PR #27277 完整报告

sgl-project/sglang

Deepseek v4: support mixed dtype compression states

合并时间：2026-06-17 15:56

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27277

执行摘要

- 一句话：DeepSeek V4 压缩状态支持混合精度 (BF16)
- 推荐动作：值得精读，尤其是 CUDA kernel 中通过 if constexpr (std::is_same_v<>) 在编译期隔离两套加载策略的设计模式，既保证了新功能不退化旧路径，又保持了代码的可维护性。此外，使用模块级函数 _get_dsv4_compress_state_dtypes 集中解析环境变量，并在 _apply_memory_pool_config 中仅为 DSV4 模型调用，是一种清晰的运行时特性封装。

功能与动机

减少压缩状态的内存带宽与存储开销；benchmark 显示大 batch C128 decode 下 BF16 相比 FP32 可达 1.365x 加速（详见 PR body 中的 Speed Tests 表格）。

实现拆解

1. CUDA kernel 模板改造(c4_v2.cuh / c128_v2.cuh)：引入模板参数 BufferFloat，使用 if constexpr (std::is_same_v<BufferFloat, InputFloat>) 在编译期分支原始 FP32 路径与混合精度路径；混合精度路径额外分配 BufferFloat 对齐的存储并独立转换。
2. 运行时环境变量(environ.py → model_runner_kv_cache_mixin.py、pool_configurator.py)：添加 SGLANG_DSV4_COMPRESS_STATE_DTYPE 解析函数，分别返回 dtype 和 dtype size；同时检查与在线压缩 (SGLANG_OPT_USE_ONLINE_COMPRESS) 的冲突。
3. 内存池参数分离(deepseek_v4_memory_pool.py)：将原来统一的 state_dtype 拆分为 c4_state_dtype / c128_state_dtype，在 _init_paged_compress_states 中按 ratio 选择 dtype。
4. JIT kernel 入口适配(compress.py)：_jit_compress_module 新增 dtype_buffer 参数，传递给 C++ 模板。
5. 测试与基准：新增 test_deepseek_v4_compress_state_runtime_shapes.py（1082 行），覆盖预设 shape 和运行时 replay shape 的 FP32/BF16 对比，并可导出 CSV 用于回归分析。更新 dsv4_attention.py 单测传递新 dtype 参数。

关键文件：

- test/registered/jit/test_deepseek_v4_compress_state_runtime_shapes.py（模块 测试与基准；类别 test；类型 test-coverage；符号 ShapePreset, BenchSpec, BenchInput, effective_bytes）：新增的 operator benchmark 与注册测试，覆盖 1082 行，包含预设

shape 和运行时 replay shape 的双精度（FP32/BF16）对比，是验证混合精度正确性和性能的核心测试。

- python/sglang/srt/model_executor/model_runner_kv_cache_mixin.py（模块 模型运行器；类别 source；类型 data-contract；符号 _get_dsv4_compress_state_dtypes）：核心运行时代码，添加 _get_dsv4_compress_state_dtypes 函数解析环境变量，并在 _init_pools 和 _apply_memory_pool_config 中传递分离后的 c4_state_dtype / c128_state_dtype。
- python/sglang/srt/model_executor/pool_configurator.py（模块 内存池配置器；类别 source；类型 data-contract；符号 _get_dsv4_compress_state_dtype_sizes）：内存池配置的核心，添加 _get_dsv4_compress_state_dtype_sizes 函数，并在 _get_bytes_per_full_token 中使用以正确计算 C4/C128 状态的字节数。
- python/sglang/srt/mem_cache/deepseek_v4_memory_pool.py（模块 内存池；类别 source；类型 core-logic）：内存池核心逻辑，将 __init__ 参数从 state_dtype 拆分为 c4_state_dtype / c128_state_dtype，并在 _init_paged_compress_states 中根据 ratio 选择 dtype。
- python/sglang/jit_kernel/dsv4/compress.py（模块 JIT 压缩内核；类别 source；类型 core-logic）：JIT kernel 入口，_jit_compress_module 新增 dtype_buffer 参数并传递给 C++ 模板，compress_forward 根据 kv_score_buffer.dtype 传入。
- python/sglang/jit_kernel/csrc/deepseek_v4/c4_v2.cuh（模块 C4 CUDA 内核；类别 other；类型 dependency-wiring）：C4 压缩算子的 CUDA 核心实现，引入 BufferFloat 模板参数并通过 if constexpr 分支混合精度路径，新增 151 行，修改 65 行。
- python/sglang/jit_kernel/csrc/deepseek_v4/c128_v2.cuh（模块 C128 CUDA 内核；类别 other；类型 dependency-wiring）：C128 压缩算子的 CUDA 核心实现，与 c4_v2.cuh 类似，引入 BufferFloat 模板分支，新增 114 行，修改 50 行。

