

# Redis 缓存技术在自动气象站资料调用中的优化应用<sup>①</sup>



詹利群<sup>1</sup>, 程海林<sup>2</sup>, 任晓炜<sup>1</sup>

<sup>1</sup>(广西壮族自治区气象信息中心, 南宁 530022)

<sup>2</sup>(成都信息工程大学, 成都 610225)

通讯作者: 詹利群, E-mail: 125484908@qq.com

**摘要:** 为了提高气象自动站资料的检索查询效率, 采用基于内存 Key-Value 结构的 Redis 数据库技术, 通过搭建 Redis 数据库集群, 把数据缓存在内存中并实现主从复制, 提出一种适合气象自动站数据特性的数据存储结构模型, 使得高频次访问的气象自动站数据能够直接从内存中读取, 有效地减少数据查询响应的的时间. 本文分析了逐小时自动气象站观测资料的调用情况和到报时间特性, 并基于 Redis 分别从业务库、索引库和时间库对逐小时自动站资料进行存储设计和具体实现, 最后通过实验比对, 得出采用 Redis 缓存逐小时自动气象站观测资料的检索效率比直接访问 CIMISS 检索效率高, 该方法也可以应用到其它资料的检索调用业务场景中, 具有较好的推广应用价值.

**关键词:** Redis; 缓存技术; 自动气象站资料; 优化应用

引用格式: 詹利群, 程海林, 任晓炜. Redis 缓存技术在自动气象站资料调用中的优化应用. 计算机系统应用, 2019, 28(5): 77-83. <http://www.c-s-a.org.cn/1003-3254/6880.html>

## Optimized Application of Redis Caching Technology in Automatic Weather Station Data Call

ZHAN Li-Qun<sup>1</sup>, CHENG Hai-Lin<sup>2</sup>, REN Xiao-Wei<sup>1</sup>

<sup>1</sup>(Meteorological Information Center, Guangxi Zhuang Autonomous Region, Nanning 530022, China)

<sup>2</sup>(Chengdu University of Information Technology, Chengdu 610225, China)

**Abstract:** In order to improve the efficiency of retrieval and query of meteorological automatic station data, Redis database technology based on memory key-value structure is adopted. By building a Redis database cluster, the data are cached in memory and the master-slave replication is realized. A data storage structure model suitable for the characteristics of meteorological automatic station data is proposed, which makes high frequency access possible. Meteorological automatic station data can be read directly from memory, effectively reducing the time of data query response. In this study, the hourly automatic weather station observation data transfer and reporting time characteristics are analyzed. Based on Redis, the hourly automatic weather station data are stored and designed from business database, index database, and time database respectively. Finally, through experimental comparison, it is concluded that the hourly automatic weather station observation data are cached by Redis, and the retrieval efficiency of proposed method is higher than that of CIMISS. This method can also be applied to other information retrieval and invocation business scenarios, and has a good application value.

**Key words:** Redis; cache technology; automatic weather station data; optimized application

① 基金项目: 广西科技重点研发计划项目 (桂科 AB16380267); 国家档案局科技项目 (2016-X-06)

Foundation item: Guangxi Major Science and Technology Research and Development Plan (AB16380267); National Archives Bureau Science and Technology Project (2016-X-06)

收稿时间: 2018-11-04; 修改时间: 2018-11-23; 采用时间: 2018-12-05; csa 在线出版时间: 2019-05-01

2016年12月,全国综合气象信息共享平台(简称CIMISS)在25个省份正式开始业务化运行,全国气象部门初步建立了基于CIMISS的国省一致的气象数据环境,各省统一了数据的来源、标准、流程和服务<sup>[1,2]</sup>,形成了良好的业务应用生态.广西壮族自治区气象局围绕CIMISS统一数据环境开展核心业务系统的集约整合和流程再造,截至2017年底已有26个核心业务系统成功对接CIMISS.随着越来越多核心业务系统的对接,CIMISS成为一个数据访问效率的瓶颈,所有对接的核心业务系统的数据源调用都是通过气象数据统一服务接口(简称MUSIC)<sup>[3,4]</sup>调用CIMISS数据.MUSIC面向气象业务和科研,提供全国统一、标准、丰富的数据访问服务和应用编程接口(API),为国、省、地、县各级应用系统提供唯一权威的数据接入服务.据2017年接口应用统计分析,用户平均每天访问接口的次数达到35万次以上,实时并发数平均达到每秒50次以上,接口耗时<1秒的统计达到97.8%以上,>1秒以上占据2.2%以上.在这些资料中,逐小时自动气象站资料具有空间分辨率高、时间间隔短的特点,一直是气象业务应用中用到最多的几种资料之一<sup>[5]</sup>,对于开展预报预测、决策服务、灾害预警等气象业务意义重大<sup>[6]</sup>,用户对该类数据的检索效率提出了越来越高的要求.传统的直接从数据库查询数据已经满足不了需求,我们迫切需要找出一种高效查询逐小时自动气象站数据的方法,以提高用户服务效率.

