**北京理工大学继续教育暨现代远程教育学院**

毕业设计（论文）

**类 别**：专升本

**专 业**：计算机科学与技术

**班 级**：北京理工大学校本部

(教学站)

**姓 名**：雷明

**毕业设计题目**：准妈妈孕期信息管理服务系统

**指导教师姓名**：张丽娜

**负责人签字：**

2018 **年** 10 **月** 10 **日**

摘 要

本文基于微服务、大数据分析和机器学习这三大热门技术，阐述本服务系统如何在现代医疗服务中实现信息管理及系统服务。

首先，阐述微服务、大数据分析和机器学习在“互联网+医疗”行业的发展现状。

其次，从系统技术架构的角度，对本服务系统进行深入地层次分析阐述。

第三，结合本服务系统中应用到的微服务、大数据分析组件进行详细分析，以及应用到的机器学习算法原理进行深入说明，并对本服务管理平台的数据处理流程，包括数据采集，数据处理，数据预测，数据反馈等核心步骤进行技术剖析， 最终得出本服务管理平台的社会价值。

最后，对微服务、大数据和机器学习在“互联网+医疗”行业的发展前景做出分析。

**关键词：**互联网+医疗；微服务；大数据分析；机器学习；

ABSTRACT

Based on Micro-services, Big Data and Machine learning, this paper mainly described and analyzed the function and characteristics of these three key technologies in Modern Medical Service System.

First of all, we expatiate the status of Micro-services, Big Data analysis and Machine learning in today’s "Internet + Medical" industry

Second, we analyzed from the different levels in infra-architecture of this system

Third, illustrate and applicate these three key technologies’ combination in this service system, algorithms for further instructions of data processing flow to our service management platform, including Data acquisition, Data processing, Data prediction, Data feedback etc. core steps. Then, we concluded the social value of the service management platform.

Finally, the development prospect of Micro-services, Big Data and Machine learning in the "Internet + Medical" industry is analyzed.

**Keywords**: Internet + Medical care; Micro-service; Big Data analysis; Machine learning;

目 录

[摘 要 2](#_Toc530943520)

[ABSTRACT 3](#_Toc530943521)

[目 录 4](#_Toc530943522)

[前 言 5](#_Toc530943523)

[**1** **行业发展现状** 6](#_Toc530943524)

[**2** **系统技术架构设计与层次分析** 7](#_Toc530943525)

[**2.1** **系统技术架构设计** 7](#_Toc530943526)

[**2.2** **系统技术架构层次分析** 8](#_Toc530943527)

[2.2.1 网络层 8](#_Toc530943528)

[2.2.2 缓存层 8](#_Toc530943529)

[2.2.3 应用层 8](#_Toc530943530)

[2.2.4 数据计算层 9](#_Toc530943531)

[2.2.5 算法层 11](#_Toc530943532)

[2.2.6 数据存储层 11](#_Toc530943533)

[**3** **核心技术在系统中应用** 13](#_Toc530943534)

[**3.1** **微服务** 13](#_Toc530943535)

[3.1.1 技术特点 13](#_Toc530943536)

[3.1.2 实践应用 13](#_Toc530943537)

[**3.2** **大数据** 16](#_Toc530943538)

[3.2.1 Kafka和Storm技术特点 16](#_Toc530943539)

[3.2.2 实践应用 16](#_Toc530943540)

[**3.3** **机器学习** 17](#_Toc530943541)

[3.3.1 建模 17](#_Toc530943542)

[3.3.2 实例说明 18](#_Toc530943543)

[**4** **发展前景** 21](#_Toc530943544)

[结 论 22](#_Toc530943545)

[致 谢 23](#_Toc530943546)

[参考文献 24](#_Toc530943547)

[附 录 25](#_Toc530943548)

前 言

在当今的互联网时代，微服务、大数据分析处理和机器学习的广泛应用已经深入到所有行业，而互联网医疗作为互联网在医疗行业的新应用，代表了医疗行业新的发展方向，有利于解决中国医疗资源不平衡和人们日益增加的健康医疗需求之间的矛盾，是国家积极引导和支持的医疗发展模式。

近年来，随着国家放开“二胎政策”，待产准妈妈逐年增加，全国大大小小的妇幼保健院的办公压力随之也逐年增加。由于这些妇幼保健院多数是早期建造的，在网络硬件设施上，承受着很大负荷。所以，移动智能办公在当今互联网时代，体现着举足轻重的作用，凭借其智能、易操作、反应快速、承载信息量大等多方面的优势，越来越受人们的青睐。同时，微服务技术的飞速发展日趋完善，很大程序推动了移动智能办工的使用。在互联网时代，以及刚刚产生的物联网都得到广泛应用。

本课题以准妈妈孕期信息服务管理系统作为平台，搭载智能手机、智能平板等移动终端设备，实现对准妈妈孕期的身体状态、孕检信息等数据的采集[1]，在PC端通过服务平台，实现对采集数据的预测，达到对准妈妈状况的实时监控和各种预测的目的。

本服务系统可基于多种类型的硬件设备及不同的操作系统实现部署的多样化，并基于大数据组件的高兼容，高并发，高性能的处理能力，提高数据采集、数据清洗、数据计算和数据反馈等核心技术能力，以及利用机器学习算法，实现智能预测及信息的准确推送。

