PR #2072 完整报告

THUDM/slime

[docker] upgrade sglang to v0.5.13

合并时间：2026-06-14 21:09

原文链接：http://prhub.com.cn/THUDM/slime/pull/2072

执行摘要

- 一句话：升级 sglang 至 v0.5.13，新增 Glm5 跨层索引共享
- 推荐动作：建议团队仔细 review sglang.patch 的变更，特别是新增的超时逻辑和 draft model 支持，确保与 v0.5.13 的兼容性。跨层索引共享的设计值得阅读，可参考 glm-train-prod 的实现作为对比。

功能与动机

升级 sglang 到 v0.5.13 以利用新版本特性和修复；为 Glm5 模型引入跨层索引共享，允许部分层跳过昂贵的 top-k 计算，复用最近计算层的索引结果，从而降低推理时延和计算量。该设计参考了 glm-train-prod 的实现，并保持与原有 DSA 路径的向后兼容。

实现拆解

1. 升级 Docker 基础镜像

在 docker/Dockerfile 中将 SGLANG_IMAGE_TAG 从 v0.5.12.post1-cu129 改为 v0.5.13-cu129，docker/version.txt 同步更新版本号。

2. 重构 sglang.patch 补丁

docker/patch/latest/sglang.patch 进行了大量调整（+737/-1081），主要变更包括：

- 模型配置：为 Glm4MoeLiteForCausalLM 添加 Glm4MoeLiteForCausalLMNextN 的 draft model 转换，使其支持 speculative decoding。
- 解聚传输：在 KVArgs 中添加 aux_buffer_names 字段；在 DecodePreallocQueue 中加入 SGLANG_DISAGGREGATION_TRANSFER_TIMEOUT 超时保护，防止 bootstrapping 阶段永久挂起。
- 其他适配：更新 tokenizer 注册、KVClass 映射，引入 apply_prefill_timing_payload 等调试工具。

3. 添加 Glm5 跨层索引共享

在 slime_plugins/models/glm5/glm5.py 中：

- 新增模块级常量 _INDEXER_SUBMODULE_NAMES 定义索引器子模块名称（wq_b, wk, k_norm, weights_proj）。

- 新增 is_skip_topk_layer(layer_number, skip_topk_offset, topk_freq) 函数：判断当前层是否为跳过层，计算公式为 (max(layer_number - skip_topk_offset, 0) % topk_freq) != 0。
- 新增 source_compute_layer(layer_number, skip_topk_offset, topk_freq) 函数：从当前层向上查找最近的属于计算层的层号。
- 在 DSAMultiLatentAttention.__init__ 中读取 index_topk_freq 和 index_skip_topk_offset 配置，计算 skip_topk 标志和 _source_layer。
- 修改 forward 方法：当 index_share 启用且当前层为跳过层时，不执行 indexer 计算，改为从 packed_seq_params._dsa_index_share_topk_holder 中获取计算层的 top-k 结果。

4. 测试与配置

本次未新增独立测试文件，但现有测试可能受影响。补丁的适配通过 CI 镜像构建验证。

关键文件：

- slime_plugins/models/glm5/glm5.py（模块 模型插件；类别 source；类型 core-logic；符号 is_skip_topk_layer, source_compute_layer, _INDEXER_SUBMODULE_NAMES, DSAMultiLatentAttention.init）：核心模型文件，新增跨层索引共享功能，是本次 PR 最重要的功能变更。
- docker/patch/latest/sglang.patch（模块 补丁适配；类别 config；类型 infrastructure；符号 KVArgs, DecodePreallocQueue, apply_prefill_timing_payload, is_slime_profiling_enabled）：sglang 补丁文件，适配 v0.5.13 并新增多项功能，是 Docker 升级的关键部分。
- docker/Dockerfile（模块 Docker 构建；类别 infra；类型 configuration）：Docker 构建文件，升级 sglang 基础镜像标签。
- docker/version.txt（模块 版本记录；类别 docs；类型 documentation）：版本记录文件，同步更新以标记新镜像版本。

关键符号：is_skip_topk_layer, source_compute_layer, DSAMultiLatentAttention.init, DSAMultiLatentAttention.forward, DecodePreallocQueue.init, DecodePreallocQueue._receive_kv