Redis是一种基于内存运行并支持持久化的NoSQL数据库,主要解决关系型数据库高并发情况下数据处理时效问题.因为是纯内存操作,Redis读写性能非常出色,每秒可以处理超过10万次的读写操作<sup>[7]</sup>.

## 1 Redis 内存数据库

### 1.1 Redis 缓存方法的性能特点

Redis是开源的高性能Key-Value存储引擎,提供多种数据结构.Redis可以用作缓存,也可以用作存储<sup>[8]</sup>.支持字符串、链表、集合、哈希等多种数据结构以及丰富的编程语言,还可将缓存数据持久化,对数据的更新将异步保存到磁盘上,除此之外,也实现了master-slave(主从)同步,提供集群操作.

### 1.2 Redis 的优势

#### 1) 性能极高

Redis直接在内存中进行读取及写入,他的速度非

常的快.读取速度是110 000次/s,写入速度是81 000次/s.

#### 2) 丰富的数据类型

Redis支持二进制案例的Strings, Lists, Hashes, Sets及Ordered Sets数据类型操作.

#### 3) 原子性操作

Redis的所有操作都是原子性的,同时Redis还支持对几个操作全并后的原子性执行.

#### 4) 具有丰富的特性

Redis还支持publish/subscribe,通知, key过期等等特性.

### 1.3 Redis 的应用场景

Redis最主要的应用场景之一便是业务缓存,将一些不经常改变但又经常访问的热点数据常驻在内存中,并在内存中完成对数据的操作<sup>[9]</sup>,有效地减少数据库读取次数,减少数据库压力,提高响应时间,增强吞吐量.

对Redis的主要操作便是对其键值的存取操作,根据数据结构的不同,提供有非常丰富的操作命令.例如,对字符串和哈希结构的简单存取命令如下<sup>[9,10]</sup>:

#### 1) 字符串操作

存储命令: SET key value

查询命令: GET key

#### 2) 哈希(hash)操作

存储命令: HSET key field value

查询命令: HGET key field

Redis还有设置时间过期的功能,通过EXPIRE key seconds命令完成对键key的过期时间设置.

## 2 自动气象站资料的调用情况和时间特性

### 2.1 自动气象站资料的调用情况

逐小时自动气象站资料作为气象业务部门日常观测的重要资料之一,也是业务应用中用到最多的几种资料之一.通过分析CIMISS—MUSIC气象数据统一服务接口中的资料访问情况,如图1所示.

通过分析得出,中国地面逐小时资料是所有资料访问中访问次数最多的,平均每天达到10万次左右,占据所有资料访问的三分之一左右.因此,对逐小时自动站资料的检索查询效率进行优化提升具有非常大的必要性.

### 2.2 自动气象站资料的时间特性

根据CIMISS数据入库时间分析地面自动气象站观测逐小时资料的到报情况,如图2所示.

