

PR #32184 完整报告

sgl-project/sglang

[Fix] Reserve the mamba pool's +1 padding slot in the memory budget solve

合并时间: 2026-07-24 05:55

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/32184>

执行摘要

- 一句话: 修复 Mamba 内存预算遗漏 padding slot 的问题
- 推荐动作: 值得快速合并, 修复清晰且已验证。建议后续补充单元测试覆盖 padding slot 的预算计算逻辑。

功能与动机

Mamba 池在物理分配时每个 slot 多分配了一个 padding slot (索引 0, 用于 cuda-graph padding 批次), 但内存预算计算未将其纳入, 导致预算低于实际分配, 部分显存被隐藏占用。作者在 PR body 中明确描述了该问题, 并给出计算公式和实际日志验证 (tp8, fp32 ssm, D=8 时不足 0.42 GiB)。

实现拆解

1. 显式 `max_mamba_cache_size` 分支 (第 1713-1732 行): 将 `capped_reqs` 改为 `capped_reqs + 1`, 对应中间缓冲区多分配的 1 个 padding slot。
2. 禁用 radix cache 分支 (第 1733-1750 行): 将 `server_args.max_mamba_cache_size` 改为 `server_args.max_mamba_cache_size + 1`, 同样补齐 padding slot。
3. 自动比率求解分支 (有 spec 解码) (第 1766-1784 行): 分子预算减去 $\text{per_req} * (1 + D)$, 分母不变, 即补齐主状态和中间状态的 padding slot; 中间状态计算中再用 `capped_reqs + 1`。
4. 自动比率求解分支 (无 spec 解码) (第 1786-1788 行): 预算减去 `per_req` 后再整除, 即减去主状态的一个 padding slot。
5. 配套更新注释, 解释 padding slot 的来源和公式。
6. 未改动测试文件, 但作者已通过 `/rerun-test` 触发相关 Mamba 测试并全部通过。

关键文件:

- `python/sglang/srt/mem_cache/kv_cache_configurator.py` (模块 缓存配置; 类别 source; 类型 core-logic; 符号 `_handle_max_mamba_cache`): 所有预算计算逻辑所在文件, 是此次 PR 的唯一变更文件。

关键符号: `_handle_max_mamba_cache`

关键源码片段

python/sglang/srt/mem_cache/kv_cache_configurator.py

所有预算计算逻辑所在文件，是此次 PR 的唯一变更文件。

```
# 文件 : python/sglang/srt/mem_cache/kv_cache_configurator.py
```

```
# 方法 : _handle_max_mamba_cache ( 部分 )
```

```
# 分支 1: 用户显式设置了 max_mamba_cache_size
```

```
if server_args.max_mamba_cache_size is not None:
```

```
    # ... 省略 override 代码 ...
```

```
    if has_spec_dec:
```

```
        ratio = self._calculate_mamba_ratio()
```

```
        capped_reqs = min(
```

```
            server_args.max_running_requests // self.ps.attn_dp_size,
```

```
            server_args.max_mamba_cache_size // ratio,
```

```
        )
```

```
        intermediate_size = (
```

```
            config.mamba2_cache_params.mamba_cache_per_req
```

```
            * (capped_reqs + 1) # 原来是 capped_reqs, 现在 +1 预留 padding slot
```

```
            * server_args.speculative_num_draft_tokens
```

```
        )
```

```
        total_rest_memory = total_rest_memory - (intermediate_size / (1 << 30))
```

```
# 分支 2: 禁用 radix cache 时使用 max_running_requests
```

```
elif server_args.disable_radix_cache and server_args.max_running_requests is not None:
```

```
    # ... 省略 override 代码 ...
```

```
    if has_spec_dec:
```

```
        intermediate_size = (
```

```
            config.mamba2_cache_params.mamba_cache_per_req
```

```
            * (server_args.max_mamba_cache_size + 1) # 原来是 server_args.max_mamba_cache_size
```

```
            * server_args.speculative_num_draft_tokens
```

```
        )
```

```
        total_rest_memory = total_rest_memory - (intermediate_size / (1 << 30))
```

```
# 分支 3: 自动比率求解
```

```
else:
```

```
    # ... 省略前面代码 ...
```

```
    if has_spec_dec:
```

```
        ratio = self._calculate_mamba_ratio()
```

```
        D = server_args.speculative_num_draft_tokens
```

```
        # 分子减去 per_req*(1+D) 即为两个 padding slot 的代价
```

```
        server_args.override(
```

```
            "mamba_pool.memory_budget_spec",
```

```
            max_mamba_cache_size=int(
```

```
                (mamba_budget_bytes - per_req * (1 + D)) # 原来是 mamba_budget_bytes
```

```
                // (per_req * (1 + D / ratio))
```

```
            ),
```

```
        )
```

```
    # ... 省略中间代码 ...
```

```
intermediate_size = per_req * (capped_reqs + 1) * D # 原来是 capped_reqs
total_rest_memory = total_rest_memory - (intermediate_size / (1 << 30))
else:
    # 无 spec 时只减一个 per_req
    server_args.override(
        "mamba_pool.memory_budget",
        max_mamba_cache_size=int((mamba_budget_bytes - per_req) // per_req), # 原来是
        mamba_budget_bytes
    )
```

评论区精华

无 review 评论或讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。变更仅为公式调整，且经实际日志验证（tp8, fp32 ssm）预算与分配完全对齐。K 值略微下降（如 134->131），但并发度不变，用户无感知。但缺少直接针对此修复的单元测试。
- 影响：影响范围：仅影响 Mamba 模型的内存预算计算。对用户：修复了显存被隐藏占用的问题，可能在某些极端情况下避免 OOM。对系统：内存预算更精确，但 K 值微降，不影响正常运行。
- 风险标记：缺少测试覆盖，核心路径变更

关联脉络

- PR #30986 [UnifiedRadixCache][mamba] Fix mamba state corruption and slot leak when load_back aborts: 同为 Mamba 缓存相关的 bugfix，涉及同一缓存子系统。
- PR #31230 [Mamba] Add a per-path cap for cached states: Mamba 缓存的功能增强，也是同一模块的变更。