

PR #27427 完整报告

sgl-project/sglang

Add GB300 base C CI suite

合并时间: 2026-06-06 17:27

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27427>

执行摘要

- 一句话: 在 CI 中添加 GB300 硬件测试套件并迁移 4-GPU 配置
- 推荐动作: 值得精读, 特别是 `test_numa_utils.py` 的重构模式和 `slash_command_handler.py` 中字段传递的设计。展示了如何在不入侵业务代码的前提下扩展 CI 硬件覆盖。对于需要新增 CI runner 的开发者有参考价值。

功能与动机

PR body 明确指出: “Add `base-c-test-4-gpu-gb300` to PR Base C and move the former 4-GPU GB200 registrations to GB300.” 目的是为新的 Grace Blackwell 硬件平台 (GB300) 提供 CI 覆盖, 取代原先的 GB200 4-GPU runner。

实现拆解

1. 新增 runner 配置: 在 `scripts/ci/runner_configs.yml` 中定义 `4-gpu-gb300` runner, 使用 DeePEP 安装器并设置 `grace_blackwell: 1`。
2. 添加 workflow 任务: 在 `.github/workflows/pr-test.yml` 中新增 `base-c-test-4-gpu-gb300` job, 并在 `pr-test-finish` 依赖列表中列入。
3. 更新套件注册: 修改 `test/run_suite.py` 中的 `PER_COMMIT_SUITES`, 将 `base-c-test-4-gpu-gb200` 替换为 `base-c-test-4-gpu-gb300`。
4. 迁移单测注册: 分别在 `test/registered/utils/test_numa_utils.py`、`test/registered/4-gpu-models/test_deepseek_v3_cuteds1_4gpu.py`、`test/registered/disaggregation/test_disaggregation_aarch64.py` 中将 `register_cuda_ci` 的 `runner_config` 从 `4-gpu-gb200` 改为 `4-gpu-gb300`。
5. 调整 slash command 处理器: 在 `scripts/ci/utils/slash_command_handler.py` 中新增 `grace_blackwell` 字段, 并在各个调度入口 (`_dispatch_err`、`_resolve_runner_config`、`detect_suite` 的 CPU 路径) 中传递该字段, 确保 `/rerun-test` 能够正确激活 GB300 安装环境。
6. 适配 rerun-test workflow: 在 `.github/workflows/rerun-test.yml` 中添加 `grace_blackwell` 输入参数, 并在安装步骤中注入环境变量。
7. 重构 NUMA 拓扑测试: 在 `test_numa_utils.py` 中将 `_get_gpu_name` 扩展为 `_get_gpu_info` (返回 GPU 名称和数量), 创建统一的 `TestGraceBlackwellNumaTopology` 类同时支持 GB200 和 GB300, 并新增

`_query_single_numa_node_for_gpu` 辅助函数以简化 NUMA 节点断言。

关键文件:

- `test/registered/utils/test_numa_utils.py` (模块 NUMA 拓扑; 类别 test; 类型 test-coverage; 符号 `_get_gpu_name`, `_get_gpu_info`, `TestGB200NumaTopology`, `_query_single_numa_node_for_gpu`): 核心测试文件, 重构了 GPU 信息获取和 NUMA 拓扑测试类, 统一支持 GB200 和 GB300。修改量最大 (+37/-31), 信号最强。
- `.github/workflows/pr-test.yml` (模块 工作流; 类别 infra; 类型 infrastructure): 主 CI 工作流, 新增 `base-c-test-4-gpu-gb300` job 并在 `pr-test-finish` 依赖中引用。
- `scripts/ci/utils/slash_command_handler.py` (模块 CI 脚本; 类别 infra; 类型 infrastructure): 协调 `/rerun-test` 命令, 新增 `grace_blackwell` 字段传递, 确保 GB300 runner 的正确调度。
- `test/registered/disaggregation/test_disaggregation_aarch64.py` (模块 解聚测试; 类别 test; 类型 test-coverage): 将解聚测试注册到 GB300 runner, 并更换模型为 Qwen3-8B。
- `test/registered/4-gpu-models/test_deepseek_v3_cuteds1_4gpu.py` (模块 模型测试; 类别 test; 类型 test-coverage): 将 DeepSeek V3 模型测试迁移到 GB300 runner。
- `test/run_suite.py` (模块 套件注册; 类别 test; 类型 test-coverage): 更新整套 CI 套件注册, 将 `base-c-test-4-gpu-gb200` 替换为 `base-c-test-4-gpu-gb300`。
- `.github/workflows/rerun-test.yml` (模块 工作流; 类别 infra; 类型 infrastructure): 为 `/rerun-test` 工作流添加 `grace_blackwell` 输入参数并注入环境变量。
- `.github/workflows/_pr-test-stage.yml` (模块 工作流; 类别 infra; 类型 infrastructure): 允许 `grace_blackwell` 从 caller 传递到 install 步骤。
- `scripts/ci/runner_configs.yml` (模块 CI 脚本; 类别 infra; 类型 infrastructure): 定义 4-gpu-gb300 runner 的具体配置, 包括 DeePEP 安装器和 `grace_blackwell` 标志。

关键符号: `_get_gpu_info`, `_query_single_numa_node_for_gpu`, `TestGraceBlackwellNumaTopology.test_gpu_numa_mapping`, `_dispatch_err`, `_resolve_runner_config`

