**武汉大学计算机学院**

**本科生课程设计报告**

**网络与分布式计算实验**

专 业 名 称 ：软件工程

课 程 名 称 ：网络与分布式计算实验

指 导 教 师 一：胡启平 职称：教授

学 生 学 号 ：2017302580198

学 生 姓 名 ：邹鑫

二○二○年 五 月

**郑 重 声 明**

本人呈交的设计报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本设计报告不包含他人享有著作权的内容。对本设计报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本设计报告的知识产权归属于培养单位。

本人签名： 日期：

摘要

（黑体小2）

XXXX实验的实验目的是XXXX。

实验设计主要遵循XXXX。

实验内容主要包括：

实验结论为XXXX

……

……

……

**关键词：**关键词1；关键词2；关键词3

（黑体小4） （宋体小4）

目录

# 1实验目的和意义

## 实验目的

本次实验的目的是编写一个网络应用程序的客户端和服务器端。在客户端和

服务器端的交互中，采用JWT实现单点登录。在用户调用微服务时，截取JWT，并根据JWT和存储的权限判断用户是否有权限使用指定的微服务。其中，用户标识和用户权限存放在Apache Ignite中；客户端使用Flutter编写。

## 实验意义

作为一门网络与分布式计算的实验课，此次实验可以让我们更加深入地了解

网络应用程序的前后端交互过程；更深入地了解目前最流行的跨域身份验证解决方案——JWT(Json Web Token)；让我们初步使用Apache Ignite——一个以内存为中心的分布式数据库、缓存和处理平台；学会初步使用Flutter——Google 开源的帮助开发者通过一套代码库高效构建多平台精美应用，支持移动、Web、桌面和嵌入式平台的UI工具包。

本次实验可以让我们了解和初步使用上述技术，对分布式计算布局与应用有更深层次的理解。

# 2技术简介

本次实验主要采用的技术有——  
● Flutter进行移动应用的前端开发

* 使用SpringBoot框架进行服务器端的开发
* 使用JWT技术来实行单点登录和服务请求认证
* 使用Apache Ignite作为数据库存储数据

## 2.1 Flutter简介[1]

Flutter 是 Google 开源的 UI 工具包，帮助开发者通过一套代码库高效构建多平台精美应用，支持移动、Web、桌面和嵌入式平台。

Flutter 可以方便的加入现有的工程中。在全世界，Flutter正在被越来越多的开发者和组织使用，并且 Flutter是完全免费、开源的。它也是构建未来的 Google Fuchsia 应用的主要方式。

Flutter 组件采用现代响应式框架构建，这是从React中获得的灵感，中心思想是用组件(widget)构建你的UI。 组件描述了在给定其当前配置和状态时他们显示的样子。当组件状态改变，组件会重构它的描述(description)，Flutter 会对比之前的描述， 以确定底层渲染树从当前状态转换到下一个状态所需要的最小更改。

## 2.2 Spring Boot简介[2]

SpringBoot是由Pivotal团队在2013年开始研发、2014年4月发布第一个版本的全新开源的轻量级框架。它基于Spring4.0设计，不仅继承了Spring框架原有的优秀特性，而且还通过简化配置来进一步简化了Spring应用的整个搭建和开发过程。另外SpringBoot通过集成大量的框架使得依赖包的版本冲突，以及引用的不稳定性等问题得到了很好的解决。

特点编辑

SpringBoot所具备的特征有：

（1）可以创建独立的Spring应用程序，并且基于其Maven或Gradle插件，可以创建可执行的JARs和WARs；

（2）内嵌Tomcat或Jetty等Servlet容器；

（3）提供自动配置的“starter”项目对象模型（POMS）以简化Maven配置；

（4）尽可能自动配置Spring容器；

（5）提供准备好的特性，如指标、健康检查和外部化配置；

（6）绝对没有代码生成，不需要XML配置。

## 2.3 Apache Ignite简介[3]

Ignite是一个以内存为中心的，可持久化、强一致和高可用的，拥有强大的SQL、键-值存储及相关的API的数据平台。

Ignite有许多特点：

● 固化内存：Ignite的固化内存组件不仅仅将内存作为一个缓存层，还视为一个全功能的存储层。这意味着可以按需将持久化打开或者关闭。如果持久化关闭，那么Ignite就可以作为一个分布式的内存数据库或者内存数据网格，这完全取决于使用SQL和键-值API的喜好。如果持久化打开，那么Ignite就成为一个分布式的，可水平扩展的数据库，它会保证完整的数据一致性以及集群故障的可恢复能力。

