学号 姓名 班级 专业 院系

岭南师范学院2015 年－2016 学年度第二学期

期末考试试题A卷

(考试时间: 120分钟)

考试科目：大数据核心技术

题 号	一	二	三	四	五	总 分	总评分人	复查人
分 值	30	16	20	24	10			
得 分								

得分	评卷人

一、单项选择题(每小题 2 分，共 30 分)
请把答案写在下表中，写在试题后无效。

题号	1	2	3	4	5	6	7	8
答案								
题号	9	10	11	12	13	14	15	
答案								

1. 下面哪个程序负责 HDFS 数据存储。 (C)

A. NameNode B. Jobtracker

C. Datanode D. secondaryNameNode

2. HDFS 中的 block 默认保存几个备份。 (A)

A. 3 份 B. 2 份

C. 1 份 D. 不确定

3. HDFS1.0 默认 Block Size 大小是多少。 (B)

A. 32MB B. 64MB

C. 128MB D. 256MB

任课教师签名： 命题教师签名： 系主任签名： 主管院长签名：

4. 下面哪个进程负责 MapReduce 任务调度。 (B)

A. NameNode B. Jobtracker

C. TaskTracker D. secondaryNameNode

5. Hadoop1.0 默认的调度器策略是哪个。 (A)

A. 先进先出调度器 B. 计算能力调度器

C. 公平调度器 D. 优先级调度器

6. Client 端上传文件的时候下列哪项正确？ (B)

A. 数据经过 NameNode 传递给 DataNode

B. Client 端将文件切分为 Block，依次上传

C. Client 只上传数据到一台 DataNode，然后由 NameNode 负责 Block 复制工作

D. 以上都不正确

7. 在实验集群的 master 节点使用 jps 命令查看进程时，终端出现以下哪项能说明 Hadoop 主节点启动成功？ (D)

A. Namenode, Datanode, TaskTracker

B. Namenode, Datanode, secondaryNameNode

C. Namenode, Datanode, HMaster

D. Namenode, JobTracker, secondaryNameNode

8. 若不针对 MapReduce 编程模型中的 key 和 value 值进行特别设置，下列哪一项是 MapReduce 不适宜的运算。 (D)

A. Max B. Min

C. Count D. Average

9. MapReduce编程模型，键值对　　的key必须实现哪个接口？　（ A ）

A. WritableComparable　　B. Comparable

C. Writable　　D. LongWritable

10. 以下哪一项属于非结构化数据。（C）

A. 企业ERP数据　　B. 财务系统数据

C. 视频监控数据　　D. 日志数据

11. HBase数据库的BlockCache缓存的数据块中，哪一项不一定能提高效率。　（ D ）

A. -ROOT-表　　B. .META.表

C. HFile index　　D. 普通的数据块

12. HBase是分布式列式存储系统，记录按什么集中存放。　（ A ）

A. 列族　　B. 列

C. 行　　D. 不确定

13. HBase的Region组成中，必须要有以下哪一项。　（ B ）

A. StoreFile　　B. MemStore

C. HFile　　D. MetaStore

14. 客户端首次查询HBase数据库时，首先需要从哪个表开始查找。　（ B ）

A. .META.　　B. -ROOT-

C. 用户表　　D. 信息表

15、设计分布式数据仓库hive的数据表时，为取样更高效，一般可以对表中的连续字段进行什么操作。　（ A ）

A. 分桶　　B. 分区

C. 索引　　D. 分表

得分	评卷人

二、判断题(每题2 分，共16 分)
请在下表中填写√或者×，写在试题后无效。

题号	1	2	3	4	5	6	7	8
答案								

1. Hadoop 支持数据的随机读写。（hbase支持,hadoop不支持）　（ 错 ）

2. NameNode 负责管理元数据信息 metadata，client 端每次读写请求，它都会从磁盘中读取或会写入 metadata 信息并反馈给 client 端。（内存中读取）　（ 错 ）

3. MapReduce 的 input split 一定是一个 block。　（默认是）　（ 错）

4. MapReduce适于PB级别以上的海量数据在线处理。　（离线）　（ 错 ）

5. 链式 MapReduce计算中，对任意一个 MapReduce作业，Map 和 Reduce 阶段可以有无限个 Mapper，但Reducer只能有一个。　（ 对 ）

