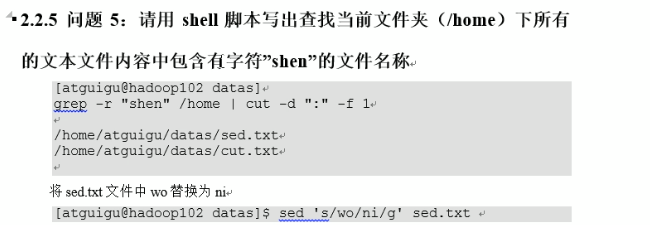
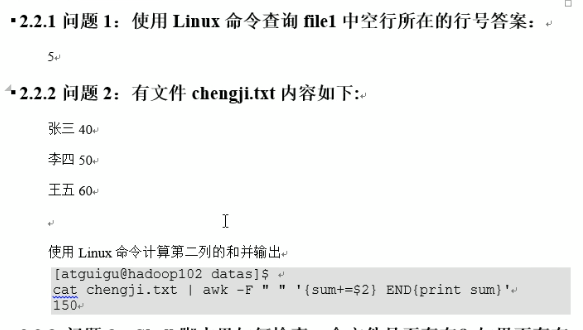
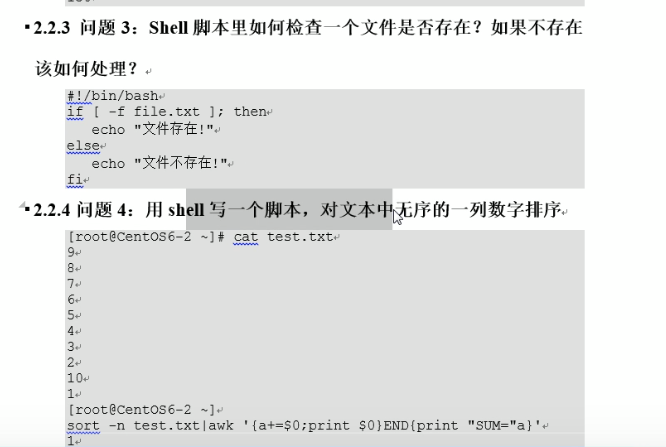
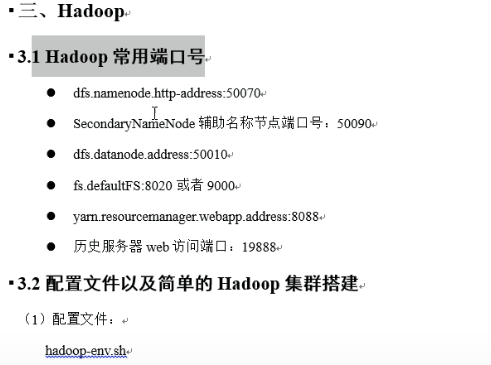
f









Hdfs读取文件流程：

启动DestributeFileSysterm 向NameNode请求下载文件，Namenode返回文件的元数据。

Hdfsclient启动FSDataInputStream向不同的DataNode发送请求Block块

，DataNode返回数据。

Hdfs写数据

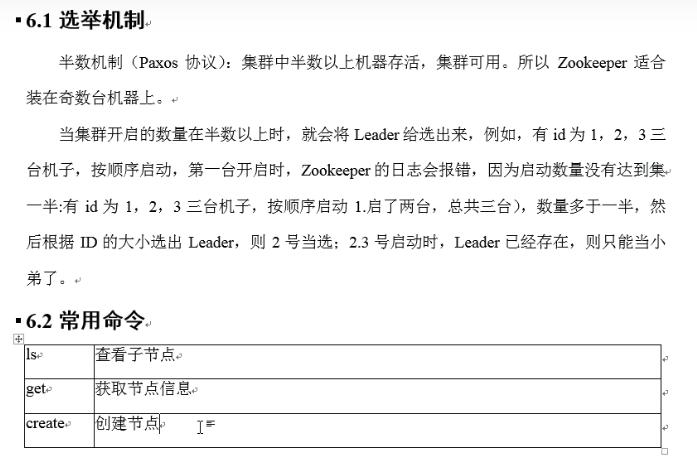
hdfsclient启动DestributeFileSysterm 向NameNode发送上传数据请求，Namenode返回可以上传，hdfsclient按次序上传128M的Block,并请求返回datanode，NM建立matestore并返回dn1，dn2，dn3，存储在哪些节点，Hdfsclient启动FSDataInputStream请求对应的DN建立连接，dn1，dn2，dn3相应连接，hdfsclietn开始向不同的dns传输数据，dn在各自的基点上启动Bytebuffer，写入128M的Block，写完之后，HdfsClient向NN发送写入完毕。

