PR #24890 完整报告

sgl-project/sglang

Port KV Compression V2 from deepseek_v4_dev

合并时间：2026-05-13 22:40

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24890

执行摘要

- 一句话：移植 DeepSeek V4 融合 KV 压缩 V2 内核到主分支
- 推荐动作：建议开发团队深入阅读 compressor_v2.py 和 compress.py 的设计，了解在线 / 离线压缩路径的统一抽象。同时关注 Compressor 内部方法暴露导致的耦合问题，未来可考虑通过策略模式或接口抽象进一步解耦。对于正在使用 DeepSeek V4 的团队，应通过全面的集成测试验证 v2 路径的端到端行为。

功能与动机

PR 旨在将 DeepSeek V4 的融合 KV 压缩 V2 内核从 deepseek_v4_dev 分支移植到主分支，让主分支能够使用优化的在线压缩、预填充和解码内核，提升模型推理性能。根据 PR 描述，该移植包含了 c4、c128 以及 online c128 三种内核。

实现拆解

1. JIT 内核模块: 新增 python/sglang/jit_kernel/dsv4/compress.py，定义了 _jit_compress_norm_rope_module、_jit_compress_module、_jit_compress_128_online_module 和 _jit_compress_plan_module 等 JIT 编译函数，以及 CompressorDecodePlan、CompressorPrefillPlan 等计划数据结构，用于管理压缩过程中所需的 CUDA 内核和元数据。
2. 压缩后端 Mixin: 新增 python/sglang/srt/layers/attention/dsv4/compressor_v2.py，提供 CompressorBackendMixin 类，封装了 _forward_compress_all_in_one 和 forward_unified 方法，统一处理在线 / 离线压缩路径，并通过 _use_online_compress 控制条件。
3. 模型端调整: 修改 python/sglang/srt/models/deepseek_v4.py，将原先分开的 RMSNorm、RoPE 和 KV 写入步骤融合为 fused_q_norm_rope、fused_norm_rope_inplace 等调用，直接写入 FlashMLA 分页缓存，移除了 bf16 中间 KV 张量。
4. Q 融合内核: 扩展 python/sglang/jit_kernel/deepseek_v4.py，新增 _jit_main_q_norm_rope_module、_jit_main_k_norm_rope_flashmla_module、_jit_main_q_indexer_rope_hadamard_quant_module 等 JIT 模块，提供 fused 的 Q 和 K 处理内核，包括 norm、RoPE、Hadamard 量化和 FlashMLA 缓存写入。
5. 测试覆盖: 新增 test_c4_v2.py 和 test_c128_v2.py，包含基于 fp64 的参考实现（_gt_compress）对比测试，覆盖 prefill no context、prefill then decode、prefill then

extend 等场景，确保内核正确性；通用上下文辅助函数放在 common.py 中。

关键文件：

- python/sglang/jit_kernel/dsv4/compress.py（模块 JIT 内核；类别 source；类型 core-logic；符号 _jit_compress_norm_rope_module, _jit_compress_module, _jit_compress_128_online_module, _jit_compress_plan_module）：核心 JIT 内核模块，定义所有压缩内核和计划数据结构的 JIT 编译入口，是 V2 压缩的底层基石。
- python/sglang/srt/layers/attention/dsv4/compressor_v2.py（模块 压缩器；类别 source；类型 dependency-wiring；符号 _use_online_compress, CompressorBackendMixin, init, _maybe_upgrade_forward_metadata）：统一压缩后端 Mixin，封装在线 / 离线压缩逻辑，连接 JIT 内核和模型层。
- python/sglang/srt/models/deepseek_v4.py（模块 模型层；类别 source；类型 data-contract；符号 _compute_kv_to_cache, _compute_kv, _compute_kv_bf16）：模型层 KV 计算路径重构，融合 norm 和 RoPE，移除 bf16 中间变量。
- python/sglang/jit_kernel/deepseek_v4.py（模块 JIT 内核；类别 source；类型 core-logic；符号 _jit_main_q_norm_rope_module, _jit_main_k_norm_rope_flashmla_module, _jit_main_q_indexer_rope_hadamard_quant_module, fused_norm_rope_inplace）：新增主路径 Q/K 融合内核的 Python 包装函数和 JIT 模块加载。
- python/sglang/jit_kernel/tests/deepseek_v4/test_c4_v2.py（模块 测试；类别 test；类型 test-coverage；符号 _gt_compress, _run_prefill, _run_decode, _make_inputs）：c4 压缩内核的全面单元测试，包含 fp64 参考实现对比。
- python/sglang/jit_kernel/tests/deepseek_v4/test_c128_v2.py（模块 测试；类别 test；类型 test-coverage；符号 _gt_compress, _make_inputs, _run_prefill, _run_decode）：c128 压缩内核的全面单元测试，覆盖 prefill 多批次等场景。
- python/sglang/jit_kernel/tests/deepseek_v4/common.py（模块 测试；类别 test；类型 test-coverage；符号 LegacyContext, num_pages, state_loc, make_prefill_plan）：测试辅助上下文，支持 legacy 和 paged 两种模式下的计划生成。