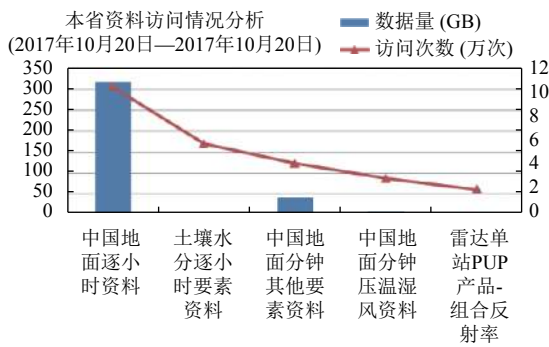


图1 广西 CIMISS 资料访问情况分析

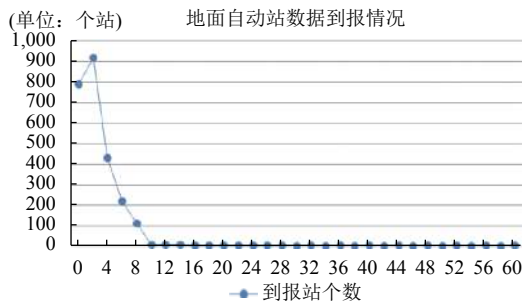


图2 2017年11月20日10时数据到报情况

可以发现, 98% 以上地面自动站的数据在正点后5分钟内已到报, 正点十分后数据相对稳定, 不经常改变, 因此可在正点十分后将相对稳定的数据缓存到 Redis 以提升检索效率.

为了确保 Redis 缓存数据与 CIMISS 数据的一致性, 需要定期的将 Redis 数据与 CIMISS 进行同步, 但如果同步频率设置太高, 如每秒或每分钟同步一次, 则从 Redis 获取数据与从 CIMISS 获取数据无差别, 对检索效率提升不大. 如果同步频率设置太低, 如每10分钟或每半小时同步一次, 缓存中的数据与 CIMISS 数据的一致性差异较大. 为此, 需要分析 CIMISS 地面自动气象站观测逐小时资料的数据更新规律, 选取了广西 2017 年的中国地面逐小时资料数据, 统计平均每  $k(k=1, 2, 3, \dots, 10)$  分钟更新次数, 结果如图 3 所示.

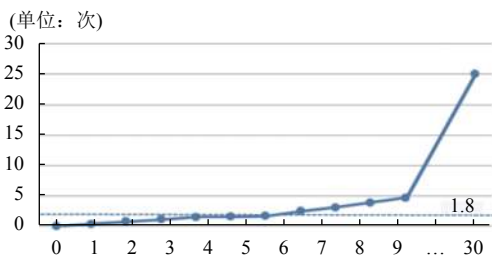


图3 CIMISS 平均每  $k$  分钟更新次数

需要选择一个较高的同步频率并且在这个频率期间数据更新尽量少. 根据逐小时资料数据的更新规律, 可选择每隔 5 分钟定期同步 CIMISS 的数据, 以此确保缓存数据的准确性.

### 3 基于 Redis 逐小时自动气象站资料存储模型

传统的关系型数据库把逻辑模型映射为一系列表来实现, 而 Redis 作为典型的 Key-Value 数据库没有表的概念, 需要将逻辑关系映射在 Key 值中或者一系列的键值中<sup>[1]</sup>. 在 CIMISS 系统中, 逐小时自动气象站资料是基于 Oracle 关系数据库进行存储的, 其检索接口方式主要包括有按时间、地区等. 根据这些业务需求, 需要对数据在 Redis 中的存储结构进行策略优化设计, Redis 数据库设计的关键在于 Key 的设计, 合理的设计能够有效提升查询效率和节省内存的开销. 因此, 针对实时查询的需求, 在 Redis 中的地面自动气象站逐小时资料, 只存储最近 1 小时的数据, 并通过设计业务数据库、索引数据库和过期时间库以提升查询效率和保证有效性.

#### 3.1 业务数据库存储模型

业务数据是存储地面自动气象站逐小时资料数据的库, 包括站名、区站号、市、县、经度、纬度、行政区划码等基础信息, 并可动态扩展温度、降水量等观测要素值. 为实现此功能, 使用站号作为 Key(固定前缀 \_surflh\_ 表示逐小时资料), 使用哈希结构作为存储数据的结构. 其存储结构模型如图 4 所示.