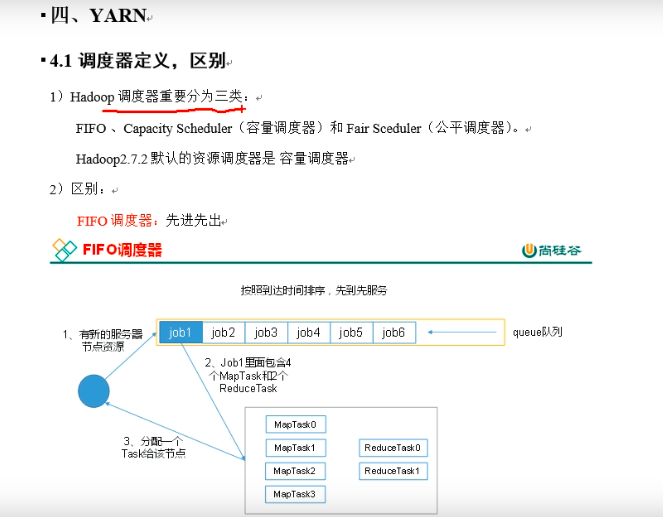
MapReduce详细流程

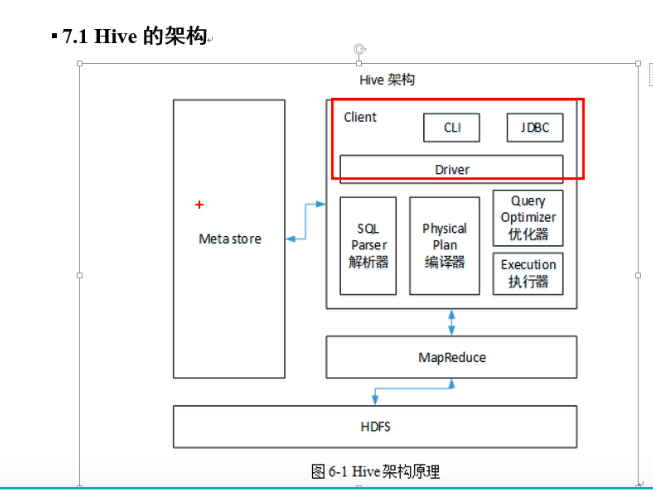
Map端根据hdfs的文件数，启动对应的partition拉取数据，建立（k(partition索引)，v(数据)），写入到还原缓冲区，缓冲区对数据进行按照key的字典快速排序，**环形缓冲区**，大小默认100M，默认到80%就会spill到磁盘，spill之前会做本地的merge的归并操作。Reduce默认时5个，拉取文件并根据key做归并排序也就是shuffle默认优先归并到内存，内存不够写入磁盘，然后进行计算写入磁盘

优化：

1. **环形缓冲区**：100->200M
2. 80%->90% spill率
3. Map端merge文件可以修改个数，一次merge多少个；combiner启动汇总
4. Map和reduce的memory ，默认1G->4-5G，CPU个数增加和mr个数
5. 小文件启用JVM重用
6. Yarn调度，调大container的内存和cpu核数
7. 传入数据启动Lzo，gzib2，切片，传输中启动snappy压缩，落入磁盘启动压缩gzib
8. 采用ConbinInputFormart作为输入，可以将切片的大量小文件数据进行合并

