

PR #28563 完整报告

sgl-project/sglang

[Fix] Reuse an already-running server in bench_one_batch_server instead of forking an orphan

合并时间: 2026-06-18 06:14

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/28563>

执行摘要

- 一句话: bench_one_batch_server 复用已有服务器, 避免孤儿进程
- 推荐动作: 虽然变更不大, 但后续若有人扩展基准测试框架, 可以借鉴 server_is_up 的复用模式。建议在复用已有服务器时增加模型 / 参数校验 (如比较 model name), 避免无意识跑错模型。

功能与动机

当 bench_one_batch_server 在已有运行服务器的 host:port 上执行 (无 --base-url) 时, 仍会 fork 一个子进程启动新服务器; 该子进程会加载模型导致 GPU OOM, 并可能因 SIGQUIT 杀死整个基准测试。PR body 中明确指出问题根因: 预备性检查被已有服务器满足, 但子进程仍在后台加载模型。

实现拆解

1. 新增 `server_is_up` 函数: 在 `python/sglang/test/bench_one_batch_server_internal.py` 中新增一个辅助函数, 通过请求 `/v1/models` 端点并检查返回状态码是否为 200 来判断指定端口是否已有服务器运行。
2. 修改 `launch_server_process` 函数: 在 fork 子进程之前, 先调用 `server_is_up` 检测目标端口 (超时 5 秒)。如果已有服务器响应, 则给出警告并直接返回 (`proc=None, base_url`), 复用已有服务器。
3. 添加子进程存活检查: 在等待服务器启动的循环中, 加入 `proc.is_alive()` 检查, 若子进程提前退出 (例如因 OOM 死亡), 立即抛出 `RuntimeError`, 快速失败。
4. 超时后杀死子进程: 若等待超时, 在抛出 `TimeoutError` 之前调用 `kill_process_tree(proc.pid)` 清理子进程, 防止其成为孤儿进程。

关键文件:

- `python/sglang/test/bench_one_batch_server_internal.py` (模块 性能测试; 类别 test; 类型 test-coverage; 符号 `server_is_up`): 所有变更集中在此文件, 新增 `server_is_up` 探测函数, 重写了 `launch_server_process` 的逻辑。

关键符号: `server_is_up`, `launch_server_process`

关键源码片段

python/sglang/test/bench_one_batch_server_internal.py

所有变更集中在此文件，新增 server_is_up 探测函数，重写了 launch_server_process 的逻辑。

```
def server_is_up(base_url: str, timeout: float = DEFAULT_TIMEOUT) -> bool:
    """Return True if a server answers /v1/models with 200 at base_url."""
    try:
        headers = {
            "Content-Type": "application/json; charset=utf-8",
        }
        response = requests.get(
            f"{base_url}/v1/models", headers=headers, timeout=timeout
        )
        return response.status_code == 200
    except requests.RequestException:
        return False


def launch_server_process(launch_server_func: Callable, server_args: ServerArgs):
    base_url = f"http://{server_args.host}:{server_args.port}"

    # 如果目标端口已有服务器，则直接复用而非 fork 新进程
    if server_is_up(base_url, timeout=5):
        print(
            f"WARNING: reusing the server already running at {base_url} "
            f"(--model and server-launch args ignored). Pass --base-url to silence."
        )
        return None, base_url

    proc = multiprocessing.Process(
        target=launch_server_internal,
        args=(
            launch_server_func,
            server_args,
        ),
    )
    proc.start()

    start_time = time.time()
    while time.time() - start_time < DEFAULT_TIMEOUT:
        # 快速失败：如果子进程启动期间退出（例如 OOM），立即报错
        if not proc.is_alive():
            raise RuntimeError(
                f"Server process exited during startup (exit code "
                f"{proc.exitcode}); see the traceback above for the cause."
            )
        if server_is_up(base_url):
            return proc, base_url
```

```
time.sleep(10)
```

```
# 超时后清理子进程，防止变成孤儿进程
```

```
kill_process_tree(proc.pid)
```

```
raise TimeoutError("Server failed to start within the timeout period.")
```

评论区精华

- 暂无高价值评论线程

风险与影响

- 风险：主要风险在于如果已有服务器与预期模型参数不同（如模型、精度等），基准测试结果可能不准确。PR 中通过打印警告 "reusing the server already running..." 并提示使用 `--base-url` 来规避此风险，但未强制校验。另外，`server_is_up` 超时时间固定为 5 秒，可能在某些慢速网络环境下误判。
- 影响：影响范围仅限于 `bench_one_batch_server_internal.py` 这个测试脚本。正向影响：防止多服务器进程争抢 GPU 资源导致 OOM 或孤儿进程，提升基准测试的健壮性。
- 风险标记：可能无意识复用错误模型，5 秒超时可能不足

关联脉络

- 暂无明显关联 PR