6. MapReduce计算过程中，相同的key默认会被发送到同一个reduce task处理。（ 对 ）

7. HBase对于空（NULL）的列，不需要占用存储空间。　（没有则空不存储）（ 对 ）

8. HBase可以有列，可以没有列族（column family）。　（有列族）　（ 错 ）

得分	评卷人

三、简答题(每小题5 分，共20 分)

1. 简述大数据技术的特点。

答：Volume（大体量）：即可从数百TB到数十数百PB、甚至EB规模。

Variety（多样性）：即大数据包括各种格式和形态的数据。

Velocity（时效性）：即很多大数据需要在一定的时间限度下得到及时处理。

Veracity（准确性）：即处理的结果要保证一定的准确性。

Value（大价值）：即大数据包含很多深度的价值，大数据分析挖掘和利用带来巨大的商业

学号＿＿＿＿＿ 姓名＿＿＿＿＿ 班级＿＿＿＿ 专业＿＿＿＿＿ 二级学院＿＿＿＿＿

…○…线…○…订…○…装…○…内…○…○…

价值。

2. 启动 Hadoop 系统，当使用 bin/start-all.sh命令启动时，请给出集群各进程启动顺序。

答：启动顺序：

namenode -> datanode -> secondarynamenode -> resourcemanager -> nodemanager

3. 简述 HBase 的主要技术特点。

答：（1）列式存储

（2）表数据是稀疏的多维映射表

（3）读写的严格一致性

（4）提供很高的数据读写速度

（5）良好的线性可扩展性

（6）提供海量数据

（7）数据会自动分片

（8）对于数据故障，hbase 是有自动的失效检测和恢复能力。

（9）提供了方便的与 HDFS 和 MAPREDUCE集成的能力。

4. Hive 数据仓库中，创建了以下外部表，请给出对应的 HQL 查询语句

```
CREATE EXTERNAL TABLEsogou_ext(
ts   STRING,   uid    STRING   keyword  STRING,
rank INT,       order INT,       url   STRING,
year INT,       month INT,       day   INT,    hour INT
)
COMMENT 'This is the sogou search data of extend data'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION '/sogou_ext/20160508';
```

（1）给出独立 uid 总数的 HQL 语句

答：select count(**distinct UID**) from sogou_ext;

（2）对于 keyword，给出其频度最高的 20 个词的 HQL 语句

答：select keyword from sogou_ext group by keyword order by order desc limit 20;

得分	评卷人

四、设计题(每小题 8 分，共 24 分)

1. 100 万个字符串，其中有些是相同的（重复），需要把重复的全部去掉，保留没有重复的字符串。请结合 MapReduce编程模型给出设计思路或核心代码。

P228

Public static class ProjectionMap extends

```
Mapper
{
Private int clo;
Project void setup(Context context) throws IOException,InterruptedException{
     Col=context.getConfiguration().getInt(“col”,0);
}
Public void map(LongWritable offset,Text line,Context context){
     RelationA record=new   RelationA (line.toString());
     Context.write(newText(record.getCol(col)),NullWritable.get());
}
}
```

REDUCE 端实现代码：

```
Public          static          class          ProjectionRedice          extends
Reducer
Public  void  reduce(Text  key,Iterable                value,Context  context)throws
IOException,InterruptedException{
Context.write(key,NullWritable.get());
}
```

2. 倒排索引设计。

有一个文档库，包含有大量的文档，现需要使用 MapReduce 编程技术对文档内容建立一个倒排索引库。要求 Reduce 最后输出键值对为<**单词，文件名#偏移量**>，并且前后输出的相同的 key 所对应的文件名是字典序的。如


```
word1    doc1#200
word1    doc2#10
word2    doc2#10
```

