

1. 根据项目中的介绍，谈谈为什么要对数仓进行分层？

数仓进行分层的一个主要原因就是希望在管理数据的时候，能对数据有一个更加清晰的掌控。

主要有以下优点：

划清层次结构：每一个数据分层都有它的作用域，这样我们在使用表的时候能更方便地定位和理解。

数据血缘追踪：简单来讲可以这样理解，我们最终给下游是直接能使用的业务表，但是它的来源有很多，如果有一张来源表出问题了，我们希望能够快速准确地定位到问题，并清楚它的危害范围。

减少重复开发：规范数据分层，开发一些通用的中间层数据，能够减少极大的重复计算。

把复杂问题简单化。将一个复杂的任务分解成多个步骤来完成，每一层只处理单一的步骤，比较简单和容易理解。而且便于维护数据的准确性，当数据出现问题之后，可以不用修复所有的数据，只需要从有问题的步骤开始修复。

屏蔽原始数据的异常。屏蔽业务的影响，不必改一次业务就需要重新接入数据。

2. 那数据仓库都分哪几层？

如果划分细致，数据仓库总共可以划分为5层：

ODS 层

ODS 层: Operation Data Store, 数据准备区，贴源层。直接接入源数据的：业务库、埋点日志、消息队列等。ODS 层是数据仓库的准备区。

DW数仓

DWD 层：Data Warehouse Details,数据明细层，属于业务层和数据仓库层的隔离层，把持和 ODS 层相同颗粒度。进行数据清洗和规范化操作，去空值/脏数据、离群值等。

DWM 层：Data Warehouse middle,数据中间层，在 DWD 的基础上进行轻微的聚合操作，算出相应的统计指标。

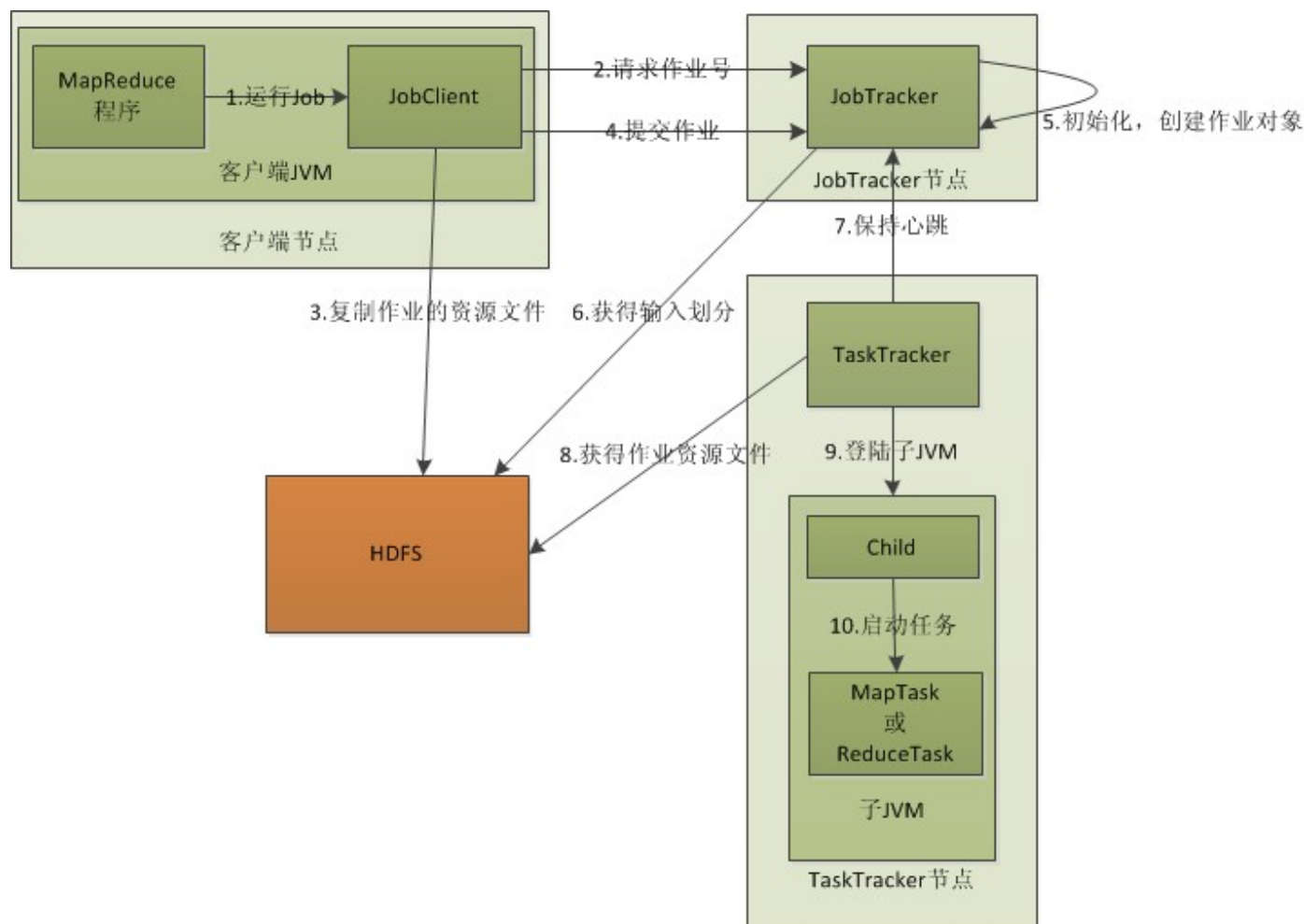
DWS 层：Data warehouse service,数据服务层，在 DWM 的基础上，整合汇总一个主题的数据服务层。汇总结果一般为宽表，用于 OLAP、数据分发等。

