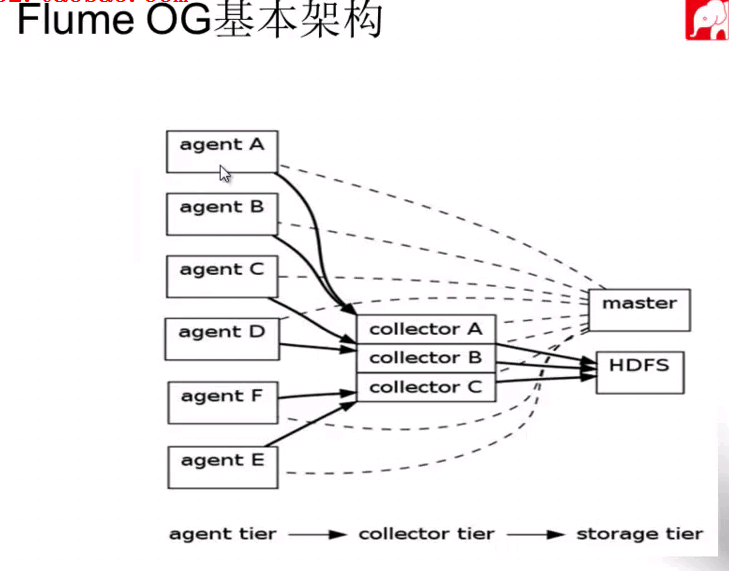


常用的两种数据来源：

1. 机器产生的数据源：
   1. 机器产生的数据
   2. 用户访问的日志
   3. 用户购买的日志
2. 传统系统中的数据
   1. 传统关系数据库：MYSQL、Oracle
   2. 磁盘阵列
   3. 磁带
3. Hadoop手机和入库基本要求
   1. 分布式
      1. 数据源多样化
      2. 数据源分散
   2. 可靠性
      1. 保证不丢失数据
      2. 允许丢失部分数据
   3. 可扩展性
      1. 数据源可能不断增加
   4. 通过并行提高性能
4. 常见的Hadoop收集与入库系统
   1. 数据收集
      1. Flume
      2. Kafka
      3. Scribe
   2. 传统数据库与Hadoop同步
      1. Sqoop
5. Hadoop数据收集系统-Flume
   1. Flume OG
      1. OG：Original Generation
      2. 0.9.x或者CDH3以及更早版本
      3. 由agent、collector、master等组件组成
   2. Flume NG
      1. NG：Next/New Generation
      2. 1.x或者CDH4以及之后的版本
      3. 由agent、client等组件组成
   3. 为什么要推出NG版本
      1. 精简代码
      2. 架构简化
6. Flume OG基本架构



