PR #24766 完整报告

sgl-project/sglang

[NUMA+Ray] Fix NUMA NVML handle resolution under shuffled CUDA_VISIBLE_DEVICES

合并时间：2026-05-10 12:18

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24766

执行摘要

- 一句话：修复 Ray 下 NUMA 绑定错选 GPU 的问题
- 推荐动作：值得合并，修复明确，风险可控。建议关注 PyTorch 版本更新对该内部 API 的影响，并及时更新 fallback 逻辑。

功能与动机

pynvml.nvmlDeviceGetHandleByIndex 按 PCI 总线顺序枚举 GPU，忽略 CUDA_VISIBLE_DEVICES。在 Ray 等分配器重排 CVD 后，numa_utils._query_numa_node_for_gpu 传递逻辑索引导致 NUMA 亲和性绑定到错误的物理 GPU，影响调度子进程性能。

实现拆解

1. 新增索引转换函数 _get_nvml_device_index（numa_utils.py）：通过 getattr(torch.cuda, "_get_nvml_device_index", None) 获取 PyTorch 内部辅助函数，该函数能正确解析 CUDA_VISIBLE_DEVICES 映射；若不可用则回退使用原始 device_id 并打印警告。
2. 修改 _query_numa_node_for_gpu 调用：将传入的 device_id（CUDA 逻辑索引）先通过 _get_nvml_device_index 转换为 NVML 物理索引，再传给 nvmlDeviceGetHandleByIndex。
3. 增强调试信息：在 configure_subprocess 的 debug_str 中附加 logical_gpu_id、physical_gpu_id 和当前 CUDA_VISIBLE_DEVICES 环境变量值，便于追踪绑定正确性。
4. 更新文档注释：将 _query_numa_node_for_gpu 参数 device_id 的注释从“GPU device index”改为“CUDA logical device index (post-CUDA_VISIBLE_DEVICES)”，明确语义。

关键文件：

- python/sglang/srt/utils/numa_utils.py（模块 NUMA 绑定；类别 source；类型 core-logic；符号 _get_nvml_device_index）：所有核心改动集中在此文件，包括新增索引转换函数、修改 NUMA 查询调用和增强调试信息。

关键符号：_get_nvml_device_index

关键源码片段

python/sglang/srt/utils/numa_utils.py

所有核心改动集中在此文件，包括新增索引转换函数、修改 NUMA 查询调用和增强调试信息。

```
# python/sglang/srt/utils/numa_utils.py importtorch
def_get_nvml_device_index(device_id: int) -> int:     # PyTorch 内部辅助函数，能正确解析
 CUDA_VISIBLE_DEVICES 映射     get_nvml_device_index = getattr(torch.cuda, "
_get_nvml_device_index", None)     if get_nvml_device_index is None:
logger.warning(             "torch.cuda._get_nvml_device_index is unavailable; falling
back to "             f"device_id={device_id}as the NVML device index. This may select "
         "the wrong physical GPU when CUDA_VISIBLE_DEVICES reorders devices "
     f"(CUDA_VISIBLE_DEVICES={os.environ.get('CUDA_VISIBLE_DEVICES','')})."          )
         return device_id # fallback 保持原有行为     return
get_nvml_device_index(device_id) # 返回物理 NVML 索引 # 修改
_query_numa_node_for_gpu 中的调用 def_query_numa_node_for_gpu(device_id: int): """
Get the NUMA node affinity list for a GPU device. Args: device_id: CUDA logical device
index (post-CUDA_VISIBLE_DEVICES). Returns: List of NUMA node IDs that have
affinity with the device. """     try:          pynvml.nvmlInit()          # 将 CUDA 逻辑索引转换
为 NVML 物理索引          nvml_device_id = _get_nvml_device_index(device_id)
handle = pynvml.nvmlDeviceGetHandleByIndex(nvml_device_id)          # ... 后续 NUMA 查
询逻辑不变
```

评论区精华

无 review 评论被记录，仅有一位批准者 ispobock 直接 approve，无公开讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 内部 API 依赖风险：torch.cuda._get_nvml_device_index 是 PyTorch 未公开的内部接口，未来版本可能变更或移除。但 PR 已通过 fallback 机制（回退并告警）缓解此风险。
  2. 回归风险低：仅修改 numa_utils.py 一个文件，逻辑独立，且 fallback 保持原有行为。主要影响 Ray + NUMA 绑定场景。 - 影响：影响范围较小，但修复了 Ray 环境下 NUMA 绑定错误导致性能下降的严重 bug。影响用户为使用 Ray 分配 GPU 且启用 NUMA 绑定的部署，其余场景不受影响。调试信息的增强对运维排查有帮助。 - 风险标记：依赖内部 API，仅单文件改动

关联脉络

- 暂无明显关联 PR