● ACID兼容：存储在Ignite中的数据，在内存和磁盘上是同时支持ACID的，使Ignite成为一个强一致的系统，Ignite可以在整个网络的多台服务器上保持事务。

● 完整的SQL支持：Ignite提供了完整的SQL、DDL和DML的支持，可以使用纯SQL而不用写代码与Ignite进行交互，这意味着只使用SQL就可以创建表和索引，以及插入、更新和查询数据。有这个完整的SQL支持，Ignite就可以作为一种分布式SQL数据库。

● 并置处理：大多数传统数据库是以客户机-服务器的模式运行的，这意味着数据必须发给客户端进行处理，这个方式需要在客户端和服务端之间进行大量的数据移动，通常来说不可扩展。而Ignite使用了另外一种方式，可以将轻量级的计算发给数据，即数据的并置计算，从结果上来说，Ignite扩展性更好，并且使数据移动最小化。

● 可扩展性和持久性：Ignite是一个弹性的、可水平扩展的分布式系统，它支持按需地添加和删除节点，Ignite还可以存储数据的多个副本，这样可以使集群从部分故障中恢复。如果打开了持久化，那么Ignite中存储的数据可以在集群的完全故障中恢复。Ignite集群重启会非常快，因为数据从磁盘上获取，瞬间就具有了可操作性。从结果上来说，数据不需要在处理之前预加载到内存中，而Ignite会缓慢地恢复内存级的性能。

## 2.4 JWT简介[4]

JWT是目前最流行的跨域身份验证解决方案。

JWT分为3部分——Header、Payload和Signature。

在使用中，JWT一般是在用户登录时服务器生成之后发送给客户端的，客户端在其每次请求中将JWT串放入其中，以此来使服务器得以鉴别其身份。

### 2.4.1 Header

Header部分是JWT的头信息，说明了JWT使用的hash算法，并说明此串的类型是JWT，例如：

{

"alg": "HS256",

"typ": "JWT"

}

此串就是JWT的头部信息，表明这个串是JWT串，且使用的hash算法是HS256算法。

### 2.4.2 Payload

Payload部分存放的是一些声明，以键值对的形式说明某些信息。其中，iss(jwt签发者)、sub(jwt所面向的用户)、aud(接收jwt的一方)、exp(jwt的过期时间，这个过期时间必须要大于签发时间)、nbf(jwt的生效起始时间)、iat(jwt的签发时间)、jti(jwt的唯一身份标识，主要用来作为一次性token,从而回避重放攻击)这几部分是jwt中预先定义的，是推荐使用但非必须的。用户也可以定义自己的声明。例如：

{

"iss": "id123",

"name": "John Doe",

"iat": 1516239022,

"exp": 1516239099

}

在上述例子中，iss表明了签发者是id123；name是自定义的声明，具体意义与其定义有关；iat表明了该JWT的签发时间，exp表明了该JWT的失效时间，只要时间超过exp，该JWT就失效了。

### 2.4.3 Signature

此部分是一个字符串。

假设使用BASE64URL编码方式将Header编码成的字符串为header=ABC，将Payload编码成的字符串为payload=XYZ，则Signature则是将字符串“ABC.XYZ”采用Header中表明的hash算法进行hash运算得到的字符串，假设为signature=OPQ，则JWT则为字符串“ABC.XYZ.OPQ”。

对Header和Payload进行签名的目的是——如果在网络传输的过程中，JWT被第三方更改了或者由于网络原因改变了，那么服务器可以知道JWT串出现了问题，从而可以要求客户端重新登录。

# 3实验设计

本次实验是要整体了解分布式计算和网络工程的一些简单应用，而主要的任

务有——

* 使用Flutter编写客户端
* 使用Spring Boot编写服务器端
* 使用JWT实现单点登录的功能，以及在请求服务时验证权限
* 使用Apache Ignite作为数据库进行存储。

所以，本次实验从整体来说分为客户端和服务器端两大部分，而服务器端又

可以分为逻辑处理层和数据持久化层。

所以，我把本次实验分为X个模块。