PR #2016 完整报告

THUDM/slime

Fully support --rollout-external-engine-addrs

合并时间：2026-06-04 22:05

原文链接：http://prhub.com.cn/THUDM/slime/pull/2016

执行摘要

- 一句话：全量支持 --rollout-external-engine-addrs 外部引擎
- 推荐动作：建议精读 slime/backends/sglang_utils/external.py 的设计，特别是如何通过 HTTP 自发现外部引擎属性并动态注册到路由器。slime/ray/placement_group.py 中外部引擎模式的 GPU 布局调整值得关注，这是将外部资源融入 Ray 集群的典型案例。slime/backends/sglang_utils/sglang_engine.py 中的路由器注册重构也值得参考。

功能与动机

之前只部分支持 --rollout-external，但外部引擎的发现和注册逻辑分散且不完整。为支持 delta 权重更新等非 NCCL 同步场景，需要让 slime 能灵活对接用户自己启动的 SGLang 引擎池（例如 PD 分离集群）。PR 通过统一的 discovery+apply 模式完整实现了外部引擎接入。

实现拆解

步骤 1：新建 external.py 模块

在slime/backends/sglang_utils/external.py中定义了ExternalEngineInfo数据类，包含url、host、port、worker_type、num_gpus、disaggregation_bootstrap_port 和 server_info 字段。实现了 normalize_external_engine_addr 地址标准化（host:port → http://host:port）。get_server_info通过HTTPGET请求/server_info或/get_server_info端点获取引擎元信息。_infer_worker_type 根据 server_info 的 disaggregation_mode 判断 worker 类型（prefill、decode、regular）。discover_external_engines 遍历地址列表，调用上述函数组装 ExternalEngineInfo 列表，并计算每个引擎的 num_gpus（优先取 server_info 中的 num_gpus，否则取 tp_size * pp_size）。apply_external_engine_info_to_args 在参数解析阶段调用 discover，并将结果写入 args.rollout_external_engine_infos、args.rollout_num_engines、args.rollout_num_gpus，同时记录日志。

步骤 2：重构 rollout.py 引擎启动逻辑

移除 ServerGroup.start_engines 方法中的 if self.args.rollout_external 分支和 _allocate_rollout_engine_addr_and_ports_external 函数。外部引擎的地址和端口信息已通过 args.rollout_external_engine_infos 传递，不再需要特殊分配路径。添加 from slime.backends.sglang_utils.external import start_external_rollout_servers 导入，用于后续创建外部引擎的 RolloutServer 包装。调整 RolloutManager.server 和 _get_updatable_server 的类型注解为 Any 以适应多 Server 类型。

步骤 3：调整 placement_group.py 的 GPU 布局

新增 _get_placement_group_layout 函数，根据运行模式计算需要多少 GPU。当 rollout_external 为 True 时，外部引擎不计入 placement group，因此 num_gpus 只取 actor 部分（actor_num_nodes * actor_num_gpus_per_node）。如果同时 debug_rollout_only，则返回 0 总 GPU（因为引擎在外部）。同时修改 _create_placement_group 支持 num_gpus=0 时返回 None 避免创建空 placement group。

步骤 4：重构 sglang_engine.py 中引擎初始化和路由器注册

将之前内联的 _get_actual_server_args（通过 HTTP 获取引擎参数）替换为调用 get_server_info（从 external.py）。提取 _register_to_router 方法，统一处理向 SGLang 路由器注册 worker 的逻辑，支持 worker_type 和 PD 模式的 bootstrap_port。对旧版路由器（≤0.2.1）只支持 regular 类型；新版路由器发送 JSON payload 包含 worker_type 和可选的 bootstrap_port。当 worker_type 为 prefill 但缺少 bootstrap_port 时抛出 RuntimeError。

