

PR #32710 完整报告

sgl-project/sglang

[Radix Cache] Add Rust TreeCore backend with shared parity tests

合并时间: 2026-09-01 00:26

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/32710>

执行摘要

- 一句话: 新增 Rust TreeCore 缓存后端与共享一致性测试套件
- 推荐动作: 值得精读的架构级 PR。重点学习: 1) FFI 边界设计——owned snapshot、tagged tuple 编解码、禁止迭代器跨 Rust 边界、inspection binding 与生产 binding 严格隔离 (wheel 内校验无 inspect_* 方法); 2) 双后端共享测试套件的一致性验证策略, 比后端各自复制测试更能暴露语义差异; 3) fail-fast 特性管理——不支持的组合在构造期显式拒绝并附 TODO(Jialin), 避免不完整语义静默运行; 4) TreeComponent trait 抽象让 Python 语义可以逐方法移植, 每个方法都带 Python 参考实现注释, 是跨语言移植的良好范例。

功能与动机

PR body 明确动机: 'Add an in-tree Rust implementation of UnifiedTreeCoreInterface for Unified Radix Cache while retaining Python cache orchestration, pools, and action application. The same cache-level unit suite now exercises both Python and Rust TreeCore implementations, exposing parity gaps through shared assertions rather than backend-specific test copies.' 即引入 Rust 后端以获得无 Python 节点对象遍历的高性能路径, 同时用共享测试套件保证与 Python 实现的语义一致性。此外, issue 讨论中 ishandhanani 与 dreamtalen 提出希望该 Rust 实现可作为独立 crate 用于 DynoSim 纯 Rust 仿真器, 降低 SGLang 与其仿真之间的语义漂移。

实现拆解

实现按 5 个步骤推进:

1. 新增 Rust 核心 crate 与 PyO3 绑定: 新增 rust/mem-cache crate, 含 node.rs (Node/NodeArena/KeyNamespace/ValueState 槽位)、unified_lru_list.rs (哨兵单元双向链表)、unified_tree_core.rs (match/insert/evict 主流程); 通过 PyO3 暴露 RustUnifiedTreeCoreBinding 等绑定类, Python 侧 extension.py 负责按环境变量选择加载。
2. Python adapter 与注册集成: 新增 adapter.py 的 RustUnifiedTreeCore 实现 UnifiedTreeCoreInterface, 核心工作是 tagged tuple 编解码 (cache_action_from_tagged/_kv_event_from_tagged/_transfer*_binding 等), Rust 只回传 (tag, ...) 元组和 owned 数组, 不跨 FFI 传迭代器; 构造器对 session radix cache、C128、自定义组件等不支持配置直接抛 ValueError。

3. 组件驱动抽象与语义移植: components/mod.rs 定义 TreeComponent trait (match validator、LRU refresh、节点分裂重分配、逐出、锁路径、insert 提交) , full.rs/swa.rs/mamba.rs 各自实现组件语义; 随后按 27 个独立 commit 分批移植 #33091 (分配感知逐出)、#36317 (Full-only load-back 所有权)、#31479 (KV 事件合并)、#31902 (拒绝 unsafe prefetch-refill)、#33639 (增量 Mamba backup)、#34808 (DCP 感知 checkpoint grid)、#32228 (部分 lock replay)、#30827 (cache-salt 命名空间)、#37091 (adopted insert ranges) 等历史行为。
4. 边界与失败语义: typed boundary errors、stale-handle 操作抛 KeyError 且不污染核心、poisoned-core fail-closed; 生产 binding 不暴露 inspect_* 方法, 测试 inspector 独立成模块且不进 wheel。
5. 测试 / 构建 / CI 配套: 共享 UnifiedRadixCache 套件 2436 例双后端通过, Rust 原生测试 831 例、adapter 集成测试 100 例; setup.py 接入 cargo workspace 元数据与扩展构建; prepare_sglang_wheel.py 新增 auditwheel repair 后的 smoke test (校验加载路径、类名、无 inspect 方法); CI 新增 native mem-cache 与 release-wheel 覆盖, 并用 per-run target dir 解决 CARGO_TARGET_DIR 并发竞态。

关键文件:

