大数据期末复习题（按知识点分类）

一、HDFS（分布式文件系统）

填空题

1. Hadoop2.X中HDFS默认块（Block）的大小为 128MB
2. NameNode的作用是 负责协调集群中的数据存储
3. DataNode的作用是 存储被拆分的数据块
4. HDFS中最小的数据存储单位是 Block（块）
5. HDFS中的Block默认保存 3份
6. 默认的NameNode Web管理端口是 9870

选择题

1. 以下哪个程序负责HDFS存储分布（D）？
   A.NameNode B.YARN C.SecondaryNameNode D.DataNode
2. SecondaryNameNode的作用是（C）
   A.监控NameNode B.管理DataNode C.合并fsimage和editlogs D.支持NameNode HA
3. Hadoop Client端上传文件的时候，下列哪项是正确的（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.Client只上传数据到多台DataNode，然后由DataNode负责Block复制工作
4. 在Hadoop 3.X的非高可用部署模式下，以下哪个组件通常与NameNode部署在同一节点？（C）
   A.DataNode B.NodeManager C.SecondaryNameManager D.ResourceManager

判断题

2. Hadoop支持数据的随机读写（×）
3. HDFS采用多副本机制，默认副本数为3，所有副本储存在同一个机架上以提升读取速度（×）
4. Hadoop 3.X支持Erasure Coding（纠删码），可在保证容错能力的同时显著降低存储开销，默认已替代三副本机制（×）

5. 在Hadoop集群中，DataNode会周期性向NameNode发送心跳和块报告，若NameNode在10min内未收到心跳，则认为该DataNode已失效（√）
6. HDFS的SecondaryNameNode是NameNode的热备份，当NameNode故障时可立即接管服务（×）
7. Hadoop中DataNode数据节点会定期向NameNode发送“心跳”信息，向NameNode报告自己的状态（√）

二、MapReduce（分布式计算框架）

填空题

4. Hadoop框架中最核心的设计是为海量数据提供储存的 HDFS 和对数据进行计算的 MapReduce
5. MapReduce的最小计算单元是 Split
6. MapReduce的shuffle过程包括三个子过程: sort、partition、merge（combine）
7. 在大数据处理框架中，Hadoop MapReduc通过将任务分布到集群中的多个节点上执行，体现了 分布式计算 的思想；每个Map和Reduce任务内部利用多线程处理数据分片属于 并行计算
8. MapReduce中Mapper类调用map方法的次数等于Split的 行数
9. MapReduce适合PB级以上海量数据的 离线 处理
10. Hadoop中类TextInputFormat是默认的FileInputFormat实现类。按行读取每条记录，键是 行偏移量，类型是 LongWritale；值是 行数据，类型是 Text
11. Reducer的数目由mapred-site.xml配置文件里的项目mapred.reduce.tasks决定。默认值为1

选择题

11. 在MapReduce中Shuffle阶段的Combiner的作用是什么（B）
    A.合并来自不同Mapper的输出 B.在Mapper端对的相同键的值进行局部聚合
    C.在Reducer端对相同键的值进行全局聚合 D.将Mapper和Reducer合并为一个阶段
12. 关于HDFS、YARN和MapReduce的协作关系，以下说法正确的是（C）
    A.MapReduce直接管理DataNode和NodeManager B.YARN从HDFS中读取输入数据并返回结果
    C.MapReduce应用向YARN申请计算资源，并从HDFS读取/写入数据
    D.NameNode负责调度Map和Reduce任务

判断题

1. MapReduce的中间结果（Map输出）会写入HDFS，以保证容错性（×）
2. MapReduce的Split大小默认和HDFS中Block大小是一样的（√）
3. MapReduce的输入数据集是静态的，不能动态变化（√）
4. 在MapReduce中，Combiner函数的作业是减少Map端输出到Reduce端的数据量，其逻辑必须与Reducer完全一致（√）

三、YARN（Hadoop资源管理）

填空题

3. Hadoop的资源管理器是 Yarn
4. 在YARN架构中，运行在每个节点上的NodeManager负责向ResourceManager汇报本节点的资源使用情况，并启动/架空容器（Container）中的任务
5. YARN使用Container（容器）作为资源抽象单位，封装了CPU、内存等计算资源，由ResourceManager统一管理和分配
6. Hadoop客户端提交应用程序到YARN后，首先由ResourceManager分配一个Container用于启动 ApplicationMaster，该组件随后向ResourceManager申请资源以运行具体的任务
7. YARN将Hadoop 1.X中 JobTracker的功能分为了两个独立的服务：全局的ResourceManager负责群资源管理，每个应用程序的ApplicationMaster 负责该应用的任务调度

