PR #31920 完整报告

sgl-project/sglang

[HiCache] Add model-aware key isolation to Mooncake Store

合并时间：2026-07-23 10:20

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31920

执行摘要

- 一句话：Mooncake Store 新增模型感知的 key 隔离机制
- 推荐动作：建议阅读该 PR，尤其是关注 HiCache 多模型隔离的读者。核心设计决策 config_prefix 组合方案清晰，但有一个未解决的 review 建议（过滤空分割片段），值得后续跟进补丁。

功能与动机

在共享 Mooncake Store 的多租户或多模型场景中，不同模型可能使用相同的逻辑 key（如 'page0'），导致 key 冲突和数据污染。本 PR 通过引入模型名称作为 key 的一部分，实现租户和模型的隔离，无需为每个模型部署独立的 Mooncake Store。

实现拆解

1. 在 MooncakeStore.__init__ 中构造 config_prefix（mooncake_store.py）：将原有的 extra_backend_tag 属性替换为 config_prefix，由两部分组成：
   - 如果 extra_config 中包含 extra_backend_tag，则将其加入前缀列表。
   - 如果 storage_config.model_name 非空，则将其斜杠替换为连字符（防止 key 解析歧义）后加入前缀列表。
   - 最终将各部分用下划线连接作为 self.config_prefix。
2. 修改 _tag_keys 方法（mooncake_store.py）：将原有的 self.extra_backend_tag 引用改为 self.config_prefix，当 config_prefix 为 None 时不加前缀，否则在每个 key 前添加 {config_prefix}_ 前缀。
3. 更新多处 key 操作入口的注释（mooncake_store.py 的 batch_get_v1、batch_set_v1、batch_exists 方法）：将所有引用 extra_backend_tag 的注释改为引用 config_prefix。
4. 调整测试基础设施（test_mooncake_group_semantics.py）：将 _make_config 和 _make_store 中的 model_name 默认值从固定的 'test' 改为可配置的参数，使得测试可以模拟不同模型。
5. 新增隔离性测试（test_mooncake_group_semantics.py）：test_model_names_isolate_the_same_logical_key 创建两个不同 model_name 的 store，写入相同的逻辑 key 'page0'，然后断言实际存储的 key 包含不同的前缀且集合不重叠。

关键文件：

- python/sglang/srt/mem_cache/storage/mooncake_store/mooncake_store.py（模块 存储层；类别 source；类型 core-logic；符号 init, _tag_keys, batch_get_v1, batch_set_v1）：核心变更所在，将 extra_backend_tag 升级为组合 config_prefix，并修改 _tag_keys 以使用新前缀。
- test/registered/unit/mem_cache/test_mooncake_group_semantics.py（模块 测试；类别 test；类型 test-coverage；符号 _make_config, _make_store, test_model_names_isolate_the_same_logical_key）：新增 test_model_names_isolate_the_same_logical_key 测试，验证不同 model_name 的 store 写入相同逻辑 key 时实际 key 隔离。同时调整测试辅助函数以支持传入 model_name。

关键符号：_tag_keys, _make_config, _make_store, test_model_names_isolate_the_same_logical_key

关键源码片段

python/sglang/srt/mem_cache/storage/mooncake_store/mooncake_store.py

核心变更所在，将 extra_backend_tag 升级为组合 config_prefix，并修改 _tag_keys 以使用新前缀。

python/sglang/srt/mem_cache/storage/mooncake_store/mooncake_store.py # 在 __init__ 中构造 config_prefix # 替换原来的 extra_backend_tag 逻辑 self.config_prefix = None config_prefix_parts = [] # 如果配置了 extra_backend_tag，则将其作为前缀的一部分 if extra_config and extra_config.get("extra_backend_tag") is not None: config_prefix_parts.append(str(extra_config["extra_backend_tag"])) # 如果配置了 model_name，则将斜杠替换为连字符后加入前缀 if storage_config is not None and storage_config.model_name: model_name = "-".join(storage_config.model_name.split("/")) config_prefix_parts.append(model_name) if config_prefix_parts: self.config_prefix = "_".join(config_prefix_parts) logger.info(f"Using Mooncake config prefix:{self.config_prefix}") # _tag_keys 方法使用 config_prefix 而非 extra_backend_tag def _tag_keys(self, keys: List[str]) -> List[str]: if self.config_prefix is None: return keys # 在原始 key 前添加 config_prefix 和分隔符，实现租户隔离 return [f"{self.config_prefix}_{key}" for key in keys]

