

PR #1870 完整报告

radixark/miles

Switch cell start/stop/recover from engine indices to cell indices

合并时间: 2026-09-04 13:05

原文链接: <http://prhub.com.cn/radixark/miles/pull/1870>

执行摘要

- 一句话: 将 cell 管理接口从 engine 索引切换到 cell 索引, 简化启动 / 停止 / 恢复操作的接口。
- 推荐动作: 此 PR 值得精读。它展示了如何通过接口简化 (从引擎索引到单元格索引) 来提升代码的内聚性和可理解性, 是模块化重构的良好实践。核心设计决策 (`_engine_indices_of_cell` 的引入、 `CellIndexer` 的简化、 `recover` 断言的移除) 值得借鉴。作为 #1837 重构链的一环, 它清晰地展示了该功能线的演进方向。

功能与动机

此 PR 是 #1837 跟踪 issue 的一部分。目标是简化 `ServerGroup` 的外部操作接口, 将 cell 视为不可分割的操作单元, 而不是暴露更细粒度的引擎索引。这样做可以简化调用方 (如 `InferenceController`) 的逻辑, 避免调用方需要理解 cell 和 engine 之间的映射关系 (例如 `nodes_per_engine`)。commit 消息明确提到: `start_engines, stop_engines and recover now take cell indices; CellIndexer carries cell_index instead of the flat engine_indices, and InferenceController.start_cell/stop_cell pass the single cell through.` 这也解决了 #1867 中提及的 `unregister-before-validation` 间隙问题, 通过使用原子性的 cell 索引进行操作。

实现拆解

变更按以下步骤展开:

1. 重构 `ServerGroup.start_engines` 的入口参数: 将方法签名中的 `start_indices: list[int] | None` 参数重命名为 `start_cell_indices: list[int] | None`。方法内部增加转换逻辑, 使用新增的 `_engine_indices_of_cell` 辅助方法将传入的 cell 索引列表转换为引擎索引列表 `start_indices`, 再用于后续的引擎分配逻辑。
2. 重构 `ServerGroup.stop_engines` 和 `recover` 方法: 将 `stop_engines` 的参数从 `engine_indices` 重命名为 `cell_indices`。方法内部现在根据 `cell_indices` 直接计算出所有相关的 `engine_indices` 用于从路由器注销, 然后直接操作对应的 `ServerCell` 进行停止, 移除了之前验证必须停止整个 cell 的复杂断言。将 `recover` 的 `filter_indices` 参数重命名为 `filter_cell_indices`, 内部逻辑改为按 cell 粒度筛选未分配的 cell (使用 `any(not engine.is_allocated for engine in cell.engines)`), 并传递给 `start_engines`。同时移除了 `recover` 方法中 `assert len(new_engine_indices) == len(start_indices)` 的断言, 因为新接口允许部分 cell 的恢复, 其 engine 数量不一定与请求完全匹配。

3. 引入 `_engine_indices_of_cell` 辅助方法：在 `ServerGroup` 中新增一个私有方法 `_engine_indices_of_cell(self, cell_index: int) -> range`，用于根据 `cell` 索引和 `nodes_per_engine` 计算该 `cell` 包含的引擎索引范围。这封装了之前分散在 `stop_engines` 和其他地方的转换逻辑。
4. 简化 `CellIndexer` 和 `InferenceController`：将 `CellIndexer` 的 `engine_indices` 字段替换为 `cell_index: int`，因为外部调用者（`InferenceController`）现在只需要知道 `cell` 的索引。相应地，`InferenceController.start_cell` 和 `stop_cell` 方法在调用 `group.recover` 和 `group.stop_engines` 时，现在传递 `[idx.cell_index]` 而不是 `idx.engine_indices`。
5. 更新所有相关测试：同步更新了 `test_server_group.py`, `test_fault_tolerance.py`, `test_server_group_router_registration.py`, `test_server_cell.py`, `test_server_group_tearardown.py` 中对重命名后方法和参数的调用，确保测试覆盖新的接口。

