PR #34926 完整报告

sgl-project/sglang

Clean deprecated DeepSeek V4 Environs

合并时间：2026-08-18 07:07

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34926

执行摘要

- 一句话：清理 DeepSeek V4 废弃环境变量，统一后端路径
- 推荐动作：值得精读，特别是 compress_hip.py 和 deepseek_v4_backend.py 的 diff，展示了如何通过删除历史开关来收敛配置面。推荐关注两点：一是 AMD 特有路径被回收时的取舍，二是 use_swizzle 从环境变量变为后端推导的依赖方向。

功能与动机

仓库近期持续推动配置系统精简（如 PR#35060 清理 environ.py）。PR 标题和提交信息表明目标是清理 DeepSeek V4 中已废弃的环境变量：随着这些开关对应的旧实现不再被推荐使用，继续保留会带来配置面膨胀和分支路径的维护负担。PR body 使用模板没有给出具体动机，但从删除清单可以看出是想把多处二选一的 fallback 收敛为单一稳定路径。

实现拆解

1. 配置注册清理：python/sglang/srt/environ.py 删除一批废弃环境变量定义，同步更新 arg_groups/overrides.py、arg_groups/deepseek_v4_hook.py、server_args.py 中的相关引用和校验逻辑，避免残留开关继续进入配置解析。
2. Attention 压缩路径收敛：compress_hip.py 删除 use_fused_compress、use_hip_fused_compress、use_fused_compress_triton 三个 cached_property 及对应的 Triton fused 分支，compress_extend_paged 和 compress_decode_paged 统一走 fused_softmax_pool_triton + fused_norm_rope_inplace_triton；compress_dispatch 不再受 SGLANG_OPT_DPSK_V4_RADIX 门控，decode / extend-without-spec 直接进入 fused 后端压缩。
3. Metadata 准备路径收敛：deepseek_v4_backend.py 和 deepseek_v4_backend_hip_radix.py 删除 SGLANG_PREP_IN_CUDA_GRAPH 条件分支，decode 固定返回 DSV4RawDecodeMetadata，target_verify 固定返回 DSV4RawVerifyMetadata；原先分散在 init_forward_metadata_target_verify_old 中的逻辑内联进主函数，needs_cpu_seq_lens 的计算也去掉该开关。
4. MoE 权重与激活路径收敛：mega_moe_sm90.py 和 mega_moe.py 删除 SGLANG_OPT_FIX_MEGA_MOE_MEMORY 分支，统一 interleave 权重布局；deep_gemm.py 的 use_swizzle 改为从 get_moe_a2a_backend().is_megamoe() 动态推导，并删除 SGLANG_OPT_SWIGLU_CLAMP_FUSION 与 SGLANG_OPT_USE_JIT_EP_ACTIVATION 断言；fused_moe.py 也统一了

silu_and_mul_clamp 路径。

5. DSpark 规划器清理：dspark_planner.py 删除 _require_prep_in_cuda_graph 方法，不再强制校验 SGLANG_PREP_IN_CUDA_GRAPH，避免与上一步的 CUDA graph 分支删除产生矛盾。
6. 测试与文档配套：test_model_overrides.py 删除被清理开关对应的测试用例；docs 侧通过 mintlify 预览更新了环境变量文档。

关键文件：

- python/sglang/srt/layers/attention/dsv4/compress_hip.py（模块 压缩器；类别 source；类型 core-logic；符号 use_fused_compress, use_hip_fused_compress, use_fused_compress_triton, compress_dispatch）：HIP 压缩器核心，删除三项 fused 开关和三套分支，compress_dispatch / compress_decode_paged / compress_extend_paged 统一到 fused_softmax_pool_triton 路径。
- python/sglang/srt/layers/attention/deepseek_v4_backend.py（模块 注意力后端；类别 source；类型 core-logic；符号 init_forward_metadata_target_verify, init_forward_metadata_decode, init_forward_metadata_target_verify_old, store_cache）：注意力后端主文件，删除 SGLANG_PREP_IN_CUDA_GRAPH 分支和 init_forward_metadata_target_verify_old，decode / target_verify 固定返回 raw metadata，并统一 fused store cache。
- python/sglang/srt/layers/attention/dsv4/compressor_v2.py（模块 压缩器；类别 source；类型 dependency-wiring；符号 forward_unified, _forward_unified_hip）：删除 HIP 专属 _forward_unified_hip 和 SGLANG_OPT_USE_JIT_NORM 分支，统一走 _forward_compress_all_in_one，同时移除非 fused store cache。
- python/sglang/srt/layers/attention/deepseek_v4_backend_hip_radix.py（模块 注意力后端；类别 source；类型 dependency-wiring；符号 init_forward_metadata_decode, init_forward_metadata_target_verify, store_cache）：HIP radix 后端同步删除 SGLANG_PREP_IN_CUDA_GRAPH 分支和 store cache 非 fused 路径。
- python/sglang/srt/layers/moe/mega_moe_sm90.py（模块 MoE；类别 source；类型 dependency-wiring；符号 build_sm90_mega_moe_experts_weights, run_sm90_mega_routed）：SM90 MegaMoE 权重构建删除 FIX_MEGA_MOE_MEMORY 开关，不再调用 deep_gemm.transform_weights_for_mega_moe_sm90，统一 interleave 布局。
- python/sglang/srt/layers/moe/moe_runner/deep_gemm.py（模块 MoE；类别 source；类型 dependency-wiring；符号 DeepGemmRunnerCore, _run_contiguous_gemm, _run_masked_gemm）：DeepGEMM runner 的 use_swizzle 从环境变量改为由 MegaMoE 后端推导，删除 swiglu clamp 与 JIT EP activation 断言，同时删除 einops import。

