文章编号：2096-1472(2021)-10-10-04
DOI:10.19644/j.cnki.issn2096-1472.2021.010.003

# 分布式论坛系统解决方案的设计

张 娜，白金峰

(大连东软信息学院，辽宁 大连 116023)
✉zhangna@neusoft.edu.cn; baijinfeng1202@126.com

**摘 要**：当今互联网应用与服务的主要问题就是如何支撑庞大的用户请求数量及并行处理海量数据。本文在详细调研了目前论坛系统处理高并发请求和海量数据的解决方案后，针对论坛系统的用户和服务特性，设计了一套分布式论坛系统的解决方案。该系统基于Spring Cloud(用于构建微服务开发的框架集合)微服务架构，为提升用户体验，采用了前后端分离的B/S(浏览器/服务器)模式；极大地提升了处理并发请求的能力，并且方便管理员对系统和内容进行管理；拓展性强，可用性高，运行稳定，使用简单，能够满足论坛用户和管理者的需求。

**关键词**：分布式；Spring Cloud；微服务；论坛系统
**中图分类号**：TP391 **文献标识码**：A

# Design of Distributed Forum System Solution

ZHANG Na, BAI Jinfeng

( *Dalian Neusoft University of Information*, *Dalian* 116023, *China*)
✉zhangna@neusoft.edu.cn; baijinfeng1202@126.com

**Abstract:** Nowadays, a main problem of Internet applications and services is how to support the huge number of user requests and to process the massive data at the same time. After a detailed investigation of solutions to handling high concurrent requests and massive data for the current forum system, this paper proposes to design a set of distributed forum system solutions according to users and service characteristics of the forum system. The system is based on the Spring Cloud (a framework collection for microservice development) Microservice Architecture. In order to improve user experience, a B/S (browser/server) model that separates the front and back ends is adopted. The system greatly improves the ability of processing concurrent requests, and facilitates administrators to manage the system and contents. With its strong expansibility, high availability, stable operation and user-friendliness, the system can meet the needs of forum users and administrators.

**Keywords:** distributed; Spring Cloud; microservice; forum system



## 1 引言(Introduction)

随着当今计算机技术的快速发展，互联网应用与服务最主要的问题就是如何处理海量的数据和支撑数以百万计的用户。目前分布式计算系统十分火爆，各种各样的分布式架构也层出不穷，但解决云计算工程中实际具体问题的论文和研究成果依然较少。

本文提出的分布式论坛系统主要解决两大问题：一是分布式技术；二是论坛系统的功能可用性。众所周知，目前Web服务架构中比较火的技术就是微服务+分布式部署结构[1]，通过该结构可以明显地改善系统的性能和可用性，所以本论文的重点研究方向之一就是如何架设微服务+分布式系统，以及解决分布式系统中容易遇到的问题。除此之外，分布式系统的功能开发方式与传统结构系统的功能开发方式不同，开发者还需要了解如何将服务进行拆分，通过模块的方式实现系统

基金项目：基于区块链技术的数字普惠金融助力中小微企业经济复苏发展研究(2020dlskzd330)。

的功能并且保证系统功能可用性。分布式论坛系统适用于用户量较大、每秒请求并发量较高和数据量较大的场景，它不仅可以优化用户体验，还可以减轻单一服务实例的负载。

## 2　系统需求(System requirements)

论坛分为用户端和管理员端。如图1所示为用户端，用户在未登录之前可以浏览板块、帖子，回复和浏览其他用户主页；若用户没有账号可以进行注册；用户登录后除可以进行未登录之前的操作外，还可以收藏帖子、发表新帖或者回复帖子、点赞、关注用户、修改个人信息和签到等。

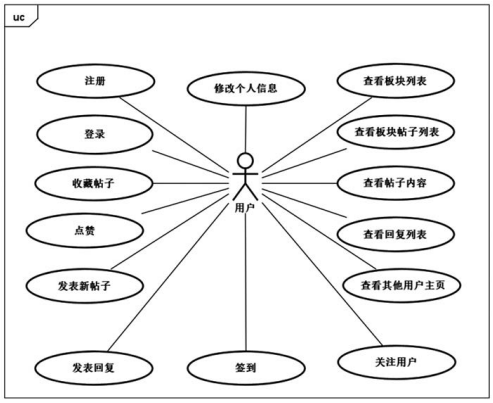


图1 用户用例描述

Fig.1 User use case description

管理员端功能用例如图2所示，管理员登录系统后，可以查看当前用户端的信息，同时还可以对板块、帖子、回复进行管理，对用户进行封禁和重置密码等操作；管理员还可以向系统中添加新的管理员账户来帮助自己管理系统，同时可以看到自己和其他管理员的操作记录，若想要收回其他管理员的权限，则可以停用某个管理员账户，同时也支持永久收回账户(删除账户)。