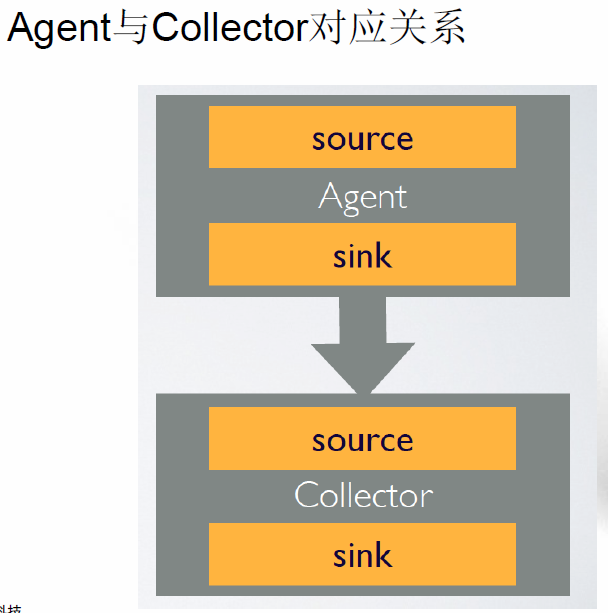
各个组件：

1. Master：
   1. Master负责配置及通信管理，是集群的控制器。
   2. Collector用于对数据进行聚合（数据收集器），往往产生一个更大的数据流，然后加载到storage（存储）中。
   3. Agent用于采集数据，Agent是Flume中产生数据流的地方，同时Agent将会产生的数据流传输到Collector
2. Agent
   1. 用于采集数据
   2. 数据流产生的地方
   3. 通常由source和sink两部分组成
      1. Source用于获取数据，可从文本文件，syslog,HTTP等获取数据；
      2. Sink将Source获得的数据进一步传输给后面的Collector
   4. Flume自带了很多Source和Sink实现
      1. SyslogTCP（5140）|agentSink（”localhost”,35853）
      2. Tail(“/etc/services”)|agentSink(“localhost”,35853)
3. Collector
   1. 汇总多个Agent结果
   2. 将汇总结果导入后端存储系统，比如HDFS，HBase
   3. Flume自带了很多collector实现

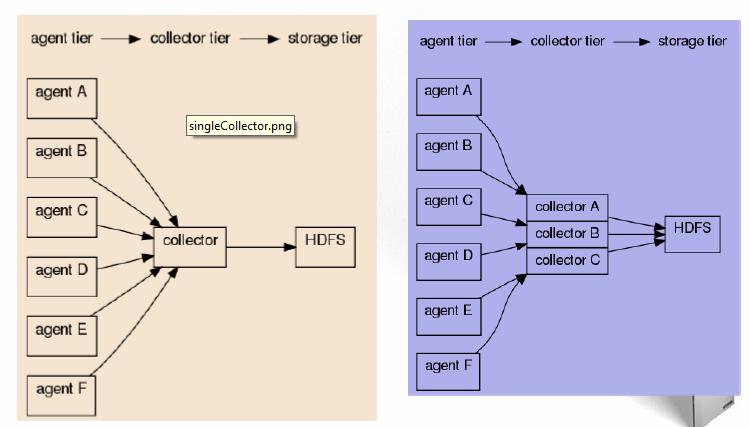
collectorSource(35853) | console

CollectorSource(35853) | collectorSink("file:///tmp/flume/collected", "syslog");

collectorSource(35853) | collectorSink("hdfs://namenode/user/flume/ ","syslog");



1. 可手动指定，也可自动匹配。
2. 自动匹配的情况下，master会平衡collector之间的负载。



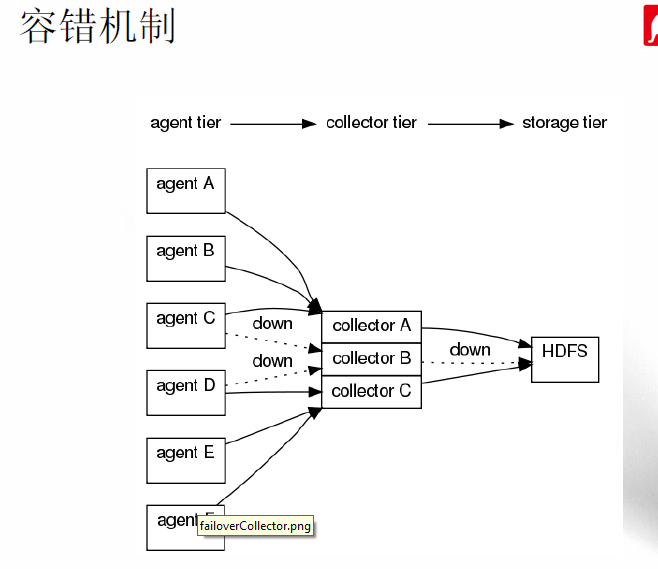
为什么引入collector？

1. 对agent数据进行汇总，避免产生过多小文件；
2. 避免多个agent连接对hadoop造成过大压力；
3. 中间件，屏蔽agent和hadoop间的异构性。

