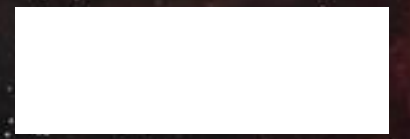
KEYSPACE

2025 / 北京

Valkey Hash Field Expiration 特性解读

陈嘉祥

阿里云研发



Background The Horsehead Nebula and its surroundings. The reflection nebula NGC 2023 in the bottom left corner. / Stephanh / License: CC BY 4.0

Valkey 过期机制

- 通过主哈希表实现 key → value 的索引，通过过期哈希表实现 key → ttl 的索引
- 过期机制：
  - 被动过期：访问某个 key 前，会检查过期，删除已过期 key
  - 主动过期：周期性扫描过期哈希表，删除已过期的 key
- 局限：只支持 key 级别设置过期时间，Hash、Set 等可能包含很多 field 的类型，无法单独为某个 field 设置过期时间

HFE 应用：用户多设备登陆管理

String Expiration

Key	Value	Expire
User_1_phone	Token	TTL
User_1_pad	Token	TTL
User_1_pc	Token	TTL
User_2_phone	Token	TTL
User_2_pad	Token	TTL
User_2_pc	Token	TTL
…	…	…

Hash Field Expiration

Key	Field	Value	Expire
User_1	phone	Token	TTL
	pc	Token	TTL
	pad	Token	TTL
User_2	phone	Token	TTL
	pc	Token	TTL
	pad	Token	TTL
…	…	…	…

Valkey

HFE 基本用法

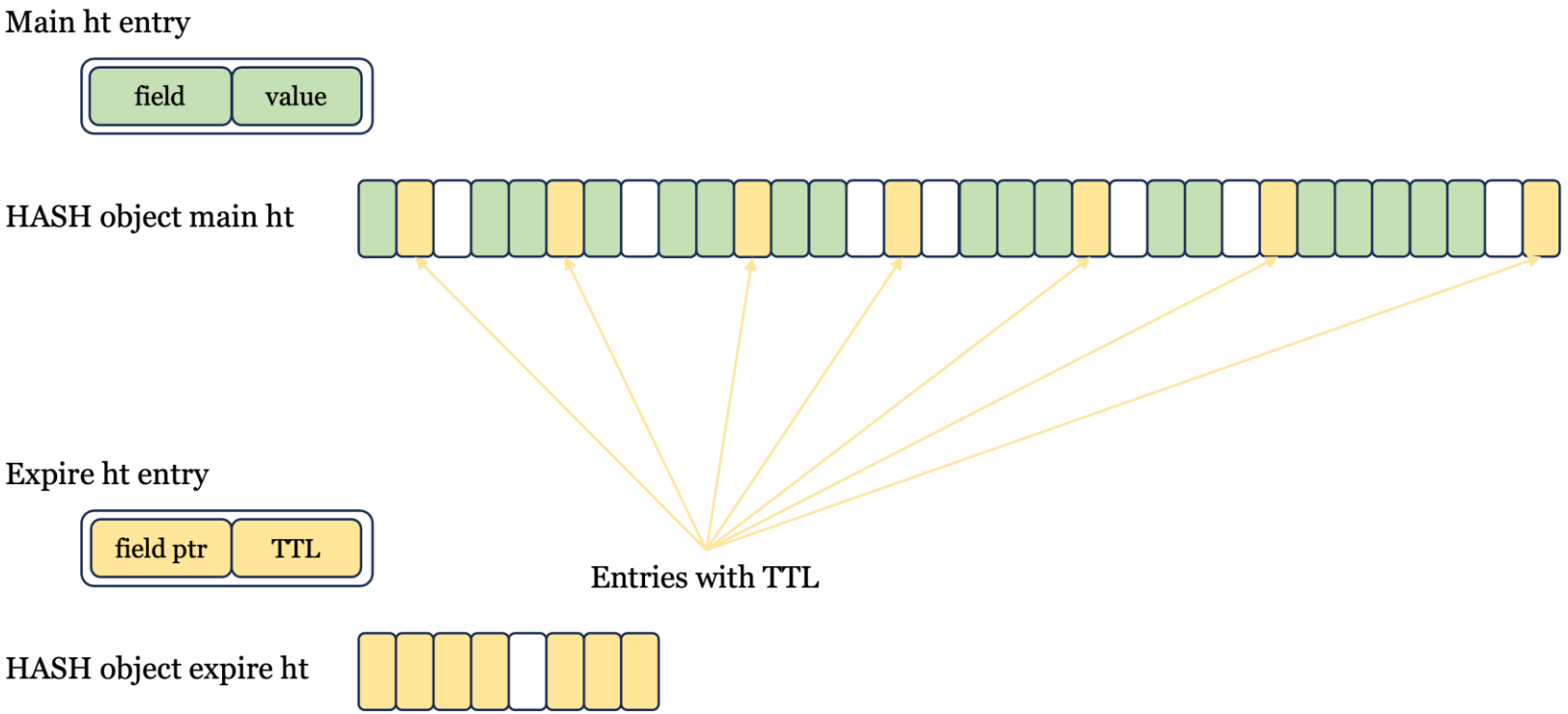
- 设置 field 过期时间：
  - HSETEX key [NX | XX] [FNX | FXX] [EX seconds | PX milliseconds | EXAT unix-time-seconds | PXAT unix-time-milliseconds | KEEPTTL] FIELDS numfields field value [field value ...]
  - HEXPIRE/HPEXPIRE key seconds/milliseconds [NX | XX | GT | LT] FIELDS numfields field [field ...]
  - HEXPIREAT/HPEXPIREAT key unix-time-seconds/milliseconds [NX | XX | GT | LT] FIELDS numfields field [field ...]
- 删除过期时间：
  - HPERSIST key FIELDS numfields field [field ...]
  - HSET key field value [field value ...]
- 查询过期时间：
  - HTTL/HPTTL key FIELDS numfields field [field ...]
  - HEXPIRETIME/HPEXPIRETIME key FIELDS numfields field [field ...]