判断题

14. YARN中的ApplicationMaster负责向ResourceManager申请资源 ，并在NodeManager上启动和监控任务容器。 （√）

四、Spark核心（RDD、DAG、调度、依赖、持久化）

填空题

2. Spark是用 Scala 语言开发的（注：原题写Scale是笔误，正确为Scala）
3. 在Spark中，一个RDD被划分为多个 分区，这些单元被调度到集群的不同节点上并行处

理，体现了“数据本地性”原则，以减少跨网络的数据传输

4. Spark从集合中创建RDD的方法有两种：Parallelize 和 makeRDD
5. Spark运行速度比MapReduce快的原因是：内存计算 和 DAG优化
6. Spark的 DAGScheduler（或任务）调度器能根据数据位置动态分配任务，优先将计算任务调度到存储该数据的节点上，从而减少网络传输，这一特性被称为 数据本地性
7. 在迭代式计算（如机器学习）场景中，Spark可通过 缓存（持久化）机制将数据缓存复用，显著减少重复计算；而MapReduce每次迭代都要重新读取输入数据并写入中间结果。
8. Spark中的RDD名为 弹性分布式数据集
9. Spark将中间结果储存在 内存中，避免了MapReduce在每个作业阶段都将数据写入 磁盘（HDFS） 带来大量的I/O开销
10. Spark使用DAG（有向无环图）执行引擎，能够将多个操作流水线化并优化执行计划；而MapReduce采用固定的 Map - Reduce两阶段模型，难以跨阶段优化
11. RDD和它依赖的父RDD的关系有两种不同的类型：宽依赖和窄依赖
12. 要使用Spark，开发者需要编写一个程序 Driver，它被提交到集群以调度运行Worker。在其中定义了一个或多个RDD，并调用RDD上的 action，Worker则执行RDD分区计算任务
13. Spark的驱动器是执行开发程序中main方法的进程。她负责开发人员编写的用来创建SparkContext对象、创建RDD。以及RDD转换操作代码的执行。如果你是使用spark shell，那么当你启动Spark shell时，系统后台自动启动了一个Spark驱动器程序，就是在Spark shell中预加载的一个叫做sc的 SparkContext对象。如果驱动器程序终止，那么Spark应用也就结束。
14. Spark Executor是一个工作进程，负责在Spark作业中运行任务，任务间互相独立。Spark应用启动时，该节点被同时启动，并且始终伴随着整个Spark应用的生命周期而存在
15. RDDs只维护血缘关系，也称之为依赖。依赖包括窄依赖：RDDs之间分区是一对一 的；另一种是宽依赖，下游RDD的每个分区与上游RDD（也成为父RDD）的每个分区都有关，是多对多的关系
16. 虽然RDD的血缘关系天然地可以实现容错，当RDD的某个分区数据失败或丢失可以通过血缘关系重建，但是对于长时间迭代型应用来说，随着迭代的进行，RDDs之间的血缘关系会越来越长，一旦在后续迭代过程中出错，则需要通过非常长的血缘关系去重建，势必影响性能。为此，RDD支持 checkpoint将数据保存到持久化的储存中，这样就可以切断之前的血缘关系，因为它之后的RDD不需要知道他的父RDD了，它可以从 checkpoint中拿到数据
17. DAG的全程叫做 有向无环图，原始的RDD通过一系列转化就形成了DAG，根据RDD之间的依赖关系将不同的DAG划分成不同的Stage，对于宽依赖，由于有Shuffle的存在，只能在父RDD处理完成后才能开始接下来的计算，因此 宽依赖是划分Stage的依据
18. Spark Core 实现了Spark的基本功能，包含任务调度、内存管理、错误恢复、与储存系统交互等模块。他还包含了对RDD的API定义

19. Spark的DAG调度器会根据依赖关系将作业划分为多个 Stage，其中 宽依赖是划分的边界
20. Spark对于频繁使用的RDD可以使用 缓存（Cache）或检查点（Checkpoint）将其保存在内存或磁盘中，避免重复计算
21. 窄依赖是指 父RDD的每个分区最多只会被子RDD的一个分区使用（注：原题描述宽依赖是笔误，此为窄依赖定义）；宽依赖通常会触发 Shuffle操作，可能导致性能瓶颈，容错性较差
22. Master是Spark集群的主节点，负责协调整个集群的资源分配和任务调度。它接收来自客户端提交的应用程序，并为这些应用程序分配资源
23. Worker 是Spark集群中的工作节点，负责执行实际的计算任务。可以启动多个 Executor进程，这些进程用于执行任务
24. Spark中每个Stage有一组可以以 流水线（pipeline）方式执行的任务组成，这些任务在不同分区上并行运行，且无需跨节点交换数据
25. 在Spark UI中， 一个Job包含若干Stage，Stage数量等于该Job的DAG中 Shuffle（宽依赖）操作的次数加一