假设在 map 阶段已经获取了当前 split 分片的文件名是 String**filename**请按要求给出设计思路或核心代码。

```
Map(){
String filename=fileSplit.getPath().getName();
String temp=new String();
String line=value.toString().toLowerCase();
StringTokenizer iter=new StringTokenizer(line);
For(;itr.hasMoreTokens();){
  Temp=iter.nextToken();
  If(!stopwords contains(temp)){
     Text word=new Text();
     Word.set(temp+” #” +fileName);
     Context.write(word,new IntWritable(1));
  }
 }
}
Reducer{
   Private IntWritable result=new IntWritable();
   Public void reduce(Text ,key,Iterable values,Context context) throws
IOException,InterruptedException{
     Int sum=0;
     For(InWritable val:values){
         Sum+=val.get();
```

```
        }
        Result.set(sum);
        Context.write(key,result);
    }

}
```

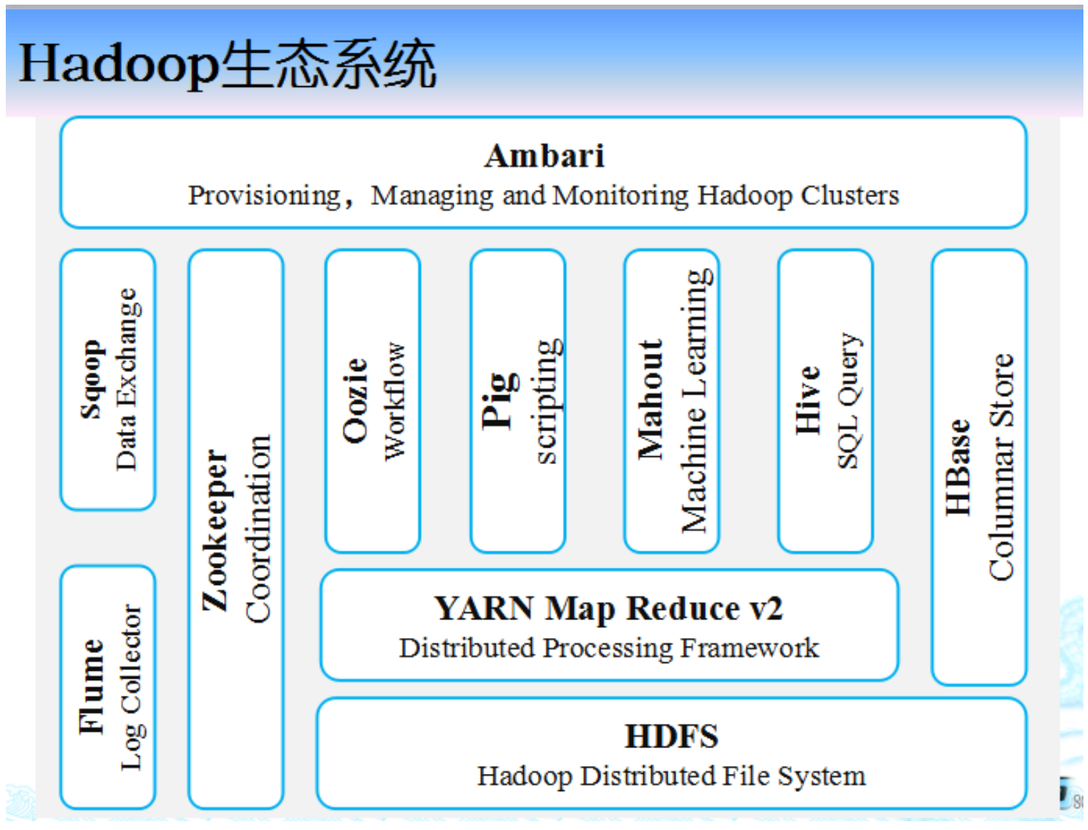
3. 请在下面程序的下划线中补充完整程序（共 8 处）。

得分	评卷人

```
public class WordCount {
    public static class TokenizerMapper extends
        Mapper<____Object_____, __Text ___ , ____Text_____,
IntWritable_____> {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();
        public void map(LongWritable key, Text value, Context context){
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken());
                context.write(word, one);
            }
        }
    }
    public static class IntSumReducer extends
            Reducer< ____Text_____ , _____IntWritable_____ , Text, IntWritable> {
        private IntWritable result = new IntWritable();
        public void reduce( __Texy_____ key, Iterable< ____IntWritable_____ > values,
Context context) {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            result.set(sum);
            context.write(key, result);
        }
    }
    public static void main(String[] args) throws Exception {
        略……
    }
}
```