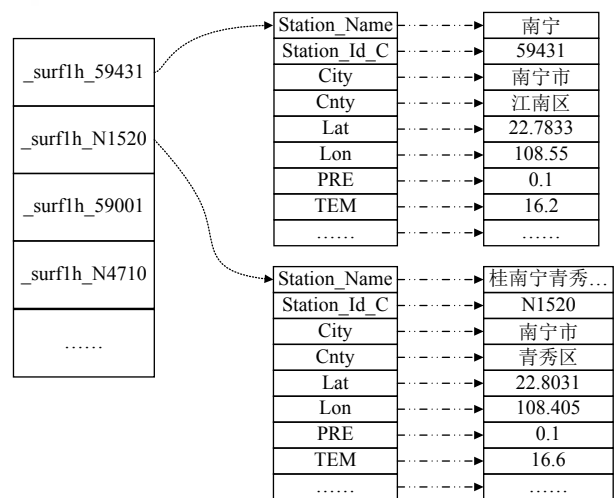


图4 逐小时资料业务数据存储结构

站名、站号等基础信息作为通用的公共数据,可在 Redis 搭建完成后便初始化并长期保存,而在业务调用中,通过哈希结构的 HSET 命令来更新要素数据或动态扩展要素,使得业务数据库能够支撑存储地面逐小时资料的所有要素信息.如广西气象自动站有 2700 个站,在 Redis 初始化时就可将全部自动站采用站号 Key 进行初始化完成,每个 Key 中的气象自动站逐小时 210 个要素字段也可以动态初始化完成,在数据到报后通过 jedis.hset("\_surf1h\_"+站号,"要素",value) 的方式进行添加覆盖数据,从而达到数据的快速存储操作.

```

.....
Redis>jedis.hset("_surf1h_59431","City","南宁")
Redis>jedis.hset("_surf1h_59431","Cnty","江南区")
Redis>jedis.hset("_surf1h_59431","Lat","22.7833")
Redis>jedis.hset("_surf1h_59431","Lon","108.55")
.....
    
```

### 3.2 索引库(地点-站号索引库、要素值-站号索引库)

为了实现能够快速的按照地区、站号等要素值检索,需要建立地区、要素值与业务数据库中 Key(即站号)的索引关系.检索数据时,先通过索引库检索到符合条件的站号,以站号为 Key 到业务数据库中业务数据.不同的索引方式根据不同的关系操作,采用不同的数据存储结构.在本文中,主要建立了地区-站号索引库,用于支持按地区检索要素.

地区-站号索引库采用哈希结构,顶级 Key 命名为 \_surf1h\_index\_area,哈希结构的 Key 使用市(县)名,值为区站号,多个使用逗号分隔,这部分数据经常被访问但几乎不会更新,因此在搭建 Redis 后即可进行初始化并长期保存.通过在初期建立 \_surf1h\_index\_area 的 Key,将广西 14 个地市和每个地市的相对应的区站号建立起来,其存储结构如图 5 所示.

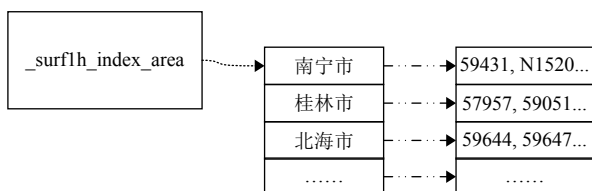


图 5 逐小时资料地区-站号索引库存储结构

### 3.3 过期时间库

为保证缓存在业务数据库中数据与 CIMISS 中的

数据是准确一致的,可以通过设计过期时间库来确保有效性.过期时间的设定与检索查询效率和数据一致性紧密相关,过期时间太短,则会频繁地查询 CIMISS,效率提升不明显,过期时间太长,缓存中的数据又与 CIMISS 差异较大,不适合业务应用.Redis 本身拥有设置时间过期的功能,即对存储在 Redis 数据库中的值可以设置一个过期时间.根据对 2.2 自动气象站资料的时间特性进行分析,在正点 10 分开始将较稳定的数据在 Redis 进行缓存,并每隔 5 分与 CIMISS 数据进行一次同步,即 Redis 中的数据只有在 5 分钟内是有效的,当超出该时间后,过期时间库中的该要素信息便会被移除,此时即表示业务数据库该要素的数据已过期.

