PR #29202 完整报告

sgl-project/sglang

[AMD] Enable draft-extend CUDA graph and reduce bubble for MTP

合并时间：2026-08-13 15:50

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29202

执行摘要

- 一句话：DSV4 MTP 在 AMD 启用 draft-extend CUDA graph，TTT 提升可达 11.7%
- 推荐动作：值得精读，尤其适合关注 ROCm/HIP 后端、投机解码（EAGLE/MTP）与 CUDA graph 捕获的工程师。三个值得借鉴的点：一是用 repeat_interleave(..., output_size=...) 消除隐式 D2H 同步的微优化技巧，以及面对未知 kernel 故障时"分路径降级、保留注释留档"的务实决策；二是 torch.profiler 引用循环与 gc.collect() 受控回收的处理模式，是 Python + pybind11 混合编程的典型坑；三是"按硬件后端局部 import、惰性加载"隔离依赖的写法。但需注意当前验证完全依赖 nightly 人工测试，合并前应保持对 target-extend 故障的敏感度。

功能与动机

PR body 列出三个独立问题：一是 HIP draft-extend CUDA-graph 门控不认识 DSV4 backend，导致可被 graph 捕获的 draft_extend 阶段退回 eager；二是 --profile 场景下对 _attach_unified_kv_prefill_meta 改动后，服务在垃圾回收期间可靠崩溃，阻塞了 MTP 路径的 profiling 验证；三是 repeat_interleave 在 unified-kv prefill 元数据构建中引发隐式 device→host 同步，拖慢整个 draft_extend 元数据构建。作者在评论中还补充：最初尝试在 target-extend 路径也消除该同步，但触发 GPU memory access fault 且未能定位是哪个 kernel 导致，因此作为 workaround 仅对 draft-extend 生效。

实现拆解

变更入口是 eagle_worker_v2.py::_capture_cuda_graphs 的 HIP 门控扩展，整体按以下 4 步推进：

1. 扩展HIPdraft-extend图捕获门控（python/sglang/srt/speculative/eagle_worker_v2.py）：将 supports_hip_aiter_draft_extend_graph 重命名为 supports_hip_draft_extend_graph，在原有 AiterMultiStepDraftBackend 判断基础上，用 or isinstance(self.draft_extend_attn_backend, DeepseekV4HipRadixBackend) 增加 DSV4 HIP radix 后端支持；DeepseekV4HipRadixBackend 采用局部 import，避免非 HIP 环境拉入依赖。这样 draft_extend 阶段在 AMD 上可被 CUDA graph 捕获，不再退回 eager。
2. 消除 repeat_interleave 的隐式 D2H 同步（python/sglang/srt/layers/attention/deepseek_v4_backend_hip_radix.py）：_attach_unified_kv_prefill_meta 新增 num_tokens: int 与 need_compress: bool = True 参数，调用方 init_forward_metadata_prefill 传入。当

need_compress=True 时保留原 repeat_interleave 用法；当 need_compress=False（即 draft-extend 路径）时传入 output_size=num_tokens，跳过读取 extend_seq_lens.sum() 回 host 的隐式同步。之所以分路径，是因为 target-extend 上同样改动会触发 GPU memory access fault。

3. 修复 profiler 停止时的 GC 崩溃（python/sglang/srt/managers/scheduler_components/profiler_manager.py）：在 _stop_profile 末尾把原先无条件的 self.torch_profiler = None 改为先判空再置 None 并立即 gc.collect()，避免 torch.profiler 的 pybind11 对象引用循环被延迟的 cyclic GC 回收时发生 C++ 析构 double-free。该修复只在 --profile 启动时执行一次，不引入持续开销。
4. 验证与配套：未新增自动化测试文件；作者手动运行 test/registered/amd/test_deepseek_v4_pro_fp4_mtp.py（nightly 套件）两轮，结果分别为 Accuracy 0.955（avg_spec_accept_length 2.914、speed 151.51 token/s）与 Accuracy 0.943（output_throughput 1353.042 token/s、acc_length 2.96、speed 124.83 token/s），gsm8k 准确率 0.95；另有一次 fix lints 提交修复 CI 的 lint 失败。

