2023-2024第1学期大数据系统开发考试说明

*本次考试的重点内容是基本原理和实验，所列出的内容要深入理解基本概念（没有列出的内容基本不考），这些基本概念包括实验中的很多内容，如：如何设置NameNode。除此之外，还要求掌握HDFS、HBase的基本操作命令。请大家在准备考试时按下面内容进行，特别是实验部分。*

1. **大数据基本概念**
2. 大数据特征

->4V，体量，多样性，速度，价值密度

1. 关系数据库与NoSQL数据库的差异

->关系数据库（Relational Database）和NoSQL数据库（Not Only SQL Database）的主要差异在于数据模型、一致性要求、可扩展性、事务支持和适用场景。关系数据库使用表格形式存储结构化数据，强调一致性和复杂查询，适用于金融和企业应用。NoSQL数据库采用多种数据模型，强调横向扩展和灵活性，适用于社交媒体、大数据分析等应用，但一致性和事务支持可变。选择数据库类型应根据具体需求。

1. 大数据库关键技术

->大数据处理涉及多种技术，包括分析技术（自然语言处理、统计与分析、数据挖掘、模型预测）、大数据技术（数据采集、数据存取、基础架构支持、计算结果展现）、以及存储技术（结构化数据、非结构化数据、半结构化数据的存储与处理）。

1. 大数据处理流程

->数据采集，数据存储，数据处理，数据挖掘与分析，数据可视化与决策

1. 大数据系统架构

->数据感知层（数据采集层），数据文件系统（数据存储层），数据仓库与计算层（数据管理与计算层），接口层、分析工具层、模型可视化层（数据分析和挖掘层），（数据展现和可视化层），业务应用层（安全与管理层，云计算和容器化）

1. **Hadoop**
2. 什么是Hadoop

->Hadoop是一个能够对大量数据进行分布式处理的软件框架。它是一个开源的、分布式计算平台，可以用于存储和处理大规模的数据，具有高可扩展性、高效性、高可靠性等特点。Hadoop的核心思想是将数据分布式存储在多个节点上，并使用MapReduce等计算模型对数据进行处理和分析。

1. Hadoop的核心架构

->Hadoop Distributed File System（HDFS）：分布式文件系统，用于存储大数据。

MapReduce：分布式计算框架，用于并行处理大规模数据集。

YARN（Yet Another Resource Negotiator）：资源管理框架，用于集群资源的管理和调度

1. Hadoop生态圈

->Hadoop生态圈是一组与Hadoop紧密集成的开源项目和工具，用于扩展和增强Hadoop的功能。生态圈包括多个子项目，如Hive（用于SQL查询）、Pig（用于数据流编程）、HBase（分布式数据库）、Spark（快速实时大数据处理，内存计算）、Zookeeper（分布式协调服务），Mahout（数据挖掘库），Oozie（作业流调度系统），MapReduce（分布式计算框架），Tez（DAG计算），Yarn（资源管理系统），Flume（日志收集），Sqoop（数据库ETL工具）和Ambari（安装部署工具）等

1. Hadoop的优点

->开源：Hadoop是开源的，源代码可自由访问和修改。

优秀的生态圈：Hadoop生态圈包括多个工具和框架，可以满足各种大数据处理需求。

高扩展性：Hadoop可以轻松扩展到数以千计的节点，适应不断增长的数据量。

高效性：Hadoop具有分布式计算能力，能够快速处理大规模数据。

高可靠性：Hadoop自动保存数据的多个副本，能够自动重新分配失败的任务。

低成本：可以使用廉价的机器构建服务器集群来处理数据，成本相对较低。

1. Hadoop处理大数据的优势

->天然优势：Hadoop在数据提取、变形和加载（ETL）方面具有天然优势，能够将大规模数据处理引擎尽可能靠近存储。

分布式架构：Hadoop的MapReduce功能能够将单个任务分解为多个碎片任务，并在多个节点上并行处理，从而提高了处理速度。

数据可靠性：Hadoop自动保存数据的多个副本，确保数据的可靠性和容错性。

1. **什么是HDFS**

**->HDFS（Hadoop Distributed File System）是一种分布式文件系统，用于处理大规模数据集。它是Apache Hadoop的核心组件之一，专门设计用于在商业硬件上运行的大型数据集。HDFS允许将单个Hadoop集群扩展到数百甚至数千个节点，提供高可靠性、高容错性、高吞吐量和可扩展性的分布式存储和数据处理。HDFS用于存储和管理大数据，同时支持文件的分布式访问和处理。**

