文章编号:1671-4598(2020)09-0212-06 DOI:10.16526/j.cnki.11—4762/tp.2020.09.042 中图分类号:TP391.3 文献标识码:A

基于 Flink 的海量医学图像检索系统设计与实现

毛亚青，王 亮，胡俊峰
（徐州医科大学 医学信息与工程学院，江苏 徐州 221000）

摘要：医学图像检索是有效利用医学资源的基础，而医学图像的海量性和增量性为图像检索带来了新的挑战和要求；为了提高医学图像检索过程的效率，设计并实现一种基于 Flink 的海量医学图像检索系统；首先，系统通过 Web 应用作为用户操作入口，在后端搭建数据平台和业务集群；其次，系统通过 HBase 对医学图像数据进行分布式存储，利用深度卷积神经网络模型提取医学图像特征；然后，将所提取的医学图像特征数据进行乘积量化编码，并通过 HBase 进行存储；最后，通过基于 Flink 的内存计算对接 Kafka 进行实时图像检索，以及对批量导入图像的特征索引编码；系统在 4 个节点的服务器上部署分布式集群，使用真实医学图像数据集进行测试，通过在 MapReduce 和 Spark 两种不同技术模块下的对比实验表明本系统具有更好的检索效率表现。

关键词：医学图像检索；Flink；Hadoop；卷积神经网络

Design and Implementation of Massive Medical Image Retrieval System Based on Flink

Mao Yaqing, Wang Liang, Hu Junfeng
(School of Medical Informatics, Xuzhou Medical University, Xuzhou 221000, China)

Abstract: Medical image retrieval is the basis of effective utilization of medical resources, and the massive and incremental medical image brings new challenges and requirements for image retrieval. In order to improve the efficiency of medical image retrieval process, a Flink based medical image retrieval system is designed and implemented. Firstly, the system uses web application as the users' operation entry, and builds data platform and business cluster at the back end. Secondly, HBase is used to store the medical image data in a distributed way, and extracts the medical image features by using the deep convolution neural network model. Thirdly, the extracted medical image feature data is multiplied and encoded, and stored by HBase. Finally, Kafka is used for real-time image retrieval through memory computing and then consumed on Flink, as well as feature index coding for batch imported images. The system deployed a distributed cluster on four nodes of servers and tested with real medical image data set. The comparison experiment under MapReduce and Spark shows that the system has better retrieval efficiency.

Keywords: medical image retrieval; Flink; Hadoop; convolution neural network

0 引言

目前，医学影像数据约占医院内部总数据的 70%[1]，它们来源于不同设备、对应不同人体组织器官和多种病症，这些海量的医学图像资源为医学图像的存储、检索带来了存储体量大、检索效率低等问题。如何对海量医学图像信息进行高效检索，从海量数据中快速并准确地搜索出满足要求的图像是目前所要解决的重要问题。然而，现阶段国内外关于医学图像检索技术的研究[2-5]依然存在三大问题：1）主要在单机环境下进行，大规模医学影像数据的检索使得该串行模式的医学图像检索技术已出现进程瓶颈；2）当前医学图像检索中主要采用对图像依次进行遍历的方式，而没有很好的索引机制来做索引，也增大了检索系统的负荷；3）传统的图像检索模式大多基于已有的数据进行定时离线构建索引，对于新增的图像检索存在时效性差的问题。

为了解决医学图像检索过程中的效率问题，本课题拟采用基于 Flink 的分布式技术提高这些海量医学图像的检索实时性，针对海量医学图像检索系统中图像特征索引方式、图像存储以及图像检索的问题，建立一个高效的医学图像检索平台，实现医学图像分布式检索，提高图像处理的实时性以及图像检索准确率，从而更好地辅助医生便捷获取和利用医学图像资源。

