

PR #27861 完整报告

sgl-project/sglang

test(xpu): add multi-feature and embedding stage-b tests

合并时间: 2026-06-12 14:36

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27861>

执行摘要

- 一句话: XPU CI 新增多特性与 Embedding 测试
- 推荐动作: 此 PR 适合作为 XPU CI 测试模板参考, 尤其是单一 fixture 覆盖多特性的设计模式。建议关注 register_xpu_ci 注册机制和 CustomTestCase 基类的使用方式。对于 Intel XPU 开发者, 可从此测试用例中学习如何编写类似的功能验证测试。

功能与动机

参照 PR body, XPU CI 的测试覆盖原本只覆盖 21 个特性中的 7 个, 此 PR 新增两个测试文件, 将覆盖数提升至 13 个, 填补关键功能缺口 (如 JSON 约束解码、Radix Cache、Embedding)。目标是在不增加过多 CI 耗时的前提下, 通过单一服务器实例测试多个特性。

实现拆解

1. 创建测试文件 `test_xpu_serving_features.py`: 继承 CustomTestCase, 在 setUpClass 中启动 Llama-3.2-1B-Instruct 并指定 --device xpu, 包含 5 个测试方法: test_openai_chat_completion、test_json_constrained_generation、test_sampling_penalty、test_radix_cache_multiturn_hit、test_reasoning_separate_parser。通过 register_xpu_ci(est_time=300, suite="stage-b-test-1-gpu-xpu") 注册到 CI 套件。
2. 创建测试文件 `test_xpu_embedding.py`: 使用 --is-embedding 模式启动一个小型 Embedding 模型, 测试单条和批量 embedding 请求 (test_embedding_single、test_embedding_batch)。注册到同一 stage-b 套件, 估计用时 120 秒。
3. 修改 CI 工作流 `pr-test-xpu.yml`: 移除已内置于基础镜像的 triton-xpu==3.7.1 显式安装; 移除 stage-b 运行的 --continue-on-error 标志, 使 PR 运行期间采用 fail-fast 策略 (此修改在合入后会因 #27860 的 --continue-on-error 标志而被覆盖还原)。
4. 根据 Review 重命名: Reviewer 建议将原 test_xpu_multi_feature.py 重命名为 test_xpu_openai_server.py 或 test_xpu_server_smoke.py, 作者接受并重命名为 test_xpu_serving_features.py, 类名同步为 TestXPUServingFeatures。

关键文件:

- test/registered/xpu/test_xpu_serving_features.py (模块 XPU 测试; 类别 test; 类型 test-coverage; 符号 TestXPUServingFeatures, setUpClass, tearDownClass, _client): 核心新测试文件, 单一 fixture 覆盖 5 个服务特性, 是 XPU CI 覆盖率提升的主要来源。

- test/registered/xpu/test_xpu_embedding.py (模块 XPU 测试; 类别 test; 类型 test-coverage; 符号 TestXPUEmbedding, setUpClass, tearDownClass, _client) : 新增 Embedding 测试, 验证 XPU 上单条和批量 Embedding 端点, 填补 Embedding 覆盖缺失。
- .github/workflows/pr-test-xpu.yml (模块 CI 配置; 类别 infra; 类型 infrastructure) : 清理 CI workflow, 移除冗余 triton-xpu 安装和临时移除 --continue-on-error 标志, 支持新测试套件的运行。

关键符号: TestXPUServingFeatures.setUpClass,
 TestXPUServingFeatures.tearDownClass, TestXPUServingFeatures._client,
 TestXPUServingFeatures.test_openai_chat_completion,
 TestXPUServingFeatures.test_json_constrained_generation,
 TestXPUServingFeatures.test_sampling_penalty,
 TestXPUServingFeatures.test_radix_cache_multiturn_hit,
 TestXPUServingFeatures.test_reasoning_separate_parser,
 TestXPUEmbedding.setUpClass, TestXPUEmbedding.tearDownClass,
 TestXPUEmbedding._client, TestXPUEmbedding.test_embedding_single,
 TestXPUEmbedding.test_embedding_batch

