# PR #25644 完整报告

sgl-project/sglang

[Speculative] [NPU] Adaptive-SD NPU support

合并时间：2026-06-01 19:19

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25644

## 执行摘要

- 一句话：NPU 自适应推测解码支持
- 推荐动作：建议审核并合并。该 PR 改动简洁、目的明确，已在 Ascend 910B 上进行准确性和性能测试，结果积极。无安全或兼容性顾虑。

## 功能与动机

作为对 #21599（自适应推测解码）的后续，此 PR 旨在为 NPU 后端提供相同的自适应推测解码能力。原实现中 build_adaptive_runtime_state 硬编码使用 CudaGraphRunner，在 NPU 上无法正确启用图执行。关联 issue #23705。

## 实现拆解

1. 扩展 NPUGraphRunner.__init__：在 python/sglang/srt/hardware_backend/npu/graph_runner/npu_graph_runner.py 中，将 __init__ 方法从接收单一 model_runner 参数改为接收三个可选关键字参数 attn_backend、speculative_num_steps 和 speculative_num_draft_tokens，并通过 super().__init__ 传递给父类 CudaGraphRunner。
2. 修改 eagle_worker.py 的 build_adaptive_runtime_state：导入 NPUGraphRunner，在构建 target_graph_runner 时，根据 _is_npu 标志选择 NPUGraphRunner 或 CudaGraphRunner，并传入必要的推测解码参数。
3. 同步修改 eagle_worker_v2.py：应用与 eagle_worker.py 完全相同的逻辑，确保两个 EAGLE 工作器版本（v1 和 v2）均支持 NPU 自适应推测解码。
4. 无测试文件变更：本次提交未添加或修改测试文件，依赖现有测试覆盖。

关键文件：

- python/sglang/srt/hardware_backend/npu/graph_runner/npu_graph_runner.py（模块 NPU 图运行器；类别 source；类型 core-logic；符号 init）：核心改动：扩展 __init__ 以接收推测解码参数，并传递给父类 CudaGraphRunner，使 NPU 图运行器支持自适应推测解码。
- python/sglang/srt/speculative/eagle_worker.py（模块 推测解码；类别 source；类型 dependency-wiring）：在 build_adaptive_runtime_state 中根据 _is_npu 选择 NPUGraphRunner 而非 CudaGraphRunner，是启用 NPU 自适应推测解码的关键连接点。
- python/sglang/srt/speculative/eagle_worker_v2.py（模块 推测解码；类别 source；类型 dependency-wiring）：与 eagle_worker.py 相同的改动，保证 EAGLE v2 工作器在 NPU

上也能使用自适应推测解码。

关键符号：init, build_adaptive_runtime_state

## 关键源码片段

### python/sglang/srt/hardware_backend/npu/graph_runner/npu_graph_runner.py

核心改动：扩展 __init__ 以接收推测解码参数，并传递给父类 CudaGraphRunner，使 NPU 图运行器支持自适应推测解码。

```
# python/sglang/srt/hardware_backend/npu/graph_runner/npu_graph_runner.py

class NPUGraphRunner(CudaGraphRunner):
    """NPU 图运行器，通过 torch.compile 和 NPU 图执行模型前向传播。"""

    def __init__(
        self,
        model_runner: ModelRunner,
        *,
        attn_backend=None, # 新增：注意力后端，用于自适应推测解码
        speculative_num_steps: Optional[int] = None, # 新增：推测步数
        speculative_num_draft_tokens: Optional[int] = None, # 新增：推测草稿 token 数
    ):
        # 将补丁函数替换为 NPU 版本
        sglang.srt.model_executor.cuda_graph_runner.patch_model = patch_model_npu
        # 将新参数传递给父类，父类会据此初始化图运行器的相关属性
        super().__init__(
            model_runner,
            attn_backend=attn_backend,
            speculative_num_steps=speculative_num_steps,
            speculative_num_draft_tokens=speculative_num_draft_tokens,
        )
        self.update_attr_name = None
        self.update_attr_type = None
        self.model_runner = model_runner
        self._init_arch_map()
        self.use_fia = get_bool_env_var("ASCEND_USE_FIA", "False")
```

### python/sglang/srt/speculative/eagle_worker.py

在 build_adaptive_runtime_state 中根据 _is_npu 选择 NPUGraphRunner 而非 CudaGraphRunner，是启用 NPU 自适应推测解码的关键连接点。

```
# python/sglang/srt/speculative/eagle_worker.py（部分）

    def build_adaptive_runtime_state(
        self, speculative_num_steps: int, speculative_num_draft_tokens: int
    ) -> SpecRuntimeState:
        # ... 省略上下文 ...
```

```
if not self.server_args.disable_cuda_graph:
    # 根据硬件后端选择正确的图运行器
    TargetGraphRunnerCls = NPUGraphRunner if _is_npu else CudaGraphRunner
    target_graph_runner = TargetGraphRunnerCls(
        target_model_runner,
        attn_backend=target_attn_backend,
        speculative_num_steps=speculative_num_steps,
        speculative_num_draft_tokens=speculative_num_draft_tokens,
    )
# ... 后续构建 SpecRuntimeState ...
```

## 评论区精华

PR 无 review 评论，仅有审核人 iforgetmyname 的两次批准。未发现设计争议或未解决问题。

- 暂无高价值评论线程

## 风险与影响

- 风险：
  1. 回归风险：对 NPUGraphRunner 的修改补充了父类所需的参数，不会影响现有 NPU 非自适应场景。但若 _is_npu 标志在特定配置下未正确设置，可能导致仍使用 CudaGraphRunner，从而在 NPU 上运行失败。
  2. 兼容性风险：改动集中在 NPU 特定代码路径，不影响 CUDA 后端。但 NPUGraphRunner 新增参数未做向后兼容处理——旧代码直接实例化 NPUGraphRunner(model_runner) 会因缺少关键字参数报错。不过该 runner 仅在 NPU 后端使用，且使用方式已更新。
  3. 性能风险：无。自适应推测解码本身可提升吞吐，PR 确保了 NPU 上能正确启用图执行，预期性能为正。
- 影响：
  1. 用户影响：在 Ascend NPU 上使用 --speculative-adaptive 时，自适应推测解码功能可正常工作，带来吞吐提升和延迟下降（PR 展示整体吞吐 +10.7%，延迟 -10.2%）。
  2. 系统影响：修改仅影响 NPU 后端，CUDA 后端无变化。
  3. 团队影响：为 NPU 后端填补了关键能力缺口，使自适应推测解码在 NPU 上达到与 CUDA 类似的效果。 - 风险标记：NPU 专用路径，无测试文件变更

## 关联脉络

- PR #21599 [Speculative] Adaptive speculative decoding: 此 PR 是 #21599 的后续，为 NPU 后端添加自适应推测解码支持。