

PR #24378 完整报告

sgl-project/sglang

fix(disagg): broadcast bootstrap port across multi-node prefill ranks

合并时间: 2026-05-14 16:39

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/24378>

执行摘要

- 一句话: 同步多节点 prefill bootstrap 端口, 修复跨节点注册失败
- 推荐动作: 此 PR 修复了多节点 disagg 部署中的静默故障, 设计简洁高效。建议阅读 `_sync_bootstrap_port_across_nodes` 方法, 了解如何使用 `torch.distributed.broadcast_object` 实现跨节点配置同步, 这种 Leader 广播模式在分布式系统中很有参考价值。

功能与动机

在 multi-node disaggregated prefill 中, 每个 prefill scheduler 进程解析自己的 `server_args.disaggregation_bootstrap_port`。某些启动器 (如 Dynamo) 会在启动时自动保留空闲端口, 导致各主机端口不同。`register_to_bootstrap` 虽然主机地址正确解析自 `dist_init_addr` (world-rank 0), 但端口使用本地端口, 非 leader 节点因此向 leader 的本地端口发送请求, 而 leader 在该端口没有监听, 导致 `Connection refused` 和后续的 `KeyError` 及 `AttributeError`。当前 workaround 是手动指定固定端口, 但默认路径静默失败。

实现拆解

实现分为以下几个步骤:

1. 在 `CommonKVManager.__init__` 的 prefill 分支中, `register_to_bootstrap()` 调用之前新增 `_sync_bootstrap_port_across_nodes` 调用, 将本地 `bootstrap_port` 替换为同步后的端口。
2. 新增 `_sync_bootstrap_port_across_nodes(local_port)` 方法, 首先判断是否为多节点 prefill (`dist_init_addr` 非空且 `nnodes > 1`), 否则直接返回 `local_port` (单节点恒等)。
3. 若为多节点, 检查 `torch.distributed` 是否已初始化, 否则抛出 `RuntimeError`, 避免静默退化。
4. 通过 `get_world_group().broadcast_object(local_port, src=0)` 将端口广播到所有节点, 并返回同步后的端口。
5. 根据 review 建议将 `import torch.distributed` 从方法内提升到文件顶部, 并增加 `nnodes == 1` 的单节点短路条件。所有改动仅涉及一个文件 `python/sglang/srt/disaggregation/common/conn.py`, 未引入新的测试文件。

关键文件:

- `python/sglang/srt/disaggregation/common/conn.py` (模块 分离式预填充; 类别 `source`; 类型 `dependency-wiring`; 符号 `_sync_bootstrap_port_across_nodes`, `CommonKVManager.init`) : 该文件是分离式预填充的核心连接管理模块, 实现了 `bootstrap` 端口跨节点同步逻辑, 是多节点修复的单一变更点。

关键符号: `CommonKVManager._sync_bootstrap_port_across_nodes`,
`CommonKVManager.init`

关键源码片段

`python/sglang/srt/disaggregation/common/conn.py`

该文件是分离式预填充的核心连接管理模块, 实现了 `bootstrap` 端口跨节点同步逻辑, 是多节点修复的单一变更点。

以下展示 `_sync_bootstrap_port_across_nodes` 方法的完整实现, 以及 `__init__` 中调用它的上下文。代码注释遵循盘古排版规则。

```
# 在 CommonKVManager.__init__ 的 prefill 分支中 (位于 register_to_bootstrap 调用前)
```

```
if self.disaggregation_mode == DisaggregationMode.PREFILL:
    self.is_dummy_cp_rank = (
        not self.enable_all_cp_ranks_for_transfer
        and self.attn_cp_size > 1
        and self.attn_cp_rank != 0
    )
    # 同步 bootstrap 端口: 在多节点 prefill 中, 各节点可能因启动器动态
    # 分配而拥有不同的本地端口, 需要统一为 leader (rank 0) 的端口。
    self.bootstrap_port = self._sync_bootstrap_port_across_nodes(
        self.bootstrap_port
    )
    self.register_to_bootstrap()
```

```
# 新增的端口同步方法
```

```
def _sync_bootstrap_port_across_nodes(self, local_port: int) -> int:
    # Broadcast world-rank-0's bootstrap port to all prefill ranks.
    # Required for multi-node prefill when the launcher auto-reserves a
    # free port per host (e.g. Dynamo's
    # `_reserve_disaggregation_bootstrap_port`): without sync, non-leader
    # ranks register to `
```

```

        'multi-node prefill mode.'
    )

world_group = get_world_group()
# 广播: src=0 的端口被发送到所有 rank, 返回值为广播源的值。
synced_port = world_group.broadcast_object(local_port, src=0)
if synced_port != local_port:
    logger.info(
        f'Synced disaggregation bootstrap port from leader: '
        f'local={local_port} -> leader={synced_port} '
        f'(world_rank={world_group.rank_in_group})'
    )
return synced_port

```

评论区精华

- 导入风格建议 (gemini-code-assist[bot], low priority) : “The manual import of torch.distributed inside a method is generally discouraged. It is better to import it at the top of the file to ensure all dependencies are resolved at module load time and to improve readability.” → 作者提交独立 commit 将 import 提升到文件顶部, 已解决。
- 单节点边界条件 (ShangmingCai) : “Should we check or self.server_args.nnodes == 1 as well? I am seeing people still set up --nnodes 1 when launching a single-node server.” → 作者回复 “Makes sense, revised!”, 并在代码中增加了该条件。
- 整体认可: 合并者 ShangmingCai 最终给出 LGTM 并合并 PR, 无其他未解决的讨论。
 - torch.distributed 导入风格 (style): 作者接受并提交了 hoist commit, 将 import 移至模块顶部。
 - 单节点条件判断增强 (correctness): 作者接受并修改, 在同步方法中加入 nnodes == 1 的短路条件。

风险与影响

- 风险:
 - 依赖分布式通信: 新方法依赖 torch.distributed.broadcast_object, 如果分布式环境未初始化或通信失败, 会在启动阶段直接抛 RuntimeError, 有助于提前暴露问题, 但要求前面的 Scheduler.init_model_worker() 已经完成初始化。
 - 单节点无影响: 单节点时条件短路, 不进行广播, 不影响现有单机部署。
 - 性能风险极低: 仅启动时广播一个整数, benchmark 验证无性能退化。
 - 缺少单元测试: PR 未包含直接针对该方法的单元测试, 回归风险依赖集成测试覆盖。
- 影响:
 - 用户影响: 多节点分离式预填充部署不再需要显式指定 --disaggregation-bootstrap-port, 默认路径即可正确运行, 简化了配置。单节点用户无影响。
 - 系统影响: 运行时行为无变化, 仅增加一次初始化阶段的广播开销。

- 团队影响: 减少了因端口不一致导致的部署故障和排查成本, 使 sglang 更好的适配 Dynamo 等动态端口分配启动器。
- 风险标记: 核心路径变更, 缺少测试覆盖, 依赖分布式通信, 单节点无影响

关联脉络

- PR #25064 [Bug Fix] Add priority property to DecodeRequest to fix AttributeError with --enable-priority-scheduling: 同为分离式预填充模块的 bugfix, 修复了解码端属性缺失导致的 AttributeError, 与本 PR 的稳定性改进属于同一功能线。