关键源码片段

test/registered/xpu/test_xpu_serving_features.py

核心新测试文件, 单一 fixture 覆盖 5 个服务特性, 是 XPU CI 覆盖率提升的主要来源。

```
class TestXPUServingFeatures(CustomTestCase):
    """One server, many features. Boots Llama-3.2-1B-Instruct once and ..."""
    @classmethod
    def setUpClass(cls):
        cls.model = DEFAULT_SMALL_MODEL_NAME_FOR_TEST
        cls.base_url = DEFAULT_URL_FOR_TEST
        # 启动一个 XPU 服务器实例, 供所有测试复用
        cls.process = popen_launch_server(
            cls.model, cls.base_url,
            timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
            other_args=["--device", "xpu"],
        )
        cls.openai_url = cls.base_url + "/v1"

    @classmethod
    def tearDownClass(cls):
        kill_process_tree(cls.process.pid)

    def _client(self) -> openai.Client:
        # 服务器无 API key, 但 openai client 需要非空字符串
        return openai.Client(api_key="EMPTY", base_url=self.openai_url)

    def test_openai_chat_completion(self):
        """验证 OpenAI Chat Completions API 在 XPU 上正常响应。"""
```

```

response = self._client().chat.completions.create(
    model=self.model,
    messages=[{"role": "user", "content": "Say hello in one word."}],
    max_tokens=8, temperature=0.0,
)
self.assertEqual(len(response.choices), 1)
self.assertEqual(response.choices[0].message.role, "assistant")
self.assertGreater(len(response.choices[0].message.content or ""), 0)
self.assertGreater(response.usage.completion_tokens, 0)

def test_json_constrained_generation(self):
    """验证 JSON Schema 约束生成在 XPU 上输出正确的 JSON。"""
    schema = {"type": "object", "properties": {"name": {"type": "string"}, "age": {"type": "integer"}},
    "required": ["name", "age"]}
    response = self._client().chat.completions.create(
        model=self.model,
        messages=[{"role": "user", "content": "Return a JSON object with fields name (string)
        and age (integer)."}],
        max_tokens=64, temperature=0.0,
        response_format={"type": "json_schema", "json_schema": {"name": "person", "schema":
        schema, "strict": True}},
    )
    text = response.choices[0].message.content
    self.assertIsNotNone(text)
    parsed = json.loads(text)
    self.assertIn("name", parsed)
    self.assertIn("age", parsed)
    self.assertIsInstance(parsed["age"], int)

def test_sampling_penalty(self):
    """验证频率/存在惩罚改变了相同 prompt 的输出。"""
    prompt = "List five different colors:"
    baseline = self._client().completions.create(
        model=self.model, prompt=prompt, max_tokens=64, temperature=0.7, seed=1,
    )
    penalized = self._client().completions.create(
        model=self.model, prompt=prompt, max_tokens=64, temperature=0.7, seed=1,
        frequency_penalty=2.0, presence_penalty=2.0,
    )
    self.assertGreater(len(baseline.choices[0].text), 0)
    self.assertGreater(len(penalized.choices[0].text), 0)
    # 惩罚必须改变输出
    self.assertNotEqual(baseline.choices[0].text, penalized.choices[0].text)

def test_radix_cache_multiturn_hit(self):
    """验证多轮对话中 Radix Cache 命中。"""
    run_multiturn_cache_hit_test(
        base_url=self.base_url, model_path=self.model,
        num_clients=4, num_rounds=3, request_length=128, output_length=64,
    )

```

)

评论区精华

Reviewer mingfeima 在 `test_xpu_serving_features.py` 第 9 行评论道: "is it more proper to change from `test_xpu_multi_feature.py` to `test_xpu_openai_server.py` or `test_xpu_server_smoke.py`?" 作者接受建议, 在最终提交中将文件重命名为 `test_xpu_serving_features.py`, 类名改为 `TestXPUServingFeatures`, 并更新了注释。

- 测试文件命名讨论 (design): 作者接受并重命名为 `test_xpu_serving_features.py` / `TestXPUServingFeatures`, 认为更准确反映服务特性覆盖范围。

风险与影响

- 风险:
 - 回归风险: PR 仅涉及新增测试文件和 CI 配置, 未修改任何生产代码, 回归风险极低。
 - 测试稳定性: XPU CI runner 可能存在资源竞争或环境不一致问题, 导致测试假阳性失败。
 - 依赖风险: 移除 `triton-xpu` 显式安装依赖于基础镜像 `intel/sglang-dev:latest` 正确预装该包; 若镜像更新不及时, CI 将失败。
 - CI 耗时增加: 两个 fixture 各启动一次服务器, 预计增加约 5 分钟 CI 总时长, 但仍在合理范围。
- 影响:
 - 用户: 无直接影响。
 - 系统: CI 验证更全面, 确保 XPU 平台的多项核心功能通过回归测试, 降低发布风险。
 - 团队: 减少手动测试验证成本, 提前发现 XPU 相关问题。
 - 风险标记: 测试稳定性依赖 XPU 环境, CI 基础设施依赖, 新增 CI 耗时

关联脉络

- PR #27860 [Infra] Cleanup XPU CI workflow: 此 PR 基于 #27860 的 CI 清理 workflow, PR body 明确说明 stacked on #27860, 建议先合入 #27860。