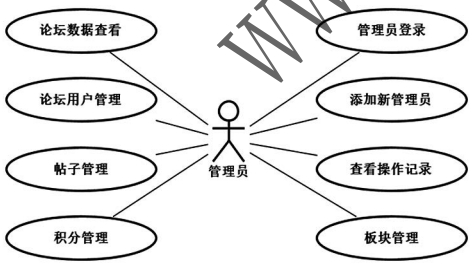


图2 管理员用例描述

Fig.2 Administrator use case description

同时，系统还有一些非用户性功能，例如定期收集在线用户数量，收集每日访问量、每日新帖发表量等，如图3所示。

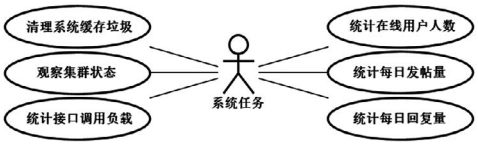


图3 系统用例描述

Fig.3 System use case description

## 3　系统设计(System design)

### 3.1　技术路线

系统采用基于B/S架构的前后端分离技术，前后端的通信使用HTTP协议，数据格式使用JSON。前端使用开放框架Vue.js，通过Vue.js将页面分为若干个组件，组件之间可以通信，当有数据变化时，只更新相应的组件。后端框架使用Spring Boot，通过此框架可以迅速搭建应用环境和进行开发，提升开发效率。

Spring Cloud[2]是一个包含了分布式系统全部解决方案的技术栈，包括分布式的注册中心、服务熔断监控、服务降级、路由网关、服务间调用等。Spring Cloud通过Spring Boot技术来实现，所以在分布式系统的集成上选用Spring Boot架设微服务组件，使用Spring Cloud技术来进行集成是非常合适的。Cloud Config用来进行配置管理，通过Cloud Config可以将每个微服务组件的配置集中管理起来，避免配置过于分散，方便管理和修改。

RabbitMQ[3]是一款消息队列中间件，该中间件可以将服务进行解耦，并且可以使两个微服务异步完成一个任务。系统使用Redis作为缓存服务器，使用MyBatis作为数据库访问层，通过MyCat进行负载均衡分发，使数据库连接请求均匀地分发到每个数据库上面。

### 3.2　系统架构

(1)整体架构

系统整体结构如图4所示。前端Web服务器使用Nginx进行内容分发。后端使用微服务分布式集群[4]的方式：Gateway为后端应用的入口，负责将请求分发给相应的微服务模块；Eureka中注册了集群中全部实例信息，方便实例间进行调用；CONFIG为集群提供配置服务，管理着集群中每一个应用的配置信息；Dash Board用来收集集群的健康信息，绘制集群负载曲线图，方便运维人员实时关注集群的运行状态；Spring Boot Admin负责管理集群中每一个微服务模块的资源用量，如内存消耗、线程池大小、连接数量等；集群内的微服务模块使用内置的Tomcat作为容器，使用Feign和RabbitMQ进行服务间的调用与重试。数据方面，系统使用MySQL集群提供数据库服务，MySQL集群通过主从复制的方式和分库的方式提高系统的可用性和数据吞吐量；同时系统还采用Redis集群作为缓存服务器，对于一些变化较少而访问量较大的数据进行缓存处理，从而减轻数据库的压力，提升系统运行效率。

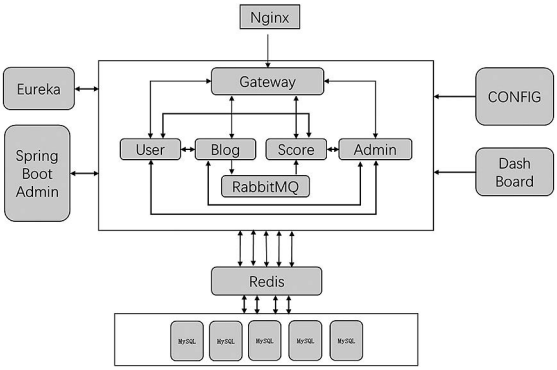


图4 系统整体结构

Fig.4 Overall structure of the system

(2)节点集群

多服务节点的分布式模式如图5所示。系统中某一微服务的节点可以为多个，它们对外部显示仍为一个服务，系统进行调用时会自动进行负载均衡，选取节点中负载最小的实例进行调用，这样搭建的集群系统吞吐量高，可以承受大量的并发请求，并且系统可用性也大大提高：即使系统中某一实例的节点宕机，其他节点仍能提供服务，当Eureka检测到服务宕机后，就会将其从注册中心中剔除，防止调用失败。