# **行业发展现状**

四年前，互联网医疗迎来了大热元年，但很快就遭遇资本寒冬。熬过生死线后，幸存的纯线上问诊难有盈利点，转型迫在眉睫。至于那些转换赛道死磕线下服务的项目，也很难说从轻问诊转型重服务何时才能奏效。2018年，沉寂两年的互联网医疗重回头条，一系列的政府动作正在释放利好信息。4月11日，李克强总理视察上海华山医院，称赞互联网远程医疗。一天之后，国务院常务会议通过了“互联网+医疗健康发展意见”。5天后国新办又公开介绍了相关“发展意见”精神。当月26日，卫健委发布“互联网+医疗健康”实施计划。在互联网医疗创业者看来，这意味着“互联网+医疗”首度从国家战略的层面被认可。

自从2014年“互联网+医疗”在短短两年内经历了过山车般的大起大落。在2018年“互联网+医疗”行业形式突然转好，再加上AlphaGo的横空出世，使得在“互联网+医疗”行业中应用机器学习、深度学习等算法技术趋于深入发展，更有可能将此技术应用于医疗行业，而且发展得会更加好。

随着各大医疗机构逐渐将此技术引入到医疗行业中，势必带来前所未有的发展和变革。自2008年起至2017年底，我国的出生人口逐年增加，再加之2016年1月1日起我国全面放开“二胎政策”，使得准妈妈成为各大医院中一股庞大的医疗人群，不仅给各大医疗机构从人员到设备等带来沉重压力，而且在孕检过程中普遍出现“预约难”、“检查难”、程序繁琐等诸多问题。所以，我们需要智能技术以解决此期间带来的各种问题，这也正是本服务系统设计并产生的初衷。

准妈妈孕期信息服务管理系统作为平台，可通过智能手机、智能平板等移动终端设备实现对准妈妈孕期的身体状态、孕检信息等数据的采集[1]，在PC端通过本服务平台，实现对采集数据的预测，达到对准妈妈状况的实时监控和各种预测的目的，并各个阶段中所产生的数据进行采集、存储和处理，医疗大数据被提到重要位置。

随着人工智能、大数据、云计算的崛起，未来数据将扮演关键的作用，数据将成为如同水电煤一样的基础设施。因此，本服务系统志在利用微服务和大数据的技术实现并帮助准妈妈在孕期过程中了解更多的知识，简化各个阶段的检查步骤，利用机器学习方法实现对准妈妈怀孕期间健康情况的指导建议，最终实现碎片化的医疗数据整合，为准妈妈提供智能，全面的医疗服务。

# **系统技术架构设计与层次分析**

## **系统技术架构设计**

本服务系统基于6层架构设计予以实现，如图2–1。采用这样的架构设计可以使得开发人员在本层和相邻层的接口定义清晰完整的前提下，专注理解本层的功能和技术，调用其相邻层所提供的程序接口即可。

系统开发过程中，功能需求不断变化，即使业务逻辑层中的需求发生变化，表现层以及数据访问层也不需要变化，大大降低系统各层之间的依赖，充分利用现有功能程序组件。将具有相对独立功能的层应用于新系统的开发，保证新系统在开发过程中，能够将重点集中于辨识和实现应用系统特有的业务功能，最终缩短系统开发周期，提高系统的质量，更能适应当前互联网行业的架构设计理念。



图2–1

## **系统技术架构层次分析**

### **网络层**

使用Nginx实现反向代理及负载均衡，平均分担负载，即将请求分摊到多个操作单元上进行执行，例如Web服务器、FTP服务器、企业关键应用服务器和其它关键任务服务器等，从而共同完成工作任务。简单而言，就是当有2台或以上服务器时，根据规则随机地将请求分发到指定的服务器上处理，负载均衡配置一般都需要同时配置反向代理，通过反向代理跳转到负载均衡。

### **缓存层**

在本服务系统中，使用Redis作为数据缓存层的存储工具，其使用的目的：

1、将热点数据如用户登录信息等，放入缓存数据是为了让客户端很少或者甚至不访问数据库服务器进行的数据查询。在高并发情况下，最大程度降低对数据库服务器的访问压力。

2、防数据穿透，对缓存查询加锁。如果Key存在，就加锁。然后查询数据库入缓存，然后解锁。其他进程如果发现有锁就等待，然后等解锁后返回数据或者进入DB查询。

### **应用层**

在本服务系统中，应用层使用Spring Boot框架，并设置用户管理、信息管理、数据管理、系统管理和日志管理模块，如图2–2所示。用户管理模块用于维护着准妈妈和医师信息，并将热点数据如准妈妈和医师的登录信息等放置于缓存数据库中。信息管理模块提供对准妈妈和医师信息，以及准妈妈检查结果信息和孕检过程等流程信息的更新和修改功能。数据管理模块提供多维度的数据统计展示。系统管理模块提供孕检预约功能、信息管理功能和检查报告进度展示功能。日志管理模块负责收集各个功能模块执行过程的日志，并提供告警和多维度统计功能。



图2–2

### **数据计算层**

#### 实时流程

在本服务系统的实时流程中使用Kafka和Storm作为数据计算层的服务组件，如图2–3所示。



图2–3

在使用Kafka时，通过创建用户信息Topic[3]、检查预约Topic、检查告警Topic和用户反馈Topic，本服务系统可根据用户的业务请求将相应的请求数据放置于相应的Topic队列中，以供后序流程使用。