ADS层

ADS 层：Application data service, 数据应用层，存放在 ES,Redis、PostgreSQL 等系统中，供数据分析和挖掘使用。

在数仓中看到使用了MapReduce。

3. 那你介绍一下 Mapreduce 工作原理？



MapReduce 工作原理分为以下 5 个步骤：

在客户端启动一个作业。

向 JobTracker 请求一个 Job ID。

将运行作业所需要的资源文件复制到 HDFS 上，包括 MapReduce 程序打包的 JAR 文件、配置文件和客户端计算所得的输入划分信息。这些文件都存放在 JobTracker 专门为该作业创建的文件夹中。文件夹名为该作业的 Job ID。JAR 文件默认会有 10 个副本（mapred.submit.replication 属性控制）；输入划分信息告诉了 JobTracker 应该为这个作业启动多少个 map 任务等信息。

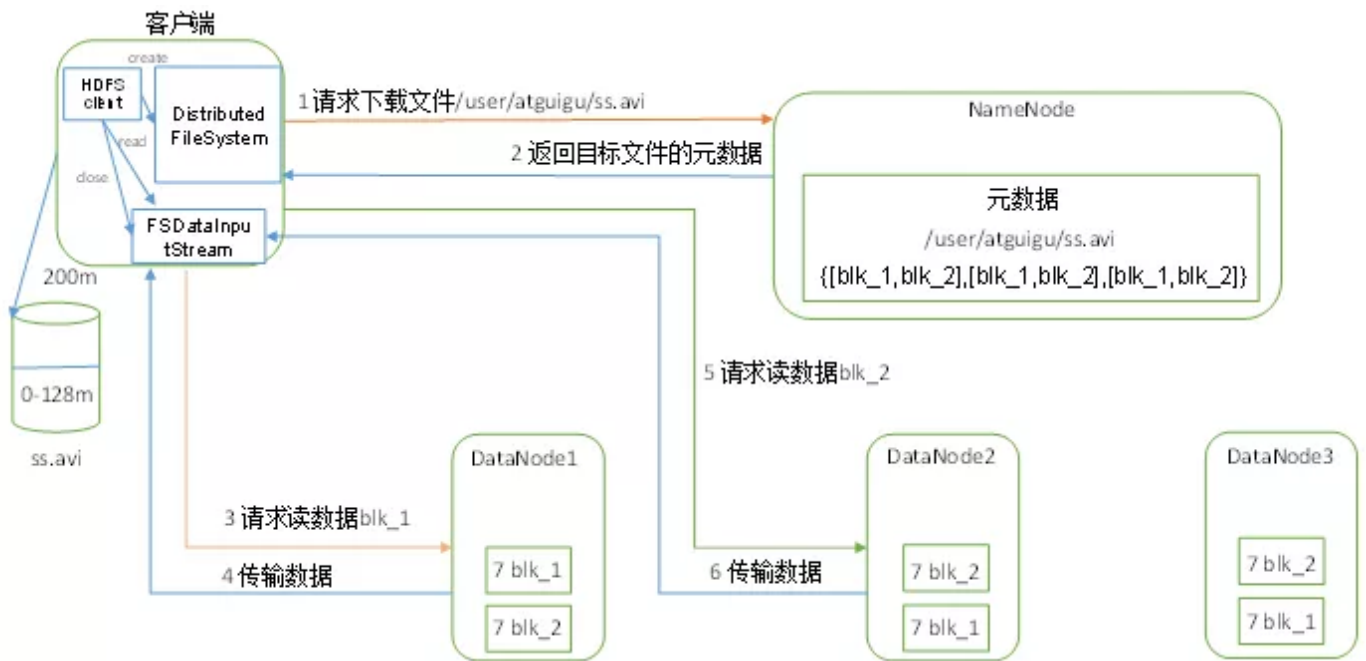
JobTracker 接收到作业后，将其放在一个作业队列里，等待作业调度器对其进行调度，当作业调度器根据自己的调度算法调度到该作业时，会根据输入划分信息为每个划分创建一个 map 任务，并将 map 任务分配给 TaskTracker 执行。对于 map 和 reduce 任务，TaskTracker 根据主机核的数量和内存的大小有固定数量的 map 槽和 reduce 槽。这里需要强调的是：map 任务不是随随便便地分配给某个 TaskTracker 的，这里有个概念叫：数据本地化（Data-Local）。意思是：将 map 任务分配给含有该 map 处理的数据块的 TaskTracker 上，同时将程序 JAR 包复制到该 TaskTracker 上来运行，这叫“运算移动，数据不移动”。而分配 reduce 任务时并不考虑数据本地化。

TaskTracker 每隔一段时间会给 JobTracker 发送一个心跳，告诉 JobTracker 它依然在运行，同时心跳中还携带着很多的信息，比如当前 map 任务完成的进度等信息。当 JobTracker 收到作业的最后一个任务完成信息时，便将该作业设置成“成功”。当 JobClient 查询状态时，它将得知任务已完成，便显示一条消息给用户。