Master

1. 管理协调agent和collector的配置信息；
2. Flume集群的控制器；
3. 跟踪数据流的最后确认信息，并通知agent
4. 通常需要配置多个master以防单点故障；
5. 借助Zookeeper管理多个Master。

容错：



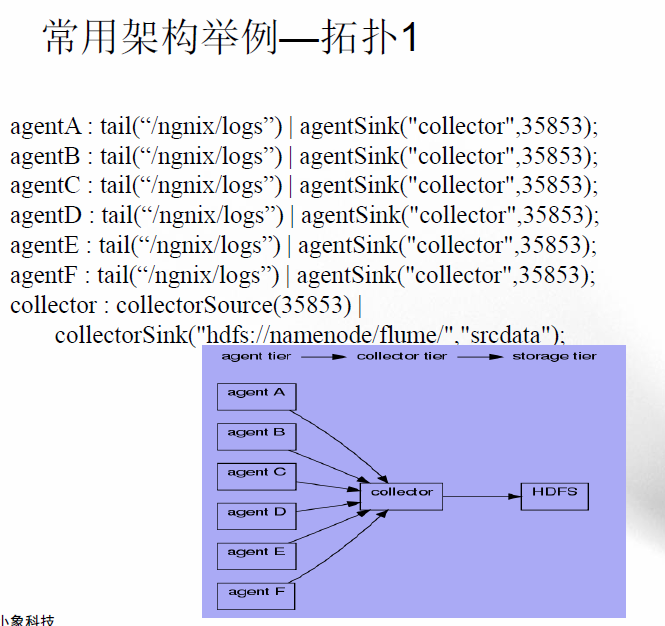
Agent优先写入到主的collector

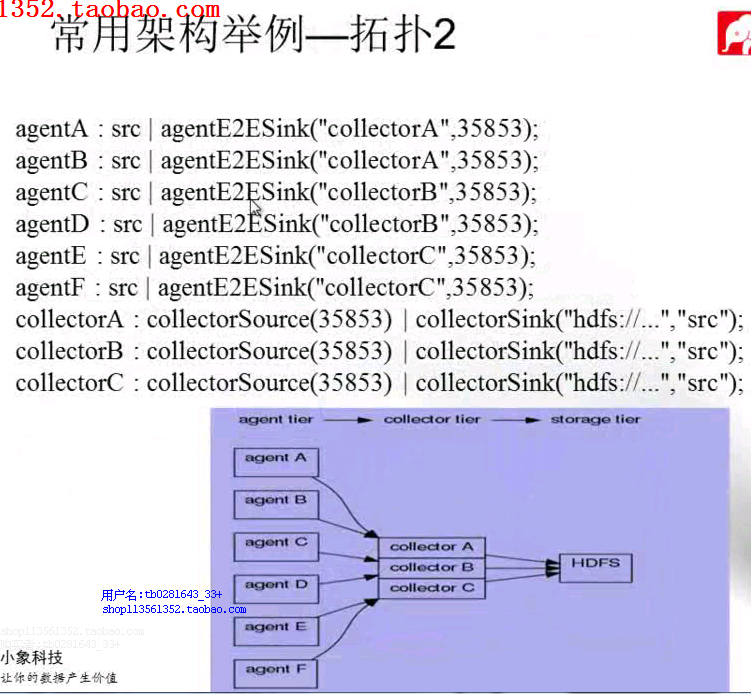
三种可靠性级别：

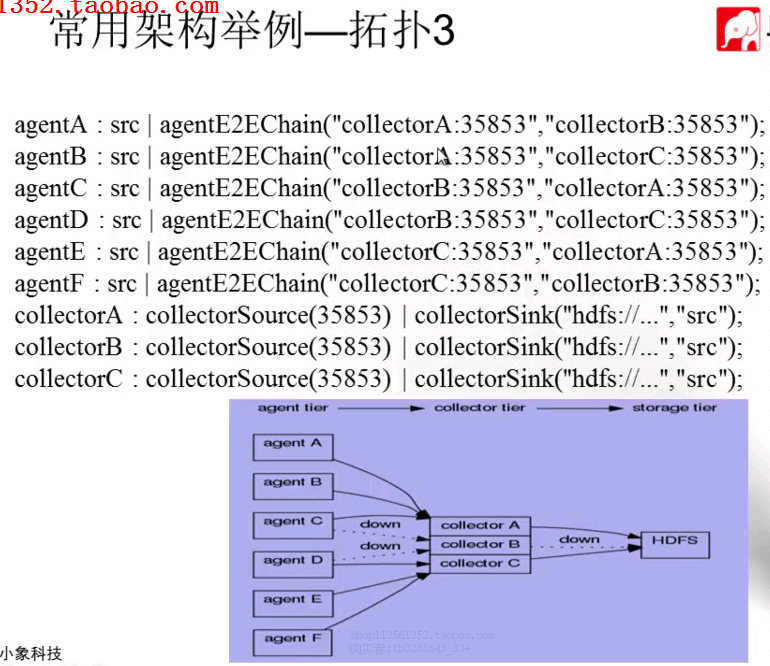
1. agentE2ESink[("machine"[,port])]
   1. agent收到确认消息才认为数据发送成功，否则重试。
2. agentDFOSink[("machine"[,port])]
   1. 当agent发现collector操作失败的时候，agent写入到本地磁盘上，当collector恢复后，再重新发送数据。用的最多
3. agentBESink[("machine"[,port])]