在使用Storm时，定制相应业务获取相应Topic队列中数据，依次进行数据清洗、格式转换、业务处理、生成日志和数据入库等操作。数据清洗用于处理无效的请求数据和垃圾数据，从而以减少占用不必要的数据处理资源。格式转换用于将Json格式的请求数据转换成关系型数据库如MySQL和搜索引擎如Elastic Search使用的数据结构。业务处理是针对应用层定制相应的业务数据的处理流程。生成日志即针对整个数据清洗层各个模块的处理操作而生成的处理日志，并进行实时监控，对发生异常的情况进行实时提醒。数据入库即当正常数据处理完成后，将有结果数据写入到最终的数据存储中。在本服务系统中，将写入到MySQL和Elastic Search。

#### 离线流程



图2–4

离线数据是目前整个数据开发的根本和基础，也是目前数据开发的主战场。在本服务系统的离线流程中包含了两大处理流程，如图2–4所示。

1、数据回溯即实时流程的全部处理过程。当服务系统发生异常，不能正常处理，造成Kafka中若干Topic队列出现积压，致使产出不正确的最终数据。在这种情况下，必须将一段时间内的原始数据通过离线流程进行数据回溯，把最终产出的数据进行数据补全，并重新写入到数据库中。而对于已经存在的数据，将进行数据更新，以供Web应用层进行正确的数据检索。

2、算法模型数据迭代，针对不同的数据如准妈妈每次检查报告数据和身体、饮食状况的数据分别进行相应业务数据挖掘，挖掘出更多报告中隐含的数据信息，并由专业的营养师结合全部数据信息对各个准妈妈给予更加专业地指导建议。

### **算法层**

之所以在本服务系统架构中设置算法层，其目的在于将专业的检查报告数据通过算法概括成点数据即综合值，并与标准值进行对比，而后再以趋势图表的方式进行展示。从而使准妈妈在没有医学基础的前提下，能大致了解每次孕检后身体变化情况。因此，可以看出算法在数据挖掘中起着很大的作用，而且也是整个数据架构的关键点，所有底层数据发挥作用都需要经过数据挖掘和计算框架输出。数据计算层既要通过不同算法满足不同需求的挖掘需要，又要根据数据需求尽可能的实时输出结果。

### **数据存储层**

在本服务系统中，采用MySQL作为数据的最终数据存储、Elastic Search作为Web应用层进行检索的数据搜索引擎。

#### MySQL



图2–5

从内容上来看，创建准妈妈的相关数据表如图2–5所示。准妈妈基础信息表、预约信息表、日常身体饮食情况记录表和孕检结果信息表及医师基础信息表等。准妈妈基础信息表存储在医院建档时准妈妈信息，如姓名、身份证号、籍贯、年龄、末次月经等信息。医师基础信息表存储医院在册的医师详细信息，准妈妈可根据自己的意向选择合适自己的医师，进行之后的每次孕检。预约信息表存储准妈妈预约下次孕检的时间与医师的信息。当准妈妈预约成功后直到当日检查，本服务系统以短信方式进行提醒，以免过期错过检查。日常身体饮食情况记录表存储准妈妈每天身体和饮食相应信息，这部分数据用于数据挖掘和预测业务，从中挖掘出更多的信息提供给准妈妈和准妈妈指定的医师。孕检结果信息表存储准妈妈每次孕检结果报告的数据，以及经过数据挖掘后的数据信息，并结合医师和营养师的建议，得出更为详细的数据报告。

#### Elastic Search

采用Elastic Search作为Web应用层的检索存储，是由于其具有实时分析的分布式搜索引擎，可将索引分拆成多个分片，每个分片可有零个或多个副本。集群中的每个数据节点都可承载一个或多个分片，并且协调和处理各种操作，负载及平衡路由在大多数情况下能够自动完成，可以扩展至上百台服务器，处理PB级别的结构化或非结构化数据，支持插件机制、分词插件、同步插件、Hadoop插件和可视化插件等。

# **核心技术在系统中应用**

## **微服务**

从微服务的独立性这一技术特点出发，深入浅出地阐述微服务技术在本服务系统中的应用。

### **技术特点**

微服务架构[4]就是将一个完整的应用从数据存储开始垂直拆分成多个不同的服务，**每个服务都能独立部署、独立维护、独立扩展，服务与服务之间通过RESTful API的方式互相调用**[5] 。通常情况下，类似系统服务的设计与开发都是把所有功能模块整合在一个项目中进行开发与部署，这样就会造成开发人员的工作交集过多，势必是有冲突产生，甚至会影响到开发过程的进度，最终导致项目延期等等各种情况。这正是区别与其他技术的特点，也正是如此，微服务技术是为了解耦以及服务复用而存在的。

