

Redis 在高速缓存系统中的应用

曾超宇¹, 李金香²

(1. 中国电子信息产业集团有限公司第六研究所, 北京 100083;

2. 渤海大学 教育与体育学院, 辽宁 锦州 121013)

摘要: 随着互联网的发展, 各种类型的应用层出不穷, 网站访问量越来越大、内容越来越多, 用户与系统的交互不断增强, 访问集中, 造成数据库负载过大, 网站显示延迟等影响。缓存技术就是解决此问题的一种方案, 缓存技术以其简单的设计、高效的存储性能得到了越来越广泛的应用, 而内存数据库则是一种优秀的缓存解决方案。主要介绍 Redis 的特性以及在系统中的应用。

关键词: Redis; 高速缓存; 内存数据库

中图分类号: TP311.1

文献标识码: A

文章编号: 1674-7720(2013)12-0011-03

Redis application in cache system

Zeng Chaoyu¹, Li Jinxiang²

(1. The 6th Research Institute of China Electronics Corporation, Beijing 100083, China;

2. College of Education and Sports, Bohai University, Jinzhou 121013, China)

Abstract: With the continuous development of Internet technology, all kinds of applications appear endlessly, Web site traffic is bigger and bigger with more and more content, the user interacts with the system more deeply. The increased access to concentration causes the database load is too large, website show delays, etc. And caching technology is a solution to solve this problem, the cache technology, with its simple design and efficient storage performance has been more and more widely used, and in-memory database, as a kind of excellent caching solutions. This paper introduces Redis features and application in the system.

Key words: Redis; cache; main memory database

随着 Internet 技术的快速发展, 网络接入速度不断提高, 各种类型的应用层出不穷, 网站访问量越来越大、内容越来越多, 用户交互不断增强, 访问集中, 造成数据库负载过大、网站显示延迟等影响, 而造成影响用户体验的主要瓶颈集中在数据库服务器承载能力方面。要让数据库服务器快速响应并能够承受越来越大的负载, 缓存技术就是解决此问题的一种方案。Cache 性能高效, 设计简单, 可以对数据库中的数据进行缓存, 降低数据库负载; 可以对 Web 页面进行缓存, 提高 Web 页面响应速度; 对复杂计算结果进行缓存, 可以减少网站服务器的传输负荷和计算速度和对用户的响应速度, 有效地提高网站性能及可扩展性。本文主要介绍开源内存数据库 Redis 的特性及其应用。

1 Redis 简介

Redis 是一种 Key-Value 类型的内存数据库产品, 全名为远程字典服务 (REmote DIctionary Server), 与 Memcache 相似, 但 Redis 支持更多的数据类型, 包括字符串 (String)、链表 (List)、集合 (set)、有序集合 (Zset)、哈希

(Hash) 等。与 Memcached 一样, 为了保证效率, Redis 数据都是缓存在内存中, 但与 Memcache 只是用来做缓存相比, Redis 适用的场景更多, 并且可以直接用于数据存储服务。

2 Redis 特性

虽然 Redis 与 Memcached 具有很多相似的特征, 可替代 Memcached 做为缓存, 但它又具有更多优秀的特性, 如支持多种数据结构、支持简单事务控制、支持持久化、支持主从复制、Virtual Memory 功能等。

2.1 Redis 数据类型

Redis 除了提供常规的数值或字符串外还提供 4 种数据类型: List、Set、Zset (Sorted Set) 和 Hset (Hash Set)。

(1) String (字符串)

Strings 数据结构是简单的 Key-Value 类型, Value 其实不仅是 String, 也可以是数字。使用 Strings 类型, 可以完全实现目前 Memcached 的功能, 并且效率更高; 还可以享受 Redis 的定时持久化、操作日志及 Replication 等功能。

(2) List (双向链表)

Lists 就是链表, 使用 Lists 结构可以轻松地实现最新

消息排行等功能。Lists 的另一个应用就是消息队列,可以利用 Lists 的 PUSH 操作将任务存在 Lists 中,随后工作线程再用 POP 操作将任务取出进行执行。Redis 还提供了操作 Lists 中某一段的 API,可以直接查询、删除 Lists 中某一段的元素。

(3)Set(集合)

Sets 是一个集合,集合的概念就是一堆不重复值的组合。Set 是 String 类型的无序集合。Redis 还为集合提供了求交集、并集、差集等操作,可以非常方便地实现如共同关注、共同喜好、二度好友等功能,对上面的所有集合操作还可以使用不同的命令选择将结果返回给客户端还是存集到一个新的集合中。

(4)Zset(有序集合)

Zset 与 Set 相似,但在 Set 的基础上增加了一个 Score 的属性。这一属性在添加修改元素时进行指定,每次指定后,Zset 会自动按新的值重新调整顺序。

(5)Hash(hash 表)

在 Memcached 中,经常将一些结构化的信息打包成 hashmap,在客户端序列化后存储为一个字符串的值,比如用户的昵称、年龄、性别、积分等,在需要修改其中某一项时,通常需要将所有值取出反序列化后,修改某一项的值,再序列化存储回去。这样不仅增大了开销,也不适用于一些可能并发操作的场合(比如两个并发的操作都需要修改积分)。而 Redis 的 Hash 结构可以实现像在数据库中 Update 一个属性一样只修改某一项属性值。