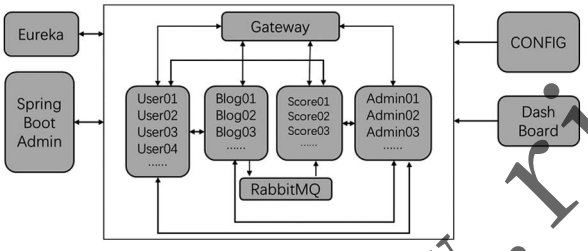


图5 多节点模式

Fig.5 Multi-node mode

(3)缓存架构

缓存是提高系统吞吐量和减轻数据库压力的重要措施，本系统为了保证缓存的可用性采用了Redis Cluster的Redis集群模式[5]，选取了四个Redis节点，两个主节点，两个从节点，当需要加入缓存时，通过哈希算法对KEY进行计算，然后存到对应的缓存容器中。当Redis Cluster中某一个主节点由于某种原因宕机时，其后备的从节点会迅速升级为主节点并继续提供服务，如图6所示。

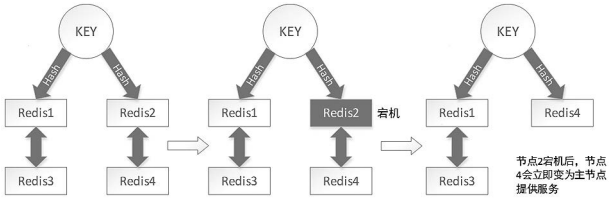


图6 Redis Cluster集群

Fig.6 Redis Cluster

### 3.3 数据库设计

通过分析系统业务需求，系统的数据库设计如图7所示。

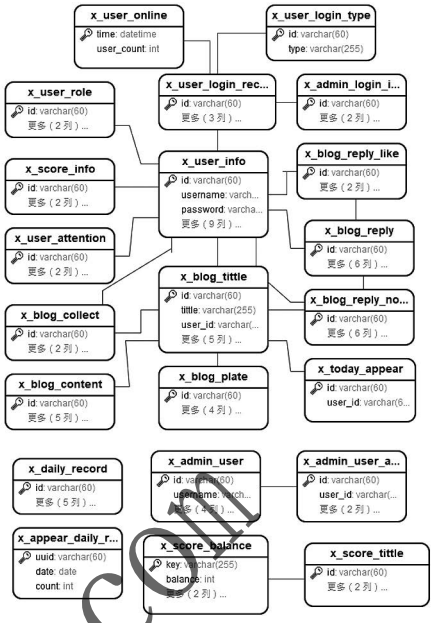


图7 数据库设计

Fig.7 Database design

### 3.4 对象设计

本系统后端使用三层架构进行设计，Controller负责拦截请求路径，请求数据与响应数据的绑定，验证数据合法性，调用Service进行业务处理；Service主要负责业务逻辑，对请求数据进行处理，将处理后的结果返回给Controller或者存入数据库；Dao层是实体数据访问层，主要负责对持久化数据的增删改查。在此基础上，还加入了微服务之间进行访问的Client、消息中间件RabbitMQ和缓存中间件Redis，Client主要用来与其他微服务进行通信，传输和获取业务数据；Rabbit用来在多个服务之间传递消息和进行失败业务的重试；Redis用来对热点数据进行缓存和分布式session管理。

### 3.5 控制流程设计

(1)前端控制流程

用户在页面上点击按钮或切换页面时会触发相应的事件，在State中对应的Action就会产生动作，将请求数据包装好，通过Ajax发送HTTP请求。请求到达Nginx服务器，由Nginx进行反向代理[6]，通过负载均衡算法发送给后端服务器网关，后端服务器处理请求。当后端服务器处理完毕后，将请求通过网关返回给Nginx，Nginx再将HTTP响应返回给Ajax，Ajax将响应体分解，取出需要的部分返回给Action，最后Action将数据绑定到指定组件上。

(2)后端控制流程

当Gateway接收到请求后，会对请求进行校验，包括请求头校验、数据合法性校验、用户权限校验等，对不符合标准的请求不予放行，最后Gateway会将过滤后的请求通过路

由的方式分发到每一个微服务上，微服务对请求进行处理[7]，通过消息中间件RabbitMQ进行消息分发和失败任务重试。部分请求可能需要缓存的支持，它们可以从缓存集群Redis Cluster中获取数据和更新缓存。如果需要对持久化数据进行增删改查操作，微服务可以通过MyCat进行路由分发，到MySQL集群中相应的数据库中操作数据。当微服务将业务逻辑处理完毕后，将响应消息响应给网关，网关会对响应中的敏感数据进行加密，然后返回给调用方。