### **实践应用**

结合本服务系统的实现，详细说明一下微服务架构在开发过程中的具体使用。在开发过程中，通常会把微服务技术的实现可以分成三个模块，如图3–1所示：

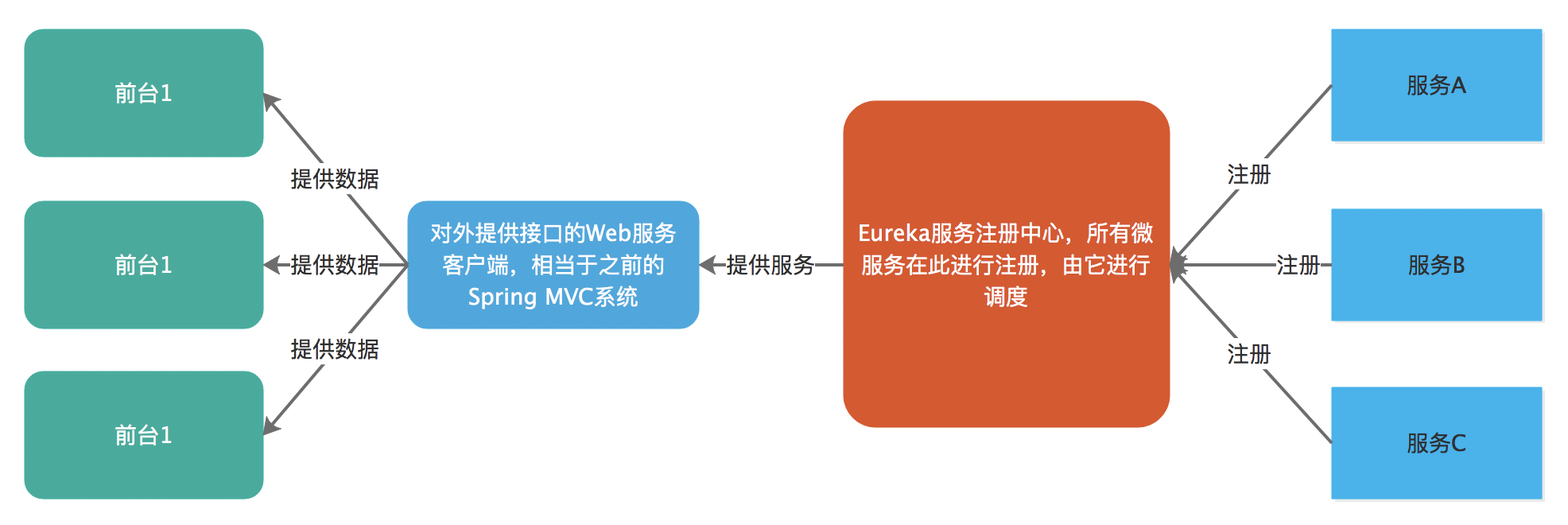


图3–1

从图3 - 1中我们可以看出，一个微服务架构可以分成三个模块，一个服务的注册中心，一个是服务的客户端（即属于Eureka Client项目），另一个则是提供所有微服务（具体的业务，如用户注册逻辑）的模块。当我们把所有的微服务注册到注册中心之后，我们需要建立一个属于Eureka Clinet的项目，可以通过Discovery Client中的API方法来调用所注册的服务。[6]



图3–2

在开发过程中，使用Eureka服务，将微服务模块连接起来，它是从属于Spring Cloud Netflix （包含服务发现、断路器和监控、智能路由、客户端负载均衡）的子项目。我们需要建立至少三个Spring Boot工程，一个是服务注册中心（Eureka Server），一个服务客户端（Eureka Client），还有一个是用户注册的微服务API（User Registion）。所以，对于其他的功能模块，我们同样只需要创建与用户注册微服务API类似的子项目，编写具体业务实现代码，便可实现服务的拆分。在部署时，将各个微服务API注册到服务注册中心如图3–2，既而通过服务客户端提供功能模块API的服务，这就是微服务架构的突出特点。

所以，从使用微服务架构技术的角度，本服务系统在开发上体现了简单、快捷、分层、业务扩展等技术特点，在部署上可不依赖如Nginx等负载均衡的技术工具，便可以快速地实现负载均衡和分布式等技术架构。这正是微服务在当今互联网技术公司中发展的原因，充分体现了微服务技术的强大，可以说是今后架构技术的重点发展方向。