关键文件：

- `miles/ray/rollout/server_group.py`（模块 Rollout 服务组；类别 source；类型 core-logic；符号 `_engine_indices_of_cell`, `stop_engines`, `recover`）：核心逻辑变更文件。实现了从 `engine` 索引到 `cell` 索引的接口切换，新增了关键转换方法，并重构了 `stop_engines` 和 `recover`。
- `miles/ray/rollout/server_cell.py`（模块 单元格索引器；类别 source；类型 data-contract；符号 `CellIndexer`, `get_cell_indexer_of_id_map`）：简化了 `CellIndexer` 数据结构，从携带 `engine_indices` 列表改为仅携带 `cell_index` 整数，与新的接口设计对齐。
- `miles/ray/rollout/inference_controller.py`（模块 推理控制器；类别 source；类型 entrypoint；符号 `start_cell`, `stop_cell`）：作为调用方，其 `start_cell` 和 `stop_cell` 方法需要适配新的、更简单的 `cell` 索引接口。

关键符号： `ServerGroup.start_engines`, `ServerGroup.stop_engines`, `ServerGroup.recover`, `ServerGroup._engine_indices_of_cell`, `InferenceController.start_cell`, `InferenceController.stop_cell`, `get_cell_indexer_of_id_map`

关键源码片段

`miles/ray/rollout/server_group.py`

核心逻辑变更文件。实现了从 `engine` 索引到 `cell` 索引的接口切换，新增了关键转换方法，并重构了 `stop_engines` 和 `recover`。

`ServerGroup` 关键变更片段

此片段展示了接口变更的核心：新增了将 `cell` 索引转换为引擎索引的辅助方法，并重构了 `stop_engines` 以使用新的 `cell_indices` 参数。

```
# 新增方法：将 cell 索引转换为该 cell 包含的所有引擎索引范围
def _engine_indices_of_cell(self, cell_index: int) -> range:
    # 每个 cell 可能对应多个引擎 (nodes_per_engine > 1)
    return range(cell_index * self.nodes_per_engine, (cell_index + 1) * self.nodes_per_engine)

# ... (其他代码保持不变)

# 重构后的方法：参数从 engine_indices 改为 cell_indices
```

```

def stop_engines(self, cell_indices: list[int]):
    logger.info(f"Killing server {cell_indices}")
    # 内部使用 _engine_indices_of_cell 将 cell 索引转换为 engine 索引，用于从路由器注销
    engine_indices = [i for cell_index in cell_indices for i in self._engine_indices_of_cell(cell_index)]
    try:
        async_utils.run(asyncio.wait_for(self.unregister_workers(engine_indices), timeout=SHUTDOWN_TIMEOUT))
    except Exception as e:
        logger.warning(f"Unregistering {engine_indices=} from the router failed, tearing down anyway (e: {e})")
    # 直接操作对应的 cell 进行停止
    for cell_index in sorted(set(cell_indices)):
        self.cells[cell_index].stop()

```

recover 方法的关键变更

recover 方法的过滤参数也从引擎索引切换到了 cell 索引，并移除了原有断言。

```

async def recover(self, port_cursors: PortCursors, filter_cell_indices: list[int] | None = None):
    # 默认过滤：选择任何引擎未分配的 cell
    if filter_cell_indices is None:
        filter_cell_indices = [
            cell_index
            for cell_index, cell in enumerate(self.cells)
            if any(not engine.is_allocated for engine in cell.engines)
        ]
    # 调用 start_engines 时传递 cell 索引
    handles, new_engine_indices = self.start_engines(port_cursors, start_cell_indices=filter_cell_indices)
    # ... ( 后续逻辑不变，但移除了 `assert len(new_engine_indices) == len(start_indices)` )

```

miles/ray/rollout/server_cell.py

简化了 `CellIndexer` 数据结构，从携带 `engine_indices` 列表改为仅携带 `cell_index` 整数，与新的接口设计对齐。

CellIndexer 和 get_cell_indexer_of_id_map 变更

`CellIndexer` 现在只记录 cell 在其所在 group 中的位置 (`cell_index`)，而不是该 cell 包含的具体引擎索引列表。这使得 `get_cell_indexer_of_id_map` 的实现更简单。

```

class CellIndexer(NamedTuple):
    srv_key: str
    group_index: int
    cell_index: int # 替换了原来的 engine_indices: list[int]

```

```

def get_cell_indexer_of_id_map(servers: dict[str, "RolloutServer"]) -> list[CellIndexer]:
    """Flatten ``servers`` into a list whose position is the cell id.