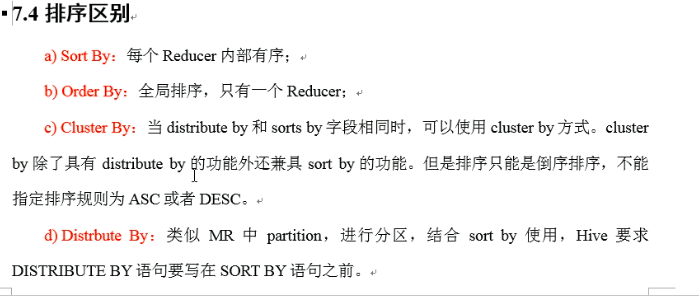




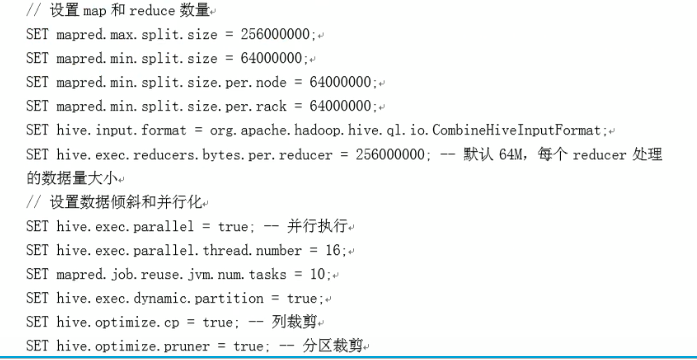
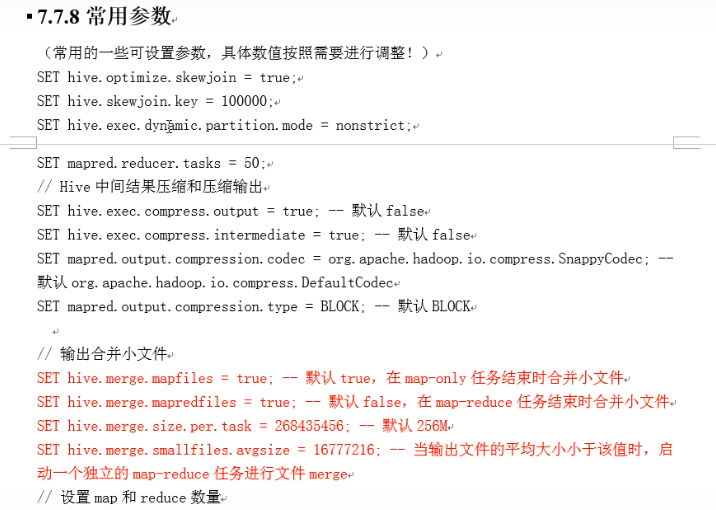
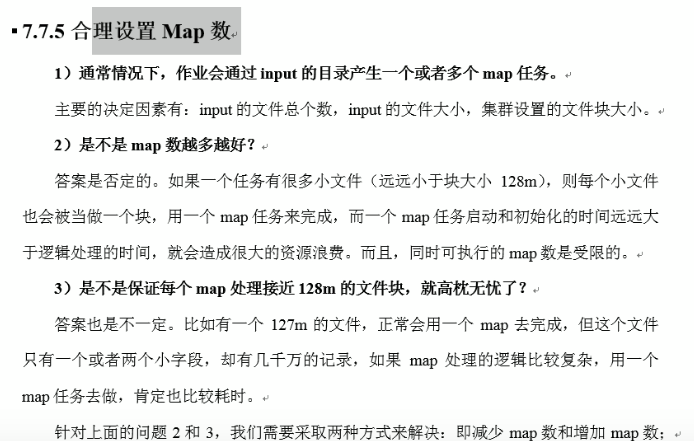