## 4 逐小时自动气象站资料查询优化实现

### 4.1 优化业务数据库

数据写入 Redis 业务数据库初期,程序员建立散列类型,通过 HSET 命令逐要素写入.设计过程中又通过 HMSET 命令同时设置多个要素字段写入,程序代码比之前简洁.另外,通过要使用 Redis 集群存储资料首先要初始化线程类:

```

new RedisTestThread("redis"+i, element, admin-Code).start()
    
```

将不同的要素、地区数据进行初始化.在线程类中实现将不同的站号进行初始化,根据资料要素的不同站号,通过 jedis.hmset(key, dataMap) 进行数据设置,其中 Key 为 "\_surf1h\_"+站号, dataMap 为站号下对应的业务数据.

最后设置数据过期时间 jedis.expire(key, Time).

### 4.2 优化索引库

省、市、县三级用户对自动气象站逐小时数据的查询主要通过地区行政编码进行分类查询,索引库的建立,可以通过采用地区行政编码作为地区分类.首先建立所有地区分类的集合,如“450000”代表全广西,则该索引下的站号为全广西的自动站站号,“450100”代表南宁市,则“450100”索引下存放整个南宁市的自动站站号.通过循环设置全广西各区市县的站号索引.在程序设计上先定义一个 HashMap 用来存放行政编码对应的站号信息,然后通过 jedis.hmset(\_surf1h\_index\_area, map) 存放业务数据, jedis.expire(\_surf1h\_index\_area, Time) 设置过期时间,达到按地区行政编码优化索引库提升索引查询性能.

### 4.3 查询流程设计与实现

完成业务数据库、索引库及过期时间库存储设计后,开始通过程序将逐小时自动气象站资料及相关的辅助信息存入库中,资料查询流程如图6所示。

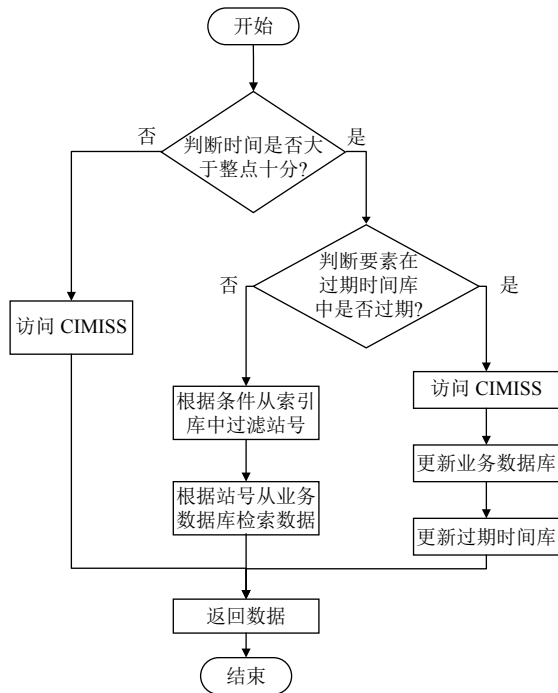


图6 基于Redis的逐小时资料检索查询流程

主要包括以下三种情况:

#### 1) 请求时间在正点后十分钟内

根据自动气象站观测规范,自动气象站观测数据在正点观测后,数据立即从分布在全区各个观测站发回观测数据文件。通常正点后的2-4分钟是地面逐小时观测数据入库的高峰期,在这个时间内数据解码入库的更新频率非常高,将数据缓存到Redis中没有意义,因此这个时间内的数据请求均是通过接口直接访问CIMISS获取数据并返回。

#### 2) 请求时间在正点十分后,Redis缓存数据已过期

此时CIMISS中数据相对稳定,但Redis中缓存数据已过期,因此需要重新访问CIMISS以获取最新有效数据,获取数据后更新业务数据库中相应要素的值,同时重新设置该要素的过期时间,如设置为5分钟,最后返回数据。

#### 3) 请求时间在正点十分后,Redis缓存数据未过期

此时缓存在Redis中的数据为有效数据,首先通过查询条件,在索引库中进行相应的关系查询筛选出符

合条件的站点数据,然后通过站号从业务数据库中检索数据并返回。

在程序设计上使用jedis.hget(areaIndexKey, admin-Code),根据地区行政编码进行数据分类获取站号索引,得到该地区的所有站点,再通过循环站点获取业务数据:

```
for (String stationId: stations) {
    //用 hmget 获取 Redis 相关业务数据
    List<String> data = jedis.hmget("_surf1h_" +
    stationId, "Station_Id_C", "Station_Name", "Admin_Code_CHN", "Province", "City", "Cnty", "Town", "Station_lvl", "Lat", "Lon", element);
    //用于将从 redis 中 hash 数据查询出的数据转换为业务所需的结构
    Map<String, String> dataMap = this.listToMap
    (data, CimissUtils.commentElements + "," + element);
    dataLists.add(dataMap);
}
```