收稿日期:2020-03-06； **修回日期**:2020-04-01。
基金项目:国家自然科学基金青年科学基金项目(31900022)；教育部产学合作协同育人项目(201801226011)；中华医学会医学教育分会、中国高等教育学会医学教育专业委员会 2018 年医学教育研究立项课题重点项目(2018A-N05065)。
作者简介:毛亚青(1995-)，男，江苏徐州人，硕士，在读研究生，主要从事医学图像处理、深度学习、大数据分布式计算方向的研究。
通讯作者:胡俊峰(1969-)，男，江苏徐州人，博士，教授，主要从事医学影像学、医学图像处理方向的研究。

1　相关工作

1.1　Hadoop 平台简介

从狭义上来说，Hadoop[2-6]是一个由 Apache 基金会所维护的分布式系统基础架构，旨在解决海量数据的存储和计算问题。而从广义上来说，Hadoop 通常指的是它所构建的 Hadoop 生态，包括 Hadoop 核心技术，以及基于 Hadoop 平台所部署的大数据开源组件和产品，如 HBase、Hive、Spark、Flink、Pig、ZooKeeper、Kafka、Flume、Phoenix、Sqoop 等。这些组件藉由 Hadoop 平台，实现大数据场景下的数据存储、数据仓库、分布式计算、数据分析、实时计算、数据传输等不同需求，从而构成 Hadoop 生态。

Hadoop 的核心技术：HDFS、MapReduce、HBase 被誉为 Hadoop 的三驾马车，更为企业生产应用带来了高可靠、高容错和高效率等特性。HDFS 是分布式文件系统（Hadoop Distributed File System），其底层维护着多个数据副本，即使 Hadoop 某个计算或存储节点出现故障也不会导致数据的丢失，所以即使部署在成本低廉的服务器上也能同样保障其可靠性和容错性。MapReduce 是 Hadoop 中并行计算编程的基本模型，能够将任务并行分配给多个节点同时工作，从而加快任务处理的速度。HBase 是一个可伸缩、分布式、面向列的数据库，和传统关系数据库不同，HBase 提供了对大规模数据的随机、实时读写访问，同时，HBase 中保存的数据可以使用 MapReduce 来处理，它将数据存储和并行计算完美地结合在一起。

1.2　Flink 平台简介

Apache Flink 是一个框架和分布式处理引擎，用于对无界和有界数据流进行有状态计算。Flink 被设计在所有常见的集群环境中运行，以内存执行速度和任意规模来执行计算。Flink 基于 Flink 流式执行模型（Streaming execution model），能够支持流处理和批处理两种应用类型。流处理和批处理所提供的服务等级协议完全不相同，流处理一般需要支持低延迟、Exactly－once 保证，而批处理需要支持高吞吐、高效处理，所以在实现的时候通常是分别给出两套实现方法，或者通过一个独立的开源框架来实现其中每一种处理方案。如：实现批处理的开源方案 MapReduce、Spark；实现流处理的开源方案 Storm；微批处理方案 Spark Streaming。

与传统方案不同，Flink 在实现流处理和批处理时，将二者统一起来：Flink 是完全支持流处理，也就是说作为流处理看待时输入数据流是无界的；批处理被作为一种特殊的流处理，即有界的数据流。这种批、流一体的架构使得 Flink 在执行计算时具有极低的延迟。

2　系统架构设计

2.1　系统总体架构设计

为了实现海量医学图像的实时检索和高效存储，设计基于 Flink 的海量医学图像并行检索系统总体架构共包括：数据采集层、数据存储层、资源管理层、数据计算层和应用层。

