

# PR #2000 完整报告

radixark/miles

Use SGLANG\_DG\_CACHE\_DIR\_PER\_PROCESS instead of hand-built cache dirs

合并时间: 2026-09-04 13:46

原文链接: <http://prhub.com.cn/radixark/miles/pull/2000>

## 执行摘要

- 一句话: 用 SGLang 原生的进程级 DeepGEMM 缓存隔离替代自建的 hacky 路径。
- 推荐动作: 该 PR 值得精读。它是一个典型的、高质量的“代码卫生”改进, 展示了如何利用上游工具的正确抽象来消除技术债。其修改清晰、范围小、有测试覆盖, 且是 #1837 大型重构跟踪 issue 的一部分。关注点在于理解为何手动路径拼接是“hacky”的, 以及 SGLang 提供的 PER\_PROCESS 隔离机制为何是更优解。这对于维护类似的分布式系统启动配置具有借鉴意义。

## 功能与动机

在 miles/ray/rollout/server\_cell.py 的 launch\_sglang\_ray\_actor 函数中, 存在一个 TODO 注释指出, 为 DeepEP/NVSHMEM 缓存设置的 SGLANG\_DG\_CACHE\_DIR 环境变量是一种“hacky”的做法。该 PR 的动机是采用 SGLang 上游推荐的方式 (SGLANG\_DG\_CACHE\_DIR\_PER\_PROCESS=1) 来实现相同的进程隔离, 从而消除这个技术债, 使代码更简洁、更健壮, 并减少因 worker\_type 和 global\_rank 变量变化导致路径错误的风险。

## 实现拆解

变更从一个被修改的核心源文件和配套测试文件开始, 集中于引擎启动函数的签名简化和环境变量配置的标准化。

1. 入口点与参数简化: 在 miles/ray/rollout/server\_cell.py 的 ServerCell.start\_engines 方法中, 调用 launch\_sglang\_ray\_actor 时移除了 global\_rank 和 worker\_type 两个参数, 仅保留 args、pg 和 gpu\_index。这直接对应于该函数签名的重构。
2. 核心配置逻辑替换: 在同一文件的 launch\_sglang\_ray\_actor 函数体中, 将构建 SGLANG\_DG\_CACHE\_DIR 环境变量的代码 (依赖 worker\_type 和 global\_rank) 替换为设置 SGLANG\_DG\_CACHE\_DIR\_PER\_PROCESS 为固定值“1”。这利用了 SGLang 的内置机制, 确保每个进程使用独立的缓存目录, 无需外部拼接路径。
3. 测试适配: tests/fast/ray/rollout/test\_addr\_allocator.py 中的两个测试辅助 mock 函数 \_launch 和 lambda 被更新, 以匹配 launch\_sglang\_ray\_actor 新的 gpu\_index 关键字参数, 确保测试能正确模拟 actor 启动。

关键文件:

- `miles/ray/rollout/server_cell.py` (模块 启动调度; 类别 `source`; 类型 `core-logic`; 符号 `launch_sglang_ray_actor`) : 核心源文件, 包含被重构的启动函数 `launch_sglang_ray_actor` 及其调用方 `ServerCell.start_engines`。这是本次缓存隔离配置变更和参数简化的主要发生地。
- `tests/fast/ray/rollout/test_addr_allocator.py` (模块 地址分配测试; 类别 `test`; 类型 `test-coverage`; 符号 `_launch`) : 测试文件, 包含用于验证 `ServerCell.start_engines` 行为的 `mock` 辅助函数。需要同步更新以匹配新的 `launch_sglang_ray_actor` 参数接口, 确保测试有效性。