通过上述代码可查询得到相应的业务数据

## 5 实验对比分析

### 5.1 实验环境搭建

#### 1) 搭建 Redis 集群

使用Redis集群不仅可以扩大缓存数据量,也能通过主从复制机制避免单节点故障导致无法继续工作。在本文中,搭建3台服务器6个节点的Redis集群,部署环境如表1所示。

序号	IP 地址	端口
1	10.158.90.57	7000
2		7001
3	10.158.90.58	7000
4		7001
5	10.158.90.59	7000
6		7001

#### 2) 建立模拟访问

使用Jedis(Redis的Java版本的客户端实现)实现对Redis集群的操作<sup>[12]</sup>,编写Java程序执行访问请求,使用多线程等技术,模拟实现对CIMISS以及对Redis的检索查询操作。

### 5.2 数据准备

选取地面中中国地面逐小时资料进行试验分析,

并选取了2018年11月26~30日每天上午7~10时和下午17~22时(业务高峰期)对实时数据(1小时内)进行缓存.资料包括站号、站名、温度、降水等210个要素,如表2所示.

表2 中国地面逐小时资料数据结构

要素代码	要素名称	要素代码	要素名称
Station_Name	站名	Station_Id_C	区站号(字符)
Province	省	Lat	纬度
City	市	Lon	经度
Cnty	区县	REP_CORR_ID	更正报标志
Town	乡镇	TEM	温度/气温
PRE_1h	过去1小时降水量	...	...

### 5.3 实验设计

为实现对基于Redis的逐小时自动气象站资料存储的查询效率进行分析,设计两组实验模拟访问最近一小时的逐小时自动气象站资料,实验周期为2018年11月26~30日每天上午7~10时和下午17~22时,访问频率为每隔一分钟进行一次请求,实验详细设计如表3所示.

表3 实验组设计

实验组	实验过程
CIMISS组	每次都直接访问CIMISS接口查询数据
Redis组	(1) 正点后十分前,直接访问CIMISS接口查询数据; (2) 正点后十分后,每隔五分钟访问一次CIMISS查询数据,并将数据缓存到Redis,其后的五分钟内则访问Redis查询数据.

实验选择检索南宁市最近一小时降雨量为例,通过记录每次开始执行访问至获取到数据时的时间差(即执行一次检索数据的耗时,时间单位为ms)作为实验结果.

### 5.4 实验结果分析

#### 1) 查询效率分析

因正点后前十分CIMISS组合Redis组均是直接访问CIMISS接口,效率一致,不必分析.选择在正点后十分后模拟基于Redis和CIMISS的数据查询,每分钟调用一次查询,分别记录每次查询耗时,结果如图7所示.

可以看出,在正点后十分后,直接从CIMISS查询的耗时总体在500ms左右,而Redis的查询耗时大多数时候在100ms左右,每隔5分钟因需从CIMISS进行数据同步而耗时在500ms左右.从Redis查询平均耗时0.2秒左右,CIMISS组平均耗时0.5秒左右,前者具有更高的查询效率,是后者的两倍以上.

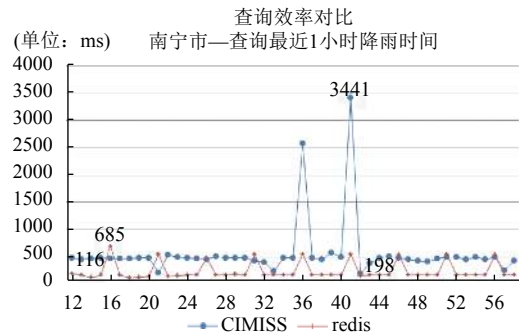


图7 基于Redis与CIMISS查询耗对比

#### 2) 数据一致性分析

缓存中的数据包括中国地面逐小时资料共210个要素,为确保数据的准备,通过对比2018年11月26~30日每天上午7~10时和下午17~22时每分钟从Redis取出的要素值与CIMISS中的要素值,记录值相同的要素个数,实验期间平均结果如图8所示.