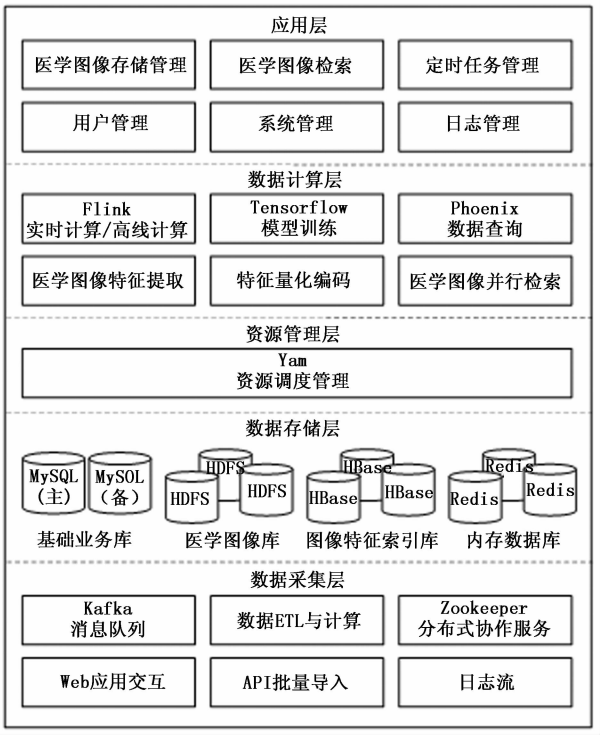

图 1　系统总体架构图

如图 1 所示为医学图像检索系统架构图，具体如下。

1）数据采集层：

系统的数据源主要包括：用户通过 Web 界面上传医学图像、通过 API 批量导入的医学图像和系统操作日志流。对于实时产生的数据首先放入 Kafka 消息队列进行缓冲中以供后续计算，通过 Zookeeper 组件对 Kafka 服务器消费生产速度进行同步。此外，还可以通过 ETL 导入数据作为系统的数据源。

2）数据存储层：

系统的数据存储根据数据类型和应用场景分为基础业务库、医学图像存储库、图像特征索引库和内存数据库。其中，基础业务库通过 MySQL 存放系统的结构化信息，如：人员列表、组织架构、图像基础信息等。医学图像存储库通过 Hadoop 平台的 HDFS 进行存储，图像 ID 对应 MySQL 中的图像基础信息表的记录。同时，该 ID 图像的特征索引存储在 HBase 数据库中。此外，服务器将经常访问的数据缓存在内存数据库 Redis 中，从而提高访问速度和计算效率。

3）资源管理层：

系统通过由 Yarn 进行资源管理，负责在有数据计算请求时根据集群状况分配计算资源和计算节点，从而提供 MapReduce、Spark、Flink 等组件的计算环境。

4）数据计算层：

系统首先对于用户输入的医学图像通过 Flink 进行特征提取，根据图像上传形式分为实时计算和离线批量计算两种；进而对图像进行特征量化编码便于检索，特征编码存

储在 HBase 中，并由 Phoenix 进行 HBase 中数据的查询计算；在用户需要检索时通过医学图像并行检索比对特征相似度计算返回检索结果。

5）应用层：

系统通过 Web 的形式提供用户交互界面，实现对医学图像的存储管理、医学图像检索操作、用户管理、系统管理和日志管理等。

2.2 系统功能结构设计

医学图像检索系统主要设计包括系统管理模块、用户管理模块和图像管理模块三部分。其中，系统管理模块包括系统管理、日志管理、异常报警；用户管理模块包括用户注册、用户登录、个人信息管理；图像管理模块包括图像上传、图像批量导入、图像查看和图像检索检索。具体功能如图 2 所示。

