PR #32348 完整报告

sgl-project/sglang

Report accelerator type in /v1/loads

合并时间：2026-07-25 08:31

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/32348

执行摘要

- 一句话：/v1/loads 返回 accelerator 字段
- 推荐动作：该 PR 作为基础设施改进值得合并。设计简单合理：使用 LRU 缓存避免重复查询硬件信息，且结果易于测试。建议后续考虑在 /v1/loads 文档中明确说明 accelerator 字段。

功能与动机

PR body 明确指出：Expose the accelerator name in the JSON /v1/loads response so clients can distinguish heterogeneous serving instances without an additional probe。

实现拆解

1. 新增 _accelerator_name 函数 (python/sglang/srt/entrypoints/v1_loads.py)：引入 functools.lru_cache 对结果进行缓存（maxsize=1），函数内部调用 sglang.srt.utils.get_device_name() 获取设备的营销名称（如 NVIDIA GB300），若不可用则返回 None。
2. 修改 get_loads 端点响应：在返回的 JSON 字典中增加键 "accelerator"，值为 _accelerator_name() 的调用结果，位于 "loads" 之前。同时更新了函数的文档字符串，将原来的 "JSON response with timestamp, version, and per-DP-rank loads" 改为包含 "accelerator"。
3. 新增单元测试 (test/registered/unit/entrypoints/test_v1_loads_aggregate.py)：添加 TestLoadsAcceleratorField 测试类，通过 unittest.mock.patch.object 模拟 _accelerator_name 返回 "NVIDIA GB300"，然后验证 JSON 响应中 accelerator 字段值等于该字符串。同时新增 from unittest import mock 和 from sglang.srt.entrypoints import v1_loads 导入。

关键文件：

- python/sglang/srt/entrypoints/v1_loads.py（模块 API 入口；类别 source；类型 core-logic；符号 _accelerator_name）：核心变更文件：新增 _accelerator_name 函数和导入，并在 /v1/loads 响应中增加 accelerator 字段。
- test/registered/unit/entrypoints/test_v1_loads_aggregate.py（模块 单元测试；类别 test；类型 test-coverage；符号 TestLoadsAcceleratorField, test_accelerator_reported_in_json）：新增 TestLoadsAcceleratorField 测试类，确保 accelerator 字段正确输出。

关键符号：_accelerator_name

关键源码片段

python/sglang/srt/entrypoints/v1_loads.py

核心变更文件：新增 _accelerator_name 函数和导入，并在 /v1/loads 响应中增加 accelerator 字段。

```
# python/sglang/srt/entrypoints/v1_loads.py

from functools import lru_cache
from sglang.srt.utils import get_device_name # 新增导入

router = APIRouter()

@lru_cache(maxsize=1) # 只缓存一次，避免重复调用
# 获取设备营销名称（如 "NVIDIA GB300"），不可用时返回 None
def _accelerator_name() -> Optional[str]:
    return get_device_name()

# ...（其他函数不变）

@router.get("/v1/loads")
async def get_loads(
    dp_rank: Optional[int] = None,
    include: Optional[str] = None,
    format: Optional[str] = None,
    tokenizer_manager=Depends(_get_tokenizer_manager),
):
    # ...（原有逻辑不变）
    return {
        "timestamp": datetime.now(timezone.utc).isoformat(),
        "version": __version__,
        "accelerator": _accelerator_name(), # 新增字段
        "loads": loads,
    }
```

test/registered/unit/entrypoints/test_v1_loads_aggregate.py

新增 TestLoadsAcceleratorField 测试类，确保 accelerator 字段正确输出。

```
# test/registered/unit/entrypoints/test_v1_loads_aggregate.py

from unittest import mock
from sglang.srt.entrypoints import v1_loads # 新增导入，用于 patch

# ...（原有测试类不变）

class TestLoadsAcceleratorField(CustomTestCase):
    def test_accelerator_reported_in_json(self):
```

```
        """Guards the response contract: the JSON envelope carries an
        "accelerator" field with the detected device name."""
        manager = _FakeHttpTokenizerManager([LoadSnapshot(dp_rank=0)])

        with mock.patch.object(
            v1_loads, "_accelerator_name", return_value="NVIDIA GB300"
        ):
            response = asyncio.run(get_loads(tokenizer_manager=manager))
            self.assertEqual(response["accelerator"], "NVIDIA GB300")
```

评论区精华

当前没有 review 评论或讨论线程。PR 由作者 cctry 自行合并，无其他审阅者参与。

- 暂无高价值评论线程

风险与影响

- 风险：该 PR 变更简单，仅在后端 JSON 响应中增加一个字段，不修改任何现有字段或行为。风险极低：
  - 没有回归风险：新增字段不会影响已有字段的解析；
  - 没有性能影响：_accelerator_name 使用 LRU 缓存，仅第一次调用时计算；
  - 没有兼容性问题：客户端可能忽略未知字段，向后兼容。
  - 影响：
  - 用户 / 客户端：任何使用 /v1/loads 接口的后端、负载均衡器或监控工具，现在可以直接从 JSON 获取加速器类型，无需额外调用硬件探测接口。
  - 系统：无影响。
  - 团队：提供了集群异构感知的基础能力，后续可基于此字段进行更智能的负载分配。
  - 风险标记：暂无

关联脉络

- PR #32245 Add prefill and decode load counters to LoadSnapshot: 同属 /v1/loads 的 observability 改进，扩展了负载接口信息。