HFE 设计与核心挑战

- 尽可能小的内存膨胀
- 保持 API 高效
  - 保持原有哈希 API 高效
  - 保证过期时间的维护操作高效
- 高效的过期机制，避免已过期 field 占用大量内存

数据结构选型：Hash + Hash

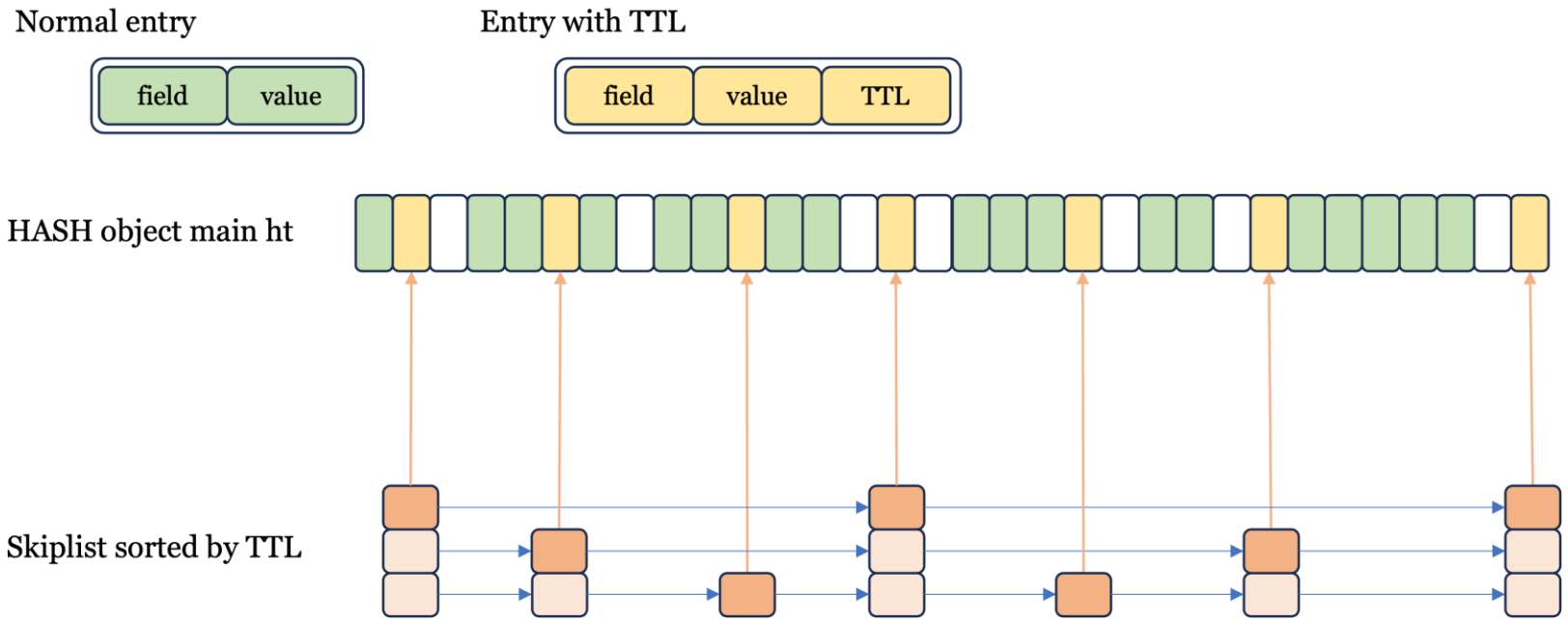
- 优点：过期时间操作 O(1) 复杂度、内存开销较低
- 缺点：过期效率低



Valkey

数据结构选型：Hash + Skiplist

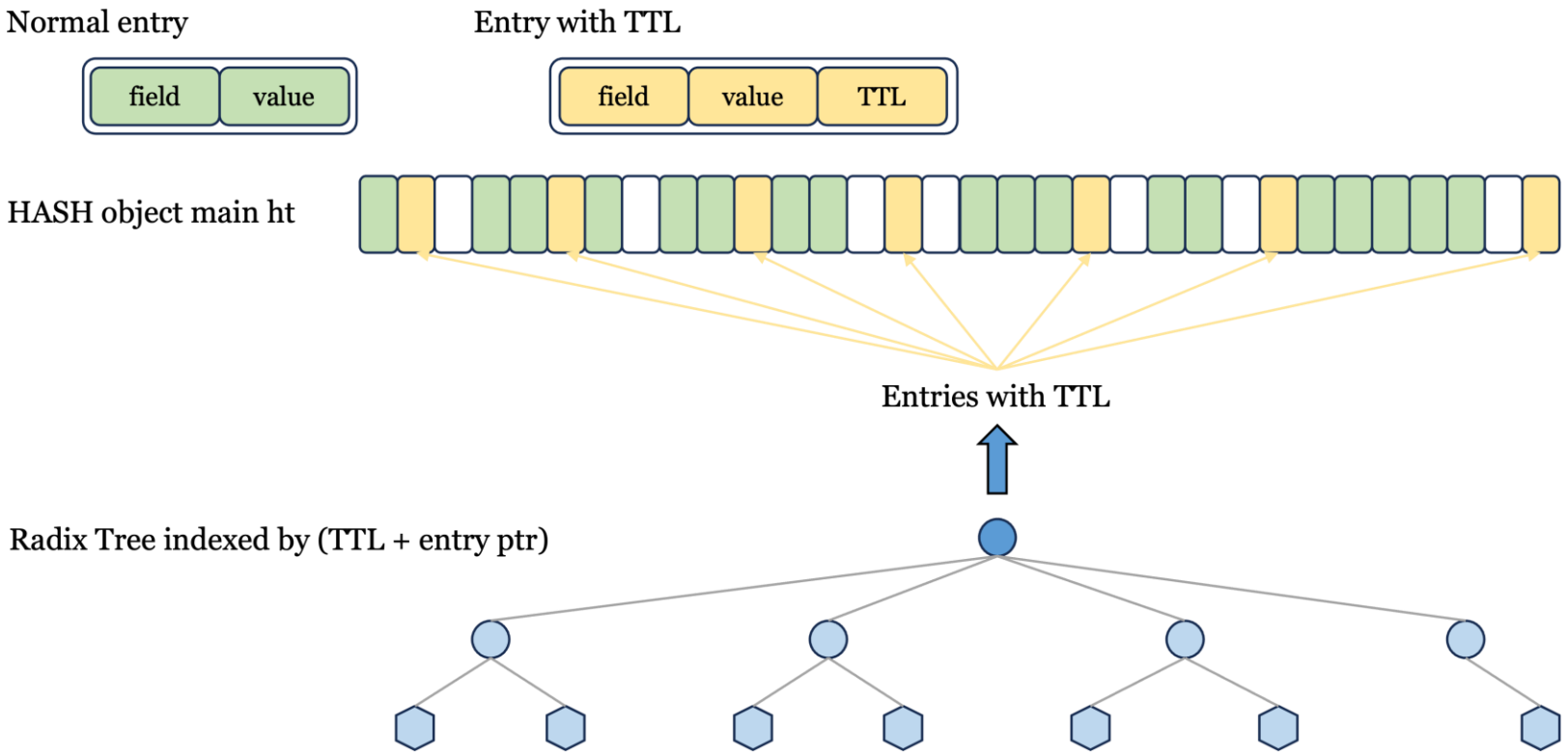
- 优点：过期效率高、内存开销可控
- 缺点：过期时间操作 O(logn) 复杂度