以上是在客户端、JobTracker、TaskTracker的层次来分析MapReduce的工作原理。

4. 那 Hdfs 读数据流程你了解吗？

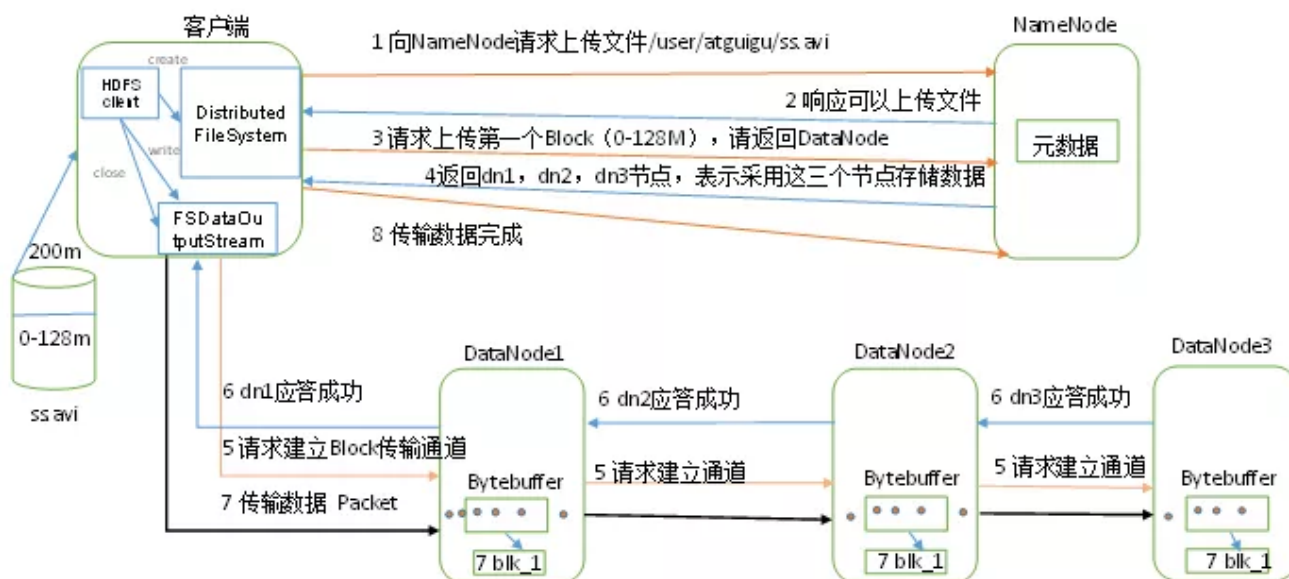
- (1) 客户端通过 Distributed FileSystem 向 NameNode 请求下载文件，NameNode 通过查询元数据，找到文件块所在的 DataNode 地址。
- (2) 挑选一台 DataNode（就近原则，然后随机）服务器，请求读取数据。
- (3) DataNode 开始传输数据给客户端（从磁盘里面读取数据输入流，以 Packet为单位来做校验）。
- (4) 客户端以 Packet 为单位接收，先在本地缓存，然后写入目标文件。



5. Hdfs 写数据流程也介绍一下

- (1) 客户端通过 Distributed FileSystem 模块向 NameNode 请求上传文件，NameNode 检查目标文件是否已存在，父目录是否存在。
- (2) NameNode 返回是否可以上传。
- (3) 客户端请求第一个 Block 上传到哪几个 DataNode 服务器上。
- (4) NameNode 返回 3 个 DataNode 节点，分别为 dn1、dn2、dn3。
- (5) 客户端通过 FSDataOutputStream 模块请求 dn1 上传数据，dn1 收到请求会继续调用 dn2，然后 dn2 调用 dn3，将这个通信管道建立完成。
- (6) dn1、dn2、dn3 逐级应答客户端。
- (7) 客户端开始往 dn1 上传第一个 Block（先从磁盘读取数据放到一个本地内存缓存），以 Packet 为单位，dn1 收到一个 Packet 就会传给 dn2，dn2 传给 dn3；dn1 每传一个 packet 会放入一个应答队列等待应答。

(8) 当一个 Block 传输完成之后，客户端再次请求 NameNode 上传第二个 Block 的服务器。（重复执行3-7步）。



6. Spark 为什么比 MapReduce 快?

Spark 是基于内存计算，MapReduce 是基于磁盘运算，所以速度快

Spark 拥有高效的调度算法，是基于 DAG,形成一系列的有向无环图

Spark 是通过 RDD 算子来运算的，它拥有两种操作，一种转换操作，一种动作操作，可以将先运算的结果存储在内存中，随后在计算出来

Spark 还拥有容错机制 Linage。

7. Spark 任务执行流程?

- SparkContext 向资源管理器注册并向资源管理器申请运行 Executor
- 资源管理器分配 Executor，然后资源管理器启动 Executor
- Executor 发送心跳至资源管理器
- SparkContext 构建 DAG 有向无环图
- 将 DAG 分解成 Stage (TaskSet)
- 把 Stage 发送给 TaskScheduler
- Executor 向 SparkContext 申请 Task
- TaskScheduler 将 Task 发送给 Executor 运行
- 同时 SparkContext 将应用程序代码发放给 Executor
- Task 在 Executor 上运行，运行完毕释放所有资源