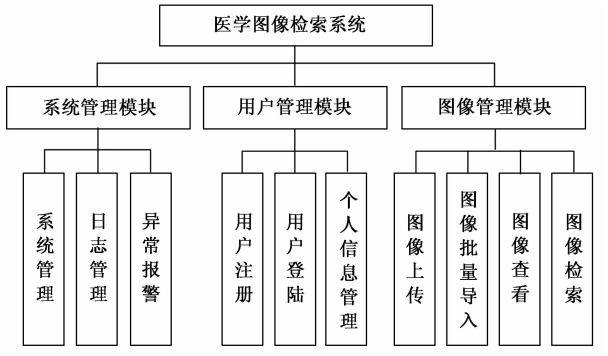

图 2 系统功能结构图

2.3 系统集群网络架构

系统集群网络架构共包括前端集群、后端业务集群和数据计算集群，医学图像并行检索系统网络架构如图 3 所示。

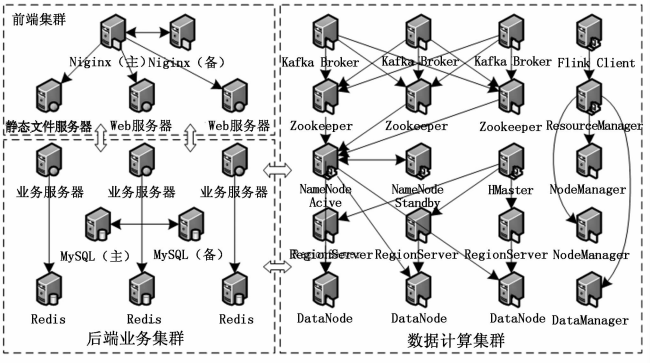

图 3 系统集群网络架构图

系统主要采用前端界面和后端业务分离的思想，在提高各个模块的内聚性的同时降低各模块之间的耦合性。在前端集群中，由 Nginx 负责请求的反向代理和负载均衡，根据用户操作分别指向静态文件服务器或 Web 服务器，实现网页相关界面的显示与交互。前端集群通过远程调用的方式与后端业务集群进行通信，实现相关业务操作、MySQL 数据库交互操作、数据计算与结果缓存到 Redis 等操作。对于后端业务操作中的数据计算环节则由数据计算集群负责，如：实时图像上传、批量图像导入、特征提取模型计算、特征编码模型计算等。

在数据计算集群中部署了 Hadoop 平台（HDFS、HBase、Yarn）以及 Flink、Kafka、Zookeeper 等组件。其中 HDFS 负责进行底层数据的存储，具体由 HDFS 的 DataNode 进行文件分片多备份存放，由 NameNode 进行元数据管理和文件操作管理，同时通过 Zookeeper 注册两个 NameNode 并实时监控状态，防止一方故障立即切换到另一个，从而保证 NameNode 的高可用性。HBase 负责对医学图像和特征编码进行存储，由 HMaster 管理多个 RegionServer 进行数据维护和查询，底层由 HDFS 进行存储。对于实时计算部分通过 Kafka Broker 接受 Kafka 生产者生产的实时消息，再通过 Kafka 消费者 Flink 进行处理计算，其中 Kafka 的生产、消费进度由 Zookeeper 进行记录。Flink 不仅提供实时计算，同时提供离线批量计算，其计算过程通过 Yarn 申请计算资源，具体由 ResourceManager 管理资源并分配到 NodeManager 上进行计算。

3 主要模块设计

3.1 医学图像存储模块

系统在接收到 Web 端的医学图像上传请求后，首先在 MySQL 业务库中创建图像记录，然后服务器后台将图像文件的字节码对应业务库中的图像 ID 存储到 HBase 中，实现海量医学图像数据的存储。在图像经过特征提取和编码后，将编码后的图像特征对应图像 ID 存储到 HBase 中，在图像检索的过程中通过 Phoenix 进行查询。

对于海量文件的存储，通常的方案是通过 HDFS 进行分布式存储。HDFS 系统在存储过程中将文件切分成多个 block 多副本存储在多个节点上，从而保障文件的可用性和拓展性，默认每个 block 的大小为 128 MB。然而，HDFS 通过 NameNode 加载每个文件的元数据信息，一般上传图像文件的尺寸都较小，在大量这种小文件的存储情况下，其每个小文件都会占用一个 block，造成 HDFS 产生大量的文件元数据信息。这些元数据信息会给 NameNode 的内存和计算带来很重的负担，从而降低系统的存储效率。为了解决 HDFS 在小文件存储方面的问题，通常的做法是先将很多小文件合并成一个大文件再保存到 HDFS，同时为这些小文件建立索引，以便进行快速存取，如 Hadoop 自带的 HAR 文件和 SequenceFile 方案。但是这两种方案均需要用户编写程序定时进行小文件的统一合并，且不支持文件追加和修改，并不适合医学图像实时上传、频繁更新业务库的场景。因此，系统将图像文件通过字节码的形式直接存储在 HBase 中，从而避免 HDFS 存储过多的小文件、影响效率的情况。