关键文件：

- python/sglang/srt/layers/attention/deepseek_v4_backend_hip_radix.py（模块 注意力后端；类别 source；类型 core-logic；符号 _attach_unified_kv_prefill_meta, init_forward_metadata_prefill）：核心性能改动：为 unified-KV prefill 元数据构建消除隐式 D2H 同步，并按 need_compress 分路径传递 output_size；同时因 target-extend 触发 GPU memory access fault 而保留双分支，是本次设计权衡的关键点。
- python/sglang/srt/speculative/eagle_worker_v2.py（模块 投机解码；类别 source；类型 dependency-wiring；符号 _capture_cuda_graphs）：draft-extend CUDA graph 门控的扩展入口：将 DeepseekV4HipRadixBackend 纳入 HIP 支持类型，使 DSV4 MTP 的 draft-extend 阶段从 eager 转为图捕获，这是性能收益的直接来源。
- python/sglang/srt/managers/scheduler_components/profiler_manager.py（模块 性能剖析；类别 source；类型 bugfix；符号 _stop_profile）：修复 --profile 下启动崩溃的独立 bugfix：在受控停止点显式释放 torch_profiler 引用并立即 gc.collect()，规避 pybind11 引用循环被延迟回收时的 double-free。

关键符号：_capture_cuda_graphs, _attach_unified_kv_prefill_meta, init_forward_metadata_prefill, _stop_profile

关键源码片段

python/sglang/srt/layers/attention/deepseek_v4_backend_hip_radix.py

核心性能改动：为 unified-KV prefill 元数据构建消除隐式 D2H 同步，并按 need_compress 分路径传递 output_size；同时因 target-extend 触发 GPU memory access fault 而保留双分支，是本次设计权衡的关键点。

```
# python/sglang/srt/layers/attention/deepseek_v4_backend_hip_radix.py

def _attach_unified_kv_prefill_meta(
    self,
    core: DSV4AttnMetadata,
```

```
    req_pool_indices: torch.Tensor,
    seq_lens: torch.Tensor,
    extend_seq_lens: torch.Tensor,
    num_tokens: int,
    need_compress: bool = True,
) -> None:
    from sglang.kernels.ops.attention.dsv4.unified_kv_kernels.env_gate import (
        is_unified_kv_triton,
    )

    if not is_unified_kv_triton():
        return
    device = req_pool_indices.device
    bs = req_pool_indices.shape[0]
    seq_lens = seq_lens.to(torch.int64)
    extend_seq_lens = extend_seq_lens.to(torch.int64)

    # token -> req index（长度 L = sum(extend_seq_lens)）。
    # 传 output_size 可跳过隐式的 sum() 返回 host 的同步，避免 draft-extend
    # 元数据构建被 D2H 卡住。但 target-extend 路径上同样的优化会触发 GPU
    # 内存访问错误，因此只对 draft-extend（need_compress=False）生效。
    if need_compress:
        bid = torch.repeat_interleave(
            torch.arange(bs, device=device, dtype=torch.int64),
            extend_seq_lens,
        )
    else:
        bid = torch.repeat_interleave(
            torch.arange(bs, device=device, dtype=torch.int64),
            extend_seq_lens,
            output_size=num_tokens,
        )
    if core.unified is None:
        core.unified = UnifiedKvMetadata()
    core.unified.pf_state_slot = req_pool_indices[bid]
    core.unified.pf_chunk_start = (seq_lens - extend_seq_lens)[bid]
    cu_q_per_req = torch.cumsum(extend_seq_lens, dim=0) - extend_seq_lens
    core.unified.pf_cu_q = cu_q_per_req[bid]
    core.unified.pf_final_pos = (seq_lens - 1)[bid]
```

