绕它展开。

1. 依赖 pin 升级: flashmla.cmake 中 FetchContent_Declare 的 URL 从 c1dee56 换到 3e18517, SHA256 同步更新, 文件头注释明确这是 DeepSeek v4.1 内核的引入。
2. 目录与文件清单重构: 上游把内核源码从 csrc/sm90、csrc/sm100 迁到 csrc/kernels/sm90、csrc/kernels/sm100, FlashMLA_SOURCES 约 50 行路径全部改写; sm90 sparse decode 新增 v4_persistent 与 v32_no_rope 系列实例化, sm100 新增 v4/v41 系列 decode 实例; 同时把 csrc/api/dense_decode.cpp、sparse_decode.cpp、sparse_prefill.cpp 纳入编译单元, 配合保留的 python_api.cpp shim 工作。

3. API 桥接层适配: flashmla_extension.cc 删除 api/dense_decode.h、api/sparse_decode.h、api/sparse_fwd.h 三个头文件包含, 改为按 csrc/python_api.cpp 的方式手工前向声明 dense_attn_decode_interface 与 sparse_attn_decode_interface; v4.1 中 sparse 接口新增 const std::optional& kv_format 形参, sgl_sparse_decode_fwd 转发时补传 std::nullopt, sgl_dense_decode_fwd 保持不变。
4. B300 原生编译支持: CUDA 13+ 下追加 -gencode=arch=compute_103a,code=sm_103a; 原先针对 csrc/utils.h 的 SM103 补丁因上游已内置而删除, 而 cutlass arch/config.h 的 SM103 补丁被保留——它此前从未被真正触发, 现在变成 load-bearing, 因为只有启用 sm_103a 后才有 TU 在 CUDA_ARCH== 1030 下编译。
5. 测试配套修正: python/sglang/test/kits/basic_decode_correctness_kit.py 的 test_ascii_ratio 把裸指令 prompt 改为 Q/A 问答式 (Q: Write a single sentence about a sunny day in the park. \nA:), 固定输出语言, 使 printable ASCII 比例真正度量输出健康度; 这是该 pin 升级后回归验证的配套保障。

关键文件:

- python/sglang/kernels/aot/cmake/flashmla.cmake (模块 构建配置; 类别 infra; 类型 core-logic; 符号 FLASHMLA_SOURCES, FLASHMLA_CUDA_FLAGS) : PR 的核心变更文件: FlashMLA pin 升级、上游 csrc 目录重构后的文件清单重写、CUDA 13+ 的 sm_103a 支持以及 v4/v4.1 内核实例化的引入全部集中于此, 约 100 行变更。
- python/sglang/kernels/aot/csrc/flashmla_extension.cc (模块 内核封装; 类别 source; 类型 dependency-wiring; 符号 sgl_sparse_decode_fwd, sgl_dense_decode_fwd, dense_attn_decode_interface, sparse_attn_decode_interface) : API 桥接层: 上游把 api/.h 改为 api/.cpp 后, 这里以手工前向声明方式继续调用 dense/sparse 解码头, 并适配 sparse 接口新增的 kv_format 参数, 是 v4.1 内核接入 sgl-kernel torch.ops 的关键一环。
- python/sglang/test/kits/basic_decode_correctness_kit.py (模块 测试工具; 类别 test; 类型 test-coverage; 符号 test_ascii_ratio) : 回归测试配套: 修正 ascii_ratio 探针在 base 检查点上因输出语言漂移而失效的问题, 改为 Q/A 框架固定输出语言, 是本次 pin 升级后验证解码正确性的重要保障。

关键符号: sgl_sparse_decode_fwd, sgl_dense_decode_fwd,
dense_attn_decode_interface, sparse_attn_decode_interface, test_ascii_ratio