1. HDFS的目标

->硬件故障管理：HDFS旨在处理常态下的硬件故障，保证数据的可靠性和可用性。

流式数据访问：HDFS适用于批处理操作，重点在数据吞吐量而不是低延迟的数据访问。

大数据集：HDFS设计用于处理大文件和大数据集，支持高聚合数据带宽。

简单一致性模型：HDFS适用于一次写入多次读取的操作模式，适合批处理和数据分析应用。

移动计算比移动数据更经济：HDFS支持在靠近数据存储位置进行计算，减少了数据传输的网络拥堵。

异构软硬件平台间的可移植性：HDFS设计支持跨平台迁移，推动大数据应用的广泛采用。

1. HDFS体系结构

->名字节点（NameNode）：维护文件系统的命名空间和元数据，记录文件和目录的层次结构。

辅助名字节点（SecondaryNameNode）：协助管理Hadoop分布式文件系统的元数据，但不提供故障转移功能。

数据节点（DataNode）：存储实际的数据块，定期向名字节点发送心跳信号和块报告。

块（Block）：HDFS将文件划分为块，典型块大小为128MB，默认情况下每个文件块有3个副本。

客户端（Client）：用于与HDFS交互，执行文件读取、写入等操作。

1. HDFS块：大小、副本

->块大小：HDFS典型的块大小为128MB，可以配置为不同大小，但典型的大小有助于提高数据吞吐量。

块副本：每个文件可以配置块副本数量，默认为3，可以在文件创建或修改时指定。副本有助于提高数据的可靠性和可读性。作用是防止因为DataNode故障而导致数据丢失。

1. HDFS读写流程

->写入流程：客户端将数据写入本地文件，当本地文件大小达到一个块的大小时，客户端通知名字节点并刷新数据块到数据节点。

1. 客户端向名字节点请求上传文件
2. 名字节点响应可以上传文件
3. 客户端请求上传第一个block，请返回数据节点
4. 名字节点返回dn节点，表示可以采用三个节点进行存储数据
5. 客户端请求于dn建立通道（一条穿三个）
6. dn3，dn2，dn1应答客户端成功
7. 传输数据
8. 客户端想名字节点表达传输完毕

读取流程：客户端通过名字节点获取文件块的位置信息，然后直接从数据节点读取数据块。

1. 客户端向名字节点请求下载指定文件
2. 名字节点返回目标文件的元数据
3. 客户端向dn1请求读数据
4. Dn1传输数据去客户端
5. 客户端继续向下一个dn2请求读数据
6. Dn2传输数据去客户端
7. HDFS创建目录、列表目录，传送文件的基本命令。

->创建目录：hdfs dfs -mkdir /目录名

列表目录：hdfs dfs -ls /目录名

传送文件：hdfs dfs -put 本地文件路径 HDFS目标路径

查看文件: hdfs dfs -cat ...

删除文件: hdfs dfs -rm ...

1. **什么是HBase**
2. HBase的特点，可扩展性、性能等。

->易扩展性： HBase可以通过增加RegionServer节点和DataNode节点来扩展运算和存储能力，从而应对不断增长的数据。

海量存储： HBase专注于PB级别数据的实时入库和快速随机访问，使其成为处理大规模数据的理想选择。

高可靠性： HBase具备高可靠性，通过WAL（Write-Ahead-Log）机制保证数据写入时不会因集群异常而导致数据丢失，同时Replication机制保证了数据的备份。

稀疏性： 在HBase的列族中，可以指定任意多的列，为空的列不占用存储空间，这使得HBase可以存储稀疏数据。

1. HBase体系结构

->HMaster： 负责管理RegionServer，实现负载均衡，管理和分配Region，管理命名空间和表的元数据以及权限控制。

RegionServer： 存放和管理本地Region，读写HDFS，管理Table中的数据，响应客户端请求。

ZooKeeper： 存放整个HBase集群的元数据和状态信息，实现HMaster主从节点的failover。

1. HBase压缩

->HBase支持数据压缩，可以通过配置来启用不同的压缩算法，以减少存储空间和提高读写性能。

1. HBase日志

->预写日志（WAL）是HBase的日志机制，用于记录所有的数据改动。它在灾难恢复时起到关键作用，通过重做WAL中记录的改动，可以恢复崩溃之前的数据。如果写入WAL失败，整个操作将被认为失败。

1. HBase读写流程

->写流程： 当客户端要写入数据时，首先会写入预写日志（WAL），然后数据被写入内存中的MemStore。当MemStore中的数据达到一定大小时，会被刷写到HDFS中。此外，HBase还使用WAL来保证数据的一致性和恢复。