系统利用 HBase 存储医学图像文件数据和特征编码等数据，进而通过 Phoenix 进行结构化查询。HBase 作为分布式数据库，通过多个 Region 对数据进行存储，在实时查询方面具有很强的优势。然而默认情况下 HBase 的表结构分

区只有一个，在写入读取时会增大单节点的负担，没有发挥集群的优势。此外，一条记录由它的 RowKey 唯一标志，并决定该条记录存储于哪个分区。因此，在设置多个分区后也需要考虑分区的分配策略，即进行合理的 RowKey 设计，从而对数据进行均匀分布，防止出现数据热点问题。

在本系统的 HBase 存储设计部分共包括创建表、预分区、RowKey 设计等环节，包括：

1）创建图像存储表，设计两个列族 MD（image data）、MI（image info）分别存放图像的字节码和图像信息（图像 id、图像特征码等），在创建表的同时进行预分区操作，设计共 9 个分区，指定每个分区的 RowKey 范围，建表语句如下：

```
create2'image_info', {NAME => 'MD'}, {NAME => 'MI'},
SPLITS => ['0000|', '0001|', '0002|', '0003|', '0004|', '
0005|', '0006|', '0007|', '0008|']
```

2）根据预分区设计，在向 HBase 中插入数据时需要对 RowKey 进行相应的格式约束，即在保证 RowKey 唯一性的同时确保其前缀格式为“000x｜”。本系统首先根据医学图像在业务库的唯一 ID 通过 MD5 加密生成 Hash 值；然后获取 RowKey 前缀：将得到的 Hash 值转成 Long 型，并根据预分区数对 9 取余，其中字符’a’～’f’替换为’10’～’15’；再取 Hash 值的后八位作为 RowKey 的后缀，拼接前缀作为最终的 Row Key。

3.2　图像特征提取与编码模块

传统树结构索引方法存储空间占用过大，且随着维度的增长空间代价成倍变大，因此需要通过对原始数据进行哈希编码压缩以节省空间。目前对哈希编码的研究主要包括数据无关哈希和数据驱动哈希：数据无关哈希方法以局部敏感哈希[7]为代表，在不考虑数据分布的情况下将原始空间中的数据投影到超平面获取相应编码。数据驱动哈希方法主要通过判别数据结构及分布信息来自动学习哈希函数，代表方法有谱哈希[8]、迭代量化[9]、乘积量化[10]、笛卡尔 K 均值[11]及组合量化[12]等。与其他编码方法相比，乘积量化模型能够有效解决聚类中心数量膨胀问题，进而提升大规模图像检索过程中的数据存储效率。

系统根据应用场景分为批量导入图像时的特征量化以及用户在实时上传图像时的特征量化。

1）批量导入图像特征量化：

系统实现医学图像的批量导入功能，用于历史或外部图像的离线导入场景。其过程如下：

（1）将图像信息记录到业务库并将图像字节码和图像 id 存储在 HBase 中；

（2）通过 DeepLearning4J 调用预训练的深度卷积神经网络 VGG－16 模型提取图像特征；

（3）使用乘积量化编码模型对提取的图像特征进行量化编码；

（4）将图像特征编码存储到 HBase 中。

2）实时图像特征量化：

在多用户同时在线的场景中，为保证用户在上传图像后能被其他用户同步到以供检索，设计实时图像检索模块，对增量上传的图像进行实时特征提取和特征量化编码，从而实现系统的时效性。其实现过程包括：