关键符号：compress_dispatch, compress_extend_paged, compress_decode_paged, init_forward_metadata_decode, init_forward_metadata_target_verify, init_forward_metadata_target_verify_old, forward_unified, _forward_unified_hip, build_mega_moe_experts_weights, _run_contiguous_gemm

关键源码片段

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

HIP 压缩器核心，删除三项 fused 开关和三套分支，compress_dispatch / compress_decode_paged / compress_extend_paged 统一到 fused_softmax_pool_triton 路径。

```
def compress_dispatch(
    self,
    kv_score: torch.Tensor,
    forward_batch: ForwardBatch,
    attn_backend: AttentionBackend,
) -> torch.Tensor:
    # 清理前受 SGLANG_OPT_DPSK_V4_RADIX 和 use_fused_compress 双重门控，
    # 现在 decode / extend-without-spec 直接走 fused 后端压缩。
    if (
        forward_batch.forward_mode.is_decode()
        or forward_batch.forward_mode.is_extend_without_speculative()
    ):
        return self.compress_fused(kv_score, forward_batch, attn_backend=attn_backend)

    # 其余模式（target_verify、extend 等）走 paged 路径
    self.compress_decode = self.compress_decode_paged
    self.compress_extend = self.compress_extend_paged
    kv_and_scores = KVAndScore(kv_score)

    if TYPE_CHECKING:
        assert isinstance(kv_and_scores, KVAndScore)

    if (
        forward_batch.forward_mode.is_decode()
        or forward_batch.forward_mode.is_target_verify()
    ):
        result = self.compress_decode(
            kv_and_scores=kv_and_scores,
            forward_batch=forward_batch,
            attn_backend=attn_backend,
        )
    elif forward_batch.forward_mode.is_extend():
        result = self.compress_extend(
            kv_and_scores=kv_and_scores,
            forward_batch=forward_batch,
            attn_backend=attn_backend,
        )
    else:
        msg = f"Forward mode {forward_batch.forward_mode} not supported in Compressor."
        raise NotImplementedError(msg)
    return result
```

python/sglang/srt/layers/attention/deepseek_v4_backend.py

注意力后端主文件，删除 SGLANG_PREP_IN_CUDA_GRAPH 分支和 init_forward_metadata_target_verify_old，decode / target_verify 固定返回 raw metadata，并统一 fused store cache。

```
  def init_forward_metadata_target_verify(
      self,
      max_seq_len: int,
      req_pool_indices: torch.Tensor,
      seq_lens: torch.Tensor,
      seq_lens_cpu: Optional[torch.Tensor] = None,
      out_cache_loc: Optional[torch.Tensor] = None,
      use_prefill_cuda_graph: bool = False,
      online_c128_state_slot_offset: int = 0,
      ragged_layout: Optional[RaggedVerifyLayout] = None,
  ) -> Union[DSV4Metadata, DSV4RawVerifyMetadata]:
      # 删除 SGLANG_PREP_IN_CUDA_GRAPH 分支后，这里无条件构造 raw verify metadata，
      # 旧的 eager 路径 init_forward_metadata_target_verify_old 被内联删除。
      assert out_cache_loc is not None
      bs = len(seq_lens)
      if self.needs_cpu_seq_lens:
          assert seq_lens_cpu is not None
          seq_lens_cpu_list = seq_lens_cpu.tolist()
      else:
          seq_lens_cpu_list = None
      if ragged_layout is None:
          # 均匀扩展：每个请求的 verify token 数相同
          self.extend_seq_lens_buffer[:bs].fill_(self.speculative_num_draft_tokens)
          extend_seq_lens = self.extend_seq_lens_buffer[:bs]
          extend_start_loc = None
          verify_lens = None
          total_verify_tokens = self.speculative_num_draft_tokens * bs
      else:
          # ragged 扩展：按布局填充每个请求的 verify_lens
          self.extend_seq_lens_buffer[:bs].copy_(ragged_layout.verify_lens)
          self.extend_start_loc_buffer[:bs].copy_(ragged_layout.extend_start_loc)
          extend_seq_lens = self.extend_seq_lens_buffer[:bs]
          extend_start_loc = self.extend_start_loc_buffer[:bs]
          verify_lens = self.extend_seq_lens_buffer[:bs]
          total_verify_tokens = ragged_layout.graph_num_tokens

      return DSV4RawVerifyMetadata(
          req_pool_indices=req_pool_indices,
          seq_lens=seq_lens,
          out_cache_loc=out_cache_loc,
          extend_seq_lens=extend_seq_lens,
          seq_lens_cpu=seq_lens_cpu_list,
          c128_compress_metadata=self._make_target_verify_c128_metadata(
```

```
            req_pool_indices,
            seq_lens,
            seq_lens_cpu_list,
            extend_seq_lens,
            use_prefill_cuda_graph,
            online_c128_state_slot_offset,
        ),
        extend_start_loc=extend_start_loc,
        verify_lens=verify_lens,
        total_verify_tokens=total_verify_tokens,
    )
```