python/sglang/srt/speculative/eagle_worker_v2.py

draft-extend CUDA graph 门控的扩展入口：将 DeepseekV4HipRadixBackend 纳入 HIP 支持类型，使 DSV4 MTP 的 draft-extend 阶段从 eager 转为图捕获，这是性能收益的直接来源。

```
# python/sglang/srt/speculative/eagle_worker_v2.py（_capture_cuda_graphs 内）

supports_hip_draft_extend_graph = False
if _is_hip:
    # 保持局部 import，避免非 HIP 环境引入 aiter / DSV4 HIP 后端依赖。
```

```
    # aiter 把 draft-extend 支持打包在 decode（multi-step）backend 中；
    # DSV4 则在 draft-extend backend 自身暴露该能力。
    from sglang.srt.layers.attention.aiter_backend import AiterMultiStepDraftBackend
    from sglang.srt.layers.attention.deepseek_v4_backend_hip_radix import (
        DeepseekV4HipRadixBackend,
    )

    supports_hip_draft_extend_graph = isinstance(
        self.draft_attn_backend, AiterMultiStepDraftBackend
    ) or isinstance(self.draft_extend_attn_backend, DeepseekV4HipRadixBackend)

# 仅在启用且未被 SGLANG_DISABLE_DRAFT_EXTEND_CUDA_GRAPH 关闭时捕获 draft-extend
graph。
if (
    self.draft_extend_attn_backend
    and not envs.SGLANG_DISABLE_DRAFT_EXTEND_CUDA_GRAPH.get()
    and (
        _is_npu
        or _is_xpu
        or supports_cuda_draft_extend_graph
        or supports_hip_draft_extend_graph
    )
):
    self.cuda_graph_runner_for_draft_extend = Device2ExtendCudaGraphRunner[
        self.target_worker.device
    ](self)
```

python/sglang/srt/managers/scheduler_components/profiler_manager.py

修复 --profile 下启动崩溃的独立 bugfix：在受控停止点显式释放 torch_profiler 引用并立即 gc.collect()，规避 pybind11 引用循环被延迟回收时的 double-free。

```
# python/sglang/srt/managers/scheduler_components/profiler_manager.py（_stop_profile 尾部）

        merge_message = self._merge_profile_traces()
        logger.info(
            "Profiling done. Traces are saved to: %s%s",
            self.torch_profiler_output_dir,
            merge_message,
        )

        # torch.profiler 的 pybind11 对象形成引用循环，若交给随机的 cyclic GC
        # 回收，C++ 析构可能 double-free 导致堆损坏崩溃。这里在受控停止点
        # 主动释放引用并立即触发 GC，保证析构发生在确定位置。
        if self.torch_profiler is not None:
            self.torch_profiler = None
            gc.collect()

        self.profile_in_progress = False
        self.profiler_start_forward_ct = None
```

return ProfileReqOutput(success=True, message=f"Succeeded.{merge_message}")

评论区精华

围绕 CI 覆盖缺口与 target-extend 故障的取舍是讨论核心：

- amd-bot 多次强调 PR CI 无法验证本 PR 改动，因为 DSV4 MTP 相关测试全部注册为 nightly："This PR's changed code is not exercised by any PR-CI test"。HaiShaw 据此要求作者补跑 nightly 套件："can you fill the coverage gap by running extra tests?"。
- 1am9trash 在 approve 时纠正了 amd-bot 的具体判断错误："the amd-bot CI summary is mistaken here: the stage-c-test-large-8-gpu-amd suite it points to does not include the V4 MTP test. That test lives in nightly, so the queued stage-c jobs would not related to this PR anyway."。
- RolaoDenthu 解释为何优化只落到 draft-extend："The earlier D2H optimization caused a GPU memory access fault on the target-extend path with the current codebase, and I couldn't detect which kernel triggered it. As a workaround, the optimization is now applied to draft-extend only."，该取舍被接受并在源码注释中留档。
- 1am9trash 提醒 lint 失败，作者以 fix lints 提交收尾。
- PR CI 覆盖缺口：DSV4 MTP 测试仅 nightly 覆盖 (testing): 作者手动运行 nightly 套件两轮并贴出结果（Accuracy 0.955 / 0.943），覆盖缺口被关闭；reviewer 接受该验证方式后合入。
- target-extend 上 D2H 优化触发 GPU memory access fault，仅应用到 draft-extend (correctness): 以 need_compress 双分支保留工作区，只在 draft-extend 使用 output_size；故障未 root-cause，但已在源码注释和 PR 讨论中留档，被 maintainer 接受。
- amd-bot 对 stage-c 套件包含 V4 MTP 测试的判断有误 (question): 确认 V4 MTP 测试在 nightly；amd-bot 关于 stage-c queued 的警告不适用，后续 amd-bot 也改为直接指出 nightly 覆盖缺口。
- lint 检查失败 (style): lint 已修复，无遗留问题。

