

PR #39423 完整报告

sgl-project/sglang

[NPU][Bugfix] Disable pinned memory to fix DeepSeek-V2 DP-attention hang

合并时间: 2026-09-15 14:59

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39423>

执行摘要

- 一句话: NPU 禁用 pinned memory, 修复 DP 挂死
- 推荐动作: 值得精读 (约 5 分钟)。NPU 平台维护者必读: 这是一个典型的「平台新类覆盖基类安全默认值导致死锁」案例, 7 行改动换来稳定性。关注三个设计决策: (1) 为什么选择无条件 return False 而不是保留 device 分支或加环境变量开关——答案是把默认值收敛到与 SRTPlatform 一致, 避免参数组合分叉; (2) 如何用两处独立复现 (#36472 作者观察 + Ascend 910C TestDPRetract 压测) 为「未验证的异步语义」背书; (3) 重开路径的条件是否应该固化成代码注释或跟踪 issue, 避免 remove comments 之后只剩一行无解释的 return False。

功能与动机

PR body 明确说明根因: #36472 让 NPU 机器解析到新的 NPUSRTPlatform, 而该类「Unlike the base default (is_pin_memory_available=False), the new class returns True, silently switching hot paths such as the next-token-logits copies and the xgrammar bitmask allocation to pinned memory with non_blocking H2D copies」。由于 torch_npu 的 pinned-memory + async-copy 语义尚未针对 CANN 验证, non_blocking H2D 依赖的流序保证在 NPU 上并不成立, 在 compute / copy / HCCL 三流并发的 DP attention 场景下 DeepSeek-V2 会间歇挂死; 作者给出两处独立复现 (#36472 作者在常规 DP attention 推理中观察到挂死, 以及本 PR 在 Ascend 910C 上用 TestDPRetract 压力复现), 并强调单步收益仅限 KB 量级 decode buffer, 因此在语义验证完成前直接关闭是安全选择。

实现拆解

1. 定位回归入口并收敛返回值: 修改 python/sglang/srt/platforms/npu.py 中 NPUDeviceMixin.is_pin_memory_available。原实现是「device 非空且字符串等于 cpu 时返回 False, 否则返回 True」, 与 SRTPlatform 基类默认的 False 相反; 现直接 return False (+1/-3), 删除 device 分支判断, 调用方无论传 npu 还是 cpu 都拿到同一个结果。
2. 影响传播路径: 该方法是平台接口层唯一的「是否允许 pinned buffer」开关, 所有通过它判断的热路径都会回退到 pageable memory + 同步 H2D, PR body 点名的两条是 next-token logits 拷贝与 xgrammar bitmask 分配; 这也意味着 DP attention 下 copy 流不再需要与 HCCL / compute 流做额外的顺序保证, 挂死条件被消除。
3. 测试同步对齐: test/registered/unit/platforms/test_platform_interface.py 的 test_pin_memory_available_for_npu_targets 把前三条断言由 assertTrue 改为

assertFalse, 并补充 3 行注释说明「Pinned memory stays disabled on NPU」, 覆盖 None、"npu"、torch.device("npu", 0)、"cpu"、torch.device("cpu") 五种入参。

- 4. 提交演进: commit 9933d13 完成主改动, 随后 commit cc1a62f (message 为 remove comments) 删除源码侧新增注释, 最终解释性文字只保留在测试文件与 PR body 中。
- 5. 配套范围: 没有配置项、文档或部署脚本改动, 不走环境变量开关, 属于直接改默认行为; PR body 建议后续「Re-enable only after the pinned-memory async-copy semantics are validated against CANN (optionally behind a debug env var first)」。

文件	角色	变更要点
python/sglang/srt/platforms/npu.py	源码 核心	NPUDeviceMixin.is_pin_memory_available 无条件返回 False, 回到 #36472 之前语义
test/registered/unit/platforms/test_platform_interface.py	测试 配套	断言全部改为 assertFalse, 加注释固化「NPU不启用pinnedmemory」的约定

关键文件:

- python/sglang/srt/platforms/npu.py (模块 NPU 平台; 类别 source; 类型 core-logic; 符号 NPUDeviceMixin.is_pin_memory_available, NPUSRTPlatform) : 唯一的源码改动点 : NPUDeviceMixin.is_pin_memory_available 从「CPU 返回 False、其余返回 True」改为无条件 False, 直接决定 NPU 上 next-token logits 拷贝与 xgrammar bitmask 分配是否使用 pinned memory + non_blocking H2D, 是 DP attention 挂死的开关。
- test/registered/unit/platforms/test_platform_interface.py (模块 平台单测; 类别 test; 类型 test-coverage; 符号 TestPlatformInterface.test_pin_memory_available_for_npu_targets) : 把「NPU 上 pinned memory 可用」的断言反向固定为不可用, 并加注释解释原因; 它是此次行为回退唯一的 in-tree 自动化守护。