读流程： 客户端从HMaster获取元数据，找到RowKey所在的RegionServer，然后进行单行查询、范围查询或全表扫描查询。

HBase写流程：

客户端发起写请求： 客户端应用程序首先连接到 HBase 集群，然后发送写请求。写请求通常包括数据的行键（RowKey）和一个或多个列值（Column Values）。

写入预写日志（Write-Ahead-Log，WAL）： HBase 使用预写日志（WAL）来记录写操作。WAL 是一种持久化的日志，它用于灾难恢复。每个写入请求都会先写入 WAL，以确保数据不会丢失。

写入MemStore： 数据被写入 MemStore，它是位于 RegionServer 内存中的数据存储结构。在 MemStore 中，数据会被排序并存储在内存中，以提供快速的读取和写入操作。

合并操作： 当 MemStore 中的数据量达到一定阈值时，HBase 会触发一个合并操作，将 MemStore 中的数据合并到磁盘上的一个数据文件（HFile）中。

数据持久化： 合并后的数据存储在 HFile 中，这是 HBase 的底层数据存储格式。HFile 存储在 HDFS 中，因此数据是持久的。

MemStore 清空： 一旦数据持久化成功，MemStore 就会被清空，为下一轮写入做好准备。

HBase 索引更新： HBase 的内部索引（Bloom Filter、Block Index 等）也会在写入时更新，以提高读取性能。

客户端响应： 客户端在数据写入成功后会收到响应，确认写操作已成功完成。

HBase读流程：

客户端发起读请求： 客户端应用程序连接到 HBase 集群，并发送读请求，请求包括所需数据的行键（RowKey）。

获取 HBase 表的元数据： 客户端会向 HMaster 请求表的元数据，这些元数据包括 Region 的位置信息。

定位 RegionServer： 客户端使用元数据确定包含所需数据的 RegionServer，并将读取请求发送到该 RegionServer。

查找 MemStore： 如果数据在 MemStore 中（尚未持久化到磁盘），则从 MemStore 中读取数据，这可以实现快速的随机访问。

查找 HFile： 如果数据不在 MemStore 中，客户端会根据数据的 RowKey 查找相应的 HFile，这通常需要进行一次磁盘访问。

合并数据： 如果数据分布在多个 HFile 中，HBase 会在内存中将数据合并，以生成完整的结果。

返回结果： HBase 会将读取的数据返回给客户端，客户端应用程序可以处理数据或将其展示给用户。

1. HBase创建表、增、删、改、查记录的命令。

->创建表： create 'table\_name', 'column\_family'

增加记录： put 'table\_name', 'row\_key', 'column\_family:column\_qualifier', 'value'

删除记录： delete 'table\_name', 'row\_key', 'column\_family:column\_qualifier', 'timestamp'

修改记录： put 'table\_name', 'row\_key', 'column\_family:column\_qualifier', 'new\_value'

查找记录： get 'table\_name', 'row\_key'

1. **MapReduce**
2. MapReduce是什么？

->MapReduce是一个软件框架，用于在大规模商用机器集群上以可靠的、具备容错能力的方式并行处理海量数据集。它允许用户编写应用程序，能够分布式处理上TB级别的数据集

1. MapReduce体系架构与处理流程

->MapReduce的体系结构包括六大过程：Input, Split, Map, Shuffle, Reduce, Finalize。输入数据被切分成多个数据块，然后经过映射（Map）和归并（Shuffle）阶段处理，最终由Reduce阶段生成结果

1. MapReduce如何在分布环境中处理词频统计、倒排索引

->在分布式环境中，MapReduce可以用于处理词频统计和倒排索引等任务。Map阶段将数据分解为键值对，然后通过Shuffle和Reduce阶段对数据进行分组和聚合，生成最终的统计结果或索引。

词频统计：

Map 阶段：输入数据被切分成多个数据块，每个 Map 任务负责处理一个数据块。在词频统计中，每个 Map 任务将文本数据拆分成单词，并为每个单词生成一个键值对，其中键是单词，值是1（表示单词出现一次）。

Shuffle 阶段：在 Shuffle 阶段，MapReduce 框架会将所有相同键的数据汇总在一起，以便后续的 Reduce 阶段能够处理相同单词的计数。这个过程需要将键值对按照键进行排序和分组。

Reduce 阶段：Reduce 任务会接收到来自多个 Map 任务的键值对，并对相同的键进行合并和计数。在词频统计中，Reduce 任务将计数累加，从而得到每个单词的总出现次数。