Valkey

数据结构选型：Hash + Radix Tree

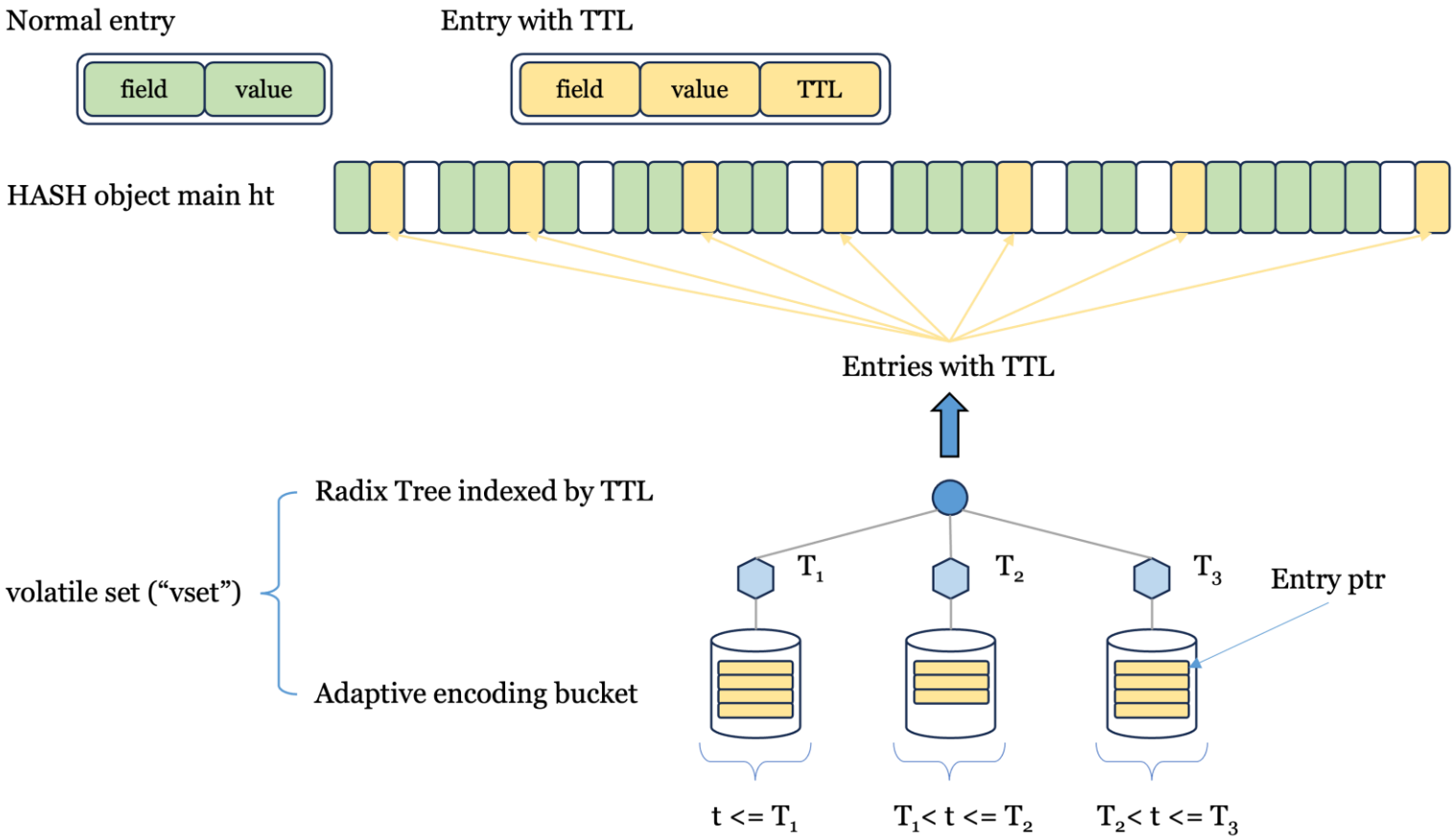
- 优点：过期时间操作 O(1) 复杂度、过期效率高
- 缺点：内存开销很大



Valkey

数据结构选型：Hash + Vset

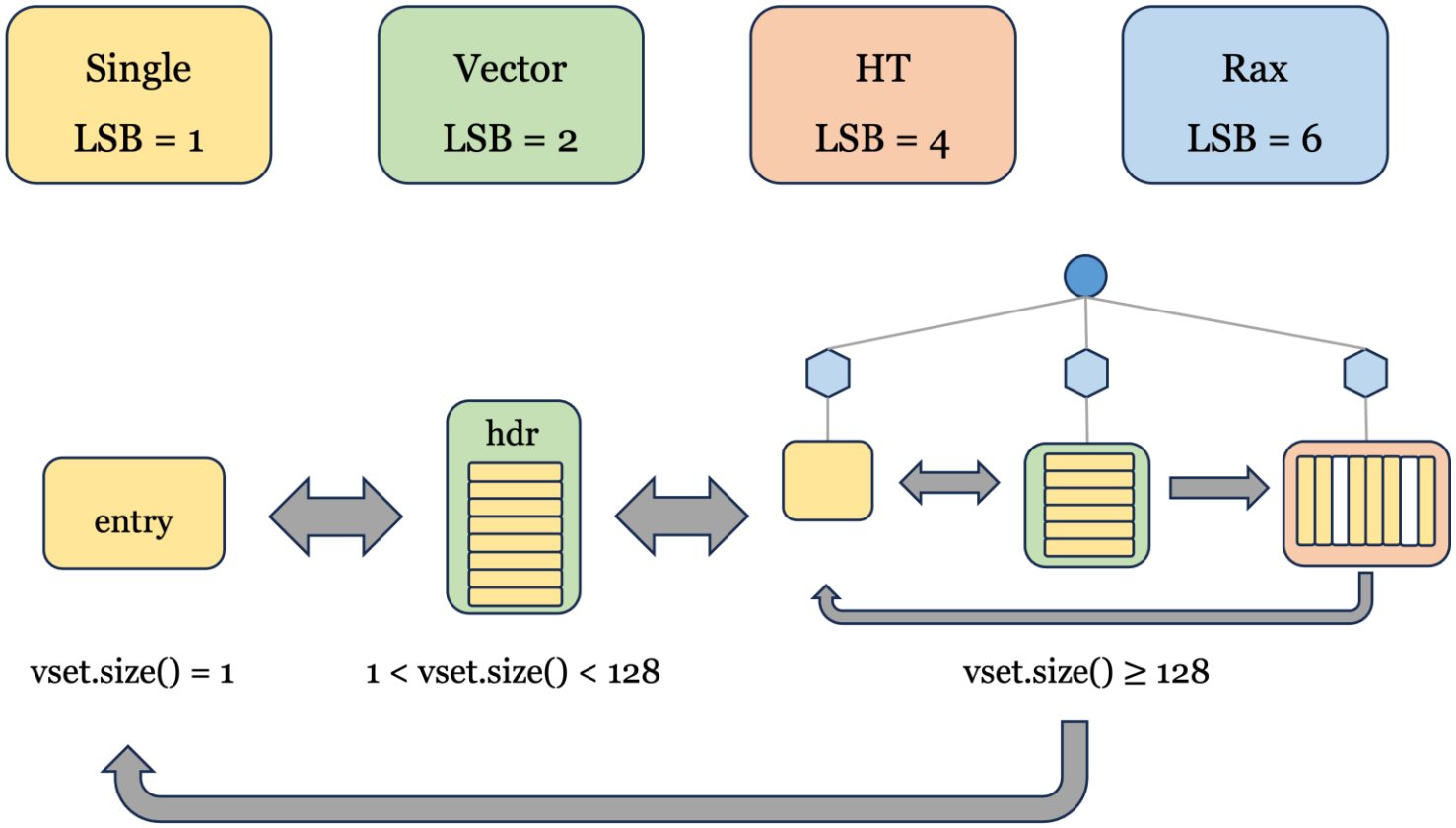
- 优点：内存开销小、过期时间操作 O(1) 复杂度、过期效率较高