关键符号: NPUDeviceMixin.is_pin_memory_available,
TestPlatformInterface.test_pin_memory_available_for_npu_targets

关键源码片段

python/sglang/srt/platforms/npu.py

唯一的源码改动点: NPUDeviceMixin.is_pin_memory_available 从「CPU 返回 False、其余返回 True」改为无条件 False, 直接决定 NPU 上 next-token logits 拷贝与 xgrammar bitmask 分配是否使用 pinned memory + non_blocking H2D, 是 DP attention 挂死的开关。

```
class NPUDeviceMixin:
    """NPU 设备能力 mixin: NPUSRTPlatform 通过它把 torch_npu 接入 SRTPlatform."""

    # 其余设备管理方法 (get_device / set_device / get_device_name /
    # empty_cache / synchronize / get_available_memory 等) 保持原样, 此处省略。

    def is_pin_memory_available(self, device=None) -> bool:
```

```

# 本 PR 的修复点：无条件返回 False，等价于回退 PR#36472 之前的行为。
# 旧实现为「device 是 "cpu" 时返回 False，否则（含 npu）返回 True」，
# 会让 next-token logits 拷贝与 xgrammar bitmask 分配走上
# pinned memory + non_blocking H2D。
# torch_npu 的 pinned + async copy 语义尚未针对 CANN 验证，
# non_blocking H2D 依赖的流序保证在 NPU 上不成立；DP attention 下
# compute / copy / HCCL 多流并发时，DeepSeek-V2 会间歇性挂死。
# device 参数被有意忽略：调用方传 "npu" 还是 "cpu" 都拿到 False，
# 避免不同入参下行为分叉而再次隐式开启 pinned memory。
# 重开条件：先在 CANN 上验证 pinned memory 的异步拷贝流序语义，
# 可考虑先用 debug 环境变量做灰度。
return False

```

```

class NPUSRTPlatform(NPUDeviceMixin, SRTPlatform):
    """默认的 in-tree NPU SRT 平台实现。"""

    def get_default_attention_backend(self) -> str:
        return "ascend"

    def get_dispatch_key_name(self) -> str:
        return "npu"

    def supports_fp8(self) -> bool:
        # NPU 量化后端位于 hardware_backend/npu/quantization
        return True

    def support_cuda_graph(self) -> bool:
        # 由 hardware_backend/npu/graph_runner 中的 NPUGraphRunner 提供
        return True

    def support_pieewise_cuda_graph(self) -> bool:
        return False

```

test/registered/unit/platforms/test_platform_interface.py

把「NPU 上 pinned memory 可用」的断言反向固定为不可用，并加注释解释原因；它是此次行为回退唯一的 in-tree 自动化守护。

```

def test_pin_memory_available_for_npu_targets(self):
    # Pinned memory 在 NPU 上保持关闭：torch_npu 的 pinned memory +
    # non_blocking H2D 路径未针对 CANN 验证（见
    # NPUSRTPlatform.is_pin_memory_available）。
    base = NPUSRTPlatform()
    # 断言由 assertTrue 收紧为 assertFalse：默认无参、显式 npu 设备、显式 cpu
    # 设备三种入参都必须为 False，防止 device 参数差异导致行为分叉，
    # 再次隐式打开 pinned memory 而复发 DP attention 挂死。
    self.assertFalse(base.is_pin_memory_available())
    self.assertFalse(base.is_pin_memory_available(device="npu"))
    self.assertFalse(base.is_pin_memory_available(device=torch.device("npu", 0)))

```

```
self.assertFalse(base.is_pin_memory_available(device="cpu"))
self.assertFalse(base.is_pin_memory_available(device=torch.device("cpu")))
```

评论区精华

本 PR 没有任何 review 评论 (review_comments_count=0)，讨论信息主要来自 PR body 与两条 issue 评论。PR body 承担了完整的技术论证：一是解释 #36472 的新类覆盖了基类安全默认值，二是给出两处独立复现作为证据，三是明确「先验证 CANN 流序语义再重开」的重开路径。评论区只有 CI 动作：sglang-npu-bot 发出 [/tag-and-rerun-ci](#)，作者 iridiumine 随后发出 [/rerun-failed-ci](#)；PR 状态区显示 Extra 流水线为失败态，但没有人对该失败项做技术判定。值得注意的是第二个 commit 只做了一件事——删除源码侧注释，最终代码里 `is_pin_memory_available` 只剩一行 `return False`，解释被转移到测试文件与 PR body。