Finalize 阶段：最终的结果会被保存到输出文件或存储系统中，以供进一步分析或查询。

倒排索引：

Map 阶段：与词频统计类似，Map 阶段首先将输入数据切分成多个数据块。然后，每个 Map 任务会处理一个数据块，并从文档中提取单词和文档标识符。对于每个单词，Map 任务生成一个键值对，其中键是单词，值是文档标识符。

Shuffle 阶段：在 Shuffle 阶段，MapReduce 框架将所有具有相同单词的文档标识符汇总在一起，以建立倒排索引。这个过程需要对键进行排序和分组。

Reduce 阶段：每个 Reduce 任务会接收到一个单词和与之相关的文档标识符列表。在倒排索引中，Reduce 任务将这些文档标识符列表合并成一个文档列表，表示包含该单词的所有文档。

Finalize 阶段：最终的倒排索引结果会被保存到输出文件或存储系统中，以供文本搜索或检索系统使用。

总之，MapReduce 提供了一个有效的方式来处理大规模数据集，并可以轻松地在分布式环境中执行词频统计和倒排索引等任务。这些任务可以利用MapReduce的并行处理和分布式计算能力来实现高效的数据处理。

1. **实验部分**
2. Hadoop全分布部署模式

**Hadoop全分布部署模式： Hadoop可以以伪分布式或全分布式模式部署。全分布式模式是指Hadoop集群的各个组件（如HDFS和YARN）在不同的物理节点上运行，充分利用了大规模集群的计算和存储能力。**

1. 各种配置文件core-site.xml、hdfs-site.xml、yarn-site.xml、zookeeper.xml、hbase-site.xml的主要作用、重点属性（少数几个，如配置hdfs端口的属性）。

**配置文件主要作用及重点属性：**

**core-site.xml：包含通用配置属性，如Hadoop集群的名称、HDFS的默认文件系统URI。**

**hdfs-site.xml：包含HDFS配置属性，如数据块大小、副本数量。**

**yarn-site.xml：包含YARN配置属性，如资源管理器地址、NodeManager配置。**

**zookeeper.xml：包含ZooKeeper配置属性，用于HBase的元数据管理和HA。**

**hbase-site.xml：包含HBase配置属性，如ZooKeeper地址、RegionServer数量。**

1. Job、task与处理的数据之间的关系

**Job、task与处理的数据之间的关系： Job是Hadoop中的一个计算任务，通常由多个Map和Reduce任务组成。Task是作业的执行单元，分为Map任务和Reduce任务，它们负责处理作业的输入数据。**

1. MapReduce的主要接口

**MapReduce的主要接口： 主要的MapReduce接口包括Mapper、Reducer、InputFormat和OutputFormat。Mapper和Reducer定义了数据处理的逻辑，InputFormat和OutputFormat用于定义数据的输入和输出格式。**

1. MapReduce实现倒排索引（原理及代码分析）

**MapReduce实现倒排索引： 倒排索引是一种用于快速查找文档的数据结构，MapReduce可以用来构建倒排索引。Mapper接收文档并生成键值对，其中键是词汇，值是文档标识。Reducer将键值对按词汇分组，并生成倒排索引。**

1. MapReduce移动计算、每个阶段的输入、输出。

**MapReduce移动计算、每个阶段的输入、输出： MapReduce通常涉及Map阶段和Reduce阶段。在Map阶段，输入数据被分割和映射到键值对；在Reduce阶段，相同键的值被汇总和处理。MapReduce可用于分布式计算，例如倒排索引构建。**

1. HBase写入记录的过程（原理与分析代码）

**HBase写入记录的过程： 写入HBase的过程包括客户端向表中插入记录，首先将数据写入WAL，然后写入MemStore，然后合并数据到HFile。这个过程确保了数据的持久性和可恢复性。**

1. 打包、运行实验程序(Java)的命令。

**打包、运行实验程序(Java)的命令： 编译Java程序并打包成JAR文件，然后使用hadoop jar命令来提交作业到Hadoop集群运行。例如：hadoop jar myjob.jar com.example.MyJob input\_path output\_path。**

1. Hadoop进行开发搜索引擎的优势

**Hadoop进行开发搜索引擎的优势： Hadoop和其生态系统提供了分布式处理和存储大规模数据的能力，这使得它在开发搜索引擎时具有优势。Hadoop的分布式计算能力可以用于构建索引、处理查询和分析用户行为，而HBase可以用于存储索引和元数据。此外，Hadoop的可扩展性和容错性也使其成为搜索引擎开发的有力工具。**

**注：不要求写属性名、代码、安装、部署等相应的命令，给出属性、命令、代码能够识别与分析，例如，能够识别Mapper接口方法、该接口的作用及使用方法（例如，分析代码）。**