Valkey

Adaptive Encoding

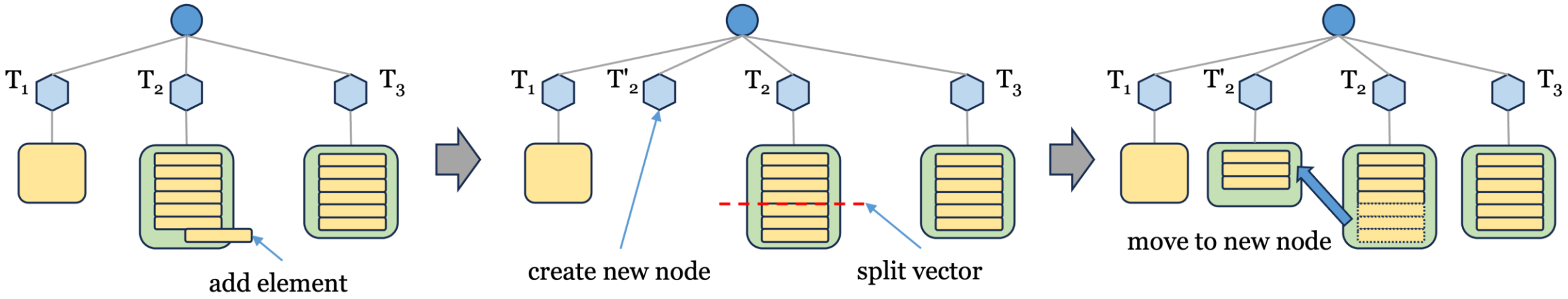
- 根据元素数量和过期时间分布选择具体编码方式，节省内存



Valkey

Vset 添加元素

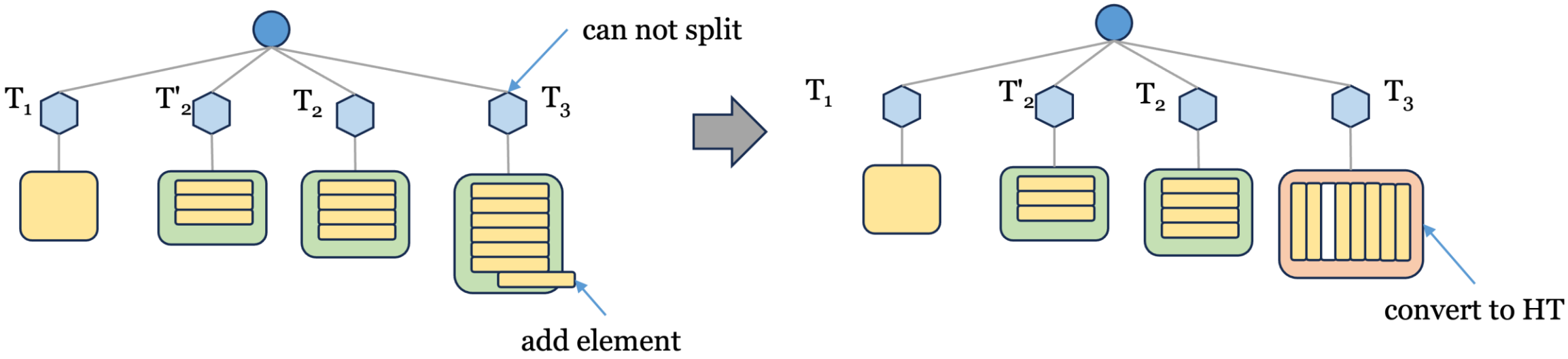
- Single: 直接赋值
- Vector: 保持有序插入