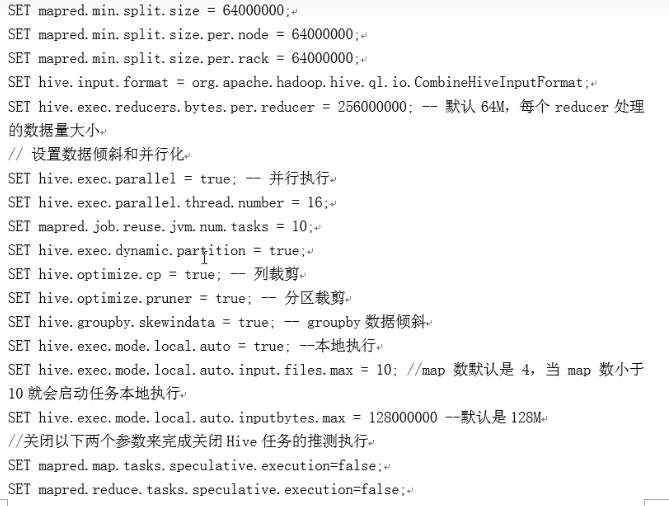
Hive于mysql的区别：



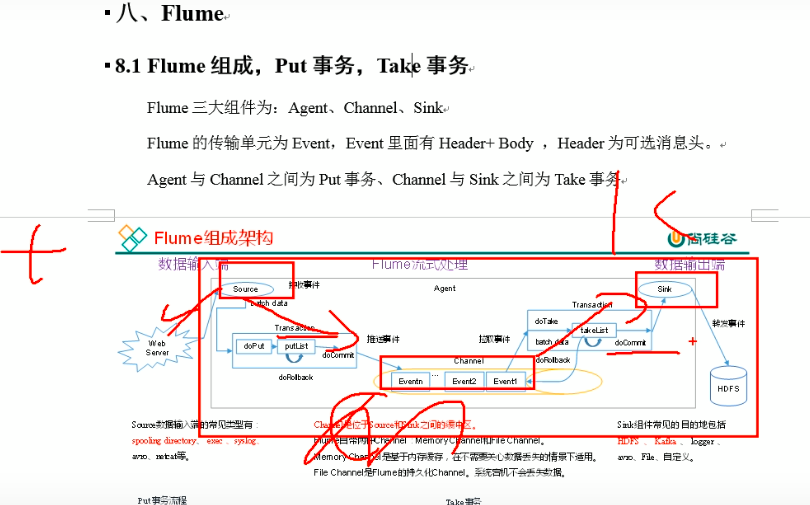


Hive优化:





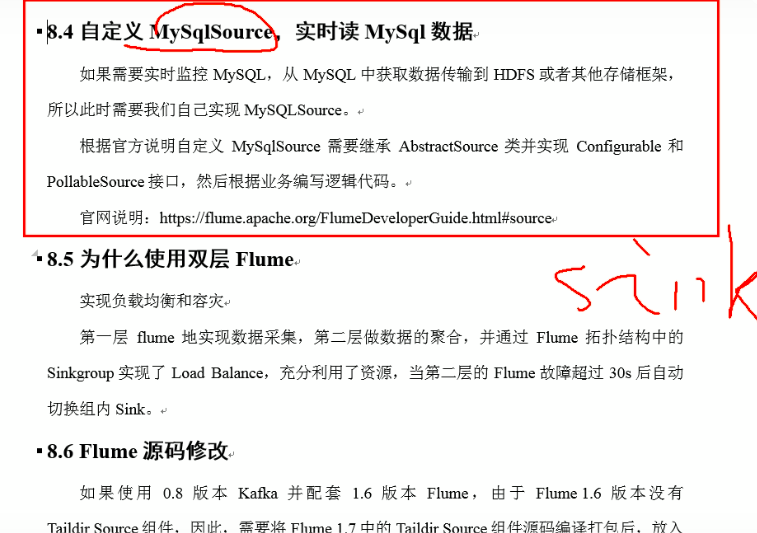
Flume知识点



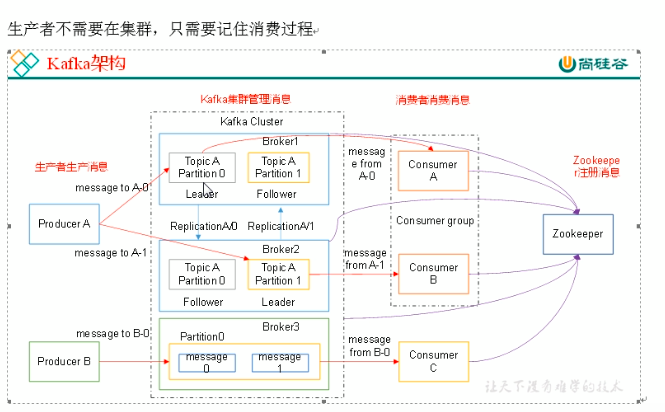
Source:Tail-di

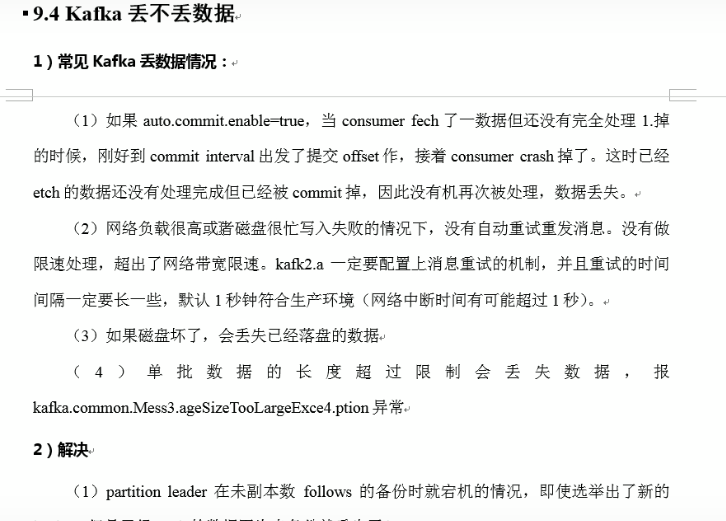
Chnnle:Memory

Sink:kafka+hdfs



Kafka知识点：



如何

如何解决：

Ack参数

acks = 0 如果设置为0，则生产者将不会等待来自服务器的任何确认，该记录将立即添加到套接字缓冲区并视为已发送。在这种情况下，无法保证服务器已收到记录，并且重试配置将不会生效（因为客户端通常不会知道任何故障），为每条记录返回的偏移量始终设置为-1。

#acks = 1 这意味着leader会将记录写入其本地日志，但无需等待所有副本服务器的完全确认即可做出回应，在这种情况下，如果leader在确认记录后立即失败，但在将数据复制到所有的副本服务器之前，则记录将会丢失。

#acks = all 这意味着leader将等待完整的同步副本集以确认记录，这保证了只要至少一个同步副本服务器仍然存活，记录就不会丢失，这是最强有力的保证，这相当于acks = -1的设置。

**为什么 Kafka 性能高？**

**顺序写磁盘**

顺序写磁盘的性能是随机写入的性能的6000倍的提升，媲美内存随机访问的性能，磁盘不再是瓶颈点。

**Page Cache**

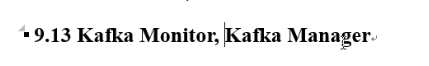
为了优化读写性能，Kafka利用了操作系统本身的Page Cache，就是利用操作系统自身的内存而不是JVM空间内存。通过操作系统的Page Cache，Kafka的读写操作基本上是基于内存的，读写速度得到了极大的提升。

**零拷贝技术**

零拷贝技术，可以有效的减少上下文切换和拷贝次数。



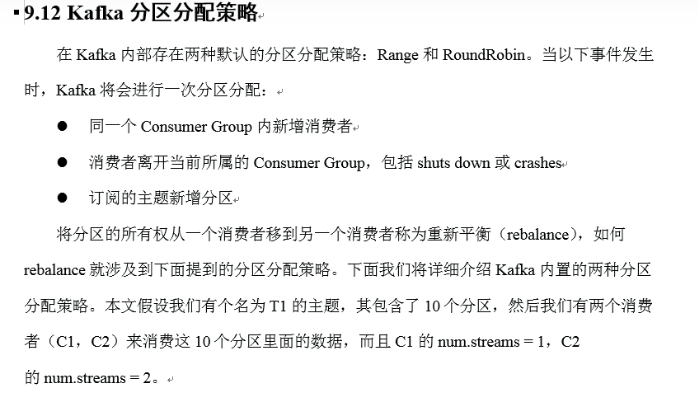
1. flume中还有没消费的信息已经进入缓存需要继续消费，可以等待kafka启动之后继续跑。
2. kafka可以设置保存30天的日志记录，能读取到kafka读到了哪里，然后kafka重启以后可以根据日志的向kafka重新灌数据
3. 如果时短期内挂掉的直接重启就可以