```

``cell\_index`` is the cell's position within its group. Order is sorted by

```

``srv_key``, so cell ids are stable across calls when the topology is
unchanged.
"""
result: list[CellIndexer] = []
for srv_key in sorted(servers):
    srv = servers[srv_key]
    for group_index, group in enumerate(srv.server_groups):
        # 直接遍历 cell 索引范围
        for cell_index in range(len(group.cells)):
            result.append(
                CellIndexer(
                    srv_key=srv_key,
                    group_index=group_index,
                    cell_index=cell_index,
                )
            )
return result

```

miles/ray/rollout/inference_controller.py

作为调用方，其 `start_cell` 和 `stop_cell` 方法需要适配新的、更简单的 cell 索引接口。

InferenceController 适配新接口

调用方现在只需传递 cell 索引，逻辑更清晰。

```

async def start_cell(self, cell_id: int):
    port_cursors = PortCursors.empty()
    idx = get_cell_indexer_of_id_map(self.servers)[cell_id]
    group = self.servers[idx.srv_key].server_groups[idx.group_index]
    # 之前是 filter_indices=idx.engine_indices, 现在是 filter_cell_indices=[idx.cell_index]
    await group.recover(port_cursors=port_cursors, filter_cell_indices=[idx.cell_index])

async def stop_cell(self, cell_id: int):
    idx = get_cell_indexer_of_id_map(self.servers)[cell_id]
    group = self.servers[idx.srv_key].server_groups[idx.group_index]
    # 之前是 engine_indices=idx.engine_indices, 现在是 cell_indices=[idx.cell_index]
    group.stop_engines(cell_indices=[idx.cell_index])

```

评论区精华

本次 PR 审核中只有一条来自 `guapisolo` 的审评论论：“Approved for the #2176 safe-merge checkpoint: this replaces partial-engine lifecycle requests with cell-indexed start/stop/recover, closing the unregister-before-validation gap from #1867; the fix remains present through the checked chain tip.” 该评论确认了变更的动机（替换部分引擎生命周期请求为 cell 索引）、解决了之前的问题（关闭 #1867 中的注销验证间隙），并表示此修复在检查链中保持有效。未发现其他讨论或争议。

- 接口重构设计与风险评估 (other): 变更被批准，认为它关闭了 #1867 中的注销验证间隙，并将部分引擎生命周期请求替换为基于 cell 索引的操作。

风险与影响

- 风险：技术风险较低。主要风险在于接口变更（参数重命名）可能导致未更新的外部调用方出现错误，但本次提交已全面更新了代码库内所有调用点（`inference_controller.py` 和多个测试文件）。移除 `recover` 中的断言 `assert len(new_engine_indices) == len(start_indices)` 是经过考虑的，因为新的基于 `cell` 的过滤逻辑允许恢复一个部分不健康的 `cell`，其启动的 `engine` 数量可能少于 `cell` 理论 `engine` 数。此变更不影响外部 API 的语义，但需确保内部实现正确处理多节点引擎场景（一个 `cell` 对应多个 `engine`）。
- 影响：影响范围：变更主要影响 `miles` 仓库内部的 `ServerGroup` 和 `InferenceController` 模块，以及与之相关的测试代码。对外部用户或 API 无直接影响。影响程度：提升了代码的可维护性和接口的一致性。新的 `cell` 索引接口更直观，降低了调用方的复杂度。变更逻辑集中，且有配套测试，回归风险小。
- 风险标记：接口参数重命名，移除原有断言

关联脉络

- PR #1837 Tracking issue for refactoring and enhancements: 本 PR 是 #1837 跟踪 issue 中列出的重构工作的一部分。
- PR #2050 Stop checking actor liveness while waiting for a tcp port: 同属 `miles/ray/rollout` 和 `miles/utils` 模块的近期重构系列，共同推进 Rollout 模块的规范化。
- PR #2049 Add a worker provider abstraction over `RayWorkerManager`: 同属 `miles/utils/workers` 和 `miles/ray/rollout` 的重构工作，都在优化工作器管理和生命周期。
- PR #2006 Launch inference engines through the inference engine worker spec: 与本 PR 同属 Rollout 模块启动逻辑的重构，都旨在通过更清晰的规范和接口来统一引擎管理。