

PR #48968 完整报告

vllm-project/vllm

[Kernel][Helion] Add numerics checks to benchmark script

合并时间: 2026-08-01 08:16

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/48968>

执行摘要

- 一句话: Helion 基准脚本新增逐 shape 数值检查
- 推荐动作: 值得精读。核心设计决策值得借鉴: 正确性与计时输入隔离 (copy.deepcopy)、数值参考与性能基线解耦 (make_correctness_baseline)、pytree 级容差比较与 FP8 one-ULP 特判、以及对上游包名漂移的容错导入。这些模式可直接复用到其他 kernel 基准脚本与 CI 正确性门禁; 若团队推广内核数值门禁, 可参考 check_kernel_correctness 的逐 case 结果收集设计。

功能与动机

PR body 说明核心动机是 "Run one numerics check for every shape before benchmarking it", 即避免基准分数来自数值错误的 kernel; 同时 "Use independent input copies for correctness and timing so validation cannot affect benchmark state" 强调验证与计时状态隔离。commit b42873a 进一步指出 nightly wheel 上 vllm.kernels.helion.ops 的 import_all_kernels 已重命名为 import_all_ops, 顶层硬导入会导致脚本启动即崩溃, 需要本地化导入来容忍名称漂移。

实现拆解

1. 数值检查核心: 新增 check_correctness(kernel, baseline_fn, inputs, case), 对输入分别 copy.deepcopy 后调用内核与基线, 再通过 _assert_close 比较返回值与变异参数, 失败时抛出带有 case 名的 AssertionError, 便于定位具体 shape。
2. 基线拆分与选择: 将原 make_autotune_baseline 拆出 make_eager_baseline (返回未编译的 autotune_baseline_fn), make_autotune_baseline 变为 eager + torch.compile(inductor); 新增 make_correctness_baseline, 依据 --numerics-with-perf-baseline 决定数值参考用 eager 还是计时基线, 实现 "数值参考" 与 "性能基线" 解耦。
3. 容差与 FP8 处理: _assert_close 用 torch.utils._pytree.tree_flatten 展平比较结构; 优先调用 helion_settings.autotune_baseline_accuracy_check_fn 自定义检查; 否则使用 autotune_baseline_atol/rtol (缺省 1e-2); 对 is_fp8_dtype 张量采用 atol=1、rtol=0 的 one-ULP 容差, 与现有 FP8 内核测试一致。
4. 批量检查与 CLI: check_kernel_correctness 逐 case 记录 CorrectnessResult 而不提前中断, 便于 CI gate 一次性报告所有失败 shape; 新增 --numerics-only (只验证不计) 参数。

5. 工程韧性: `import_all_kernels` 改为本地函数, 导入 `vllm.kernels.helion.ops` 后按 `import_all_kernels / import_all_ops` 依次尝试显式导入器, 容忍跨版本名称漂移; `log_versions` 记录 `torch / helion / triton` 版本便于复现。
6. 测试配套: 新增 `tests/kernels/helion/test_benchmark_script.py` (90 行), 用 `_FakeKernel` 模拟 "原地写输出 + 返回新张量" 的内核, 覆盖单次运行不变异基准输入、变异输入不匹配报错、返回值不匹配报错、`log_versions` 输出顺序四个场景; 无 Helion 环境时 `module-level skip`。

关键文件:

- `scripts/benchmark_helion_kernels.py` (模块 基准脚本; 类别 `source`; 类型 `core-logic`; 符号 `import_all_kernels`, `log_versions`, `pkg_version`, `make_autotune_baseline`): 唯一源码改动文件, 新增数值检查核心 `check_correctness / _assert_close / check_kernel_correctness`, 拆分 `eager` 与 `autotune` 基线, 加入版本日志与容错内核导入 (+249/-10)。
- `tests/kernels/helion/test_benchmark_script.py` (模块 基准测试; 类别 `test`; 类型 `test-coverage`; 符号 `_FakeKernel`, `init`, `call`, `test_check_correctness_runs_once_without_mutating_benchmark_inputs`): 新增测试覆盖 `check_correctness` 的单次运行、输入隔离、两类不匹配报错及 `log_versions` 输出, 是数值检查逻辑的主要质量保障。

关键符号: `check_correctness`, `_assert_close`, `make_correctness_baseline`, `make_eager_baseline`, `make_autotune_baseline`, `import_all_kernels`, `log_versions`, `check_kernel_correctness`

