ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.11, No.7, March 2015
E-mail:kfyj@dnzs.net.cn
http://www.dnzs.net.cn
Tel:+86-551-65690963 65690964

Redis在订阅推送系统中的应用

吴霖,刘振宇,李佳

(南华大学 计算机科学与技术学院,湖南 衡阳 421000)

摘要:订阅推送系统是为解决在移动互联网下,由服务端向客户端进行主动推送图文消息的需求而提出的。基于Redis的高速缓存机制在订阅推送系统中的应用进行了研究,详细说明了如何在程序开发中有效地利用Redis中的各种数据结构。

关键词:Redis;内存数据库;订阅;推送

中图分类号:TP311 文献标识码:A 文章编号:1009-3044(2015)07-0292-03

DOI:10.14004/j.cnki.ckt.2015.0256

Redis Application in Subscription and Push System

WU Lin, LIU Zhen-yu, LI Jia
(College of Computer Science and Technology, University of South China, Hengyang 421000, China)

Abstract: Subscription and push system is proposed for solving the problem of how to push diagram text combined messages from the server to the client actively in Mobile Internet. This article studied the applications in subscription and push system, based on the caching mechanism of Redis, and explain how to effectively make use of the data structures of Redis in program development detailedly.

Key words: Redis; memory database; subscription; push

近年来,随着智能手机的普及和移动互联网的快速发展,越来越多的移动APP应用如春风般涌现出来。而在移动APP应用中,经常会有由服务端向客户端进行主动推送图文消息的需求,如向用户推送优质资源、问卷调查、APP使用教程等。而目前移动APP应用的用户规模往往都是千万级别,大规模地批量推送对程序性能有很高的要求,因此,如何合理地设计一个订阅推送系统非常重要。

本文首先简单地介绍了Redis及其特性;其次,概要描述了作者最近开发的一个订阅推送系统及其系统架构;最后,详细地介绍了Redis的各种数据结构在本系统中的应用。

1 Redis

1.1 Redis简介

Redis是一个基于内存的高性能key-value数据库,其全名为远程字典服务(Remote Dictionary Server)[1]。与Memcached一样,为了保证效率,数据都是缓存在内存中,即以空间换取时间,从而达到提高性能的目的。而与Memcached的最大区别在于,Redis会周期性地把更新的数据写入磁盘或者把修改操作写入追加的记录文件,并在此基础上实现了主从同步。

1.2 Redis的特性

1.2.1 支持持久化

Redis和Memcached一样都是把数据存在内存中。不同的是,Memcached把数据全部存放在内存之中,服务器重启后数据会消失,而且缓存的数据不能超过内存大小;而在Redis中不是所有的数据都一直存储在内存中,Redis会根据数据的更新量和更新间隔时间定期进行异步更新,将数据保存在磁盘上,实现数据的持久化操作。可以通过配置文件配置持久化操作的时间间隔和临界数据量,这样Redis就会自动在一定的时间间隔或者当数据超过临界值时进行持久化操作[2]。此外,服务器重启后,数据也可以很快恢复使用。

1.2.2 支持丰富的数据结构

Redis的数据结构非常丰富。Redis支持简单的key-value类型数据,其中key是string类型,而value除了常规的string(字符串)之外,还包括list(链表)、set(集合)、zset(sorted set有序集合)和hash(哈希类型)。这些数据类型都支持push/pop、add/remove、取交

收稿日期:2015-01-26

作者简介:吴霖(1989—),男,广东潮州人,硕士,主要研究方向为软件工程;刘振宇(1962—),男,湖南衡阳人,教授,硕士生导师,主要研究方向为软件分析与建模、软件测试、信息化工程;李佳(1987—),男,安徽固镇人,硕士,主要研究方向为数据挖掘。

集、并集和差集等操作,而且这些操作都具有原子性,它还支持各种不同的排序能力[2]。

1.2.3 支持主从复制

Redis的主从复制实现简单却功能强大,支持多级Master/Slave,一个master支持多个slave连接,slave可以接受其他slave的连接。主从同步时master和slave都是非阻塞的。

利用Redis的主从复制特性,可以实现以下功能:1) 实现读写分离,如用主服务实现读操作,从服务实现写操作;2) 备份数据,利用主从服务器的方便性来备份,专门做台从服务器用于备份功能。

2 系统模型

2.1 订阅推送系统简介

订阅推送系统主要面向两类不同的用户:订阅号用户和移动端用户。

2.1.1 订阅号用户

订阅号用户主要是在该系统中使用富文本编辑器编辑图文消息,再将图文消息推送给其订阅用户。

目前有三种类别的订阅号用户,其分类依据是:用户默认是否订阅、用户能否退订。这三种类别分别是:1) 官方号。用户默认订阅,且不能退订;2) 合作号。用户默认订阅,但可以退订;3) 第三方申请号。用户默认不订阅,必须主动订阅。

而对于不同类别的订阅号,其推送的"目标用户"也不同:1) 官方号会推送给所有用户;2) 合作号则推送给除主动退订的用户外的所有用户;3) 第三方申请号则仅推送给主动订阅的用户。