关键源码片段

python/sglang/kernels/aot/cmake/flashmla.cmake

PR 的核心变更文件: FlashMLA pin 升级、上游 csrc 目录重构后的文件清单重写、CUDA 13+ 的 sm_103a 支持以及 v4/v4.1 内核实例化的引入全部集中于此, 约 100 行变更。

```
# B300 即 sm_103; sm_100f 是家族兼容架构 (能运行), 但 ptxas 面向原生
# 架构生成 SASS 更优。FlashMLA 自身的 setup.py 也以相同理由优先选
# sm_100a + sm_103a 而不选 sm_100f。
if(${CUDA_VERSION} VERSION_GREATER_EQUAL "13.0")
    list(APPEND FLASHMLA_CUDA_FLAGS
        "-gencode=arch=compute_103a,code=sm_103a"
```

```

)
# 下游 cutlass/arch/config.h 的 SM103 补丁由此成为 load-bearing:
# 此前没有任何 TU 在 __CUDA_ARCH__ == 1030 下编译, 现在必须依赖它。
# (补丁内容: 在 "// SM101 and SM101a" 锚点前插入 SM103 架构定义)
endif()

# 上游 3e18517 把 csrc/ 目录重构为 csrc/kernels/, 并把 api/*.h 头文件改成
# api/*.cpp 翻译单元 (其 pybind 注册被 FLASH_MLA_LIBTORCH_ONLY 编译掉)。
# FlashMLA_SOURCES 因此全部按新结构重写, 并为 v4.1 内核新增实例化文件。
set(FlashMLA_SOURCES
    # 与 sgl-kernel torch.ops API 兼容的 shim 与注意力入口
    ${repo-flashmla_SOURCE_DIR}/csrc/python_api.cpp
    ${repo-flashmla_SOURCE_DIR}/csrc/api/dense_decode.cpp
    ${repo-flashmla_SOURCE_DIR}/csrc/api/sparse_decode.cpp
    ${repo-flashmla_SOURCE_DIR}/csrc/api/sparse_prefill.cpp
    # sm90 稀疏 decode: 新增 v4 与 v32 no-rope 变体
    ${repo-flashmla_SOURCE_DIR}/csrc/kernels/sm90/decode/sparse/instantiations/v4_persistent_
h64.cu
    ${repo-flashmla_SOURCE_DIR}/csrc/kernels/sm90/decode/sparse/instantiations/v4_persistent_
h128.cu
    ${repo-flashmla_SOURCE_DIR}/csrc/kernels/sm90/decode/sparse/instantiations/v32_no_rope_
persistent_h64.cu
    ${repo-flashmla_SOURCE_DIR}/csrc/kernels/sm90/decode/sparse/instantiations/v32_no_rope_
persistent_h128.cu
    # 其余 dense/sparse prefill 与 sm100 实例化文件同理按新路径列出
)

```

python/sglang/kernels/aot/csrc/flashmla_extension.cc

API 桥接层: 上游把 api/.h 改为 api/.cpp 后, 这里以手工前向声明方式继续调用 dense/sparse 解码头, 并适配 sparse 接口新增的 kv_format 参数, 是 v4.1 内核接入 sgl-kernel torch.ops 的关键一环。

```

// FlashMLA 的注意力入口只由 csrc/api/*.cpp 导出, 而其自带 pybind 层被
// FLASH_MLA_LIBTORCH_ONLY 宏编译掉, 因此这里按 csrc/python_api.cpp 的方式
// 手工前向声明, 供 sgl_kernel 的 torch.ops 绑定层直接调用。
std::tuple<at::Tensor, at::Tensor, std::optional<at::Tensor>, std::optional<at::Tensor>>
dense_attn_decode_interface(
    at::Tensor& q,
    const at::Tensor& kcache,
    const int head_size_v,
    const at::Tensor& seqLens_k,
    const at::Tensor& block_table,
    const float softmax_scale,
    bool is_causal,
    std::optional<at::Tensor>& tile_scheduler_metadata,
    std::optional<at::Tensor>& num_splits);

```

// v4.1 版本在 sparse 入口上新增了 kv_format 参数; 这里保持非必填语义,
// 调用时传 std::nullopt, 让内核走默认的 KV 格式推断路径。

```

std::tuple<at::Tensor, at::Tensor, std::optional<at::Tensor>, std::optional<at::Tensor>>
sparse_attn_decode_interface(
    const at::Tensor& q,
    const at::Tensor& kv,
    const at::Tensor& indices,
    const std::optional<at::Tensor>& topk_length,
    const std::optional<at::Tensor>& attn_sink,
    std::optional<at::Tensor>& tile_scheduler_metadata,
    std::optional<at::Tensor>& num_splits,
    const std::optional<at::Tensor>& extra_kv,
    const std::optional<at::Tensor>& extra_indices,
    const std::optional<at::Tensor>& extra_topk_length,
    int d_v,
    float sm_scale,
    const std::optional<std::string>& kv_format);

// sgl_sparse_decode_fwd 是暴露给 torch.ops 的薄封装，转发时补齐
// 新加入的 kv_format 形参，其余参数原样透传。
static std::tuple<at::Tensor, at::Tensor, std::optional<at::Tensor>, std::optional<at::Tensor>>
sgl_sparse_decode_fwd(
    const at::Tensor& q,
    const at::Tensor& kv,
    const at::Tensor& indices,
    const std::optional<at::Tensor>& topk_length,
    const std::optional<at::Tensor>& attn_sink,
    std::optional<at::Tensor> tile_scheduler_metadata,
    std::optional<at::Tensor> num_splits,
    const std::optional<at::Tensor>& extra_kv,
    const std::optional<at::Tensor>& extra_indices,
    const std::optional<at::Tensor>& extra_topk_length,
    int64_t d_v,
    double sm_scale) {
return sparse_attn_decode_interface(
    q,
    kv,
    indices,
    topk_length,
    attn_sink,
    tile_scheduler_metadata,
    num_splits,
    extra_kv,
    extra_indices,
    extra_topk_length,
    static_cast<int>(d_v),
    static_cast<float>(sm_scale),
    std::nullopt);
}

