PR #23275 完整报告

sgl-project/sglang

fix: add back priorty as radix cache policy

合并时间：2026-04-21 01:04

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23275

执行摘要

- 一句话：在基数树缓存淘汰策略选项中重新添加 priority 策略。
- 推荐动作：建议快速浏览：
  - 对于核心开发者：这个 PR 值得关注，因为它涉及缓存淘汰策略的配置完整性。虽然变更简单，但确认了 priority 策略的官方支持状态。
  - 对于新开发者：可以学习如何维护配置常量和命令行参数的一致性，以及如何通过运行现有测试来验证配置变更。
  - 值得关注的设计决策：系统支持多种缓存淘汰策略（lru、lfu、slru、priority），这反映了对多样化工作负载需求的考虑。priority 策略的引入表明系统需要支持基于请求优先级的缓存管理。

功能与动机

从 PR 标题 "fix: add back priorty as radix cache policy" 和提交信息 "miss" 可以看出，这是一个修复性变更。动机是重新添加之前可能被误删或遗漏的 priority 策略选项到基数树缓存淘汰策略列表中，确保系统配置的完整性和一致性。虽然没有关联 Issue 或详细 PR body 说明，但从变更内容推断，priority 策略原本应该是支持的，但可能在之前的某个变更中被意外移除。

实现拆解

1. 更新策略选项常量：在 python/sglang/srt/server_args.py 文件中，修改 RADIX_EVICTION_POLICY_CHOICES 常量，将 "priority" 添加到选项列表中。 - 变更前：RADIX_EVICTION_POLICY_CHOICES = ["lru", "lfu", "slru"] - 变更后：RADIX_EVICTION_POLICY_CHOICES = ["lru", "lfu", "slru", "priority"] - 原因：确保系统配置支持基于优先级的缓存淘汰策略。
2. 更新命令行帮助文本：在同一文件中，更新 --radix-eviction-policy 参数的 help 文本，添加对 priority 策略的说明。 - 变更前：帮助文本仅描述 lru、lfu 和 slru 策略。 - 变更后：在原有描述基础上添加 "and 'priority' evicts lower-priority requests first." - 原因：为用户提供清晰的文档，说明 priority 策略的行为是优先淘汰低优先级请求。
3. 测试验证：从 PR 评论中看到，作者手动运行了相关测试（test/registered/unit/server_args/test_server_args.py 和 test/registered/unit/mem_cache/test_evict_policy.py）来验证修复。虽然没有直接修改测试文件，但通过运行现有测试确保配置变更不会破坏现有功能。

关键文件：

- python/sglang/srt/server_args.py（模块 服务器参数；类别 source；类型 configuration ；符号 RADIX_EVICTION_POLICY_CHOICES）：这是唯一的变更文件，包含了基数树缓存淘汰策略的配置选项定义和命令行参数设置。修复了策略选项列表，确保 priority 策略可用。

关键符号：未识别

关键源码片段

python/sglang/srt/server_args.py

这是唯一的变更文件，包含了基数树缓存淘汰策略的配置选项定义和命令行参数设置。修复了策略选项列表，确保 priority 策略可用。

```
# 基数树缓存淘汰策略的可选值列表
# 修复：将 "priority" 重新添加回选项列表中，确保支持基于优先级的淘汰策略
RADIX_EVICTION_POLICY_CHOICES = ["lru", "lfu", "slru", "priority"]

# ... 其他配置常量 ...

# 在 add_cli_args 函数中定义命令行参数
parser.add_argument(
    "--radix-eviction-policy",
    type=str,
    choices=RADIX_EVICTION_POLICY_CHOICES, # 使用更新后的选项列表
    default=ServerArgs.radix_eviction_policy,
    help="The eviction policy of radix trees. 'lru' stands for Least Recently Used, "
        "'lfu' stands for Least Frequently Used, 'slru' stands for Segmented Least Recently Used,
        "
        "and 'priority' evicts lower-priority requests first.", # 更新帮助文本，说明 priority
        策略的行为
)
```

评论区精华

本次 PR 没有 review 评论，讨论主要发生在 PR 评论中：

- 作者 ishandhanani 在评论中执行了 /rerun-test 命令，指定运行两个测试文件来验证修复。
- GitHub Actions 报告测试通过，显示两个测试在 ubuntu-latest 环境下成功运行。
- 作者进一步说明 CPU CI 当前有问题，但手动运行测试全部通过，并附上了 pytest 输出结果。
- 讨论焦点是验证修复的正确性，没有出现设计争议或技术分歧。
- 测试验证修复 (testing): 修复通过了相关测试，验证了配置变更的有效性。

风险与影响

- 风险：低风险：
  1. 配置兼容性风险：仅修改配置选项常量，不影响现有代码逻辑。只要底层实现确实支持 priority 策略，就不会引入运行时错误。
  2. 回归风险：由于只是恢复一个原本应该存在的选项，且通过了相关测试，回归风险较低。

3. 性能风险：无性能影响，纯配置变更。
4. 安全风险：无安全影响。

潜在风险：如果底层缓存实现并未真正实现 priority 策略的逻辑，那么添加这个选项可能只是表面支持，实际使用时可能无法按预期工作。但从 PR 标题使用 "add back" 来看，priority 策略应该是之前已实现但选项被误删。

- 影响：影响范围有限但重要：
  1. 对用户的影响：用户现在可以在启动服务器时使用 --radix-eviction-policy priority 参数来选择基于优先级的缓存淘汰策略。这对于需要区分请求优先级的场景（如高优先级请求应保留在缓存中更久）提供了支持。
  2. 对系统的影响：恢复了系统配置的完整性，确保所有支持的缓存淘汰策略都能通过命令行正确配置。
  3. 对团队的影响：这是一个小的配置修复，维护了配置选项与底层实现的一致性。团队需要注意这个变更，确保文档和配置示例同步更新。
  4. 影响程度：低到中。虽然变更很小，但涉及核心配置选项，对于依赖 priority 策略的用户来说是必要的修复。 - 风险标记：配置完整性修复 ，依赖底层实现

关联脉络

- PR #23202 [core] Always-on StreamingSession in UnifiedRadixCache: 都涉及基数树缓存（RadixCache）的配置或行为调整。PR 23202 重构了缓存中的 StreamingSession 集成，而当前 PR 修复了缓存的淘汰策略选项。
- PR #23145 integrate streaming session into UnifiedRadixCache: 都涉及 UnifiedRadixCache 的改进。PR 23145 将流式会话集成到缓存中，当前 PR 则修复缓存淘汰策略的配置选项。
- PR #22983 [KV-Events] Fix kv events events publishing for CP: 都涉及 KV（键值）缓存相关的修复。PR 22983 修复 KV 事件发布问题，当前 PR 修复缓存淘汰策略配置。