（1）通过命令行创建 Kafka 消息订阅的 topic，表示一条图像上传的实时记录，设计 topic 名为 imageupload，定义副本数 2 个，分区数 9 个。

```
bin/kafka－topics. sh －－zookeeper node01:2181 －－create －
－replication－factor 3 －－partitions 9 －－topic imageupload
```

（2）Web 服务响应用户的上传请求，首先将图像基本信息存入 MySQL 库中；接着创建 Kafka 的生产者，由 Kafka 生产者将图像上传消息类进行序列化，包括图像的基本信息及图像文件字节码；然后，通过 KafkaProducer 类发送到 imageupload 的 topic 中；同时，利用回调函数监测是否发送成功，异常则触发报警。因为系统对图像上传顺序的要求不高，因此 Kafka 的消息按照轮询的方式进行存放在每个分区中。

（3）创建 Flink on Yarn 任务实现 Kafka 的消费者进行实时处理。Flink 任务实现记录图像信息到业务库、提取图像特征、存储图像字节码和图像特征编码到 HBase 中。

3.3　图像并行检索模块

在医学图像并行检索的 Flink 任务执行过程中，对于输入查询的医学图像，本文首先利用 CNN 模型进行深度特征提取，然后对哈希编码后的特征向量采用非对称距离度量进行距离计算，最终输出距离最近的相似医学图像。利用非对称距离度量的优势在于能够避免直接计算查询医学图像深度特征向量与数据库中每个向量的欧式距离，从而减少查询时间、提高检索效率。

图 4 是医学图像并行检索过程的示意图。通过事先计算深度特征哈希数据库中每个聚类中心与其子向量的距离建立检索查找表；对于需要查询的医学图像深度特征向量 q，计算其与数据库中聚类中心 x_i' 的距离，即为该向量与其他图像向量之间的非对称距离；通过比较 q 与聚类中心的距离找出最近的聚类 c，设距离为 l；最后，遍历查找表将 c 聚类中每个向量与聚类中心的距离与 l 相加，即获得 q 与该聚类中所有向量的距离，筛选距离排序获得最近似的特征向量并返回对应的医学图像。

4　实验结果与分析

4.1　实验环境

为了实现医学图像特征提取模型的高效训练和并行检索模型的分布式执行，本文将模型训练和集群应用两部分实验分在不同的环境中执行。其中，模型训练过程环境选用 GPU 型号为 Tesla K80、12 GB 内存的 Google 云服务器，

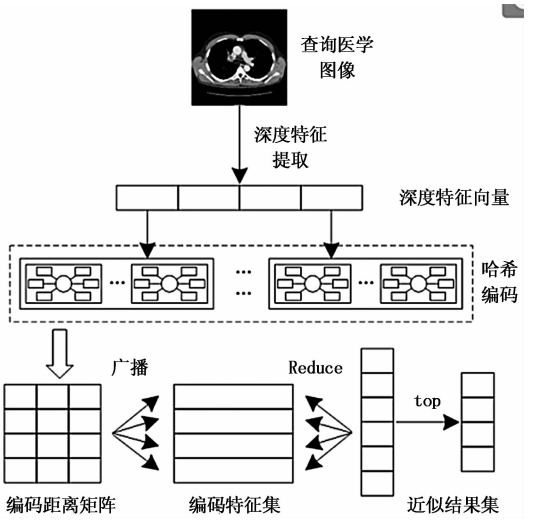

图 4 医学图像并行检索过程

并采用 Python 3.6 和 Tensorflow 1.7 的深度学习框架。集群应用环境选用 1 个主节点和 3 个计算节点，各节点配置情况如表 1 所示。

表 1 分布式节点配置情况

配置名称	配置信息
CPU	Intel(R) Xeon CPU 2.4 GHz 4Core
内存	24 GB
硬盘	1 TB
操作系统	CentOS 7.3 64 位
Java JDK 版本	JDK 1.8
Hadoop 版本	Hadoop 2.6.0－cdh5.14.2
Flink 版本	Flink 1.9.0
MySQL 版本	MySQL 5.6