```

评论区精华

本 PR 没有任何代码评审意见，真正的技术讨论都沉淀在提交说明里，值得单独提炼。核心交锋点有三个：

- sm_103a 该不该加：提交者给出量化结论，不加会导致 pin 升级在 B300 上回归约 4% (bs=1) 和 9% (bs=32)，加了反而提升 3-8%，理由是 ptxas 面向原生架构生成更优 SASS，且 FlashMLA 自己的 setup.py 也做了同样选择。
- cutlass 补丁的价值：以前 SM103 patch 从未被任何编译单元触发，加入 sm_103a 后它成为必需，这说明过去它只是防御性补丁，本次升级让它变成事实依赖。
- 测试探针的缺陷：ascii_ratio 探针在 base 检查点上测的是续写语言而非输出健康度，因为裸指令没有固定答案，DeepSeek-V4-Flash-0731 会续写出语言不定的 JSON 记录。

唯一可见的 PR 评论是作者上报 CI 状态：base 测试 (Run #34738064047) 与 extra 测试 (Run #34738063907) 全部通过；AMD ROCm 10 测试 (Run #34738064037) 为失败状态，作者未作说明。

- CI 验证状态 (other): base 与 extra CI 全绿，PR 由作者自行合并；AMD ROCm 10 测试 (Run #34738064037) 仍显示失败 (x)，作者未在评论中说明原因。

风险与影响

- 风险：
 1. 构建配置对上游目录高度敏感：flashmla.cmake 中约 50 行是纯路径改写，上游 v4.2 若再次调整 csrc 目录结构，需要再次人工同步，容易漏改导致链接失败。
 2. 前向声明是脆弱耦合：flashmla_extension.cc 绕过头文件直接声明上游符号，参数 const 属性、optional 引用等细节不一致时，问题可能延迟到链接期甚至运行期才暴露；sparse 接口本次已新增一个参数，后续上游再改签名这里要同步跟进。
 3. 新增编译目标成本：sm_103a 会让 AOT wheel 的 SASS 体积与编译时长增加；CUDA 12.x 用户跑 B300 仍只能走 sm_100f 兼容路径，拿不到 3-8% 收益。
 4. ROCm CI 失败未解释：AMD ROCm 10 测试在本 PR 上显示失败 (x)，作者没有说明，合并前应确认与 FlashMLA 升级无关。
 5. 探针修正范围有限：ascii_ratio 只修了英文 Q/A 探针，其他语言或模型若出现类似输出漂移，该探针仍可能产生噪音。 - 影响：对 DeepSeek V4.1 / MLA 推理用户：默认获得 v4.1 新内核 (sm90 v4 sparse decode、v32 no-rope 变体、sm100 v41 decode 实例)，B300 上 decode 吞吐提升约 3-8%。对构建与发布：AOT 包新增 sm_103a SASS，wheel 体积与构建时间上升，CUDA 13+ 成为 B300 性能路径的前置条件。对维护团队：与 FlashMLA fork 的同步节奏、cutlass SM103 补丁的维护成为持续负担；同时表明 sgl-kernel 的回归测试 kit 开始承担外部分核升级的验证职责，测试基础设施在发挥兜底作用。 - 风险标记：上游接口脆弱耦合，构建配置大改，AMD ROCm CI 未通过，依赖 pin 升级有回归风险

关联脉络

- PR #38942 (标题未在上下文中提供, 本 PR 为其 rebase 版本): PR body 明确说明 "It's just #38942 rebased on main branch", 本 PR 的全部变更内容均来自该 PR 在 main 分支上的重基。
- PR #39324 chore: bump sgl-kernel version to 0.4.7: 同属 kernels/aot 构建域的外部分核依赖升级, 反映 sgl-kernel 正在持续跟随 FlashMLA 与 sgl-kernel 双线 pin 的更新节奏; 未来 DeepSeek v4.1 的 MLA 相关功能将建立在本 PR 的 pin 之上。