下面将以采集血常规数据到推送至终端设备的整个数据流转过程为例进行详细说明。

在本服务系统中数据采集途径共有两种：一种是移动终端实时采集准妈妈身体状况及营养信息，另一种是医院检查部门将准妈妈各种检查报告推送到本服务平台。

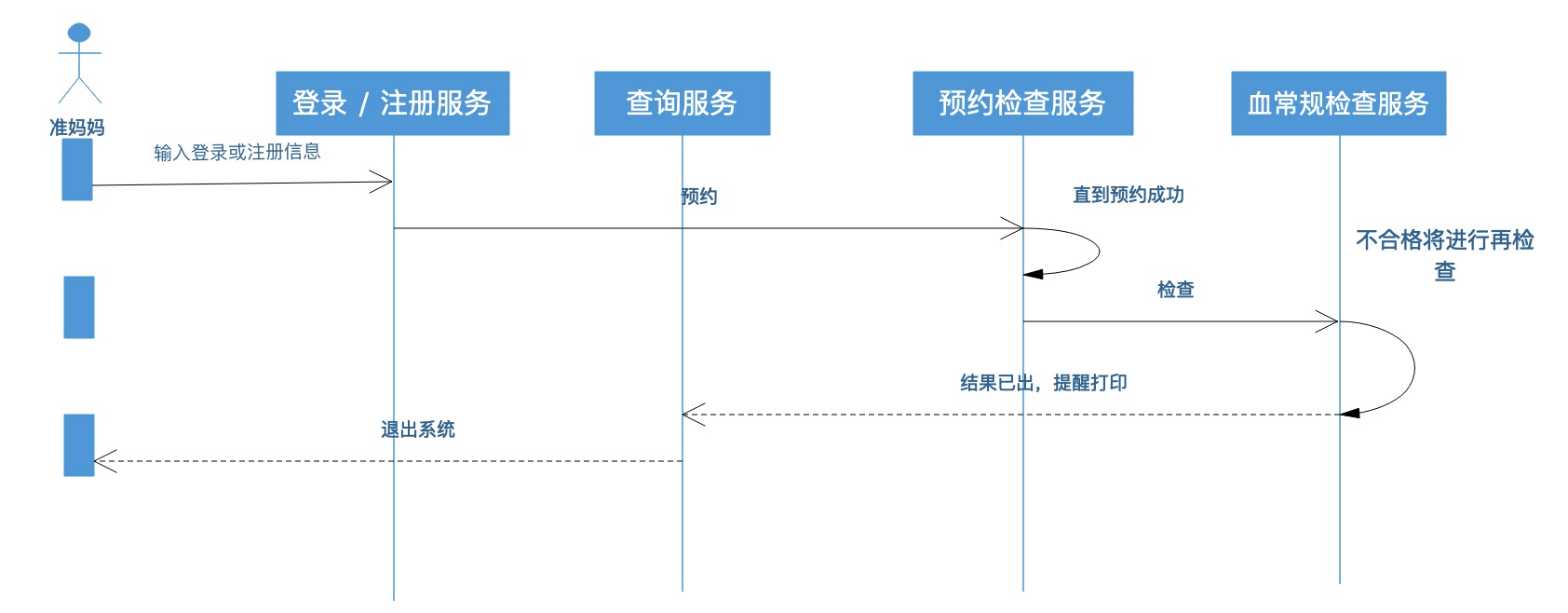


图3–3

准妈妈在本服务系统进行登录后，需要进行一系列的操作，具体流程如图3–3。当准妈妈为初次登录，则需要完成注册后再登录才可以使用本系统。进入系统后，准妈妈需要通过查询服务查看下一次预约时间。若没有预约，需要通过本服务系统进行预约检查，直到系统确认成功后才可以在预约当日进行血常规检查。本系统会在预约之日前通过短信方式以提醒准妈妈检查时间，以防止出现逾期错过检查的问题。在检查当日，当完成检查后，本服务系统会再次通过短信方式，提醒准妈妈预约下次检查的时间和打印检查报告，并可以在本服务系统中，查看到本次检查流程各个步骤的状态。

