PR #32788 完整报告

sgl-project/sglang

[Kernel] Add inventory guards and clean benchmark layout

合并时间：2026-07-30 09:03

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/32788

执行摘要

此 PR 在统一的 sglang.kernels 命名空间迁移完成后，为内核注册系统添加了仅 CPU 的库存检查，增强冲突检测，并清理了基准测试布局。变更主要集中在注册逻辑增强、新增库存测试和文件迁移上，旨在提升内核系统的健壮性和可维护性。

功能与动机

关联 Issue #29630 提出的统一 sglang.kernels 命名空间迁移已经完成。此 PR 作为后续清理，旨在防止未来冲突注册导致不确定行为，并为内核树提供结构性检查，确保操作组与包目录一致，注册目标属性正确声明。PR body 明确指出要“为操作组、注册目标、JIT 源和 AOT 编译单元添加仅 CPU 的库存检查”并“拒绝冲突注册”。

实现拆解

1. 增强注册冲突检测：在 KernelRegistry.register() 中，将原有的幂等替换改为严格检查——相同 (op, backend) 但 target 不同的注册会抛出 ValueError，只有完全相同的 spec 才保持幂等。
2. 新增仅 CPU 库存检查：创建 test/registered/kernels/test_kernel_inventory.py，利用 AST 分析遍历内核包目录，验证 ops.__all__ 与目录名一致、注册目标属性声明正确、JIT 源模式匹配等，这些测试在 CI 中以 CPU 模式运行。
3. 基准测试和配置脚本迁移：将 python/sglang/kernels/ops/attention/flash_attn/cute/ 下的独立基准测试和配置搜索脚本移到 benchmark/kernels/ 对应子目录，并更新导入路径。删除无外部引用的 bench_utils.py，其 flops、带宽计算等功能不再属于内核包。
4. 测试调整：更新 test_fused_op.py 和 test_kernels_namespace.py，将原测试 test_registry_reregister_replaces 拆分为幂等性测试和冲突拒绝测试，并匹配新语义。
5. 对齐测试目录结构：将 test/registered/kernels/benchmark/attention/bench_add_constant.py 等移动到 benchmark/elementwise/，test/registered/kernels/ops/model/test_inkling_rel_proj.py 移动到 ops/attention/，确保每个操作组的测试都在对应目录下。

以下是核心注册冲突检测的实现（python/sglang/kernels/registry.py）：

```
class KernelRegistry:
    """Maps ``"<group>.<name>"`` operator ids to their :class:`KernelSpec` list."""

    def __init__(self) -> None:
        self._by_op: Dict[str, List[KernelSpec]] = defaultdict(list)
```

```
    def register(self, spec: KernelSpec) -> KernelSpec:
        """Register ``spec``.

        幂等注册（完全相同的 spec）始终安全返回；对同一
        ``(op, backend)`` 但 ``target`` 不同的注册会立即
        拒绝，因为静默替换会使得选定实现依赖于不可控的
        模块加载顺序。
        """
        existing = self._by_op[spec.op]
        for other in existing:
            if other.backend == spec.backend:
                if other != spec:
                    raise ValueError(
                        f"Conflicting kernel registration for op {spec.op!r}, "
                        f"backend {spec.backend.value!r}: "
                        f"{other.target!r} != {spec.target!r}"
                    )
                return spec
        existing.append(spec)
        return spec
```

新增的库存检查测试使用 AST 分析确保操作组与包目录一致（test/registered/kernels/test_kernel_inventory.py）：

```
"""CPU-only structural checks for the unified kernel tree."""

from __future__ import annotations

import ast
from pathlib import Path

import sglang.kernels as kernels

REPO_ROOT = Path(__file__).resolve().parents[3]
OPS_ROOT = REPO_ROOT / "python" / "sglang" / "kernels" / "ops"


def _directory_names(root: Path) -> set[str]:
    """返回 root 下包含 .py 文件的子目录名集合，不包含隐藏目录和 ``__`` 包。"""
    return {
        path.name
        for path in root.iterdir()
        if path.is_dir()
        and not path.name.startswith((".", "__"))
        and any(path.rglob("*.py"))
    }


def test_declared_operator_groups_match_packages():
    """检查 ``kernels.ops.__all__`` 是否与 OPS_ROOT 下的实际包目录一一对应。"""
```

```
    assert set(kernels.ops.__all__) == _directory_names(OPS_ROOT)
```

配套的测试调整覆盖了新冲突检测行为（test/registered/kernels/ops/layernorm/test_fused_op.py）：

```
def test_registry_reregister_is_idempotent():
    """验证完全相同的 spec 再次注册不会导致错误或重复条目。"""
    reg = KernelRegistry()
    spec = _spec(target="math:sqrt")
    reg.register(spec)
    reg.register(spec)
    assert reg.get("g.n") == [spec]


def test_registry_rejects_conflicting_backend_registration():
    """验证不同 target 的同一 (op, backend) 注册抛出 ValueError。"""
    reg = KernelRegistry()
    reg.register(_spec(target="math:sqrt"))
    with pytest.raises(ValueError, match="Conflicting kernel registration"):
        reg.register(_spec(target="math:floor"))
```

评论区精华

本 PR 未收到公开 Review 讨论，作者直接合并。所有 CI 状态为通过。

风险与影响

- 风险：register() 语义变更可能影响依赖旧幂等行为的模块重载；移动文件可能遗留内部引用；AST 库存检查可能因 Python 版本差异出现兼容问题。
- 影响：对内核开发者和 CI 流程有直接影响——新增的库存检查将固化操作组与目录的一致性要求；冲突拒绝避免因导入顺序导致的隐蔽 bug；基准测试因搬迁更易于发现和执行。

关联脉络

本 PR 是 RFC #29630（统一 sglang.kernels 命名空间）迁移完成后的清理和强化步骤。RFC 已经经过多个阶段（#30044, #30784-#30795, #31292, #31307, #32072）实现了内核组织、JIT 基础设施和测试的全面重组。此 PR 进一步通过库存检查和冲突拒绝来确保长期一致性，标志着内核系统从“迁移期”进入“维护期”。