8. Flink 的四大基石都有哪些？

Flink四大基石分别是：Checkpoint（检查点）、State（状态）、Time（时间）、Window（窗口）。

9. watermark 的作用有啥？如何保证数据不丢失？

WaterMark 的作用是用来触发窗口进行计算，解决数据延迟、数据乱序等问题。

水印就是一个时间戳（timestamp），Flink 可以给数据流添加水印。

- 水印并不会影响原有 Eventtime 事件时间。
- 当数据流添加水印后，会按照水印时间来触发窗口计算,也就是说 watermark 水印是用来触发窗口计算的。
- 设置水印时间，会比事件时间小几秒钟,表示最大允许数据延迟达到多久。
- 水印时间 = 事件时间 - 允许延迟时间（例如：10:09:57 = 10:10:00 - 3s）

要保证数据不丢失，需要使用：

WaterMark + EventTimeWindow + Allowed Lateness 方案（侧道输出），可以做到数据不丢失。

allowedLateness(lateness:Time)—设置允许延迟的时间，该方法传入一个 Time 值，设置允许数据迟到的时间。

10. 写一个算法

删除排序链表中的重复元素_II Leetcode83题

83. 删除排序链表中的重复元素

难度 简单

👤 706

☆ 收藏

🔗 分享

🌐 切换为英文

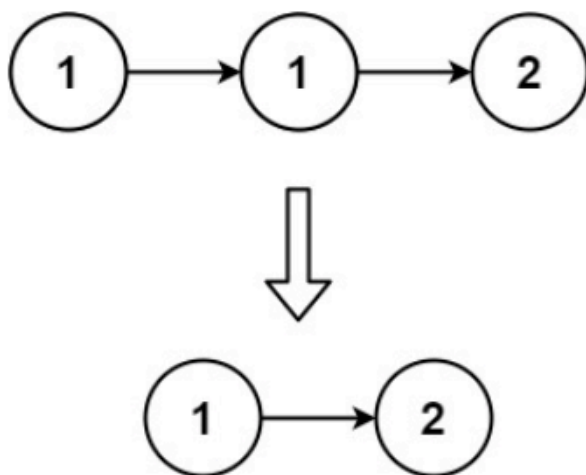
🔔 接收动态

💡 反馈

存在一个按升序排列的链表，给你这个链表的头节点 `head`，请你删除所有重复的元素，使每个元素 只出现一次。

返回同样按升序排列的结果链表。

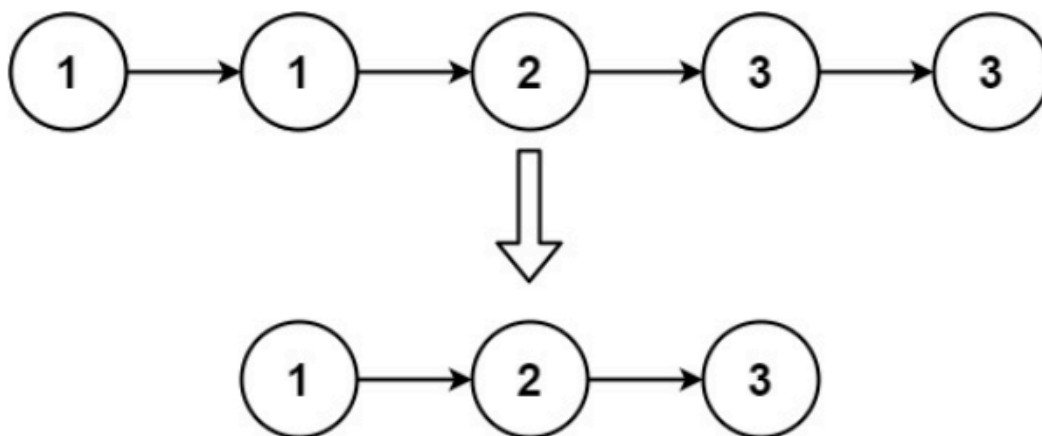
示例 1：



输入: `head = [1,1,2]`

输出: `[1,2]`

示例 2：



输入: `head = [1,1,2,3,3]`

输出: `[1,2,3]`

提示：

- 链表中节点数目在范围 `[0, 300]` 内
- `-100 <= Node.val <= 100`
- 题目数据保证链表已经按升序排列

注：资料来源于网络。