PR #25669 完整报告

sgl-project/sglang

[EPD][Perf] Async image preprocessing and cross-request ViT batching for encode_server

合并时间：2026-06-01 16:52

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25669

执行摘要

- 一句话：异步图像预处理提升 EPD 编码器吞吐
- 推荐动作：建议 EPD 相关开发者阅读本 PR，了解如何通过异步化在不改变模型输出的情况下显著提升编码器性能。线程池隔离、环境变量配置、以及根据 review 缩小范围与其他 PR 分工的协作方式值得学习。

功能与动机

EPD 编码服务器在处理并发请求时，图像预处理（PIL 调整大小、归一化）在 asyncio 事件循环上同步执行，导致所有预处理串行化，每个请求必须等待前一个完成。此外，预处理完成后每个请求独立调用 ViT 前向，未利用并发请求的批处理机会。本 PR 解决第一个问题：将预处理卸载到线程池，使预处理并行化，释放事件循环。

实现拆解

1. 在 encode_server.py 的 __init__ 中创建独立的 self.preproc_executor（ThreadPoolExecutor），大小由环境变量 SGLANG_ENCODER_PREPROC_WORKERS 控制（默认 8），与已有的 self.executor（处理 ZMQ 发送）分离，避免高并发下资源竞争。
2. 修改 _process_image_items、_process_video_items、_process_audio_items 三个方法，将 image_processor()、video_processor()、audio_processor.feature_extractor() 的同步调用改为 await asyncio.get_running_loop().run_in_executor(self.preproc_executor, functools.partial(...))，使预处理在线程中异步执行，不阻塞事件循环。
3. 在 sglang/srt/environ.py 的 Envs 类中添加环境变量 SGLANG_ENCODER_PREPROC_WORKERS = EnvInt(8)，便于用户配置线程池大小。
4. 根据 reviewer 反馈，移除了原先的 ViT 批处理逻辑及相关测试（test_encode_server_perf.py），因为 ViT 批处理已由 PR #25964 在另一层级实现，本 PR 专注于异步预处理，二者互补。

关键文件：

- python/sglang/srt/disaggregation/encode_server.py（模块 编码器；类别 source；类型 core-logic；符号 init, _process_image_items, _process_video_items, _process_audio_items）：核心变更文件，添加独立预处理线程池，修改三个预处理方法使其异步执行。

- python/sglang/srt/environ.py（模块 配置；类别 source；类型 configuration；符号 SGLANG_ENCODER_PREPROC_WORKERS）：新增环境变量 SGLANG_ENCODER_PREPROC_WORKERS，控制预处理线程池大小。

关键符号：init, _process_image_items, _process_video_items, _process_audio_items

关键源码片段

python/sglang/srt/disaggregation/encode_server.py

核心变更文件，添加独立预处理线程池，修改三个预处理方法使其异步执行。

```
# 在 __init__ 中，与 self.executor (ZMQ 发送 ) 分离，创建专用预处理线程池
self.preproc_executor = concurrent.futures.ThreadPoolExecutor(
    max_workers=envs.SGLANG_ENCODER_PREPROC_WORKERS.get()
)

# _process_image_items 使用 run_in_executor 异步执行图像预处理
async def _process_image_items(self, mm_items, model_preprocessor):
    if not (self.image_processor or model_preprocessor):
        raise ValueError("No image processor available")
    images = await self._flatten_and_load_images(mm_items)
    if model_preprocessor:
        return model_preprocessor(images, Modality.IMAGE, self.vision_config)
    image_config = self.vision_config.get("image", {})
    if self.model_type in ["kimi_k25", "kimi_vl"]:
        images = self._normalize_kimi_encoder_images(images)
    # 原先直接 return self.image_processor(...) 会阻塞事件循环
    # 现在通过 run_in_executor 在线程池中执行，不阻塞事件循环
    return await asyncio.get_running_loop().run_in_executor(
        self.preproc_executor,
        functools.partial(self.image_processor, images=images, **image_config),
    )
```

评论区精华

Review 讨论要点：

- ZhengWG 建议将异步预处理扩展到视频和音频，作者立即应用相同模式。
- ZhengWG 建议 ViT 批处理使用真正的批处理调用而非循环，作者改进后因 PR #25964 覆盖而删除整个块。
- ShangmingCai 认为默认线程数 32 过大，建议 4 或 8，作者改为 8 并补充性能说明。
- ShangmingCai 建议将环境变量迁移至 sglang.srt.environ 统一管理，作者照办。
- ShangmingCai 认为新增测试过于简单无实际价值，作者同意并删除。
- Apply async preprocessing to video/audio (correctness): 作者立即应用，统一了三种模态的处理。
- ViT batching design and subsequent removal (design): 删除 ViT 批处理，专注于异步预处理。

- Default preproc workers should be lower (performance): 作者改为 8，并在 PR 描述中补充性能说明。
- Move env var to environ module (design): 作者照做，改为 envs.SGLANG_ENCODER_PREPROC_WORKERS.get()。
- Trivial test should be removed (testing): 作者同意并删除测试文件。

风险与影响

- 风险：风险较低： 1) 线程池大小配置不当可能导致 CPU 资源争抢或预处理不足；默认值 8 经过基准测试验证，但仍需在不同负载下确认。 2) 与 ZMQ 发送线程池分离设计避免了资源竞争，但如果预处理线程中存在 GPU 操作（目前无），可能引入同步问题。 3) 移除了测试文件，新增功能无单元测试覆盖（仅依赖 CI 集成测试）。 4) 环境变量 SGLANG_ENCODER_PREPROC_WORKERS 需要补充文档。
  - 影响：对用户：使用 EPD 编码器时，图像 / 视频 / 音频预处理不再阻塞事件循环，高并发场景下吞吐提升约 23%，延迟降低 19%。对系统：增加一个默认 8 线程的线程池，内存开销不大。对团队：与 PR #25964 分工协作，本 PR 专注于预处理异步化，未来 ViT 批处理由另一 PR 提供。整体正面影响。
  - 风险标记：线程池资源竞争风险，默认值影响吞吐，缺少单元测试覆盖

关联脉络

- PR #25964 EncoderScheduler with cross-request ViT batching: 覆盖了本 PR 原本计划的 ViT 批处理，两者协同，本 PR 聚焦异步预处理。
- PR #22235 Async encode recv on scheduler side: 解决 EPD 管道中调度器侧轮询阻塞问题，与本 PR 互补。