- Rax: 找到第一个 `rax` 时间戳大于或等于 `field` 过期时间的桶，执行插入操作：



Valkey

Vset 添加元素

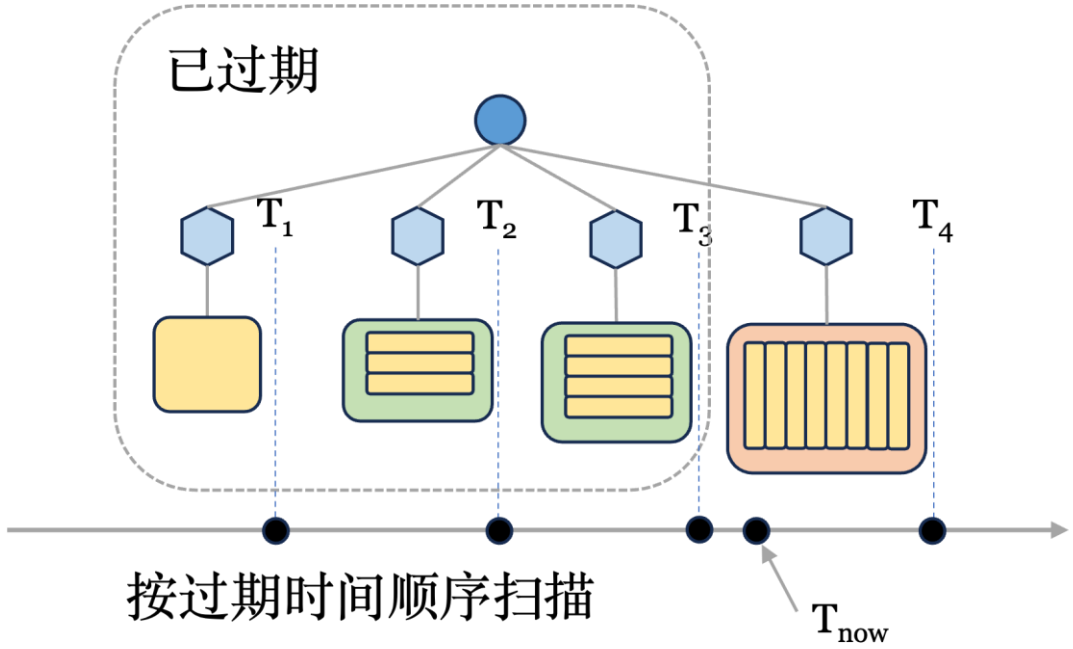
- Single: 直接赋值
- Vector: 保持有序插入
- Rax: 找到第一个 `rax` 时间戳大于或等于 `field` 过期时间的桶，执行插入操作：