关键源码片段

`scripts/benchmark_helion_kernels.py`

唯一源码改动文件, 新增数值检查核心 `check_correctness / _assert_close / check_kernel_correctness`, 拆分 `eager` 与 `autotune` 基线, 加入版本日志与容错内核导入 (+249/-10)。

```
def _assert_close(actual: object, expected: object, atol: float, rtol: float) -> None:
    # 用 pytree 展平比较返回值与变异参数, 结构不一致时直接报错
    actual_flat, actual_spec = tree_flatten(actual)
    expected_flat, expected_spec = tree_flatten(expected)
    if actual_spec != expected_spec:
        raise AssertionError(
            f'Output structure mismatch: {actual_spec} != {expected_spec}'
        )

    for actual_leaf, expected_leaf in zip(actual_flat, expected_flat, strict=True):
        # FP8 张量允许 1 ULP 误差 (atol=1, rtol=0),
        # 与 vLLM 现有 FP8 内核测试的约定保持一致
        is_fp8 = isinstance(actual_leaf, torch.Tensor) and is_fp8_dtype(
            actual_leaf.dtype
        )
```

```

helion_assert_close(
    actual_leaf,
    expected_leaf,
    atol=1 if is_fp8 else atol,
    rtol=0 if is_fp8 else rtol,
)

def check_correctness(
    kernel: Any,
    baseline_fn: Callable,
    inputs: tuple[Any, ...],
    case: str,
) -> None:
    # 正确性与计时使用独立输入副本，防止内核的原地写污染后续计时结果
    kernel_inputs = copy.deepcopy(inputs)
    baseline_inputs = copy.deepcopy(inputs)

    kernel_output = kernel(*kernel_inputs)
    baseline_output = baseline_fn(*baseline_inputs)

    settings = kernel.helion_settings
    try:
        # 优先使用内核注册的自定义精度检查函数
        custom_check = getattr(settings, 'autotune_baseline_accuracy_check_fn', None)
        if custom_check is not None:
            custom_check(kernel_output, baseline_output)
            custom_check(kernel_inputs, baseline_inputs)
            return

        # 否则使用内核配置的容差，缺省回退 1e-2
        configured_atol = getattr(settings, 'autotune_baseline_atol', None)
        configured_rtol = getattr(settings, 'autotune_baseline_rtol', None)
        atol = 1e-2 if configured_atol is None else configured_atol
        rtol = 1e-2 if configured_rtol is None else configured_rtol
        _assert_close(kernel_output, baseline_output, atol=atol, rtol=rtol)
        _assert_close(kernel_inputs, baseline_inputs, atol=atol, rtol=rtol)
    except AssertionError as e:
        # 带上 case 名，便于定位具体失败的 shape
        raise AssertionError(f'Numerics check failed for case {case}:\n{e}') from e

```

评论区精华

该 PR 没有实质 review 评论线程（claude[bot] 因 fork 自动跳过 review），关键讨论集中在 issue 评论区：

- mergify[bot] 提示 pre-commit 失败，要求运行 `uv pip install pre-commit...` 后执行 `pre-commit run --all-files`；后续 commit 5a47012 以 ruff format 修复。

- yushangdi 说明 CI 失败原因: "need a latest helion release for using the `helion.autotuner.accuracy.assert_close / is_fp8_dtype` APIs", 即新代码依赖 Helion 新版 API。
- 合并前 xiaohongchen1991 两次 APPROVED、zou3519 APPROVED, 无未解决疑虑。
- CI 失败与 Helion 新版本 API 依赖 (question): 最终合并, 后续 #50307 Bump Helion to 1.4.0 升级提供所需 API, 依赖问题解决。

风险与影响

- 风险:
 1. Helion 版本依赖: `scripts/benchmark_helion_kernels.py` 顶层 `try` 块导入 `helion.autotuner.accuracy`, 旧版 Helion 未提供该模块时会触发 `ImportError` 并 `sys.exit(1)`, 基准脚本整体不可用; 本次 CI 失败已实证该风险, 需与 Helion 版本升级 (如 #50307) 联动。
 2. FP8 one-ULP 容差漏检: FP8 比较采用 `atol=1`、`rtol=0`, 误差恰落在 1 ULP 内的真实退化可能不被报告; 这是沿用现有 FP8 内核测试约定的有意识取舍。
 3. eager 数值参考的误报可能: 默认用未编译 eager 参考做数值对比, 而计时走 `torch.compile/autotune`, 算子融合与计算顺序差异可能引入超容差偏差, 对量化类内核需关注。
 4. 深拷贝一次性开销: `check_correctness` 对每个 shape 深拷贝输入, 大 shape 下有额外内存与拷贝耗时, 但仅一次且不计入计时。
 5. 测试覆盖条件: 新增测试在无 Helion 环境时 `module-level skip`, 未安装 Helion 的 CI 无法覆盖该脚本逻辑。 - 影响: 用户与线上系统零感知: 该 PR 只改动独立开发基准脚本与对应测试, 不进入 `serving` 或模型输出路径。对团队的直接影响是 Helion 内核开发者每次运行基准前自动获得数值正确性信号, 防止虚假加速比进入性能报告; `log_versions` 让结果可复现; 同时脚本运行前提变为 Helion 新版 (含 `helion.autotuner.accuracy` API), 旧环境需先升级。 - 风险标记: 依赖 Helion 新版 API, FP8 one-ULP 容差可能漏检, 无 Helion 环境测试被跳过, 深拷贝一次性开销

关联脉络

- PR #50307 Bump Helion to 1.4.0: 本 PR 的数值检查依赖 `helion.autotuner.accuracy.assert_close / is_fp8_dtype` 新 API, 而 #50307 将 Helion 升级到 1.4.0 并提供该 API (其描述亦提到修复内核数值问题), 是 issue 评论中 CI 失败所指的版本升级配套。