### 3.6 异常处理设计

一个健壮的系统应该对业务逻辑异常或者其他一些常见的异常有着可应对的处理方式。本系统将系统内异常分为应用程序异常、业务处理异常、未授权异常、资源未找到异常、远程调用权限不足异常、参数异常、远程服务异常共七种可预见异常。

## 4 系统实现(System implementation)

### 4.1 核心功能实现

系统的核心功能[8]依据系统设计方案，采用所选技术路线完全可以实现，由于本文的重点在于分布式论坛系统解决方案的设计，因此不再给出具体编码内容和实现截图。

### 4.2 分布式功能实现

(1)配置集中管理

由于系统采用分布式结构，如果每个服务的配置文件都由自己携带，配置文件过于分散，不利于管理和修改。为了解决这些问题，系统将所有配置文件集中在一起，使用Git进行版本控制，便于修改和管理。当每一个微服务模块启动时，通过指定的应用名称和配置文件名称，从配置服务器直接拉取其所需的配置即可。

(2)代码自动编译、打包

系统的代码和jar包依赖由maven进行管理，故可以通过maven命令进行编译和打包处理。

(3)自动化集群启动

集群支持自动化启动，通过脚本命令直接启动服务，启动时仅需要指定配置文件即可。

(4)系统健康监视

为了保证系统健康运行，项目中加入了集群监视功能，主要监视内容为接口负载情况、微服务健康状态、数据库连接数、微服务线程池运行线程数、内存使用率等关键指标。

## 5 系统测试(System test)

### 5.1 功能与非功能测试

系统使用迭代模式进行开发，在模块开发过程中进行了业务测试、单元测试，在模块间整合时进行了集成测试和点对点测试，最后在整个系统完成后又进行了全面的功能测试，同时对系统性能进行了压力测试。

发帖功能主要测试系统对用户做出操作是否能给予正确的响应和提示；点赞功能主要测试用户点赞记录是否并发操作幂等性；关注用户功能主要的测试点是并发幂等性和关注、取消关注顺序问题；删除帖子功能的主要测试点为当管理员删除帖子后，帖子的评论、用户收藏、评论通知等信息是否会同步删除。

性能测试采用分布式集群方式进行，业务节点均为两个节点，非业务节点均为一个节点。集群测试测试了单一节点模式下服务宕机后的服务情况和分布式模式下节点宕机后的服务情况。

### 5.2 测试结果

系统共计设计了功能性测试用例37 个，成功执行37 个；设计了非功能性测试11 个，成功执行11 个；设计了性能测试共10 项，在两个服务节点的状态下，服务成功率达99%，系统资源占用基本稳定，没有内存泄露和CPU占用率过高的情况；设计集群测试用例2 个，成功执行2 个。

## 6 结论(Conclusion)

本文提出的分布式论坛系统与传统结构的论坛系统相比，能承载更高的用户量，系统可用性也大大提高，不会因为某一个点出现问题而导致整个系统不可访问。本系统还具备强稳定性、高容灾性、基于云配置、完善的异常捕获与处理、数据更安全、界面更简洁等优势。

## 参考文献(References)

[1] 李春阳,刘迪,崔蔚,等.基于微服务架构的统一应用开发平台[J].计算机系统应用,2017,26(4):43−48.

[2] 王方旭.基于Spring Cloud和Docker的微服务架构设计[J].中国信息化,2018(3):53−55.

[3] 徐震,焦文彬.RabbitMQ小消息确认机制优化[J].计算机系统应用,2018,27(3):252−257.

[4] 陈乐,余粟,王盟.基于分布式集群的高可用日志分析系统的设计[J].中国电子科学研究院学报,2020,15(5):420−426.

[5] 张杰,刘凯,周立军.采用Redis高并发应用系统设计与实现方法[J].计算机与数字工程,2020,48(5):1222−1226.

[6] 戴伟,马明栋,王得玉.基于Nginx的负载均衡技术研究与优化[J].计算机技术与发展,2019,29(3):77−80.

[7] STEFANO T. Architect blockchain applications as microservices[J]. MSDN Magazine, 2018, 33(9):24−30.

[8] 高云峰.对基于Java的BBS论坛系统的设计与实现分析[J].电子技术与软件工程,2018(04):202.

## 作者简介：

张　娜(1978−)，女，硕士，教授.研究领域：Java Web前后端开发.

白金峰(1997−)，男，本科生.研究领域：分布式系统开发.