步骤 5：更新测试

新增 tests/test_external_sglang_engines.py 单元测试，使用 monkeypatch 模拟 requests.get，验证 discover、apply 函数在 regular、PD 混合场景下的正确性，以及缺少地址时的异常。新增 tests/test_qwen3_4B_external_pd.py 端到端测试，在单机 6 GPU 环境启动 4 GPU 训练 + 2 个外部 SGLang 引擎（1 prefill + 1 decode），通过 delta weight update + disk 传输完成权重同步。新增 tests/test_placement_group.py 测试新的布局函数。

关键文件：

- slime/backends/sglang_utils/external.py（模块 外部引擎；类别 source；类型 core-logic；符号 ExternalEngineInfo, is_pd_worker, to_dict, normalize_external_engine_addr）：核心新增模块，定义外部引擎发现、元信息获取、参数注入等完整流程，是 PR 的基石。
- slime/ray/rollout.py（模块 Rollout 核心；类别 source；类型 core-logic；符号 server, _get_updatable_server, _allocate_rollout_engine_addr_and_ports_external, start_rollout_servers）：移除旧的 --rollout-external 分支，改为统一的地址分配路径；调整 server 类型声明，引入 start_external_rollout_servers。
- slime/backends/sglang_utils/sglang_engine.py（模块 引擎适配；类别 source；类型 dependency-wiring；符号 _get_actual_server_args, _register_to_router）：重构 _register_to_router，支持 PD 模式和 bootstrap_port；复用 get_server_info 替代内联实现。
- slime/ray/placement_group.py（模块 布局调度；类别 source；类型 core-logic；符号 _get_placement_group_layout）：新增 _get_placement_group_layout 函数，正确处理外部引擎模式下的 GPU 分配；支持零 GPU 的 placement group。
- tests/test_external_sglang_engines.py（模块 外部引擎测试；类别 test；类型 test-coverage；符号 _Response, init, raise_for_status, json）：单元测试覆盖 discover 和 apply 函数在 regular、PD 等场景下的正确性，是外部引擎行为的重要保障。

关键符号：discover_external_engines, apply_external_engine_info_to_args, get_server_info, normalize_external_engine_addr, _register_to_router, _get_placement_group_layout, start_rollout_servers, _get_updatable_server,

get_rollout_num_engines

关键源码片段

slime/backends/sglang_utils/external.py

核心新增模块，定义外部引擎发现、元信息获取、参数注入等完整流程，是 PR 的基石。

```
# 文件：slime/backends/sglang_utils/external.py

# 外部引擎发现：根据地址列表批量获取引擎元信息
def discover_external_engines(addrs: list[str], timeout: float = 30.0) -> list[ExternalEngineInfo]:
    infos = []
    for addr in addrs:
        # 标准化地址（补充 http 前缀）
        url = normalize_external_engine_addr(addr)
        parsed = urlparse(url)
        assert parsed.hostname is not None and parsed.port is not None
        # 通过 HTTP 查询 /server_info 端点
        server_info = get_server_info(url, timeout=timeout)

        # 解析并行度参数，计算 GPU 数量
        pp_size = int(server_info.get("pp_size") or server_info.get("pipeline_parallel_size") or 1)
        tp_size = int(server_info.get("tp_size") or server_info.get("tensor_parallel_size") or 1)
        num_gpus = int(server_info.get("num_gpus") or server_info.get("num_gpus_per_engine") or
        tp_size * pp_size)
        bootstrap_port = server_info.get("disaggregation_bootstrap_port")
        bootstrap_port = int(bootstrap_port) if bootstrap_port is not None else None

        infos.append(
            ExternalEngineInfo(
                url=url,
                host=parsed.hostname,
                port=parsed.port,
                worker_type=_infer_worker_type(server_info),
                num_gpus=num_gpus,
                disaggregation_bootstrap_port=bootstrap_port,
                server_info=server_info,
            )
        )
    return infos


# 将外部引擎信息应用到参数对象，供后续 RolloutManager 使用
def apply_external_engine_info_to_args(args, logger=None) -> None:
    addrs = args.rollout_external_engine_addrs
    if not addrs:
        raise ValueError("apply_external_engine_info_to_args requires --rollout-external-engine-
        addrs.")
```

```
    infos = discover_external_engines(addrs)
    if not infos:
        raise ValueError("--rollout-external-engine-addrs did not contain any engines.")

    args.rollout_external_engine_infos = [info.to_dict() for info in infos]
    args.rollout_num_engines = len(infos)
    args.rollout_num_gpus = sum(info.num_gpus for info in infos)

    if logger is not None:
        summary = [
            {
                "url": info.url,
                "worker_type": info.worker_type,
                "num_gpus": info.num_gpus,
                "disaggregation_bootstrap_port": info.disaggregation_bootstrap_port,
            }
            for info in infos
        ]
        logger.info(f"Detected external SGLang engines: {summary}")
```