关键符号: `launch_sglang_ray_actor`

## 关键源码片段

### `miles/ray/rollout/server_cell.py`

核心源文件, 包含被重构的启动函数 `launch_sglang_ray_actor` 及其调用方 `ServerCell.start_engines`。这是本次缓存隔离配置变更和参数简化的主要发生地。

```
# miles/ray/rollout/server_cell.py
# 重构后的 launch_sglang_ray_actor 函数签名及关键环境变量配置

def launch_sglang_ray_actor(
    *,
    args: Any,
    pg: Any,
    gpu_index: int,
) -> ray.actor.ActorHandle:
    pg, reordered_bundle_indices, _ = pg

    # ... (scheduling_strategy 构建逻辑省略)

    env_vars = {name: "1" for name in NOSET_VISIBLE_DEVICES_ENV_VARS_LIST} | {
        key: os.environ.get(key, default_val)
        for key, default_val in {
            # DeepEP/NVSHMEM's internal NCCL conflicts with our NCCL and hangs under CUDA
            # graphs.
            "NVSHMEM_DISABLE_NCCL": "1",
            "SGLANG_JIT_DEEPGEMM_PRECOMPILE": "false",
            # 替换原有的 hacky 路径拼接, 使用 SGLang 官方支持的每进程缓存隔离
            "SGLANG_DG_CACHE_DIR_PER_PROCESS": "1",
            # ... (其他环境变量省略)
        }.items()
    }
    env_vars.update(dumper_utils.get_sglang_env(args))

    RolloutRayActor = ray.remote(CommandActor)
    return RolloutRayActor.options(
        num_cpus=num_cpus,
        num_gpus=num_gpus,
```

```
scheduling_strategy=scheduling_strategy,  
runtime_env={  
    "env_vars": env_vars,  
},  
)..remote()
```

## 评论区精华

该 PR 的审查较为直接。评审者 guapisolo 在“stack context”中确认了变更的正确性，并指出“SGLang-owned per-process DeepGEMM cache isolation is correctly wired; no actionable findings”。这意味着对改动的核心逻辑——即 SGLang 缓存隔离机制的接入方式——没有异议，变更被视为一次干净的、符合预期的改进。

- SGLang DeepGEMM 缓存隔离机制的正确性确认 (correctness): 变更被批准，核心逻辑被认为正确无误。

## 风险与影响

- 风险:

1. 环境变量变更风险: 将 SGLANG\_DG\_CACHE\_DIR 改为 SGLANG\_DG\_CACHE\_DIR\_PER\_PROCESS 属于 SGLang 接口的使用方式变更。如果 SGLang 版本不支持此变量，或其行为与旧有手动拼接路径不完全一致，可能引发运行时问题。但鉴于这是上游推荐方式，且 PR 描述明确其为正确解法，风险较低。
2. 函数签名变更风险: launch\_sglang\_ray\_actor 函数的参数列表发生了非向后兼容的变更。当前代码库中的所有调用方（主要是 ServerCell.start\_engines）已同步更新。需要确认没有其他未被测试覆盖到的调用点。从提供的文件和历史 PR 看，此函数主要由 ServerCell 内部使用，影响范围可控。

- 影响:

1. 用户影响: 无直接用户可见功能变化。此变更提升了底层推理引擎启动过程的健壮性和可维护性。
2. 系统影响: 改善了 SGLang 引擎进程间缓存隔离的实现方式，消除了一个已知的技术债。这使得未来的重构（如进一步简化 launch\_sglang\_ray\_actor）更容易。
3. 团队影响: 简化了核心启动函数的签名和逻辑，降低了新成员理解这段代码的认知负荷。测试的同步更新也为后续类似的参数调整提供了良好范例。 - 风险标记: 依赖上游工具接口，函数签名非向后兼容变更

## 关联脉络

- PR #1837 Tracking issue for refactoring and enhancements: 本 PR 在 body 中明确标注为 "Part of #1837"，是该大型重构跟踪 issue 的一个子项。
- PR #2002 Stop passing a random seed and let each sglang engine pick its own: 同属于一系列简化 SGLang 引擎启动参数和配置的重构 PR，旨在减少 Miles 代码对 SGLang 内部行为的不必要干预和维护。

- PR #2004 Extract `compute_inference_engine_env_vars` and forbid `compute_server_args` defaults: 更早期的 PR，同样致力于清理 SGLang 引擎启动时的环境变量和参数构建逻辑，与本 PR 的重构方向一致。