五、开放题(每小题 10 分，共 10 分)

1. 谈谈对 Hadoop 系统的组成及其基本工作原理的理解。



1. MapReduce 并行计算框架

MapReduce并行计算框架是一个并行化程序执行系统。它提供了一个包含Map和Reduce两阶段的并行处理模型和过程，提供一个并行化编程模型和接口，让程序员可以方便快速地编写出大数据并行处理程序。MapReduce以键值对数据输入方式来处理数据，并能自动完成数据的划分和调度管理。在程序执行时，MapReduce并行计算框架将负责调度和分配计算资源，划分和输入输出数据，调度程序的执行，监控程序的执行状态，并负责程序执行时各计算节点的同步以及中间结果的收集整理。MapReduce框架提供了一组完整的供程序员开发MapReduce应用程序的编程接口。

2.分布式文件系统HDFS

HDFS（Hadoop Distributed File System）是一个类似于Google GFS的开源的分布式文件系统。它提供了一个可扩展、高可靠、高可用的大规模数据分布式存储管理系统，基于物理上分布在各个数据存储节点的本地Linux系统的文件系统，为上层应用程序提供了一个逻辑上成为整体的大规模数据存储文件系统。与GFS类似，HDFS采用多副本（默认为3个副本）数据冗余存储机制，并提供了有效的数据出错检测和数据恢复机制，大大提高了数据存储的可靠性。

3.分布式数据库管理系统HBase

为了克服HDFS难以管理结构化/半结构化海量数据的缺点，Hadoop提供了一个大规模分布式数据库管理和查询系统HBase。HBase是一个建立在HDFS之上的分布式数据库，它是一个分布式可扩展的NoSQL数据库，提供了对结构化、半结构化甚至非结构化大数据的实时读写和随机访问能力。HBase提供了一个基于行、列和时间戳的三维数据管理模型，HBase中每张表的记录数（行数）可以多达几十亿条甚至更多，每条记录可以拥有多达上百万的字段。

4.公共服务模块Common

Common是一套为整个Hadoop系统提供底层支撑服务和常用工具的类库和API编程接口，这些底层服务包括Hadoop抽象文件系统FileSystem、远程过程调用RPC、系统配置工具Configuration以及序列化机制。在0.20及以前的版本中，Common包含HDFS、MapReduce和其他公共的项目内容；从0.21版本开始，HDFS和MapReduce被分离为独立的子项目，其余部分内容构成Hadoop Common。

5.数据序列化系统Avro

Avro是一个数据序列化系统，用于将数据结构或数据对象转换成便于数据存储和网络传输的格式。Avro提供了丰富的数据结构类型，快速可压缩的二进制数据格式，存储持久性数据的文件集，远程调用RPC和简单动态语言集成等功能。

6.分布式协调服务框架Zookeeper

Zookeeper是一个分布式协调服务框架，主要用于解决分布式环境中的一致性问题。Zookeeper主要用于提供分布式应用中经常需要的系统可靠性维护、数据状态同步、统一命名服务、分布式应用配置项管理等功能。Zookeeper可用来在分布式环境下维护系统运行管理中的一些数据量不大的重要状态数据，并提供监测数据状态变化的机制，以此配合其他Hadoop子系统（如HBase、Hama等）或者用户开发的应用系统，解决分布式环境下系统可靠性管理和数据状态维护等问题。

7.分布式数据仓库处理工具Hive

Hive是一个建立在Hadoop之上的数据仓库，用于管理存储于HDFS或HBase中的结构化/半结构化数据。它最早由Facebook开发并用于处理并分析大量的用户及日志数据，2008年Facebook将其贡献给Apache成为Hadoop开源项目。为了便于熟悉SQL的传统数据库使用者使用Hadoop系统进行数据查询分析，Hive允许直接用类似SQL的HiveQL查询语言作为编程接口编写数据查询分析程序，并提供数据仓库所需要的数据抽取转换、存储管理和查询分析功能，而HiveQL语句在底层实现时被转换为相应的MapReduce程序加以执行。