服务器集群采用 4 个节点（node00～node03）并采用 CDH 进行部署管理，包括 HDFS、Yarn、Zookeeper、Kafka、HBase 等组件的部署、监控管理。此外，还部署了 Flink 计算组件、MySQL 主备节点、Redis 内存数据库等，具体组件子服务的分配状况如表 2 所示。

实验数据集选用由美国国立卫生研究院临床中心（NIHCC）的团队开发的医学图像数据集 DeepLesion[13]，是来自 4 427 个患者的多类别、病灶级别标注临床医疗 CT 图像开放数据集。该数据库中目前已有 32 735 张 CT 图像及病变信息，去除重复记录后共有已标记的病变图像 9 624 个，包括：肺（2 370）、腹部（2 119）、纵隔（1 640）、肝脏（1 257）、骨盆（843）、软组织（660）、肾（488）和骨（247）共 8 种损伤类型。

本文实现的医学图像检索方法在 DeepLesion 数据集上进行医学图像特征提取和分布式并行检索。根据给定医学图像实现数据集中相同病灶、相似损伤的其他医学图像的检索，从而有效地辅助医疗诊断过程。

4.2 图像检索效率分析

为了验证基于 Flink 的医学图像检索系统的图像检索效率，本文分别使用 MapReduce、Spark 和 Flink 三种分布式计算组件实现并行检索环节，并对比不同组件在不同图像数据量下进行医学图像检索的时间。

表 2 服务器集群组件分配情况

服务名称	子服务	node00	node01	node02	node03
Cloudera Manager	CDH				√
HDFS	NameNode	√			
	DataNode	√	√	√	√
	Secondary NameNode		√		
Yarn	Node Manager	√	√	√	√
	Resource Manager			√	√
Zookeeper	Zookeeper Server	√	√	√	
Kafka	Kafka	√	√	√	√
HBase	HMaster	√			
	Region Server	√	√	√	√
MySQL	MySQL			√	√
Redis	Redis	√	√	√	
Flink	Job Manager	√			
	Task Manager	√	√	√	√

各组件检索时间对比如图 5 所示。在使用 3 种组件时，并行检索的时间均随着数据量的增大而增加。其中 MapReduce 效果相对更差，且使用时间增长也较快。Spark 和 Flink 使用时间相差不多，总体来说 Flink 的处理效果更好，且随着处理数据量的增加，Flink 的计算效率明显更优于 Spark。因此，使用 Flink 进行分布式图像检索的计算更具优势。

