

Redis 开发规范

版本历史

版本/状态	责任人	起止日期	备注
V0.1/草稿	栗博	2018/7/6	完成初稿

目 录

1. 前言	1
1.1 文档说明	1
1.2 术语定义	1
2. 键值设计.....	1
2.1 KEY	1
2.2 VALUE	1
3. 命令使用.....	2
4. 客户端	2

1. 前言

1.1 文档说明

本文档制定 Redis 命名规范和使用规则，从而统一 Redis 的使用，并降低后期维护，扩容，管理的成本。主要参考阿里巴巴 Redis 规范和其他网上建议，以及相关项目经验。

1.2 术语定义

Redis: 一个内存 key-value 存储系统，提供了一些丰富的数据结构，包括 String, List, Hash, Set 和 Ordered Set。

2. 键值设计

2.1 Key

- 1) 【强制】为增加可读性和可管理性，所有 Key 的命名按照以下格式：{模块}:{功能}:{名称}。

说明：如果功能较复杂，可以将功能拆分为多级目录，统一以冒号分隔。

- 2) 【强制】全部使用小写或数字，{模块}和{功能}部分必须以小写字母开头。
- 3) 【强制】不能包含空格，换行、单双引号以及其他转义字符。
- 4) 【建议】Key 的长度尽量短，可以使用英文缩写。

2.2 Value

- 1) 【建议】尽量使用较短的长度以节省内存，String 类型不得超过 10K 字节，严格控制 List, Hash, Set 和 Ordered Set 的包含的数据量，不能超过 5000 个。

说明：如果数据量过大，请按功能分割。

- 2) 【建议】要合理控制和使用数据结构内存编码优化配置，在以下情况中，Redis 会使用占用内存较小的数据结构，请在业务中尽量控制：

String 类型：小于等于 39 字节。

List 类型：所有字符串元素长度小于 64 并且元素数量小于 512。

Hash 类型：所有键和值的字符串长度都小于 64 并且元素数量小于 512。

Set 类型：所有元素都是整数值并且元素数量不超过 512。

Ordered Set: 元素数量小于 128 个并且所有元素成员的长度都小于 64 字节。

- 3) 【建议】尽量使用 Hash 键而不是字符串键来储存键值对数据，很多情况下 Hash 键管理方便、能够避免键名冲突、并且还能够节约内存。
- 4) 【建议】尽量不要存储不过期的数据，使用 expire 设置过期时间，采取机制使 key 的过期时间错开，防止集中过期。

3. 命令使用

- 1) 【强制】给 Map, List, Set 设置过期时间时，一定要保证有元素。

说明：没有元素时，设置过期时间是无效的。

- 2) 【建议】避免使用 keys 遍历，可能会阻塞。

说明：可以使用 scan 方式渐进式处理。

- 3) 【建议】O(N)命令关注 N 的数量。

说明：例如 hgetall、lrange、smembers、zrange、sinter 等并非不能使用，但是需要明确 N 的值。有遍历的需求可以使用 hscan、sscan、zscan 代替。

- 4) 【建议】合理使用 select。

说明：redis 的多数据库较弱，使用数字进行区分，同时多业务用多数据库实际还是单线程处理，不易维护。

- 5) 【建议】使用批量操作提高效率。

说明：连续多次操作 Redis，请使用 pipeline 批量操作来提高效率，批量操作比逐个操作的性能要高的多。或者使用 mget, hmget, lpush 等批量命令。

- 6) 【建议】Redis 事务功能较弱，不建议使用。

说明：Redis 的事务功能较弱(不支持回滚)，而且集群版本要求一次事务操作的 key 必须在一个 slot 上。

- 7) 【建议】排序、并集、交集等操作尽量放在客户端执行。

- 8) 【建议】合理使用持久化方式。

说明：RDB 占用磁盘较少，但在数据量较大时效率很低，并且故障停机时丢失数据较多。AOF 占用磁盘较多，数据恢复时较慢。

- 9) 【建议】避免使用 flushall、flushdb 数据库清空操作。

4. 客户端

- 1) 【建议】客户端尽可能使用长连接或连接池，避免频繁创建销毁连接。

- 2) 【建议】避免多个应用使用同一个 Redis 实例。

- 3) 【建议】设置合理访问权限和密码。

- 4) 【建议】客户端和服务端要在一台机器或者一个网段。

- 5) 【强制】使用异步操作时，请注意异步操作带来的问题。

说明：例如 1.set 完立即 get 可能会获取失败。2.空的 map，set，list 在异步操作后调用 expire 可能会无效。