风险与影响

- 风险：主要风险集中在验证缺口与未 root-cause 的故障上：

1. 无自动化测试覆盖：三处改动都只在 AMD + DSV4 + MTP + HIP 组合下生效，唯一相关测试 test_deepseek_v4_pro_fp4_mtp.py 是 nightly 套件，PR CI 不执行，回归只能靠人工跑 nightly 兜底。
2. target-extend 故障未 root-cause：deepseek_v4_backend_hip_radix.py 中 need_compress 分支的存在意味着同一函数在 target-extend 与 draft-extend 下行为不同，未来重构若误删分支或把 output_size 推广到 target-extend，可能重现 GPU memory access fault。
3. profiler GC 修复仅缓解症状：profiler_manager.py 的 gc.collect() 只在 --profile 停止时执行一次，开销可忽略，但 torch.profiler pybind11 引用循环的 double-free 是 torch 侧生命周期问题，其他平台或新版本 torch 仍可能暴露。

4. CUDA graph 捕获的显存开销：draft-extend 进入 graph 捕获后，捕获阶段会有额外显存占用，显存受限的 AMD 部署可用 SGLANG_DISABLE_DRAFT_EXTEND_CUDA_GRAPH 关闭。
5. 合并噪音：分支含 9 个 merge commit，eagle_worker_v2.py 多次冲突，虽经 review 确认最终 diff 干净，但合并过程本身有引入错误的窗口。 - 影响：影响范围严格限定在 AMD（HIP/ROCm）上的 DeepSeek-V4 模型 + MTP/EAGLE 投机解码场景，NVIDIA 路径不受影响（reviewer 1am9trash 明确确认）。对 AMD DSV4 MTP 用户，draft-extend 从 eager 变为 CUDA graph 捕获、消除 D2H 同步后，TTT 提升 5.7%-11.7%、ITL 提升 9.7%-13.5%，属于直观的延迟与吞吐收益；本轮实测 accuracy 0.955/0.943 与 gsm8k 0.95 无精度回退。对团队而言，该 PR 确立了 HIP 上按 attention backend 扩展 draft-extend graph 门控的写法，后续新 backend 只需在 supports_hip_draft_extend_graph 中追加类型即可复用；同时留下一处待 root-cause 的隐患记录，提醒后续优化需谨慎。 - 风险标记：缺少测试覆盖，未 root-cause 的已知故障，平台特定变更，CUDA graph 显存开销

关联脉络

- PR #32755 [Perf] Occupancy tuning for DSA indexer fp8-quant Q kernel: 同属 DeepSeek-V4（DSV4）推理性能优化系列，说明 DSV4 在 kernel 与后端路径上的调优是持续主题。
- PR #34642 Revert "[Kimi K3] Fuse MLA gate projection into QKV-A GEMM": 同为投机解码相关 kernel 优化，因长序列回归被回滚，与本 PR' target-extend 故障后分区降级 ' 的处理思路形成对照，说明此类优化需要强验证。
- PR #34653 Enable unified cache out-of-window slot freeing by default: unified KV cache 默认开启的演进与本 PR 的 _attach_unified_kv_prefill_meta（unified-kv prefill metadata）处于同一功能线。
- PR #34644 [Fix] Snapshot req.prefix_indices when the prefix cache is disabled: 同属 radix cache / prefix cache 相关的调度正确性修复，与本 PR 所在的 DSV4 radix 后端路径相关。
- PR #32941 [minimax m3][npu]Adaptation of Minimax M3(w8a8) for NPU platforms [1/2]: 同为 sparse attention + 投机解码（Eagle3）的多后端适配，可对照本 PR 在 HIP 上的图捕获门控模式。