关键源码片段

slime_plugins/models/glm5/glm5.py

核心模型文件，新增跨层索引共享功能，是本次 PR 最重要的功能变更。

```
# slime_plugins/models/glm5/glm5.py

# Names of indexer submodules on a DSA model with cross-layer index sharing.
# On "computing" layers these exist; on "skip" layers they are dropped.
_INDEXER_SUBMODULE_NAMES = ("wq_b", "wk", "k_norm", "weights_proj")


def is_skip_topk_layer(
    layer_number: int, skip_topk_offset: int, topk_freq: int
) -> bool:
```

```
    """判断当前层是否为跳过层（复用前一计算层的 top-k）。

    当 ``max(layer_number - skip_topk_offset, 0) % topk_freq == 0`` 时，
    该层为计算层；否则为跳过层。
    """
    return (max(layer_number - skip_topk_offset, 0) % topk_freq) != 0


def source_compute_layer(
    layer_number: int, skip_topk_offset: int, topk_freq: int
) -> int:
    """返回跳过层所复用的计算层编号。"""
    layer = layer_number
    while is_skip_topk_layer(layer, skip_topk_offset, topk_freq):
        layer -= 1
    return layer
```

docker/patch/latest/sglang.patch

sglang 补丁文件，适配 v0.5.13 并新增多项功能，是 Docker 升级的关键部分。

```
# docker/patch/latest/sglang.patch ( 关键变更节选 )

# KVArgs 新增 aux_buffer_names 字段
@dataclass
class KVArgs:
    kv_data_ptrs: List[int]
    kv_data_lens: List[int]
    kv_item_lens: List[int]
    aux_buffer_names: List[str] # 新增：辅助缓冲区名称列表
    aux_data_ptrs: List[int]
    aux_data_lens: List[int]
    aux_item_lens: List[int]

# Decode 预分配队列超时机制
class DecodePreallocQueue(DecodeHiCachePreallocMixin):
    def _receive_kv(self, rids_to_check=None):
        bootstrap_timeout = float(
            os.environ.get("SGLANG_DISAGGREGATION_TRANSFER_TIMEOUT", "600")
        )
        now = time.perf_counter()
        for i, (decode_req, poll) in enumerate(zip(self.queue, polls)):
            if rids_to_check is not None and decode_req.req.rid not in rids_to_check:
                continue
            if poll == KVPoll.Bootstrapping:
                entry_time = decode_req.req.time_stats.decode_prealloc_queue_entry_time
                if entry_time > 0 and now - entry_time > bootstrap_timeout:
                    # 超时日志并跳过
                    logger.error(
                        f"Decode prealloc timeout for request rank={self.tp_rank} "
```

```
                f"{decode_req.req.rid=} after {bootstrap_timeout}s"
            )
```

评论区精华

该 PR 没有公开的 review 评论，作者在提交信息中标记了两次 'bugfix' commit 用于修正，但未展开讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
  - sglang 版本升级风险：新版本可能引入 API 或行为变更，与现有部署配置不兼容。需确认 v0.5.13 与旧补丁的匹配性，本次补丁进行了大幅重写，需仔细 review。
  - 跨层索引共享正确性：is_skip_topk_layer 和 source_compute_layer 的逻辑依赖于 skip_topk_offset 和 topk_freq 的正确配置。若 offset 或 freq 取值不当（如负值或零），可能导致死循环或索引越界。代码中通过 getattr(config, ...) or 1 做了兜底，但缺少边界校验。
  - 性能风险：索引共享虽然减少计算量，但如果 skip 层数量过多，可能影响模型精度或收敛。该功能默认关闭（freq=1），需用户显式开启。
  - 补丁规模：sglang.patch 变更行数大（+737/-1081），涉及多个 sglang 组件，合并后可能与其他正在进行的 docker 相关 PR（如 #2070）产生冲突。
- 影响：
  - 用户：使用 Docker 镜像的用户将自动获得 sglang v0.5.13；使用 Glm5 模型且配置了 index_topk_freq 的用户可以启用跨层索引共享优化，降低推理延迟。
  - 系统：Docker 镜像构建将拉取新的基础镜像，镜像大小可能变化。版本号更新便于追踪。
  - 团队：需要验证补丁在 v0.5.13 下的正确性，并测试跨层索引共享的精度影响。
  - 风险标记：跨层索引共享缺少独立测试，sglang 补丁变更量大需仔细 review，核心路径变更影响模型正确性

关联脉络

- PR #2070 [docker] expose sglang load inflight details: 同为 Docker 相关 PR，涉及 sglang 补丁修改，可能与本 PR 共享同一套 Docker 基础设施。
- PR #2041 [docker] always re-register mooncake addr during offloading: 同样是修改 sglang.patch 的 Docker PR，与本 PR 的补丁变更范围重叠，需注意冲突合并。