评论区精华

该 PR 没有实质 review 评论，唯一的评论是 mintlify 的文档预览通知。从 19 个 commit 的提交信息可以看出多次迭代：先移除 RADIX 与压缩相关开关（'clean SGLANG_OPT_DPSK_V4_RADIX'、'remove amd triton fused compress'），再清理 MoE 与 store cache（'remove some moe flags'、'remove fused store cache flag'），最后删除 cuda graph 开关（'remove prepare in cuda graph flag'）并修复后端守卫（'Fix DSV4 backend guards after env cleanup'）。值得注意的是 'remove amd triton fused compress' 表明这次清理是有意撤回 AMD 特有的 fused Triton 压缩路径，统一到通用实现。

- 暂无高价值评论线程

风险与影响

- 风险：具体到文件：
  - compress_hip.py：HIP 路径删除 use_hip_fused_compress 分支后，fused_softmax_pool_triton 成为唯一路径，若该 kernel 在 ROCm 上存在精度或性能问题将没有回退选项。
  - compressor_v2.py：删除 _forward_unified_hip 后，HIP 上 compress 不再走单独的 PyTorch fallback，统一走 _forward_compress_all_in_one，需要验证 AMD 上 JIT kernel 与 norm 的兼容性。
  - deepseek_v4_backend.py：删除 SGLANG_PREP_IN_CUDA_GRAPH 分支后，decode / target_verify 的 metadata 始终是 raw 结构，依赖 capture 路径的调用方行为会变化；同时 needs_cpu_seq_lens 的计算也去掉了该开关，可能影响 cuda graph 捕获的 CPU 同步点。
  - deep_gemm.py / mega_moe.py：use_swizzle 改为依赖 a2a 后端类型，若 MegaMoE 后端未被正确识别，将进入非 swizzle 路径导致权重布局不匹配。
  - dspark_planner.py：移除 _require_prep_in_cuda_graph 校验后，若用户仍关闭 prep-in-cuda-graph，DSpark 每步的 verify_lens_cpu 主机读会回到关键路径，性能可能回退，但不再有显式报错。
  - 影响：影响范围：DeepSeek V4 部署用户不再需要（也不应）设置这些环境变量，设置后会被忽略或直接删除；内部团队受益于更小的配置面。工程上，attention 压缩、MoE 权重构建和 DSpark 规划三条链路的代码路径减少，后续维护和调试成本下降。同时也为后

续 config 系统重构（如 PR#35022-35027 的 bags 化）扫清障碍。

- 风险标记：AMD 路径统一风险，CUDA graph 分支删除，配置面收窄

关联脉络

- PR #35060 Clean up environ.py: remove dead env vars, unify deprecation handling, move examples to a unit test: 同一波环境变量清理，本 PR 继续删除 DeepSeek V4 专属开关，与 environ.py 的清理方向一致。
- PR #35022 config: retire the multi-engine accommodation in the runtime context: 配置精简浪潮的延续，移除历史兼容路径，与本 PR 的配置收敛目标相同。
- PR #35025 config: the DP/EP topology reads come from the parallel bag: 同期的配置读取重构，本 PR 为后续 bags 化扫清环境变量依赖。
- PR #35026 config: the per-instance families read the bags: 配置收敛趋势的一部分，与本 PR 共同推动配置面简化。