2.2 Redis 事务控制

Redis 可以通过 MULTI/EXEC 来支持简单的事物控制。Redis 只能保证事务中的所有命令串行执行,在事务的执行过程中不会为其他客户端发起的请求提供服务。当一个 client 使用 multi 命令时,这个连接会进入一个事务上下文,该连接后续的命令会放在一个队列中,当此连接收到 exec 命令后,Redis 会顺序地执行队列中的所有命令,但是如果事务中的一个命令失败了,并不回滚其他命令。

2.3 Redis 持久化机制

Redis 是一个能支持持久化的内存数据库,它通过将内存中的数据保存到磁盘来持久化。Redis 有两种持久化方式,一种是 Snapshotting(快照),另一种是 append-only file(缩写 aof)的方式。下面分别介绍:

(1)Snapshotting 方式

快照是默认的持久化方式。这种方式是将内存中数据以快照的方式写入到二进制文件 dump.rdb 中。可以通过配置 redis.conf 来设置自动做快照持久化的方式。

(2)aof 方式

如果对数据要求很高,可以采用 aof 持久化方式。因为在使用 aof 持久化时,Redis 会将每个命令都追加到 appendonly.aof 文件中,当 Redis 出现意外关闭后,重启后

会通过执行 appendonly.aof 文件中的命令来在内存中重建整个数据^[1]。

2.4 主从复制

Redis 主从复制配置非常简单,可以通过配置 redis.conf 文件中的 Replication 段来实现主从复制。Redis 主从复制特点:

(1)支持一个 master 可以拥有多个 slave,同时 slave 还可以接收其他的 slave。

(2)主从复制不会阻塞 master 和 slave,在同步数据时, master 和 slave 都可以接收 client 请求。

通过主从复制的特性可以做到以下 3 个方面:

(1)可以做到读写分离,提供系统伸缩性和系统性能,如 master 主服务用来写数据,slave 服务用来读数据。

(2)可以做到备份数据分离,如 slave 服务器群中的一个或两个服务器用来备份数据。

(3)虽然 Redis 宣称主从复制无阻塞,但是由于 Redis 使用单线程服务,而与 slave 的同步是由线程统一处理,因此,对性能有影响。在 slave 第一次与 master 做同步时,如果 master 快照文件较大,相应快照文件的传输将耗费较长时间,文件传输过程中 master 会造成访问延迟。

2.5 Virtual Memory 功能

Redis 的 Virtual Memory(虚拟内存)通过配置可以让用户设置最大使用内存,当超出这个内存时,通过 LRU 类似算法,将一部分数据存入文件中,在内存中只保存使用频率高的数据来提高 Redis 性能。

3 Redis 应用分析

通过上述 Redis 的特征分析,可以看到当作 Cache 工具时,Redis 与其他 Cache 相比更多的优势性能和内存使用效率相差无几却拥有更多的数据结构并支持更丰富的数据操作,同时还支持持久化。例如在某博客系统中,整个系统结构图如图 1 所示,客户端通过 Redis proxy 访

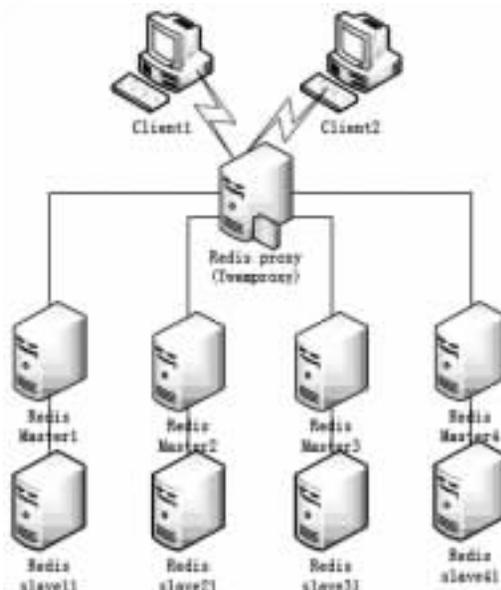


图 1 博客系统结构图

问 Redis。目前的 Redis 本身都不具备分布式集群特性,当有大量 Redis 时,通常只能通过客户端的一些数据分配算法(比如一致性哈希)来实现集群存储。而 Twemproxy 通过引入一个代理层,可以将其后端的多台 Redis 实例进行统一管理,使应用程序只需要在 Twemproxy 上进行操作,而不用关心后面具体有多少个真实的 Redis 存储,这样就可以通过平行扩展 Redis Master 服务器来无限扩展 Redis。

Redis-Master1 与 Redis-slave11 通过主从复制连接。同时,Redis-Slaver11 通过 aof 负责持久化。Twemproxy 可以配置结点故障问题,当某一个结点出现故障后,用 Redis Master 相应的 Redis Slave 切换成 Master。

在系统运行中要处理各种业务逻辑,因而需要访问大量的数据库资源,在数据库上进行读写操作,数据库压力会较大,性能较低。通常把更新不频繁的数据进行缓存。例如下面几种操作。