   1. 效率最好，agent不写入本地任何数据，如果在collector发现处理失败，直接删除消息。通常用于测试。

构建基于Collector均可以动态配置：

1. Agent和Collector均可以动态配置
2. 可通过命令行或者Web界面配置
3. 命令行配置
   1. 在已经启动的master节点上，依次输入”flume shell””connect localhost ” 如执行 exec config a1 ‘tailDir(“/data/logfile”)’ ‘agentSink’
4. Web界面
   1. 选中界面，填写source、sink等信息。







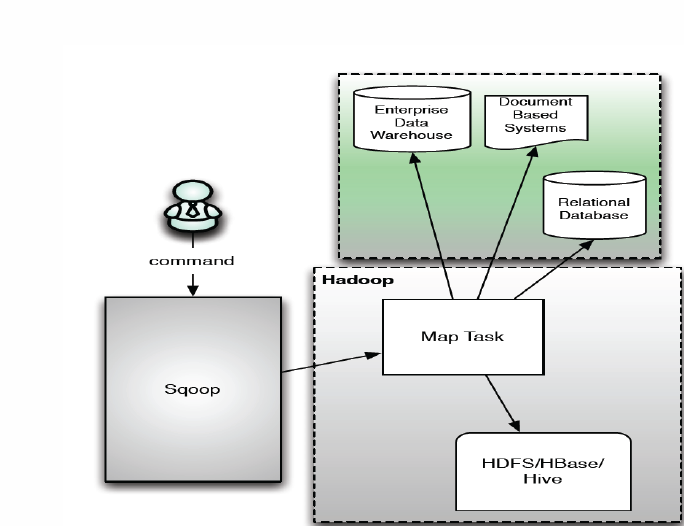
可靠、容错

Sqoop是什么：（吞吐率和可靠性）

1. Sqoop：SQL-to-Hadoop
2. 连接传统数据库和Hadoop的桥梁
   1. 把关系型数据库的数据导入到Hadoop系统（如HDFS、Hbase和Hive）中。
   2. 把数据从hadoop系统里抽取并导入到关系型数据库里面。
3. 利用MapReduce加快数据传输速度。
4. 批处理方式进行数据传输。