选择题

6. 关于Spark，以下论述错误的是（D）
   A.transformations操作延迟执行 B.Action操作触发执行 C.RDD由一系列partition组成 D.每个RDD和其他RDD没有依赖关系
7. 执行PySpark代码：`sc.parallelize([1, 2, 3, 4, 5]).map(lambda x: x+1).collect()`，结果为（B）
   A.list(list(1,2,3,4,5),1) B.list(2,3,4,5,6) C.list(1,2,3,4,5,1) D.16
8. 下面算子属于Action算子的是（D）
   A.map B.filter C.reduceByKey D.count
9. 下列关于reduceByKey和GroupByKey正确的是（B）
   A.两者都只产生窄依赖 B.reduceByKey会在map端进行预聚合，减少shuffle数据量
   C.groupByKey比reduceByKey更高效 D.两者都不会触发Shuffle
10. 下面那个算子会触发宽依赖（shuffle）？ （B）
    A.union B.distinct C.flatmap D.sample
11. 关于cache()和persist()，以下说法正确的是（C）
    A.他们是Action算子，会立即触发计算并将结果写入磁盘
    B.他们只能将RDD存储在内存中，不支持磁盘存储
    C.他们是Transformation算子，用于标记RDD为可重用，并指定存储级别
    D.调用cache()后，RDD会在第一次被使用时持久化到所有Executor的本地磁盘。

判断题

6. Spark弹性分布式数据集RDD是可以修改的（×）
7. 在Spark中每个Stage包含多个Task（√）
8. Spark的RDD是可变的分布式对象集合，创建后可以修改（×）
9. Spark的容错机制依赖于血缘关系（Lineage）（√）
10. 在Spark中，collect()操作会将RDD所有数据拉取到Driver节点，适用于大数据集的返回（×）
11. map()和filter()操作会触发宽依赖，导致Shuffle的发生（×）

五、Spark SQL & Spark Streaming

填空题

23. Spark最新的SQL查询起始点是 SparkSession
24. Spark SQL新增了两种数据抽象：DataFrame和DataSet
25. Spark Streaming使用 微批次架构， 把流式计算当作一系列连续的小规模批处理来对待
26. Spark最新的SQL查询起点是 SparkSession，内部封装了SparkContext
27. SparkSession是Spark最新的SQL查询起点，实际上是 SparkContext和 HiveContext 的封装
28. Spark Streaming 用于流式数据的处理。Spark Streaming支持的数据输入源很多，例如 Kafka、HDFS、ZeroMQ、Flume、和简单的TCP套接字等

选择题

7. 与Spark Stream相似的流计算组件是（C）
   A.Flume B.Kafka C.Storm D.Hive

判断题

18. Spark SQL的DataFrame底层仍然基于RDD实现但是通过Catalyst优化器提升了执行效率（√）

六、Hadoop/Spark部署与架构

填空题

36. Spark有三种部署方式：Standalone、Spark on Mesos和 Spark on YARN

判断题

5. hadoop-env.sh文件提供了Hadoop中JAVA_HOME的运行环境（√）

七、PySpark编程实践

填空题

11. 有以下一段PySpark程序，请完成填空：

```
from functools import reduce

lst = [1, 2, 3, 4, 5]
fold_left = reduce(lambda x,y: x-y, lst, 0) # 从左到右遍历可迭代对象
fold_right = reduce(lambda x,y: y-x, reversed(lst), 0)  # 从右到左遍历可迭代对象
```

fold_left结果为 -15 fold_right结果为 3

19. 使用PySpark完成以下任务：在整数数组筛选出偶数并乘以二：

```
rdd = sc.parallelize([10, 2, 33, 24, 50, 62, 77, 81])
result = rdd.filter(lambda x: x % 2==0) # 筛选
            .map(lambda x: x * 2)   # 操作
            .collect()  # 执行操作
```

46. 设arr是一个一维数组，用foreach遍历该数组并打印每个元素的PySpark语句为：

```
sc.parallelize(arr).foreach(lambda x: print(x))
```

55. 执行PySpark代码：

```
t = [1, 2, 3, 5, 5]
result = reduce(lambda x,y: x-y,t)
print(result)
```

结果为：-14

八、云计算与大数据融合

填空题

31. 云计算为大数据处理提供了按需获取的 弹性（虚拟化）资源（如计算、存储、网络），使得企业无需自建昂贵的物理集群即可展开大规模数据分析。
32. 数据应用常采用云计算的 PaaS（平台即服务）服务模式，直接使用托管的Hadoop、Spark等平台，降低运维复杂度。
33. 云计算采用 按需计费的计费模式，使得大数据分析任务只需要为实际使用的计算时长和存储空间付费

总结

1. 核心知识点可分为8大类：HDFS、MapReduce、YARN、Spark核心、Spark SQL/Streaming、部署架构、PySpark编程、云计算融合，复习时可按类别逐个突破。
2. 高频考点集中在：HDFS的块/副本机制、MapReduce的Shuffle/Combiner、Spark的RDD/宽/窄依赖/DAG/持久化、YARN的资源管理模型。
3. 易错点需重点关注：RDD不可变、SecondaryNameNode非热备、collect()不适合大数据集、reduceByKey比groupByKey高效的原因。

编程题：

MapReduce编写

1.wordcount

2.Top10成绩排序

3.倒排索引

4.气温排序

PySpark RDD API编写

PySpark Dataframe API编写

词频统计

电影数据分析

PySpark SQL API编写

词频统计

电影数据分析