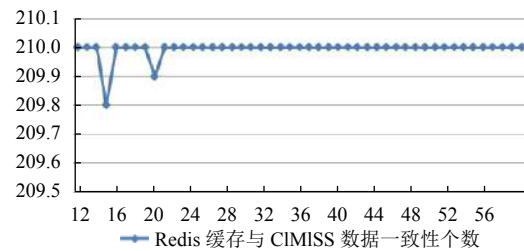


图8 Redis与CIMISS数据值相同的要素个数统计图

通过分析,中国地面逐小时资料共210个要素在1小时内缓存数据与CIMISS数据的一致性达到了99%以上,能确保数据的完整性,满足业务应用要求.

## 6 结论

本文利用Redis是开源的高性能Key-Value存储引擎,提供多种数据结构利用Redis开源高效内存数据库的特点,结合地面自动气象站逐小时资料在实际业务中的应用,通过Redis数据库结构模型设计和优化,并通过实验对Redis提高了逐小时自动气象站资料的检索效率和数据一致性进行验证,为用户在不同气象资料查询检索方面提供了一套可操作的技术参考样例.由于自动气象站观测资料更正报等实际情况,可能导致缓存在Redis中的数据及时性和一致性降低,这方面可以考虑采用基于消息的机制更新缓存数据或缩短缓存时间等来解决这个问题.在气象业务中,还存在很

多对查询要求较高的观测资料, 本文提出的逐小时地面自动站资料数据结构模型并通过 Redis 进行缓存能有效的提升检索效率, 并可以应用到其它资料的检索调用业务场景中, 具有较好的推广应用价值.

### 参考文献

- 1 国家气象信息中心. CIMISS: 支撑气象核心业务系统的数据生态. 中国气象报, 2017-01-10(003).
- 2 熊安元, 赵芳, 王颖, 等. 全国综合气象信息共享系统的设计与实现. 应用气象学报, 2015, 26(4): 500-512.
- 3 曾行吉, 李涛, 詹利群, 等. 基于 MUSIC 的特色数据与产品回写 CIMISS 方法研究. 气象研究与应用, 2018, 39(1): 111-114. [doi: 10.3969/j.issn.1673-8411.2018.01.026]
- 4 徐拥军, 何文春, 倪学磊, 等. 基于 MUSIC 接口的 CIMISS 数据环境故障应急切换设计与实现. 气象科技进展, 2018, 8(1): 213-218, 231. [doi: 10.3969/j.issn.2095-1973.2018.01.041]
- 5 王晨珏. 苏皖地面自动站资料质量控制及其同化研究[硕士学位论文]. 南京: 南京信息工程大学, 2016.
- 6 张志富, 任芝花, 张强, 等. 自动站小时气温数据质量控制系统研究. 气象与环境学报, 2013, 29(4): 64-70. [doi: 10.3969/j.issn.1673-503X.2013.04.010]
- 7 曾超宇, 李金香. Redis 在高速缓存系统中的应用. 微型机与应用, 2013, 32(12): 11-13. [doi: 10.3969/j.issn.1674-7720.2013.12.004]
- 8 闫明. 高可用可扩展集群化 Redis 设计与实现[硕士学位论文]. 西安: 西安电子科技大学, 2014.
- 9 刘高军, 王帝澳. 基于 Redis 的海量小文件分布式存储方法研究. 计算机工程与科学, 2013, 35(10): 58-64. [doi: 10.3969/j.issn.1007-130X.2013.10.007]
- 10 刘俊龙, 刘光明, 张黛, 等. 基于 Redis 的海量互联网小文件实时存储与索引策略研究. 计算机研究与发展, 2015, 52(S1): 148-154.
- 11 唐诚. Redis 数据库在微博系统中的实践. 厦门城市职业学院学报, 2012, 14(3): 55-59. [doi: 10.3969/j.issn.2095-2724.2012.03.012]
- 12 邱书洋. Redis 缓存技术研究及应用[硕士学位论文]. 郑州: 郑州大学, 2015.