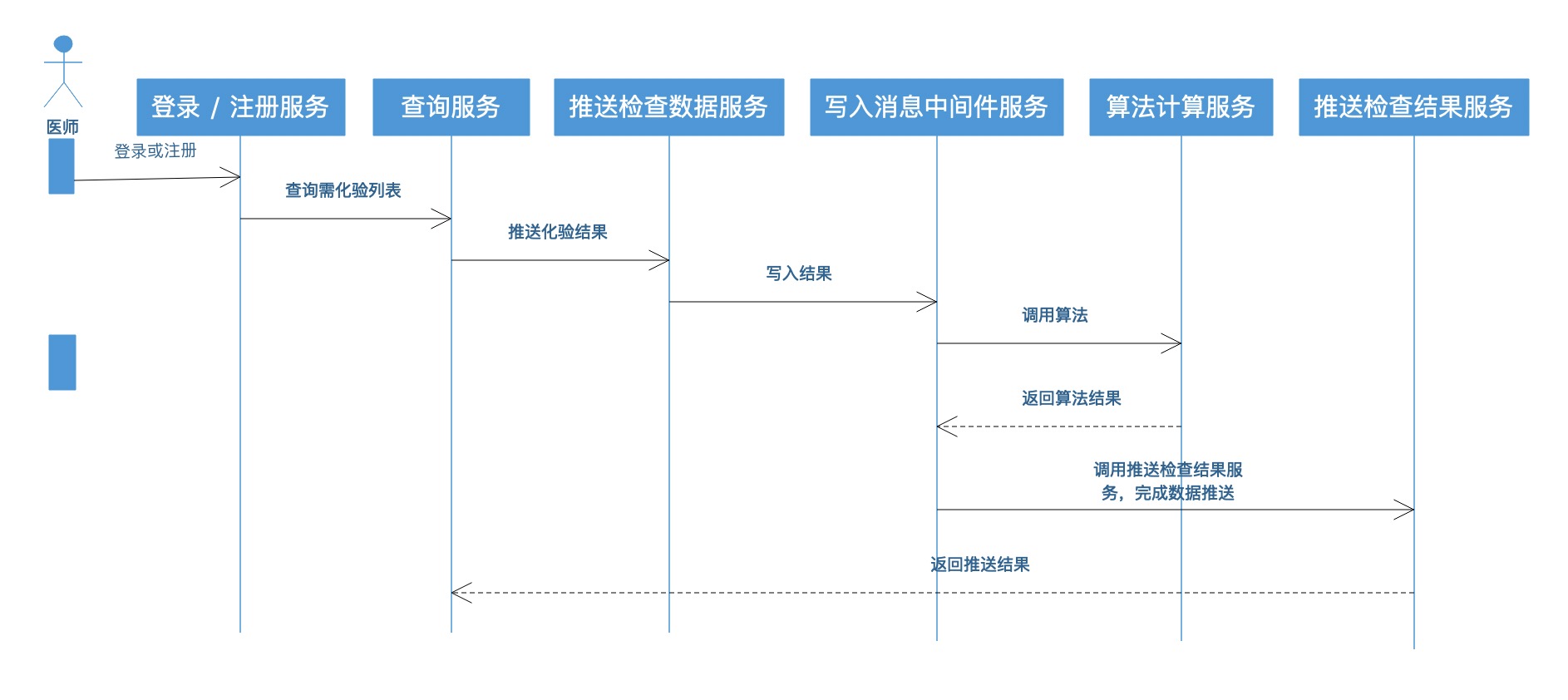


图3–4

主治医师在登录本服务系统后，同样需要进行一系列的操作流程，详细流程如图3–4。在查看到准妈妈的化验报告后，需要针对检查结果报告进行专业分析，并在本服务系统中记录该结果报告的相关医嘱。然后，将此检查结果报告通过推送检查数据服务，推送到在线实时流中进行数据处理和预测，系统会自动将预测结果和医嘱数据，再通过推送服务将数据推送给准妈妈，并以短信的方式进行提醒。

至此，准妈妈便完成了一次血常规检查过程，也了解了血常规检查数据的整条流转过程。

## **大数据**

大数据技术是当今互联网公司应用及处理大数据量信息最常用的实现技术。尽管大数据组件很多，使用任何类似的组件都可能实现自己的业务需求，因此，在设计本服务系统的架构时，就需要选择适合自己的业务场景的组件来实现。所以，针对本服务系统的实现场景，最终选择了Kafka和Storm的两个组件来实现数据处理流程。

### **Kafka和Storm技术特点**

使用Kafka是因为在本服务系统的架构设计中，考虑到本服务系统中各种消息结构的不同复杂程度，使用不同的Topic进行分类，与传统消息队列系统不同，在于Kafka的数据持久性、高扩展、高吞吐、高可用、低延迟的特性。

使用Storm是因为考虑到对本服务系统的消息处理上需要采用实时处理，实现低延迟的处理效果。而且在数据计算层的消息体基本上都是采用短消息体，吞吐量基本上在每秒3000至5000个消息，吞吐数据量较小，而且需要在数据处理流程中进行连接计算和API的调用。所以，基于这些业务特点的考虑，在技术选型中，选择Storm，而非Spark Streaming的原因。

### **实践应用**

下面以本服务系统在线实时流程[7]对于准妈妈预约检查的消息处理过程作为实例，深度说明一下大数据技术在本服务系统中具体实现。

在这个处理流程中，有两个不同的业务处理。一个是通过微服务API接收消息后将消息写入到Kafka中，另一个是订阅不同Topic进行消费数据进行数据处理和调用其他计算API的过程。

首先，说明第一个业务处理过程，通过微服务的功能模块API接收到消息后，本服务系统会将这些消息根据分类写入到Kafka的Topic中，代码参见附录F。这样的设计目的是使用消息队列实现了技术解耦，不将写数据库的操作直接暴露给用户，以防止用户通过SQL注入等黑客技术破解应用系统。而且可以通过简短的代码实现，便可以实现数据接收的业务功能。

其次，第二个业务处理流程才是真正的数据处理过程，是进行数据清洗、数据写入、接口调用的流处理全过程。在实现这个业务过程中，需要通过使用Storm的相关API连接到Kafka的Topic上，并设定并行处理参数，便可以实现多线程处理数据的能力。如下面代码所示：

TopologyBuilder builder = new TopologyBuilder();

builder.setSpout("kafkaSpout", new KafkaSpout(kafkaConfig));

builder.setBolt("OrderIRichBolt", new OrderBolt(), 2).localOrShuffleGrouping("kafkaSpout");

builder.setBolt("DBIRichBolt", new DBBolt(), 3).localOrShuffleGrouping("OrderIRichBolt");

在这部分实现代码中，指定了订阅Kafka的预约检查Topic及Kafka集群的访问地址，OrderBolt类是处理准妈妈检查预约消息的清洗、业务逻辑的实现类，DBBolt类是将最终预约消息写入数据库的实现类。通过指定三个实现类的处理顺序，并在三个业务实现类中编写相应业务处理逻辑代码，便可以将写入到Kafka Topic中的消息进行消费处理，完成整个针对准妈妈预约检查业务的清洗和数据写入等过程，得到我们最终需要的数据。[8]

## **机器学习**

此部分将以我爱人血常规检查的结果报告数据为事例进行详细的阐述。

### **3.3.1 建模**

本服务系统的最终使用人群是面向准妈妈，她们并不能了解检查报告中各个检查项的含义。所以，在本服务系统中，致力于将复杂专业的数据简单化，尽量使用图表加文字的方式将检查结果反馈给准妈妈。在这里，仅针对准妈妈血常规检查报告，进行算法建模，主要使用主成分分析法[9]，将多维数据集降维成点数据， 数据集的每个样本都可以用一个值表示，最终形成一条趋势线呈现给准妈妈。

下面是我针对血常规报告数据进行建模后，得到的计算公式，如图3–3：

图3–3

一、参数说明：

为第个检测样本报告

为第个测试项的权重，取值范围为0–1，即所有测试项权重总和为1。而相同测试项在不同检查报告中，因起到的作用可能不同。所以，此权重会根据模型的不同，调整测试项的权重。