test/registered/unit/mem_cache/test_mooncake_group_semantics.py

新增 test_model_names_isolate_the_same_logical_key 测试，验证不同 model_name 的 store 写入相同逻辑 key 时实际 key 隔离。同时调整测试辅助函数以支持传入 model_name。

```
# test/registered/unit/mem_cache/test_mooncake_group_semantics.py

def test_model_names_isolate_the_same_logical_key(self):
    # 创建两个不同 model_name 的 store
    store_a, fake_store_a = _make_store(
        enable_group_semantics=False, model_name="org/model-a"
    )
    store_b, fake_store_b = _make_store(
        enable_group_semantics=False, model_name="org/model-b"
```

```
)
store_a.register_mem_pool_host(FakeHostKVCache(objects_per_page=2))
store_b.register_mem_pool_host(FakeHostKVCache(objects_per_page=2))

# 写入相同的逻辑 key 'page0'
self.assertEqual(store_a.batch_set_v1(["page0"], torch.tensor([0])), [True])
self.assertEqual(store_b.batch_set_v1(["page0"], torch.tensor([0])), [True])

# 验证实际存储的 key 包含不同的 model 前缀且不重叠
keys_a = fake_store_a.batch_put_calls[0]["keys"]
keys_b = fake_store_b.batch_put_calls[0]["keys"]
self.assertEqual(keys_a, ["org-model-a_page0_0_k", "org-model-a_page0_0_v"])
self.assertEqual(keys_b, ["org-model-b_page0_0_k", "org-model-b_page0_0_v"])
self.assertTrue(set(keys_a).isdisjoint(keys_b))
```

评论区精华

Gemini Code Assist 机器人提出了一条中等优先级的建议：在拼接 model_name 时，应该过滤掉分割后可能的空片段，防止出现连续的连字符或前导 / 尾随连字符。例如，如果 model_name 为 'org//model'，直接 / 分割后会产生空字符串，导致前缀中出现 'org--model'。建议使用生成器表达式 "-".join(p for p in storage_config.model_name.split("/") if p)。该建议未被采纳或回复，PR 已合并。

- 模型名称中的空片段过滤 (correctness): 未明确回应，PR 已合并。潜在风险未消除。

风险与影响

- 风险：相对安全，但存在以下风险点：
  1. key 格式变更导致兼容性问题：如果已有部署依赖 extra_backend_tag 前缀格式，升级后 key 会多出模型名称部分，可能导致缓存无法命中或数据读取失败。不过该功能属于 HiCache 实验性特性，影响可控。
  2. model_name 中有特殊字符：当前仅处理斜杠替换为连字符，未过滤其他字符，可能在 Mooncake Store 内部 key 系统中引发解析问题。机器人已提及空片段问题，但未修复。
  3. 测试覆盖有限：仅测试了单 token 的场景，未覆盖多 key、不同 tp_rank 等复杂情况。 - 影响：影响范围仅限于启用 HiCache 并使用 Mooncake Store 的部署，尤其是共享存储的多模型场景。变更后，同一 Mooncake Store 实例可以安全地为多个模型提供服务而不会 key 冲突，降低了运维成本。对于单模型用户，如果之前未使用 extra_backend_tag，则行为不变（config_prefix 为 None）。若有使用 extra_backend_tag，则 key 前缀会额外追加 model_name，需注意升级不一致可能导致的缓存短暂失效。 - 风险标记：配置兼容性，未处理 review 反馈，测试覆盖有限

关联脉络

- PR #31250 [XPU][GDN] add XPU path for causal_conv1d_fn and causal_conv1d_update: 同为 HiCache/Mooncake 相关 PR，涉及 GDn backend 的键隔离（extra_backend_tag 的使用）。

- PR #31863 [NPU]remove duplicate code: 同样修改了 GDN backend，涉及 ssm_states 和键隔离逻辑，与本 PR 的 config_prefix 设计潜在关联。
- PR #32029 [Fix] Unify pinned host pool release on graceful shutdown: 同为 HiCache 相关 PR，修改了 memory pool 和前缀缓存等组件，可能与本 PR 的存储层变更交互。