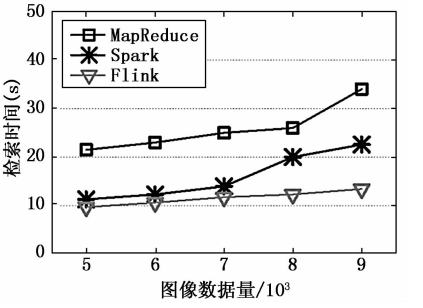

图 5 各组件检索时间对比图

4.3 系统功能效果

进入医学图像检索系统，用户首先通过用户名、密码、验证码的方式进行登陆。如果忘记密码可以通过邮箱找回密码。具体操作如图 6 所示。

同时，新用户进入系统可以通过注册账号，填写用户名、姓名、邮箱、设置密码等表单信息进行新用户的申请。具体操作如图 7 所示。

进入桌面化的系统界面后选择打开桌面上的应用进行操作，系统根据权限不同提供医学图像管理、用户信息管理、日志管理、系统状态管理、多媒体应用等应用。

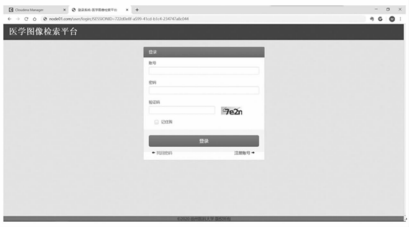

图 6　系统用户登陆界面

图 7　新用户账号注册界面

在医学图像管理模块，用户通过医学图像上传应用对图像信息、图像描述、图像文件进行填写，图像上传后存储在 HBase 平台中。具体操作如图 8 所示。

图 8　医学图像上传应用操作界面

用户可以通过医学图像列表应用查看医学图像记录，可以根据图像类型、开始时间和结束时间进行简单筛选。具体操作如图 9 所示。

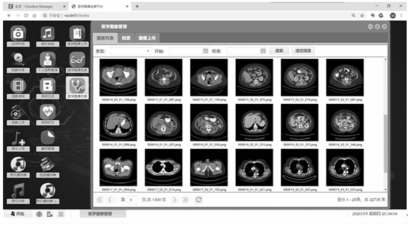

图 9　医学图像列表应用操作界面

在医学图像检索应用中，用户可以通过以图搜图的方式对医学图像库中已有的图像进行检索，检索结果按相似度从小到大进行排序。具体操作如图 10 所示。

5　结束语

针对海量医学图像存储、检索的效率问题，本文设计并实现一种基于 Flink 的医学图像检索系统。系统利用 Flink 批流一体的架构提高并行检索效率，实现实时图像编码、批量图像上传编码和图像并行检索。图像检索过程使

图 10　医学图像检索应用操作界面

用深度卷积神经网络模型提取图像特征并利用乘积量化编码模型进行特征编码。同时，通过 Web 应用作为用户操作入口，系统通过 HBase 存储医学图像数据和图像特征编码数据。实验结果表明，本系统具有更好的检索效率表现，满足实际应用需求。

参考文献：

[1] 于　凡，万艳丽，胡红濮. 医学图像检索技术发展现状 [J]. 中华医学图书情报杂志，2017，26 (7)：31－5.

[2] 王　倩，谭永杰，秦　杰，等. 基于 Hadoop 分布式平台的海量图像检索 [J]. 南京理工大学学报 (自然科学版)，2017，41 (4)：442－7.

[3] 王　立，陈军峰. Hadoop 分布式的海量图像检索 [J]. 现代电子技术，2018，41 (9)：70－5.

[4] 丁　灿，侯春萍，王宝亮. 基于 Hadoop 平台的图像检索分布式算法的改进研究 [J]. 南开大学学报：自然科学版，2017，4)：46－51.

[5] 范　敏，徐胜才. 基于 Hadoop 的海量医学图像检索系统 [J]. 计算机应用，2013，33 (12)：3345－9.

[6] 王　铭，田　茂，赵　鑫，等. 基于 Hadoop 平台的数据迁移方法研究实现 [J]. 计算机测量与控制，2018 (4)：225－230.

[7] Gionis A. Similarity Search in High Dimensions via Hashing [J]. Vldb. 1999，99 (6)：518－529.

[8] Weiss Y，Torralba A，Fergus R. Spectral Hashing [A]. proceedings of the International Conference on Neural Information Processing Systems，F [C]. 2008.

[9] Gong Y，Lazebnik S. Iterative quantization：A procrustean approach to learning binary codes [A]. proceedings of the Computer Vision & Pattern Recognition，F [C]. 2011.

[10] J Gou H，Douze M，Schmid C. Product Quantization for Nearest Neighbor Search [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence，2010，33 (1)：117－128.

[11] Norouzi M，Fleet D J. Cartesian K－Means [A]. proceedings of the Computer Vision and Pattern Recognition (CVPR)，2013 IEEE Conference on，F [C]. 2013.

[12] Wang J，Zhang T. Composite Quantization [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence，2017，(99)：1－1.

[13] Yan K，Wang X，Lu L，et al. DeepLesion：Automated Deep Mining，Categorization and Detection of Significant Radiology Image Findings using Large－Scale Clinical Lesion Annotations [D]. Cornell University，2017.