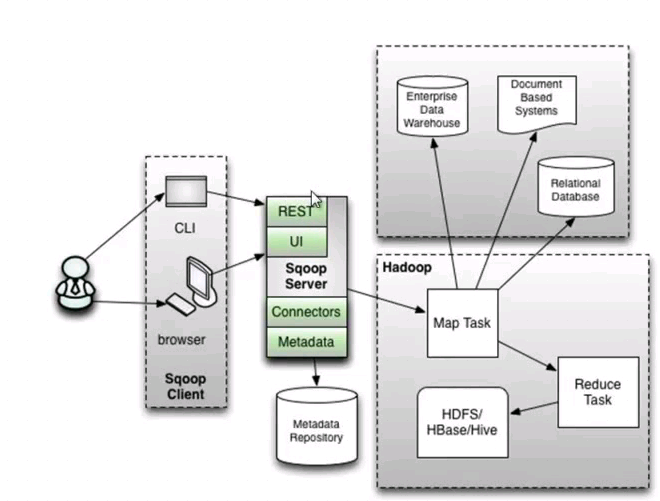
Sqoop优势：

1. 高效、可控地利用资源
   1. 任务并行度、超时时间等。
2. 数据类型映射与转换
   1. 可自动进行、用户也可自定义
3. 支持多种数据库
   1. MySQL
   2. Oracle
   3. PostgreSQL
4. Sqoop1架构：



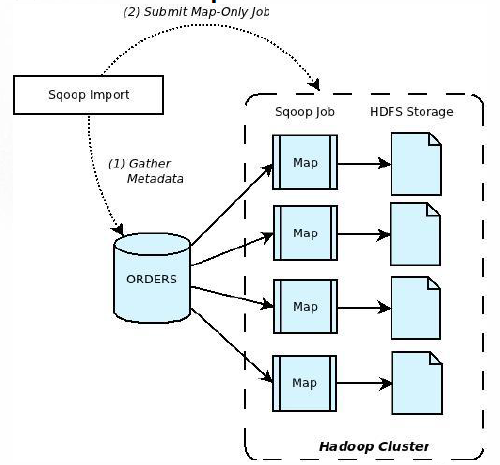
只有Map作业

Sqoop2架构：



Sqoop Import：

1. 将数据从关系型数据库导入Hadoop中
   1. Sqoop与数据库Server通信，获取数据库表的元数据信息；
   2. Sqoop启动一个Map Only作业，利用元数据信息行将数据写入hadoop。



Sqoop import使用：









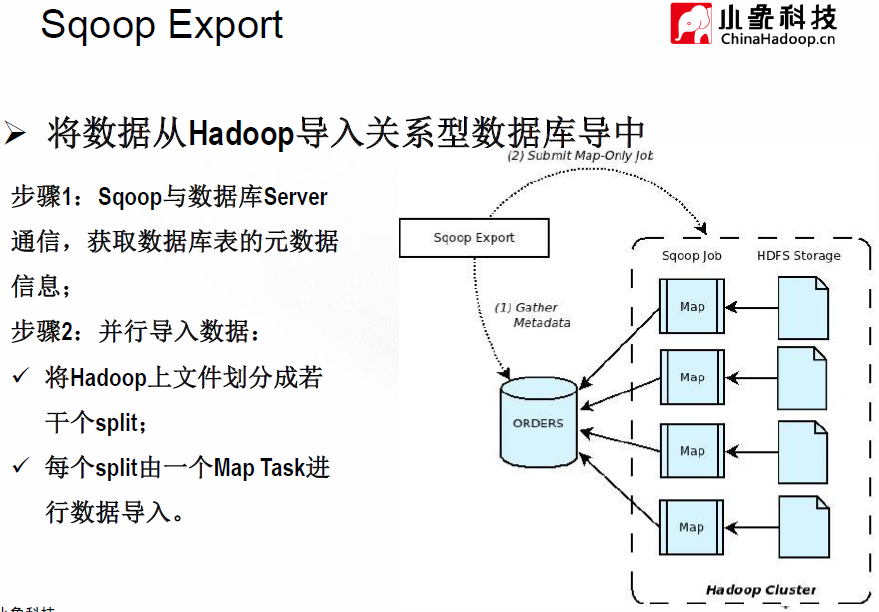




自动记录上次的ID



Sqoop Export：









允许覆盖的增量导入。