slime/ray/placement_group.py

新增 _get_placement_group_layout 函数，正确处理外部引擎模式下的 GPU 分配；支持零 GPU 的 placement group。

```
# 文件：slime/ray/placement_group.py

# 根据运行模式评估 placement group 需要的 GPU 总数和 rollout 偏移
def _get_placement_group_layout(args) -> tuple[int, int]:
    actor_num_gpus = args.actor_num_nodes * args.actor_num_gpus_per_node

    if args.debug_train_only:
        return actor_num_gpus, 0

    if args.rollout_external:
        # 外部引擎自备 GPU，不占用 trainer 节点资源
        if args.debug_rollout_only:
            return 0, 0
        return actor_num_gpus, actor_num_gpus

    if args.debug_rollout_only:
        return args.rollout_num_gpus, 0

    if args.colocate:
        return actor_num_gpus, 0

    # 标准模式：actor GPU + rollout GPU，offset 为 actor GPU 数
    return actor_num_gpus + args.rollout_num_gpus, actor_num_gpus
```

评论区精华

该 PR 未产生 review 评论，无公开讨论记录。不过通过 17 次提交的演进可以看到作者多次修复了参数移除导致的问题、delta 权重同步适配以及 CI 相关问题。

- 暂无高价值评论线程

风险与影响

- 风险： 1) 依赖外部 SGLang 引擎的 /server_info 端点，如果 SGLang 版本接口不兼容将导致发现失败； 2) 移除 --rollout-external 参数，旧用户脚本需要更新为 --rollout-external-engine-addrs； 3) 外部引擎模式下 placement group 的 GPU 计数改为只计 actor 部分，可能影响 colocate 或 debug 模式混合场景的资源配置； 4) 外部引擎无法参与 NCCL 集合通信，必须使用 delta weight update + disk 传输，如果用户配置其他传输方式（如 NCCL）会导致同步失败； 5) 端到端测试依赖 bond IP 和 IB 设备，在 CI 中可能因网络环境差异被跳过，外部模式回归覆盖不足。
  - 影响：用户影响：需要学习新的参数并部署外部 SGLang 引擎；旧的 --rollout-external 不再生效。系统影响：外部引擎模式不再占用 trainer GPU，placement group 资源分配更合理；路由器注册逻辑支持 PD 分离。团队影响：新增 external.py 核心模块，需维护与 SGLang server_info 接口的兼容性；测试覆盖了单元和端到端，但依赖特定硬件。
  - 风险标记：外部依赖裸奔，配置破坏，测试环境依赖

关联脉络

- PR #1991 [ci] Add e2e test for delta weight update: 该 PR 为 delta 权重更新添加了 e2e 测试，PR#2016 的外部引擎端到端测试也使用了 delta 同步，依赖该机制的实现。
- PR #1993 Patch sglang 0.5.12.post1 for delta sync: 更新 sglang patch 支持 delta 同步，为外部引擎的权重更新提供了底层支持。