关键符号：_get_dsv4_compress_state_dtypes, _get_dsv4_compress_state_dtype_sizes, compress_forward, _init_paged_compress_states, _get_bytes_per_full_token, _init_pools, _jit_compress_module, c4_forward, c128_forward

关键源码片段

test/registered/jit/test_deepseek_v4_compress_state_runtime_shapes.py

新增的 operator benchmark 与注册测试，覆盖 1082 行，包含预设 shape 和运行时 replay shape 的双精度（FP32/BF16）对比，是验证混合精度正确性和性能的核心测试。

```
@dataclass(frozen=True)
class ShapePreset:
    """定义一组预设 shape 参数，用于 C4/C128 压缩算子基准测试。"""
    name: str
    hidden_size: int
    num_attention_heads: int
    index_topk: int
    ratios: tuple[Literal[4, 128], ...]
    decode_batch_sizes: dict[ShapeTier, tuple[int, ...]]
    prefill_batch_sizes: dict[ShapeTier, tuple[int, ...]]
    decode_seq_lens: dict[ShapeTier, tuple[int, ...]]
```

```
    prefill_extend_lens: dict[ShapeTier, dict[Literal[4, 128], tuple[int, ...]]]

# DeepSeek-V4-Flash 配置：hidden_size=4096, num_attention_heads=64, index_topk=512
SHAPE_PRESETS: dict[str, ShapePreset] = {
    "flash": ShapePreset(
        name="flash",
        hidden_size=4096,
        num_attention_heads=64,
        index_topk=512,
        ratios=(4, 128),
        decode_batch_sizes={
            "ci": (16,),
            "smoke": (16, 128),
            "full": (1, 2, 4, 8, 16, 32, 64, 128),
        },
        prefill_batch_sizes={
            "ci": (1,),
            "smoke": (1, 2, 4, 8, 16, 32),
            # ...
        },
        # ...
    ),
}
```

评论区精华

1. FP32 性能回归讨论：Reviewer DarkSharpness 指出“引入 mixed-dtype 会导致寄存器压力翻倍，恶化默认 FP32 性能”。作者 DaZhUUU 通过 if constexpr 在编译期分离两套加载路径，确保 FP32 路径生成与原始代码相同的 SASS，并通过 cuobjdump 验证。最终结论：原始 FP32 路径无性能损失。
2. 测试位置调整：DarkSharpness 要求将测试与基准移至 test/registered/jit 目录并复用 JIT kernel utilitie，作者照做并更新命令。

- FP32 性能回归风险与编译期分支隔离 (performance): FP32 路径保持原实现，无性能损失；同时在新路径中只分配额外寄存器用于 mixed-dtype 转换。
- 测试基准位置与 CI 注册 (testing): 测试移动完成，并注册到 base-b-kernel-unit-1-gpu-large CI suite。

风险与影响

- 风险：
  1. FP32 默认路径回归：尽管已用编译期分支分离且验证 SASS 一致，但若修改 C++ 模板时意外影响分支结构则可能导致性能退化（风险低，有测试覆盖）。
  2. BF16 数值精度：当前仅通过模型级 GSM8K/MMLU/GPQA 测试，未对压缩状态本身的数值误差进行逐层分析（风险中等）。
  3. 与在线压缩的冲突：_get_dsv4_compress_state_dtypes 和 _get_dsv4_compress_state_dtype_sizes 均在 BF16 路径下抛出 ValueError，但若用

户未设置环境变量则不会触发（风险低）。

4. draft worker 路径：draft worker 虽不拥有压缩状态池，但仍会调用 _get_dsv4_compress_state_dtypes 初始化 dtype 属性，环境变量缺失时使用 FP32 默认（风险低）。 - 影响：影响范围限定于使用 DeepSeek V4 模型的用户：设置 SGLANG_DSV4_COMPRESS_STATE_DTYPE=bf16 即可在 C4/C128 压缩状态上获得存储减半和带宽降低，尤其大 batch decode 场景加速明显；默认 FP32 路径完全不变。对非 DSV4 模型无任何影响。内存池接口变更（state_dtype → c4_state_dtype / c128_state_dtype）仅影响直接实例化 DeepSeekV4TokenToKVPool 的代码，但仓库内所有调用点已同步更新。 - 风险标记：FP32 路径回归已验证无影响，BF16 精度未充分逐层验证，与 online compress 不兼容已检查，环境变量解析失败时默认回退 FP32

关联脉络

- PR #24041 Dev branch for DeepSeek V4 mixed dtype (not provided in history, referenced in PR body): 该 PR 的主分支 PR，包含初始实现，作者在本 PR 中引用并协作开发。