(1)根据条件得到最新数据的操作,以时间为权重

比较典型的取最新博文的数据代码如图 2 所示。可以将最新的 5 000 条博客的 ID 放在 Redis 的双向链表(list)集合中,使用 LPUSH global:blogpostids 命令向双向链表(list)集合中插入数据,插入完成后再用 LTRIM global:latest.blogs 0 5000 命令使其永远只保存最近 5 000 个 ID,然后在客户端获取最新博文时可以用下面的逻辑(伪代码)了。如果还有其他的取最新数据需求,比如“模拟技术”分类的最新 50 条数据,则可以再建一个按此分类的双向链表(list),将博文的 ID 插入此双向链表(list)中。

```

1: list blogPostInfoKeys;
2: using (var redisClient = new RedisClient(redisHost, redisPort, redisAuth));
3:
4: // 初始化 Redis 双向链表
5: List<string> latestBlogPostIds = SortedSet<string>(redisClient, 10);
6: List<string> redisBlogPostInfoKeys = SortedSet<string>();
7: latestBlogPostIds.ForEach(x =>
8: {
9:     redisBlogPostInfoKeys.Add(string.Format("blog/{0}/info", x));
10: });
11: using (var redis = redisClient.AsyncClient())
12: {
13:     // 初始化 Redis 双向链表
14:     latestBlogPostInfoKeys = SortedSet<string>(redis, 10);
15: }

```

图 2 取最新博文代码

(2)根据条件得到排行榜应用,相当于数据中取 TOP N 操作,例如访问量最高的博文、用户的相互关注等。

以某个条件为权重得到数据,如果博文按浏览的次數排序,这时可以使用 sorted set 将浏览次数值设置成 sorted set 的 score,将具体的博文 ID 设置成相应的 value,每次新增博文时只需要执行一条 ZADD blog:getblogpostrank 0 1000,每当博文被浏览时,使用 ZINCRBY blog:getblogpostrank 1 1000,取访问量最高的博文时就可以使用 ZREVRANGE blog:getblogpostrank 0 -1,来按照浏览次数高低排行了。

(3)网站计数的应用,比如在博客系统中浏览次数、评论数等。

Redis 的命令都是原子性的,可以使用 INCRBY、

ZINCRBY 命令来增加计数,使用 DECRBY、ZINCRBY 命令来减少计数。如果给博文 ID 号为 1 000 的增加一次浏览,则使用 INCRBY blogpost:1000 1 命令。

(4)构建反垃圾信息(spam)系统,比如评论系统中的反垃圾信息等。

在博客系统中评论是必不可少的,同时各种攻击 spam 也少不了(如垃圾评论、广告、刷排名等),可以针对这些 spam 制定一个规则,例如一分钟评论不得超过 2 次,5 分钟不能评论多于 5 次。这可以使用 Redis 的 Sorted Set 来将最近一天用户操作记起来。使用系统时间为 score,这样执行一条 ZADD user:1000:operation 54561227854 “发表评论”,得到用户最近一分钟的操作如下 ZRANGEBYSCORE user:1000:operation 54561227854 +inf。这样如果一分钟有两次评论,就可以判定为 spam(每个网站的规则不一样)。

(5)可以使用 Redis 系统中 Pub/Sub 和 Ajax 长链接来构建实时消息系统。

Redis 的 Pub/Sub 系统作为一种消息通信模式,类似于设计模式中的观察者模式,发布和订阅机制可以很方便地实现简单的聊天功能,可以应用于实时聊天(通过与 Ajax 长链接)、异步消息处理(如注册后给用户邮件通知)等场景。将不同的频道分布到不同服务器上,也可以很方便地实现服务器的扩展以应对用户量的增长。

(6)使用 List 或者 Zset 构建队列系统。

可以使用 List 构建队列系统,加入队列可以使用 rpush global:queue addblogpost,退出队列使用 lpop global:queue。使用 Zset 来构建有优先级的队列系统,加入普通级别的列队命令 zadd global:zqueue 1 addblogpost,加入高优先级的列队可以通过提高 score 的值来提高优先级。

为了高速缓存在系统应用的不同需求,对 Redis 的主要特征进行了分析,同时对其作为缓存解决方案的应用进行了举例说明。随着 Redis 的发展,官方 roadmap 在 Redis 3 版本中将加入集群功能,Redis 作为内存数据库,可以通过水平扩展来延伸系统的物理内存,同时又支持持久化,完全可以将所有数据存放在 Redis 内存数据库中,这样才能充分发挥内存数据库的优势。

参考文献

- [1] Khan.redis 学习笔记之持久化[EP/OL].[2011-02-07] (2013-03-15).<http://www.cnblogs.com/xhan/archive/2011/02/07/1949640.html>.

(收稿日期:2013-05-02)

作者简介

曾超宇,男,1982 年生,学士,助理工程师,主要研究方向:软件架构、软件工程。

李金香,女,1982 年生,学士,主要研究方向:计算机软件。

《微型机与应用》2013 年 第 32 卷 第 12 期