关键源码片段

`test/registered/utils/test_numa_utils.py`

核心测试文件, 重构了 GPU 信息获取和 NUMA 拓扑测试类, 统一支持 GB200 和 GB300。修改量最大 (+37/-31), 信号最强。

```
def _get_gpu_info(): # 替代原来的 _get_gpu_name, 同时返回 GPU 名称和数量
    try:
        import pynvml
        pynvml.nvmlInit()
        handle = pynvml.nvmlDeviceGetHandleByIndex(0)
        name = pynvml.nvmlDeviceGetName(handle)
        if isinstance(name, bytes):
            name = name.decode() # 确保字符串类型
        count = pynvml.nvmlDeviceGetCount() # 获取 GPU 数量
        pynvml.nvmlShutdown()
```

```

        return name, count
    except Exception:
        return "", 0

# 模块级调用, 获取 GPU 信息和数量, 用于判断硬件类型
_gpu_name, _gpu_count = _get_gpu_info()

@unittest.skipUnless(
    ("GB200" in _gpu_name or "GB300" in _gpu_name) and _gpu_count == 4,
    "Requires 4-GPU Grace Blackwell hardware",
)
class TestGraceBlackwellNumaTopology(unittest.TestCase):
    """硬件测试: 验证 4 GPU 的 Grace Blackwell 系统上 NUMA 拓扑符合预期
    (GPU 0-1 归节点 0, 2-3 归节点 1) 。"""

    def test_gpu_numa_mapping(self):
        self.assertEqual(_gpu_count, 4)
        expected = {0: 0, 1: 0, 2: 1, 3: 1}
        for gpu_id, expected_node in expected.items():
            result = _query_single_numa_node_for_gpu(gpu_id) # 确保返回单节点
            self.assertEqual(
                result,
                expected_node,
                f"GPU {gpu_id}: expected NUMA node {expected_node}, got {result}",
            )

```

scripts/ci/utils/slash_command_handler.py

协调 `/rerun-test` 命令, 新增 `grace_blackwell` 字段传递, 确保 GB300 runner 的正确调度。

```

def _dispatch_err(suite, msg): # 创建包含错误信息的默认调度字典
    return {
        "runner_label": None,
        "install_script": "",
        "install_timeout": "",
        "grace_blackwell": "0", # 新增: 默认不启用 Grace Blackwell
        "rdma_devices": "",
        "is_cpu": False,
        "suite": suite,
        "error": msg,
    }

def _resolve_runner_config(rc, full_path, suite):
    # 从 YAML 读取的配置
    cfg = ...
    info = {
        "runner_label": runs_on,
        "install_script": install_script,
    }

```

```
"install_timeout": str(cfg["install_timeout"]),
"grace_blackwell": str(cfg.get("grace_blackwell", "0")), # 从 YAML 读取, 缺省为 "0"
"rdma_devices": cfg.get("rdma_devices", ""),
"is_cpu": False,
"error": None,
}
return info
```

评论区精华

- Gemini Code Assist Bot 指出在 `test_numa_utils.py` 中, 硬编码的 GPU 名称检查 ("GB200" in `_gpu_name`) 在新 GB300 上会导致测试跳过或崩溃, 建议同时检查 GPU 名称和数量。
- 作者 Fridge003 在 review 中提议将 `register_cuda_ci` 改为 `4-gpu-b200`, 但最终采用了更通用的方案: 通过 `_get_gpu_info` 返回名称和计数, 并合并为 `TestGraceBlackwellNumaTopology`, 同时支持 GB200 和 GB300。
- 该讨论已解决, 最终代码实现了 bot 的建议。
- GB300 runner 上 NUMA 测试可能失败 (`correctness`): 作者通过将 `_get_gpu_name` 重构为 `_get_gpu_info` (返回名称和计数), 并创建统一的 `TestGraceBlackwellNumaTopology` 类, 同时检查名称和 GPU 数量 (`==4`) 来解决。

风险与影响

- 风险:
 - 测试迁移覆盖丢失: 原有 GB200 4-GPU runner 已停用, 如果 GB300 硬件不可用或安装脚本有误, 会导致这些测试被跳过。需确保 GB300 runners 已上线并稳定。
 - 配置不一致: `grace_blackwell` 字段分布在 `runner_configs.yml`、`slash_command_handler.py`、`rerun-test.yml` 等多个文件中, 漏传或默认值错误可能导致安装步骤不生效。
 - NUMA 测试重构回归: `TestGraceBlackwellNumaTopology` 替换了原有的 `TestGB200NumaTopology`, 若 `_query_numa_node_for_gpu` 在新硬件上返回异常, 可能误报失败或隐藏错误。
 - 临时变量清除: 移除了 `FORCE_REBUILD_DEEPEP` 环境变量, 若依赖方未及时适配可能影响 `DeePEP` 构建。
 - 影响: 用户: 无直接用户影响。系统: CI 基础设施新增 GB300 runner 支持, 原有 GB200 4-GPU 任务迁移到 GB300, 影响 CI 执行效率和硬件利用率。团队: 需要维护两组 runner 配置 (GB300 和 B200), 并确保 `grace_blackwell` 环境变量正确传递。带来对 `DeePEP` 安装器版本的新约束。
- 风险标记: 新硬件依赖, 测试迁移覆盖丢失, 配置传递一致性, 临时变量清除影响

关联脉络

- PR #27344 [CI] Isolate CUDA coredump dir per run to fix tracker mis-attribution: 同为 CI 基础设施改进, 涉及 runner 配置和测试隔离, 体现了 CI 系统的持续演进。