PR #30602 完整报告

sgl-project/sglang

[Fix] Prevent silent VLM server crash when /dev/shm is exhausted during multimodal feature transport

合并时间：2026-07-09 15:44

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/30602

执行摘要

- 一句话：防止 VLM 因 /dev/shm 耗尽而静默崩溃
- 推荐动作：建议精读本 PR，特别是 posix_fallocate 的使用模式和 fallback 函数的设计，对于处理共享内存资源稀缺的场景具有参考价值。

功能与动机

在多模态请求突发场景下，ShmPointerMMData 在拷贝中途可能耗尽 /dev/shm（tmpfs 页在 ftruncate 后延迟分配），导致进程被 SIGBUS 杀死。该 PR 旨在将静默崩溃转变为可优雅降级的错误，并修复哈希函数对混合列表的支持。

实现拆解

1. 导入新增：在 mm_utils.py 中添加 import os 和 import sys，用于平台判断和 posix_fallocate 调用。
2. ShmPointerMMData.__init__改造：在创建共享内存后、拷贝数据前，针对 Linux 平台调用 os.posix_fallocate(shm._fd, 0, nbytes) 预分配 tmpfs 页，使得 /dev/shm 不足时抛出 OSError（ENOSPC），而不是在拷贝时触发 SIGBUS。
3. 新增 fallback 函数 _wrap_shm_or_inline：尝试构造 ShmPointerMMData，如果捕获到 OSError 则打印警告并直接返回原始 tensor，调度器后续会将其 pickle 传输。
4. _wrap_tensor_or_list 迁移：将内部的 ShmPointerMMData 构造调用替换为 _wrap_shm_or_inline，确保单一 tensor 或列表元素均能触发 fallback。
5. hash_feature 修复：将列表类型检查从 isinstance(f[0], ShmPointerMMData) 改为 any(isinstance(x, ShmPointerMMData) for x in f)，以支持混合了 ShmPointerMMData 和普通 tensor 的列表。

关键文件：

- python/sglang/srt/managers/mm_utils.py（模块 多模态工具；类别 source；类型 dependency-wiring；符号 _wrap_shm_or_inline, ShmPointerMMData.init, hash_feature, _wrap_tensor_or_list）：核心变更文件：新增 import、改造 ShmPointerMMData、引入 fallback 函数、修复 hash_feature。

关键符号：_wrap_shm_or_inline, ShmPointerMMData.init, hash_feature, _wrap_tensor_or_list

关键源码片段

python/sglang/srt/managers/mm_utils.py

核心变更文件：新增 import、改造 ShmPointerMMData、引入 fallback 函数、修复 hash_feature。

```
# 关键片段：ShmPointerMMData.__init__ 中预分配 tmpfs 页
class ShmPointerMMData:
    def __init__(self, tensor: torch.Tensor, precomputed_hash: Optional[int] = None):
        if not tensor.is_cpu:
            tensor = tensor.cpu()
        if not tensor.is_contiguous():
            tensor = tensor.contiguous()
        self.shape = tensor.shape
        self.dtype = tensor.dtype
        self.precomputed_hash = precomputed_hash
        nbytes = tensor.numel() * tensor.element_size()
        shm = shared_memory.SharedMemory(
            create=True, size=nbytes, name=make_shm_name("mm")
        )
        try:
            if sys.platform == "linux":
                # 预分配 tmpfs 页，将 SIGBUS 转化为可捕获的 OSError
                os.posix_fallocate(shm._fd, 0, nbytes)
            dst = torch.frombuffer(shm.buf, dtype=torch.uint8)
            dst.copy_(tensor.view(torch.uint8).reshape(-1))
        except BaseException:
            shm.close()
            shm.unlink()
            raise
        # ...

# 新增 fallback 函数：尝试创建 ShmPointerMMData，失败则返回原始 tensor
# 调用方（如 _wrap_tensor_or_list）会将其视为内联传输

def _wrap_shm_or_inline(tensor: torch.Tensor, precomputed_hash: Optional[int] = None):
    try:
        return ShmPointerMMData(tensor, precomputed_hash=precomputed_hash)
    except OSError as e:
        print_warning_once(
            f"Failed to allocate shared memory for multimodal feature transport "
            f"({e}); falling back to inline transport. "
            f"Consider increasing /dev/shm size."
        )
        return tensor

# hash_feature 修复：支持混合列表
def hash_feature(f):
    if isinstance(f, list):
```

```
            # 当列表混合 ShmPointerMMData 和普通 tensor 时，统一处理
            if len(f) > 0 and any(isinstance(x, ShmPointerMMData) for x in f):
                return tensor_hash(
                    [x.tensor if isinstance(x, ShmPointerMMData) else x for x in f]
                )
            if len(f) > 0 and isinstance(f[0], torch.Tensor):
                return tensor_hash(f)
            return data_hash(tuple(flatten_nested_list(f)))
        # ... 其余分支不变
```

评论区精华

该 PR 的 review 评论较少，主要讨论围绕 fallback 策略：作者选择在构造 ShmPointerMMData 时立即 fallback，而不是在 _wrap_tensor_or_list 中统一处理，这样每个元素独立降级，粒度更细。此外，posix_fallocate 仅在 Linux 上调用，避免了其他平台上的兼容性问题。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. posix_fallocate 依赖文件系统支持：某些非 Linux 平台或不支持 fallocate 的文件系统可能引发 AttributeError 或 OSError，但代码已通过 sys.platform == "linux" 保护。
  2. 性能影响：posix_fallocate 会增加一次系统调用，但在共享内存分配后、数据拷贝前执行，对整体延迟影响很小。
  3. fallback 路径正确性：内联传输通过 pickle 序列化 tensor，可能导致更大的内存拷贝开销，但这是降级路径，不应在正常流程中触发。 - 影响：直接效果：在 /dev/shm 不足时，多模态 VLM 服务器不再静默崩溃，而是打印警告并降级为内联传输，保证服务可用性。影响范围：所有使用 ShmPointerMMData 的多模态请求流程（涉及多个进程间 tensor 传输）。影响程度：中等，属于稳定性增强，不改变正常路径的行为。
- 风险标记：核心路径变更，依赖于 Linux-specific syscall

关联脉络

- PR #28534 [AMD] Enable JIT staged HiCache write-back and fix CPU-index crash: 同样涉及共享内存管理（HiCache）和 JIT 写回策略，共享内存的稳定性改进与此 PR 有协同效应。
- PR #30461 [DSV4] Fix draft SWA transfer for disaggregated MTP: 涉及进程间数据传输的修复，与本 PR 的多模态共享内存传输有类似的稳定性关注。