关键符号：_jit_compress_norm_rope_module, _jit_compress_module, _jit_compress_128_online_module, _jit_compress_plan_module, CompressorDecodePlan.generate, CompressorDecodePlan.generate_legacy, CompressorBackendMixin._forward_compress_all_in_one, CompressorBackendMixin.forward_unified, _compute_kv_to_cache, _compute_q_b, fused_q_norm_rope, fused_norm_rope_inplace, fused_q_indexer_rope_hadamard_quant, fused_k_norm_rope_flashmla, _gt_compress, test_prefill_no_context, test_prefill_then_decode, test_prefill_then_extend

关键源码片段

python/sglang/srt/models/deepseek_v4.py

模型层 KV 计算路径重构，融合 norm 和 RoPE，移除 bf16 中间变量。

```
    def _compute_q_b(
        self, q: torch.Tensor,
```

```
        positions: torch.Tensor,
        q_out: Optional[torch.Tensor] = None,
    ) -> torch.Tensor:
        # wq_b 线性映射
        q, _ = self.wq_b(q)
        q = q.view(-1, self.n_local_heads, self.head_dim)
        if q_out is None:
            q_out = torch.empty_like(q)
        # 融合：warp-per-(token, head) 的 rmsnorm-self + RoPE，直接写入 q_out
        fused_q_norm_rope(q, q_out, self.eps, self.freqs_cis, positions)
        return q_out

    def _compute_kv_to_cache(
        self,
        x: torch.Tensor,
        positions: torch.Tensor,
        forward_batch: ForwardBatch,
        qkv_a: Optional[torch.Tensor] = None,
    ) -> None:
        # 融合路径：rmsnorm + RoPE + 直接写入 FlashMLA 分页缓存。
        # 替代旧的 bf16 中间 KV 张量路径。除 NSA prefill-CP 场景外均使用此路径。
        if qkv_a is not None:
            kv = qkv_a[..., self.q_lora_rank :]
        else:
            kv, _ = self.wkv(x)
        token_to_kv_pool = forward_batch.token_to_kv_pool
        # 调用底层 fused 内核完成 norm + RoPE + 缓存写入
        fused_k_norm_rope_flashmla(
            kv,
            self.kv_norm.weight,
            self.kv_norm.variance_epsilon,
            self.freqs_cis,
            positions,
            token_to_kv_pool,
            forward_batch,
            self.layer_id,
        )
```

评论区精华

在审查中，gemini-code-assist[bot] 指出将 Compressor 的内部方法 _get_state_pool 改为公开的 get_state_pool，并添加注释说明被 compressor_v2 使用，这违反了封装原则，建议重构避免直接访问内部方法。该评论未收到回复或进一步的修复，PR 仍以此方式合并，表明团队在当前阶段优先追求功能对齐。

- 封装违反：Compressor 内部方法暴露导致紧耦合 (design): 未看到后续处理或重构，PR 已合并，该问题未解决。

风险与影响

- 风险：
  - 耦合风险：CompressorBackendMixin 直接调用 Compressor.get_state_pool，导致两个模块紧密耦合，未来修改 Compressor 内部实现时可能连带影响 v2 后端。
  - 兼容性风险：新的 v2 压缩路径与原有 v1 路径共存，但模型端 deepseek_v4.py 的修改去除了部分 v1 逻辑（如 overlap_store_cache 标记失效），可能对依赖旧行为的用户造成意外。
  - JIT 编译风险：新增了大量 JIT 内核（c4_v2.cuh、c128_v2.cuh、fused_norm_rope_v2.cuh 等），JIT 编译可能在某些 GPU 架构上失败或产生次优性能。
  - 测试覆盖局限：虽然单元测试较全面，但缺少端到端 E2E 测试验证 v2 路径在完整模型推理中的正确性。
- 影响：
  - 用户影响：DeepSeek V4 用户将自动受益于 KV 压缩性能提升和显存减少（当新内核被启用时）。
  - 系统影响：新增 CUDA 源码文件 (~1000+ 行) 和 JIT 编译模块，增加了构建时间和部署包体积。
  - 团队影响：需要维护两套压缩后端（v1 和 v2），增加了长期维护成本。
  - 风险标记：核心路径变更，模块耦合风险，JIT 编译可靠性，兼容性风险

关联脉络

- PR #25120 [env] Make max KV chunk capacity configurable via SGLANG_MAX_KV_CHUNK_CAPACITY：同属 DeepSeek V4 优化系列，修改 deepseek_v4.py 和 environ.py，与 KV 压缩容量配置相关，可能影响 V2 压缩路径的行为。