Valkey

主动过期机制

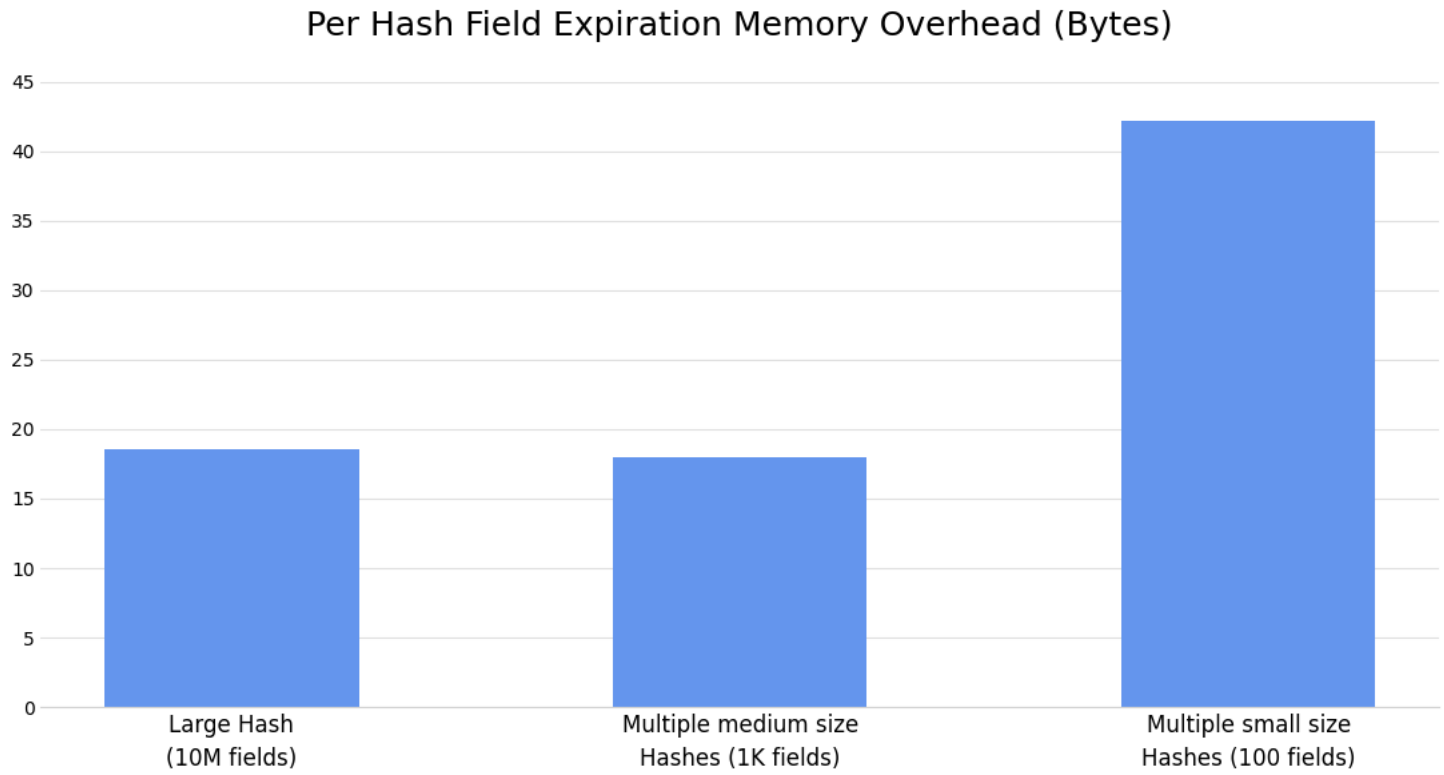
- 额外维护一个 Field 过期哈希表保存所有带有 field 过期的 Hash obj
- 周期性扫描 Field 过期哈希表，执行每个 Hash obj 内部的主动过期
- Key 级别过期和 Field 级别过期交替进行



Valkey

基准测试：内存占用

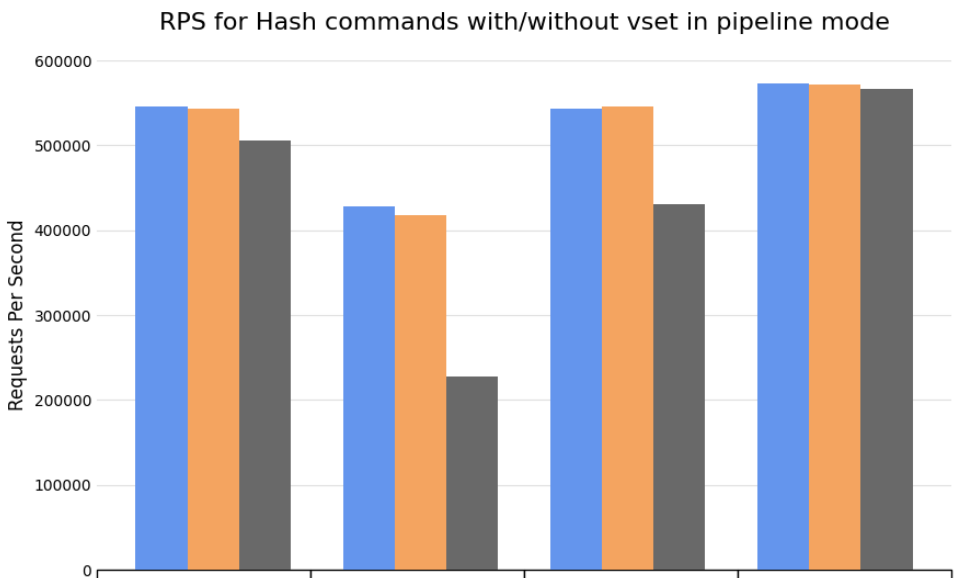
- 每个带过期的 field 至少增加 16B(8 ttl + 8 entry ptr) 内存
- 小 Hash 无 field 过期可以使用 listpack 编码，有 field 过期会直接转为 HT