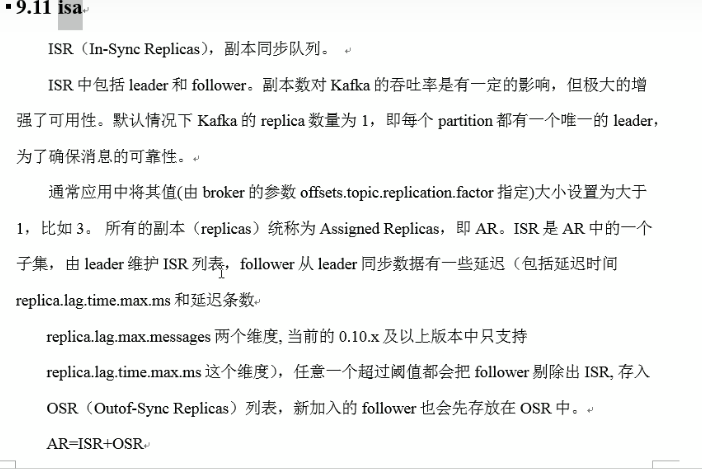
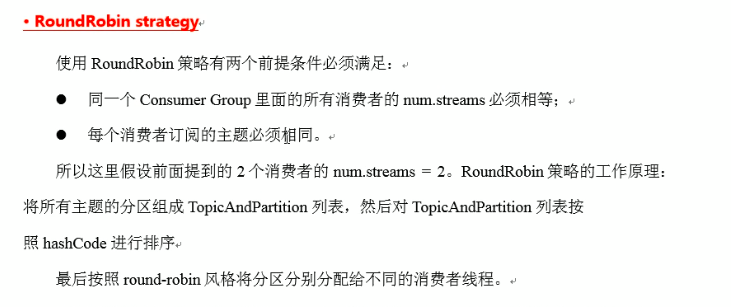
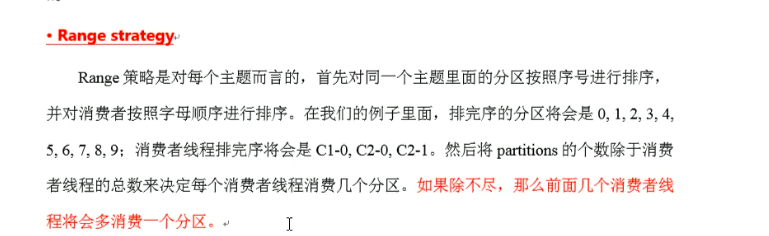


KafkaOffsetMonitor是Kafka的一款客户端消费监控工具，用来实时监控Kafka服务的Consumer以及它们所在的Partition中的Offset，我们可以浏览当前的消费者组，并且每个Topic的所有Partition的消费情况都可以一目了然。

为了简化开发者和服务工程师维护Kafka集群的工作，yahoo构建了一个叫做Kafka管理器的基于Web工具，叫做 Kafka Manager。这个管理工具可以很容易地发现分布在集群中的哪些topic分布不均匀，或者是分区在整个集群分布不均匀的的情况。它支持管理多个集群、选择副本、副本重新分配以及创建Topic。同时，这个管理工具也是一个非常好的可以快速浏览这个集群的工具，有如下功能：

1.管理多个kafka集群  
2.便捷的检查kafka集群状态(topics,brokers,备份分布情况,分区分布情况)  
3.选择你要运行的副本  
4.基于当前分区状况进行  
5.可以选择topic配置并创建topic(0.8.1.1和0.8.2的配置不同)  
6.删除topic(只支持0.8.2以上的版本并且要在broker配置中设置delete.topic.enable=true)  
7.Topic list会指明哪些topic被删除（在0.8.2以上版本适用）  
8.为已存在的topic增加分区  
9.为已存在的topic更新配置  
10.在多个topic上批量重分区  
11.在多个topic上批量重分区(可选partition broker位置)





**Hbase的优化：**