8.数据流处理工具Pig

Pig是一个用来处理大规模数据集的平台，由Yahoo!贡献给Apache成为开源项目。它简化了使用Hadoop进行数据分析处理的难度，提供一个面向领域的高层抽象语言Pig Latin，通过该语言，程序员可以将复杂的数据分析任务实现为Pig操作上的数据流脚本，这些脚本最终执行时将

被系统自动转换为 MapReduce任务链，在 Hadoop 上加以执行。Yahoo!有大量的 MapReduce作业是通过 Pig实现的。

9.键值对数据库系统 Cassandra

Cassandra是一套分布式的 K-V 型的数据库系统，最初由 Facebook 开发，用于存储邮箱等比较简单的格式化数据，后 Facebook 将 Cassandra贡献出来成为 Hadoop 开源项目。Cassandra以 Amazon 专有的完全分布式 Dynamo 为基础，结合了 Google BigTable基于列族（Column Family）的数据模型，提供了一套高度可扩展、最终一致、分布式的结构化键值存储系统。它结合了 Dynamo 的分布技术和Google 的Bigtable数据模型，更好地满足了海量数据存储的需求。同时，Cassandra变更垂直扩展为水平扩展，相比其他典型的键值数据存储模型，Cassandra提供了更为丰富的功能。

10.日志数据处理系统 Chukwa

Chukwa 是一个由 Yahoo! 贡献的开源的数据收集系统，主要用于日志的收集和数据的监控，并与 MapReduce 协同处理数据。Chukwa 是一个基于 Hadoop 的大规模集群监控系统，继承了 Hadoop 系统的可靠性，具有良好的适应性和扩展性。它使用 HDFS 来存储数据，使用 MapReduce来处理数据，同时还提供灵活强大的辅助工具用以分析、显示、监视数据结果。

11.科学计算基础工具库 Hama

Hama 是一个基于 BSP 并行计算模型（Bulk Synchronous Parallel 大同步并行模型）的计算框架，主要提供一套支撑框架和工具，支持大规模科学计算或者具有复杂数据关联性的图计算。Hama类似 Google 公司开发的 Pregel，Google 利用 Pregel 来实现图遍历（BFS）、最短路径（SSSP）、PageRank等计算。Hama 可以与 Hadoop 的 HDSF 进行完美的整合，利用 HDFS 对需要运行的任务和数据进行持久化存储。由于 BSP 在并行化计算模型上的灵活性，Hama 框架可在大规模科学计算和图计算方面得到较多应用，完成矩阵计算、排序计算、PageRank、BFS 等不同的大数据计算和处理任务。

12.数据分析挖掘工具库 Mahout

Mahout 来源于 Apache Lucene子项目，其主要目标是创建并提供经典的机器学习和数据挖掘并行化算法类库，以便减轻需要使用这些算法进行数据分析挖掘的程序员的编程负担，不需要自己再去实现这些算法。Mahout 现在已经包含了聚类、分类、推荐引擎、频繁项集挖掘等广泛使用的机器学习和数据挖掘算法。此外，它还提供了包含数据输入输出工具，以及与其他数据存储管理系统进行数据集成的工具和构架。

13.关系数据交换工具 Sqoop

Sqoop 是 SQL-to-Hadoop 的缩写，是一个在关系数据库与 Hadoop 平台间进行快速批量数据交换的工具。它可以将一个关系数据库中的数据批量导入 Hadoop 的 HDFS、HBase、Hive 中，也可以反过来将 Hadoop 平台中的数据导入关系数据库中。Sqoop 充分利用了 Hadoop MapReduce的并行化优点，整个数据交换过程基于 MapReduce实现并行化的快速处理。

14.日志数据收集工具 Flume

Flume 是由 Cloudera 开发维护的一个分布式、高可靠、高可用、适合复杂环境下大规模日志数据采集的系统。它将数据从产生、传输、处理、输出的过程抽象为数据流，并允许在数据源中定义数据发送方，从而支持收集基于各种不同传输协议的数据，并提供对日志数据进行简单的数据过滤、格式转换等处理能力。输出时，Flume 可支持将日志数据写往用户定制的输出目标。