为第个测试项的标准平均值。由于每个检查项取值都是一个范围值，所以，在此取检查值范围最大值和最小值的平均值。

为第个测试项的结果值，即为实际检查结果。

二、计算过程：

将检查报告中，各个检查项定义为特征，检查项权重值设为，标准平均值设为，对应的检查结果为。检查综合值的得出是由每个检查项权重值乘以检查结果值减去检查平均值的差值后，再进行求和的结果。

这样，我们将二维数据转换成线性数据，便可以知道真实值和预测值的匹配程度，更为清晰直观地展示趋势走向[10]。

### **3.3.2 实例说明**

图3–4 显示为我爱人从怀孕4.5个月至6个月期间共14次血常规检查报告结果的综合值趋势。图例显示为：Range Maximun为检查项范围最大值的综合值，Range Mininum为检查项范围最小值的综合值，Final Result为检查项实际结果的综合值。

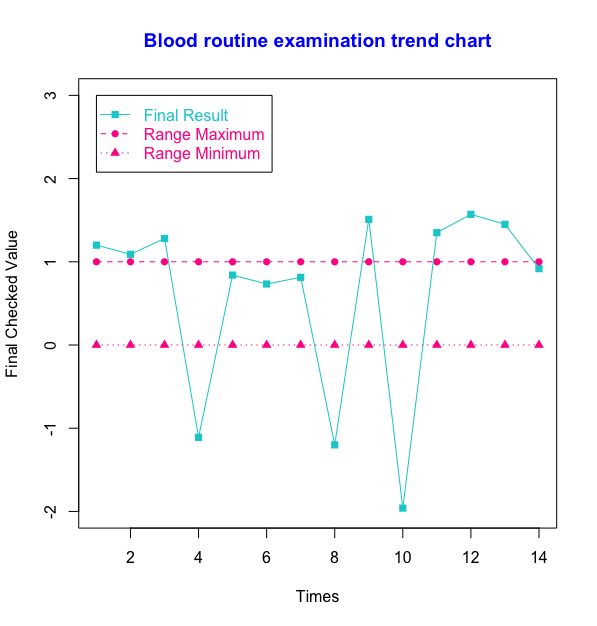


图3–4

结果说明：第1，2，3，9，11，12，13次的检查结果综合值略高于Range Maximum的点为血粘稠度增加，会影响子宫胎盘的正常血循环，甚至能增加胎盘血栓形成的机会；第4，8，10次检查结果综合值低于Range Minimum的点则表示为营养欠缺，铁供给量不足，易形成缺铁性贫血。第5，6，7，14次的检查结果综合值虽在正常范围里，但略偏高。所以，得出营养不够均衡，需要增加含铁量高的食物，亦可详细咨询营养师具体建议。

由此可见，机器学习在整个系统中起着举足轻重地作用。在现在和未来，作为现实世界科学探索的一个工具和技术，将不断地探索和发现包括不确定性系统产生的大量客观规律，以便更好地服务于医疗行业的应用实践。

# **发展前景**

医疗服务行业可以利用“互联网+医疗”从研发支撑平台、成果转化和工程应用、人才和能力建设等方面重点突破，创新建设体系与应用模式，全面引领和支撑我国医疗互联网领域的发展，形成完善的产业生态系统。

以本服务系统为例，我们可以建立以医疗大数据应用技术为基础的准妈妈孕检创新研发平台，组织开展医疗微服务互联互通、医疗大数据整合管理以及机器学习应用等研究，进行关键技术的研发、产品化和工程化，快速提高我国关于准妈妈医疗技术水平，着力探索医疗数据资源的统一标准，形成从多级别的数据资源全面整合共享和规范化，推动医疗数据的互联互通和融合共享。

通过互联网技术，培育医疗大数据行业自主创新能力，促进行业领先企业和科研单位的强强联合，发挥技术研发的导向作用，建设开放式、包容式技术研发创新平台和产业联盟，为国内外优势力量提供参与协同创新的沃土，开展医疗大数据关键核心技术和新产品的战略性、前瞻性研发，发展一批具有创新技术路线的医疗大数据的技术模式。

建立微服务、大数据和机器学习应用技术的人才培养平台，为医疗行业中创新工作提供源动力，形成一支结构合理、紧密协作的专病研究、人群队列、公共卫生、生物组学、临床医学、计算机科学等多学科团队。

利用微服务、大数据和机器学习技术，面向行业共性问题和需求，研发一系列医疗服务产品，形成数据驱动的医疗服务决策能力，服务于健康、医药、医保、医疗、大数据多个上下游产业链，提升我国医疗相关产业整体水平。

综上所述，在未来三到五年，“互联网+医疗”行业将有着更大的发展空间，能够将医疗行业的技术得以质的提高，并且这三项技术将更大化推动“互联网+医疗”行业的发展。

结 论

本论文所取得的成果

在这次毕业设计里，深入地接触微服务架构的搭建、大数据平台环境的搭建和机器学习的深入学习。最终，设计并制件出本服务系统，体验到移动互联网技术应用于医疗行业所带给我们的便利。

孕期检查是国内各大妇产医院面临的最大问题，而以互联网化为手段，优化就诊流程，提升患者的就医体验，将会成为未来“互联网+医疗”一直关注的问题。因此，一切围绕这个目的的服务会一直存在，可能贯穿医疗服务的全过程，具体可涵盖医疗资源查找与匹配、网上挂号、在线问诊、远程诊疗、医药电商、移动医疗等领域。