Valkey

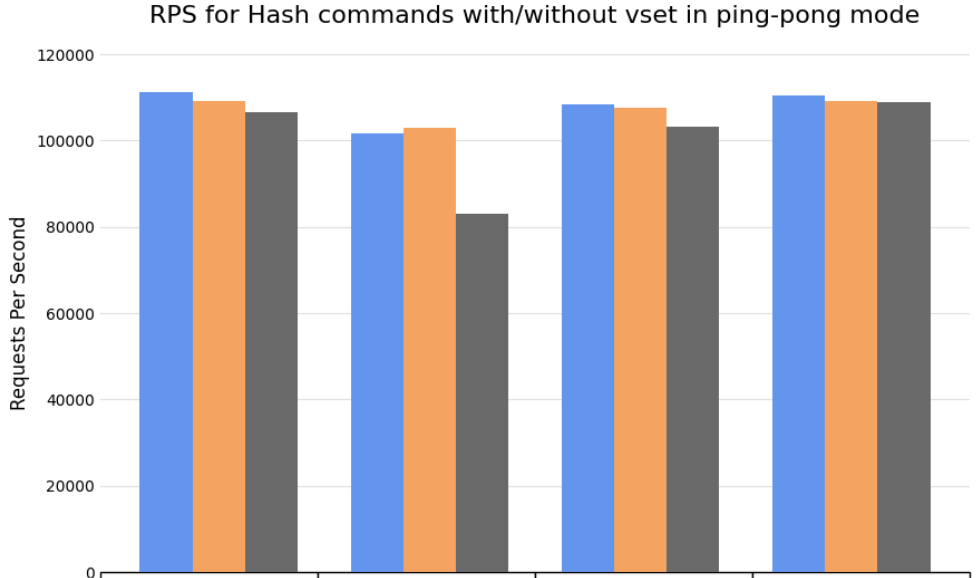
基准测试：API 性能

- Pipeline 模式



	HGET	HSET/HSETEX	HDEL	HEXISTS
Valkey without HFE feature	545,047	427,822	542,678	573,216
Fields without expiration	543,085	418,083	545,981	571,671
Fields with expiration	504,998	228,085	430,317	566,574

- Ping-pong 模式

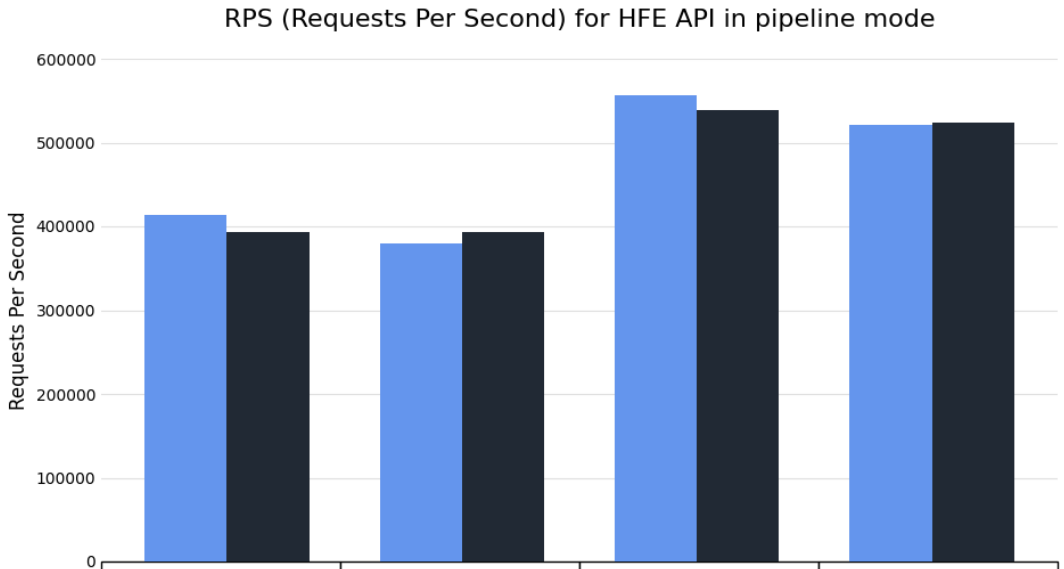


	HGET	HSET/HSETEX	HDEL	HEXISTS
Valkey without HFE feature	111,302	101,571	108,470	110,446
Fields without expiration	109,037	102,973	107,675	109,192
Fields with expiration	106,660	83,136	103,144	108,753