- python/sglang/srt/mem_cache/rust_tree_core/adapter.py (模块 适配层; 类别 source; 类型 dependency-wiring; 符号 RustUnifiedTreeCore, _radix_key_buffer, _kv_event_from_tagged, _cache_action_from_tagged): Python 侧适配层, RustUnifiedTreeCore 实现 UnifiedTreeCoreInterface, 承担 tagged tuple 编解码、句柄校验、不支持配置 fail-fast, 是双后端注册与语义对齐的枢纽。
- rust/mem-cache/src/components/mod.rs (模块 组件框架; 类别 source; 类型 core-logic; 符号 TreeComponent, component_type, needs_incremental_backup, refresh_lru): Rust 侧组件驱动框架, 定义 TreeComponent trait, 每个方法与 Python tree_component.py 一一对应并带参考实现注释, 是语义移植的骨架。
- rust/mem-cache/src/components/mamba.rs (模块 Mamba 组件; 类别 source; 类型 core-logic; 符号 MambaComponent, new, least_common_multiple, has_value): Mamba 组件驱动, 含 checkpoint grid 对 LCM 计算、tombstone refill、per-path state cap 逐出等最复杂的移植语义, 是所有组件中行为分支最多的一个。
- rust/mem-cache/src/components/full.rs (模块 FULL 组件; 类别 source; 类型 core-logic; 符号 FullComponent, component_type, create_match_validator, finalize_match_result_in_tree_core): FULL 组件驱动, 实现 match validator、host hit length 统计、节点分裂重分配与设备逐出堆 / 游标逻辑, 是最基础的组件路径。
- rust/mem-cache/src/components/swa.rs (模块 SWA 组件; 类别 source; 类型 core-logic; 符号 SwaComponent, new, maybe_split_leaf_for_swa_lock_, has_value): SWA 组件驱动, 实现滑动窗口锁、窗口内 MRU 刷新、prefetch 提交与 tombstone 填充, 是本 PR 中窗口语义最精细的组件。
- rust/mem-cache/src/node.rs (模块 节点结构; 类别 source; 类型 core-logic; 符号 KeyNamespace, KeyNamespaceRef, KeyNamespaceData, key_namespace_hash): Rust 侧节点数据结构核心, 定义 KeyNamespace (cache-salt 命名空间) 哈希方案、per-component 值槽位数组、LRU 计数器与 hash chain。

- python/sglang/srt/mem_cache/buffer_mode/pipeline.py (模块 备份管线; 类别 source; 类型 dependency-wiring; 符号 _UnifiedBackupIntent, _backup_parent_covered, enqueue_backup_intent, _build_aux_staging_transfers) : HiCache buffer 备份管线从节点对象访问重构为 BufferBackupSnapshot 快照与句柄校验, 是 Python 侧配合 Rust 后端的关键改动。
- scripts/release/prepare_sglang_wheel.py (模块 打包脚本; 类别 source; 类型 dependency-wiring; 符号 _single_wheel, _metadata, _smoke_test_tree_core, _write_github_outputs) : 新增 wheel repair 与 smoke test, 校验生产 wheel 内不含 inspection binding、Rust TreeCore 模块加载路径正确, 是发布链路的关键保障。
- test/registered/unit/mem_cache/test_rust_tree_core_integration.py (模块 集成测试; 类别 test; 类型 test-coverage; 符号 _tree_core, _key, _insert, _binding) : 驱动真实编译的 Rust mem_cache 扩展的集成测试, 覆盖空树匹配、插入回读、stale handle 不污染核心、DFS 权重排序等关键契约。

关键符号: RustUnifiedTreeCore.init, cache_action_from_tagged, _kv_event_from_tagged, _insert_step_from_binding, _match_result_from_binding, TreeComponent::create_match_validator, TreeComponent::refresh_lru, TreeComponent::commit_insert_component_data, TreeComponent::evict_component, TreeComponent::redistribute_on_node_split, FullComponent::finalize_match_result_in_tree_core, FullComponent::evict_device_next_node, MambaComponent::commit_insert_component_data, MambaComponent::evict_excess_path_states, SwaComponent::commit_prefetch, SwaComponent::maybe_split_leaf_for_swa_lock_, KeyNamespaceRef::new, UnifiedLRUList::reset_node_and_window_ancestors_mru, snapshot_buffer_backup, validate_buffer_backup, dfs_weight_order

评论区精华

review 中的核心讨论集中在四个方面:

1. 独立 crate 用于仿真: ishandhanani 提出 'being able to use this rust radix tree in simulation studies without needing to pull in libtorch or other deps'; dreamtalen 补充 DynoSim 是纯 Rust 离散事件仿真器, 'If the Rust TreeCore in this PR could be made available as a standalone crate, we could use it as DynoSim's radix tree. The main benefit would be reducing semantic drift between SGLang and its simulation.' 该需求未在本 PR 落地, 作为后续协作方向。
2. 双后端等价性测试: ispobock 建议 'drive the same op sequence through both and assert equal device_indices, LRU eviction order, and CacheActions', 最终以共享 UnifiedRadixCache 测试套件 + inspector 快照方式实现, 而非后端各自复制测试。
3. 重复 backup 缺陷: alphabetc1 指出 'A double-backup issue' 并附参考修复, 已通过独立 commit 'fix: prevent duplicate full kv backups' 修复。
4. CI 构建竞态: ispobock 定位到 ci_install_dependency.sh 共享 CARGO_TARGET_DIR, CUDA job 的 rm -rf 会打断同机 Rust 编译, 最终采用 'building in a per run target dir under RUNNER_TEMP' 解决。

- Rust TreeCore 独立 crate 用于 DynoSim 仿真 (design): 未在本 PR 落地, 作为后续协作方向; PR 当前仍通过 PyO3 绑定 tch (libtorch)。
- Python/Rust 后端等价性测试 (testing): 以共享 UnifiedRadixCache 测试套件 (2436 例) + Python/Rust inspector 快照方式实现, 而非后端各自复制测试。
- 重复 Full KV backup 问题 (correctness): 已通过独立 commit 'fix: prevent duplicate full kv backups' (作者 alphabetc1) 修复。
- Cargo 缓存目录并发竞态 (other): 最终 commit 'build rust exts in a per run target dir' 解决; 此前也加了 'guard cargo target drop with a lock'。

风险与影响

- 风险:
 1. 语义漂移风险 (高): Rust 后端是基于 Python 行为快照的移植, Python 侧后续新增语义 (session、C128、external linker、custom component) 尚未移植; 共享测试套件覆盖 2436 例, 但未覆盖这些组合, 后续 Python 侧改动需同步移植并有回归风险。
 2. 部署兼容性风险 (中): PR 明确 CUDA 13.4 / PyTorch 2.15 未启用, 打包版本限定 PyTorch 2.11-2.13; AMD/MLX/NPU 生产部署未验证, 只能 fail-fast。
 3. 构建链复杂性 (中): tch 0.24 依赖 Torch 2.13 兼容头、cargo 并发构建、wheel repair 链路 (prepare_sglang_wheel.py 新增 smoke test) 都可能成为发布阻塞点。
 4. 回归面 (中): buffer_mode/pipeline.py 从 UnifiedTreeNode 对象访问改为 BufferBackupSnapshot 快照 + validate_buffer_backup, 影响 HiCache 备份管线; unified_tree_core.py 新增接口方法, Python 后端行为也有调整。
 5. 性能未验证: PR 动机包含性能, 但未给出基准数据, Rust 后端的实际收益需后续 benchmark 确认。- 影响: 对用户: 默认路径仍是 Python 后端, Rust 后端通过环境变量 /registry 选择, 不支持的配置会显式报错而非静默错误, 现有部署无破坏性变更。对系统: 缓存核心新增一套无 Python 节点对象遍历的高性能后端候选, 为 PD-decode HiCache、DFS 权重调度等热路径去除 Python 开销铺路。对团队: 后续缓存语义迭代需同步维护双后端, 共享测试套件降低回归风险但增加功能开发成本; CI 增加 Rust 构建、wheel smoke test 等环节。对外部社区: 为 DynoSim 等 Rust 仿真生态提供了可复用 crate 的基础。- 风险标记: 核心缓存路径双后端化, 大规模语义移植需持续对齐, session/C128/linker 组合未覆盖, cargo 构建并发与缓存竞态, AMD/MLX/NPU 部署未验证, 性能收益未附基准数据

关联脉络

- PR #37151 [Unified Cache Linker][3/N]: Add backend-independent linker core: PR body 明确将 external cache linker (#37091/#37151) 列为 Rust TreeCore 推迟项: Rust 保留 adopted insert ranges 但未实现 linker 持久化与转移语义, 启用时显式拒绝并附 TODO(Jialin)。
- PR #37164 [mem_cache] Move mamba state and retraction_backup into ReqKvInfo: PR body 说明已同步该 PR 的 Mamba request-state 访问变更, 避免双后端状态访问语义分叉; 本 PR 的 commit 'Follow ReqKvInfo Mamba ownership' 即对应此工作。