互联网环境下，医疗服务逐步数字化，这将极大地提高医生与患者直接的相互了解，例如透过先进的影像获取和存储技术，利用大数据分析，获得特定个人的病灶变化情况及同类病例治疗的比较结果，这将为精准治疗提供决策依据。将患者的医疗服务需求精准推送给医生，将医疗服务项目精准提供给患者，实现医疗沟通过程中的双向精准化，才能真正达成精准治疗。

在“互联网+医疗”的驱动下，医疗机构将建立起以患者为中心的全新医疗服务模式，以改善就医体验为目的，逐步实现医疗诊治精准化、医疗组织协同化、医疗服务个性化，将医疗服务扩展到更大范围。未来医疗新模式将在信息技术的推动下向共享、协作、个性化方向发展，“互联网+医疗”将会涌现出更多的应用和模式，而这需要进一步的研究和探索。

致 谢

在论文完成之际，谨向为此文倾注了大量心血和帮助的老师和家人表示深深地谢意。

向我的指导老师张丽娜致以衷心地感谢，课题的设计以及论文的撰写自始至终都是在张丽娜老师的悉心指导下进行。张丽娜老师在学习和科研过程中给予我无微不至的关怀和指导，使我能够迅速地掌握本专业知识，并解决一个个实际的工程问题，能够迅速地给出处理流程和分析思路，对各方向知识的涉猎。

最后，感谢我的家人，尤其是我的爱人宋晓鹏能够提供孕期数据，并抽出时间给予我在论文上的帮助。

参考文献

[1] 张思莱.张思莱育儿手记.中国妇女出版社.2017年5月第6次出版

[2] 黄健宏.Redis设计与实现.机械工业出版社.2014年6月第1次出版

[3] Neha Narkhede, Gwen Shapira, Todd Palino. Kafka: The Definitive Guide.

[4] 郑天民. 微服务设计原理与架构. 人民邮电出版社. 2018年5月第1次出版

[5] [美] Sanjay Patni 著 郭理勇 译. RESTful API开发实战. 清华大学出版社. 2018年2月第1次出版

[6] John Carnell. Spring Microservices IN Action. 2018.

[7] 朱松岭. 离线和实时大数据开发实战. 机械工业出版社. 2018年5月第一次出版

[8] 罗聪翼, 龚成志译. Storm应用实践：实时事务处理之策略. 机械工业出版社. 2018年1月第1次出版

[9] 周志华. 机器学习. 清华大学出版社. 2016年1月第1次出版

[10] 李航. 统计学习方法. 清华大学出版社. 2012年3月第1次出版

附 录

**附录A：服务注册中心Eureka Server工程源代码**

Eureka Server主函数源程序：

@SpringBootApplication

@EnableEurekaServer

public class EurekaServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaServerApplication.class, args);

}

}

**附录B：服务中心Euraka Client客户端**

Eureka Client主函数源程序：

@SpringBootApplication

@EnableDiscoveryClient

public class UserregApplication {

public static void main(String[] args) {

SpringApplication.run(UserregApplication.class, args);

}

}

**附录C：服务中心Euraka Client客户端UserController业务类**

用户注册源代码：

@RestController

public class UserController {

@RequestMapping("/userreg")

public Map<String,String> reg()

{

Map<String,String> map=new HashMap<String, String>();

map.put("status","用户注册成功");

return map;

}

}

**附录D：对外提供接口数据的客户端**

主函数源代码：

@SpringBootApplication

@EnableEurekaClient

public class MywebApplication {

public static void main(String[] args) {

SpringApplication.run(MywebApplication.class, args);

}

}

**附录E：服务注册中心的用户注册接口**

用户注册接口对外实现类源代码：

@RestController

public class Myweb {

@Autowired

DiscoveryClient discoveryClient;

@RequestMapping("/userreg")

public Map userreg() {

List<ServiceInstance> list = discoveryClient.getInstances("USERREG");

String serviceUrl = list.get(0).getUri().toString();

System.out.println(serviceUrl);

RestTemplate restTemplate = new RestTemplate();

return restTemplate.getForObject(serviceUrl + "/reg", Map.class);

}

}

**附录F：Kafka写操作源程序**

写入指定Topic源代码：

object ProduceOrderMsg {

    def BROKER\_LIST = "localhost:9092"

    def TOPIC = "t-order"

    def main(args: Array[String]): Unit = {

        val props = new Properties()

        props.put(ProducerConfig.BOOTSTRAP\_SERVERS\_CONFIG, BROKER\_LIST)

        props.put(ProducerConfig.KEY\_SERIALIZER\_CLASS\_CONFIG, classOf[StringSerializer].getName)

        props.put(ProducerConfig.VALUE\_SERIALIZER\_CLASS\_CONFIG, classOf[StringSerializer].getName)

        val producer = new KafkaProducer[String, String](props)

        while ( true ) {

            val ret: Future[RecordMetadata] = producer.send(new ProducerRecord(TOPIC, "uid-" + uid + "|time-" + timestamp, orderMsg))

            val metadata = ret.get  // 打印出 metadata

            println("i=" + i + ",  offset=" + metadata.offset() + ",  partition=" + metadata.partition())

        }

        producer.close

    }

}