- 是否应直接关闭 NPU 端 pinned memory (回退 #36472 的默认值) (design): 选择保守回退: `is_pin_memory_available` 无条件返回 `False`，理由是单步收益仅 KB 量级 decode buffer，不值得承担未验证的流序风险；重开需先验证 CANN 异步拷贝语义，可先用 debug env var 灰度。无 reviewer 提出反对意见。
- 单元测试断言与新的默认行为对齐 (testing): 测试与新行为一致，成为该回退唯一的 in-tree 自动化守护；但解释性文字只留在测试文件，源码侧已无注释。
- CI 状态与重跑流程 (other): 没有 reviewer 对失败项做技术判断，合入依据以人工压测 + 单测为主，最终由 `sglang-npu-bot` 合并。

风险与影响

- 风险:
 1. 性能回退 (NPU 专属) : `is_pin_memory_available=False` 会让 next-token logits 拷贝与 xgrammar bitmask 分配退回 pageable memory + 同步 H2D，这两条都在每步 decode 热路径上。PR body 声称收益只有 KB 量级，但 Checklist 中未勾选 speed benchmark，实际损失未量化。
 2. 缺少端到端回归测试：真正触发挂死的场景是 DP attention + 小 KV 预算 + radix retraction 压力 (`SGLANG_TEST_RETRACT=1` 对应的 `TestDPRetract`)，这一复现没有进入 in-tree registered CI；单测只锁住 `is_pin_memory_available` 返回 `False` 这一布尔值，若未来有人重开 pinned memory，挂死风险不会有自动化拦截。
 3. CI 证据偏弱：PR body 的 CI States 显示 Extra 流水线失败，review 记录为空，合并由 `sglang-npu-bot` 完成，说明合入依据主要是人工压测而非绿灯 CI。
 4. 知识留存风险：源码侧注释被 `remove comments commit` 删除，`python/sglang/srt/platforms/npu.py` 里看不到为何关闭；重开条件 (CANN 验证 + 流序保证) 只存在于 PR body，未落到代码注释、环境变量或跟踪 issue 中。
 5. 潜在调用方分支切换：任何在 NPU 上依赖 `is_pin_memory_available()==True` 而走 pinned 分支的调用方 (拷贝、buffer 分配) 现在都会走另一条路径，需要确认这些分支在 pageable 语义下同样正确。
 6. 回归面受控：改动仅限 `NPUDeviceMixin`，CUDA / ROCm / XPU 平台不经过该实现，跨平台回归风险低。- 影响：用户侧：Ascend 910C 等 NPU 部署在 DeepSeek-V2 +

DP attention（尤其叠加 retraction / 小 KV 预算 / 受限解码）场景下不再间歇挂死，可用性显著提升；代价是 NPU decode 路径失去 H2D 拷贝与计算的 overlap，理论上有小幅时延上升，但未见量化数据。系统侧：无 API、无配置、无 schema、无部署脚本变更，属于默认行为回退，部署方零改动即可受益。团队侧：补上了 #36472 in-tree NPU 平台收敛的一个缺口，并给出一个通用教训——新平台类最危险的改动是覆盖基类的安全默认值；后续若要为 NPU 重新启用 pinned memory，需要先做 CANN 异步拷贝流序验证、再以 debug env var 灰度，并配套端到端挂死回归测试。 - 风险标记：核心路径行为回退，缺少端到端挂死回归测试，NPU decode 热路径性能未量化，CI 证据偏弱（Extra失败），源码侧缺少重开条件注释

关联脉络

- PR #36472 引入 in-tree NPUSRTPlatform（本 PR 点名的挂死根因 PR）：PR body 明确说明 #36472 让 NPU 机器解析到新的 NPUSRTPlatform，并把基类默认的 `is_pin_memory_available=False` 覆盖为 `True`，是本 PR 要回退的回归来源；#36472 作者也在常规 DP attention 推理中观察到 DeepSeek-V2 挂死。该 PR 不在本次提供的历史列表中，标题按其在 body 中的角色概括。
- PR #39366 fix: stop shadowing the DSpark shared-experts fusion guard: 同一时间窗口的 NPU 正确性修复（标签含 `npu / deepseek / moe`），与本 PR 共同构成 in-tree NPU 平台近期的一批稳定性收尾工作，可对照看 NPU 侧 bugfix 的模式。
- PR #39389 [NPU] [DOC] Rename NPU hardware to Ascend A2/A3 Series product: 同样是 NPU 平台的收敛类改动（文档与产品名统一），与本 PR 一起反映 NPU 支持从外部混入走向 in-tree 平台化的演进脉络。