2.1.2 移动端用户

移动端用户可以订阅一个或多个订阅号。移动端用户主要是阅读订阅号所推送过来的图文消息。

2.2 系统架构

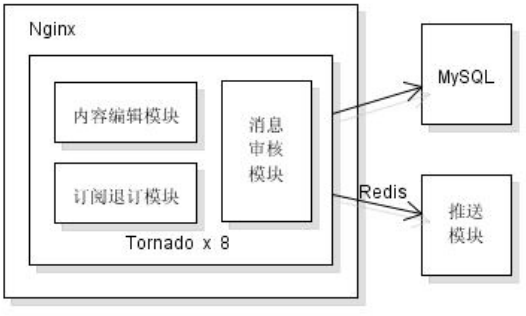

图1 订阅推送系统架构图

本系统是基于著名的Python Web框架Tornado进行开发的。正式部署环境下,启动了8个Tornado进程共同提供web服务,并使用Nginx作为其反向代理(提供静态资源服务和请求转发服务)。使用MySQL作为关系数据库,存储订阅/退订关系、图文消息的数据内容等。使用Redis作为key-value数据库,主要用于数据统计、session缓存、进程间协作通信等。

由图1所示,该系统主要分为四大模块,各模块的主要功能如下:

1) 内容编辑模块:供订阅号用户在该平台登录后,使用富文本编辑器编辑图文消息,作为推送素材。

2) 消息审核模块:为了防止订阅号用户将一些不合规的内容推送给订阅用户,故图文消息在推送之前,必须申请审核,经审核通过的图文消息才可以推送给用户。

3) 订阅退订模块:每个订阅号用户所推送的图文消息,仅会推送给订阅了该订阅号的用户,用户可以订阅或退订某订阅号。该模块负责维护订阅号和用户之间的订阅/退订关系。

4) 推送模块:考虑到推送操作耗时较长,为了避免Tornado长期阻塞而无法处理其他请求,故将推送操作的任务交由另一进程PushProcess完成。Tornado进程与推送进程PushProcess是一个生产者-消费者模型,两者间通过Redis消息队列进行协作通信。

3 Redis的应用

3.1 使用Redis缓存作为多进程间的数据共享:服务端Session

众所周知,HTTP协议是无状态的。在Web开发中,主要通过服务端session和客户端cookie这两大技术来保存用户登陆信息。在单进程下,可以将session保存在服务器进程的进程空间中。但是,实际生产环境中,单个进程所能提供的最大服务容量往往有限,故需要在单台服务器中开启多个进程,或者在多台服务器中开启多个进程,通过多进程同时提供服务以提升系统的最大服务容量。此时,就不能再将服务端的session存储于某个进程内,而需要将session存储在所有进程都能够共享访问到的位置,而Redis恰好就能提供这种服务。

本系统的部署环境中,共启动了8个Tornado进程提供Web服务,并使用Nginx作为其反向代理。每个Http请求到达后,服务器进程会从根据请求中的cookie信息来判断该用户是否已经登陆。若用户还未登陆,则跳转登陆页面,当用户成功登陆后,会在Re-

dis中保存该用户的登陆信息,并设置相应信息至用户浏览器的cookie中,以便在后续的Http请求中带上相应的cookie信息 若用户已经成功登陆,则服务器进程能够根据cookie中所带来的相关信息,从Redis中找到该用户的相关信息。

3.2 使用Redis消息队列作为协作进程间的通信

根据高内聚低耦合的程序设计原则,应尽量让不同的进程提供不同的服务。本系统中,Nginx进程提供静态资源服务和请求转发服务,Tornado进程提供动态资源服务和业务逻辑处理服务。考虑到实际的推送操作可能会耗时较长,特别是第1、2种类型的订阅号,其推送的目标用户多达数百万,因此,我们将推送业务的处理操作从Tornado进程中提取出来,交由另一单独的进程Push-Process负责。

如此一来,就涉及Tornado进程与推送进程PushProcess间的协作通信,本系统中通过Redis消息队列实现。当订阅号用户提交推送请求后,由Tornado进程将本次推送的关键信息写入Redis消息队列中。而推送进程PushProcess则长期阻塞监听该消息队列,一旦有推送请求入队,推送进程就能成功获取该请求,并进行实际推送操作,可能会导致该进程一段时间内繁忙,而Tornado进程则不会受到影响。另外,也可同时启动多个推送进程,以共同提供推送服务。

3.3 使用Redis Bitmap作为基于user id的数据统计

对于移动APP应用来说,进行一些常见的数据统计和分析非常必要,能够帮助提升产品。在订阅推送系统中,一个常见的需求就是统计每次推送的客户端打开率。那么,就需要对每条消息的推送用户总数,以及用户收到消息后的打开情况进行统计,即统计每条消息推送给哪些用户,这些用户在接收到消息后是否打开阅读了。

本系统中,使用两个Redis Bitmap来记录每条消息的推送情况。Bitmap是一种基于偏移位进行置0、置1的数据结构,本系统中使用用户的user id作为偏移位。

第一个Bitmap用于在推送过程中,每推送一个用户,就在该用户的相应user id位置1。最后,只要统计Bitmap总共有多少位置为1,即可知道该条消息共推送了多少个用户。另外,即使在推送过程中发生意外导致推送终止,在修复故障继续推送时,也可根据该Bitmap得知该条消息是否已经推送给某一用户,这样也就不会导致对同一个用户多次推送同一条消息。

第二个Bitmap用于在用户打开阅读推送消息时,触发Http请求,服务器在收到该请求后,在该Bitmap的相应用户user id位置1,这样一来,也就可以知道哪些用户打开阅读过该消息,以及总共有多少个用户阅读过该消息。

4 结束语

总之,Redis数据库凭其自身的特点以及高效的读写效率在频繁的大数据处理上有一定的优势。Redis所提供的丰富的数据结构,能够灵活地应用在不同进程间的协作通信,以及不同编程语言间的程序设计。

参考文献:

[1] 维基百科. Redis [EB/OL]. (2014-06-23). http://zh.wikipedia.org/wiki/Redis.

[2] 王心妍. Memcached和Redis在高速缓存方面的应用[J]. 无线互联科技, 2012(9):8-9.