Valkey

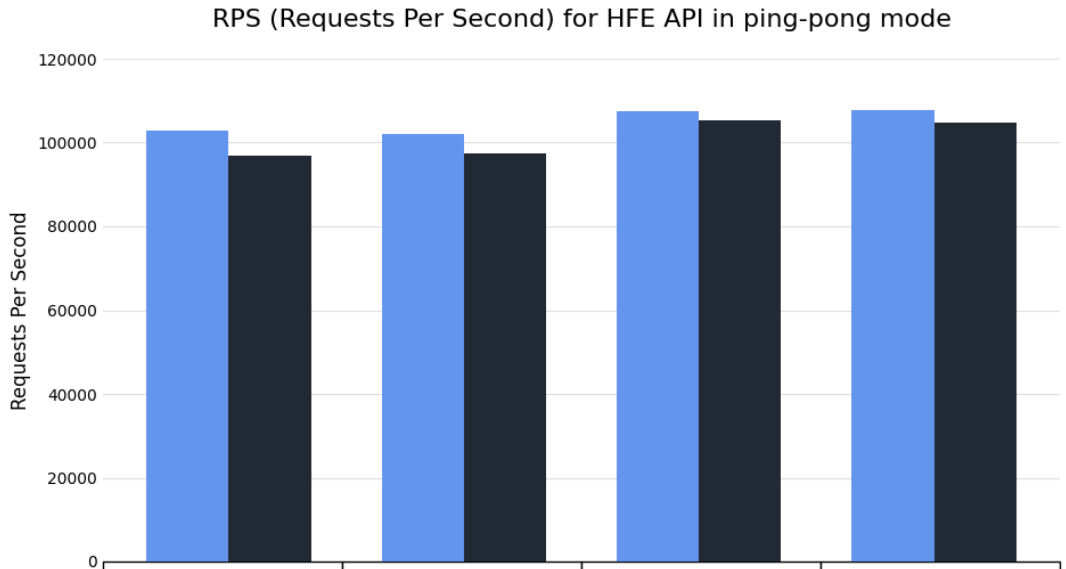
基准测试：API 性能

- Pipeline 模式



	HEXPIRE	HGETEX	HTTL	HPERSIST
QPS - Single HFE	413570.00	379607.00	556134.00	520419.00
QPS - Multiple HFEs	392595.00	392945.00	538937.00	523566.00

- Ping-pong 模式

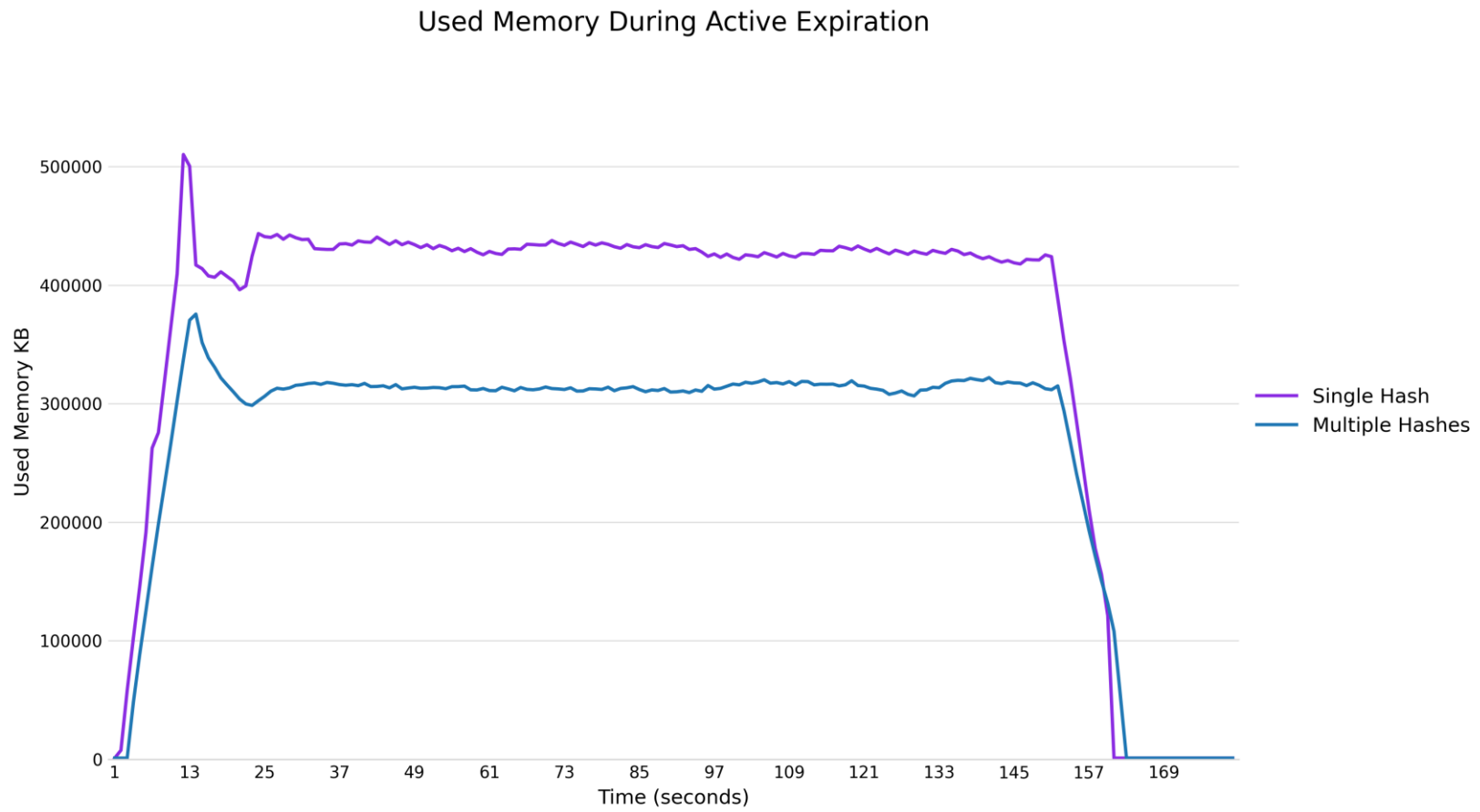


	HEXPIRE	HGETEX	HTTL	HPERSIST
QPS - Single HFE	102807.00	102118.00	107395.00	107740.00
QPS - Multiple HFEs	96822.00	97360.00	105286.00	104766.00

Valkey

基准测试：主动过期效率

- HSETEX 设置 field 过期时间为 10 秒，监控内存变化



Valkey

Future Works

- 使用紧凑编码，减少小 HFE 的内存开销
- Set 支持元素级别过期
- 使用 内存预取/SIMD 等能力加速 HFE 相关操作
- 主 key 级别的主动过期优化

Thanks !

Any Questions?