好程序员
从平凡到卓越，为梦想而拼搏
GoodProgrammer.org

课程目标

COURSE CONTENTS

01 Flume体系结构

02 Flume OG架构

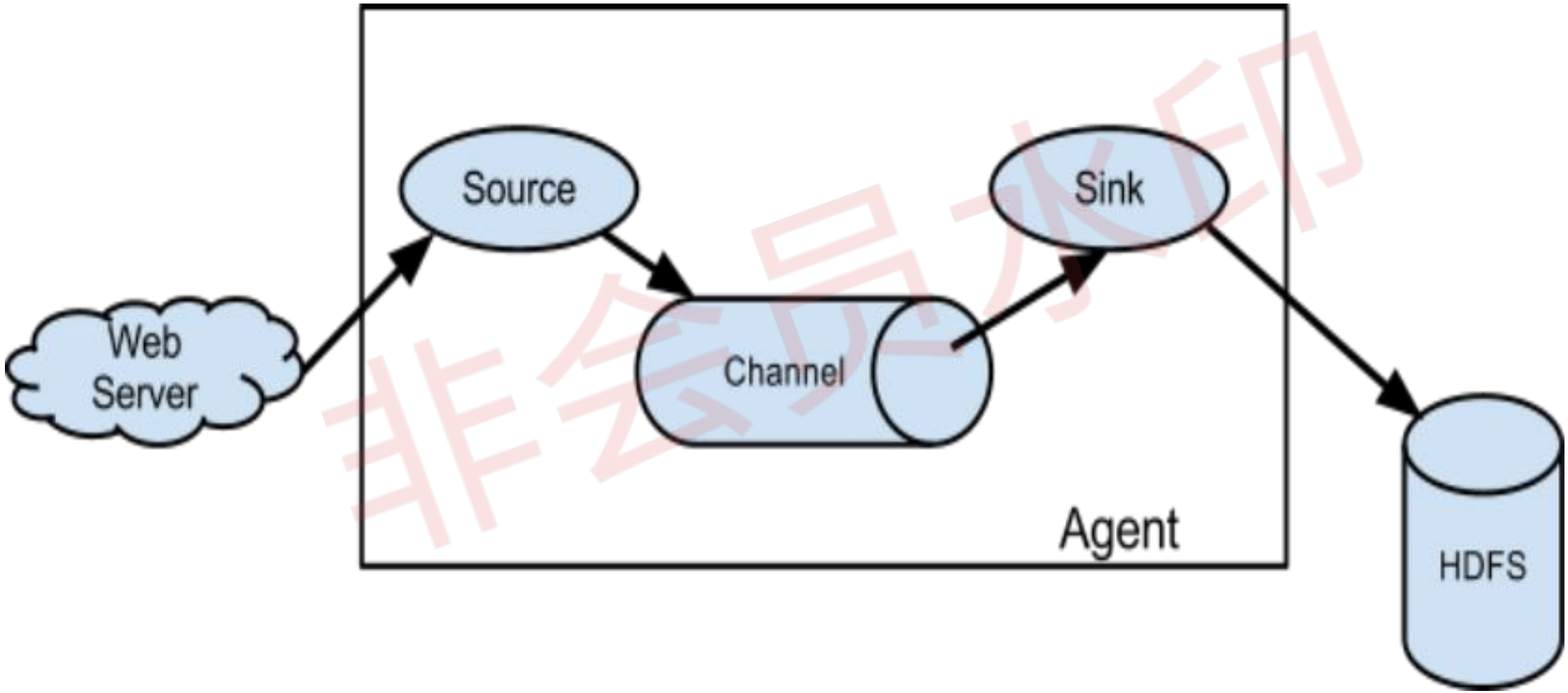
03 Flume NG架构

04 Flume之Agent

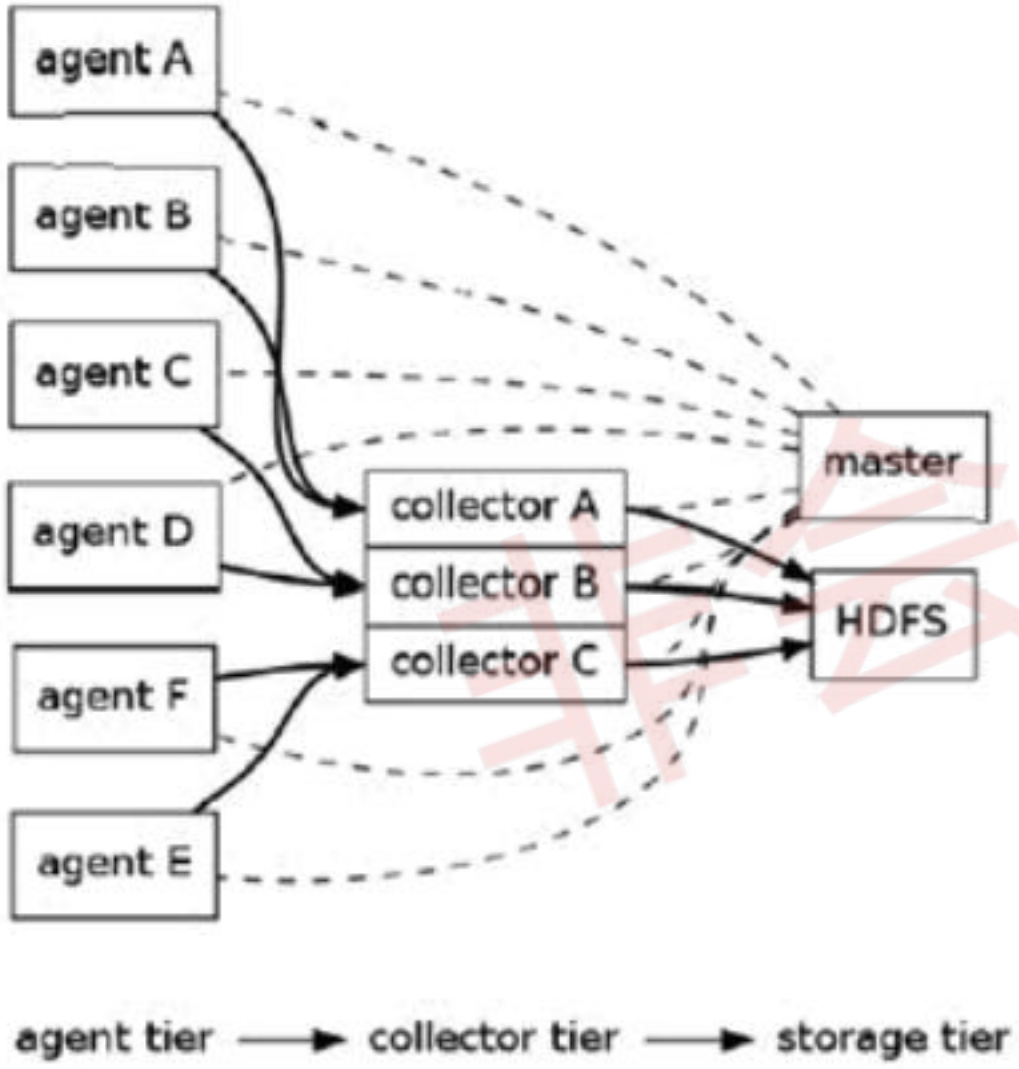
05 Flume组件之Sink

06 Flume相关面试题汇总

Flume体系结构



Flume OG架构



Flume逻辑上分三层架构：Agent，Collector，Storage。

Flume OG采用了多Master的方式。为了保证配置数据的一致性，Flume引入了ZooKeeper，用于保存配置数据，ZooKeeper本身可保证配置数据的一致性和高可用，另外，在配置数据发生变化时，ZooKeeper可以通知Flume Master节点。Flume Master间使用gossip协议同步数据。（ Gossip协议是一个通信协议,一种传播消息的方式）

FLUM OG 的特点是：

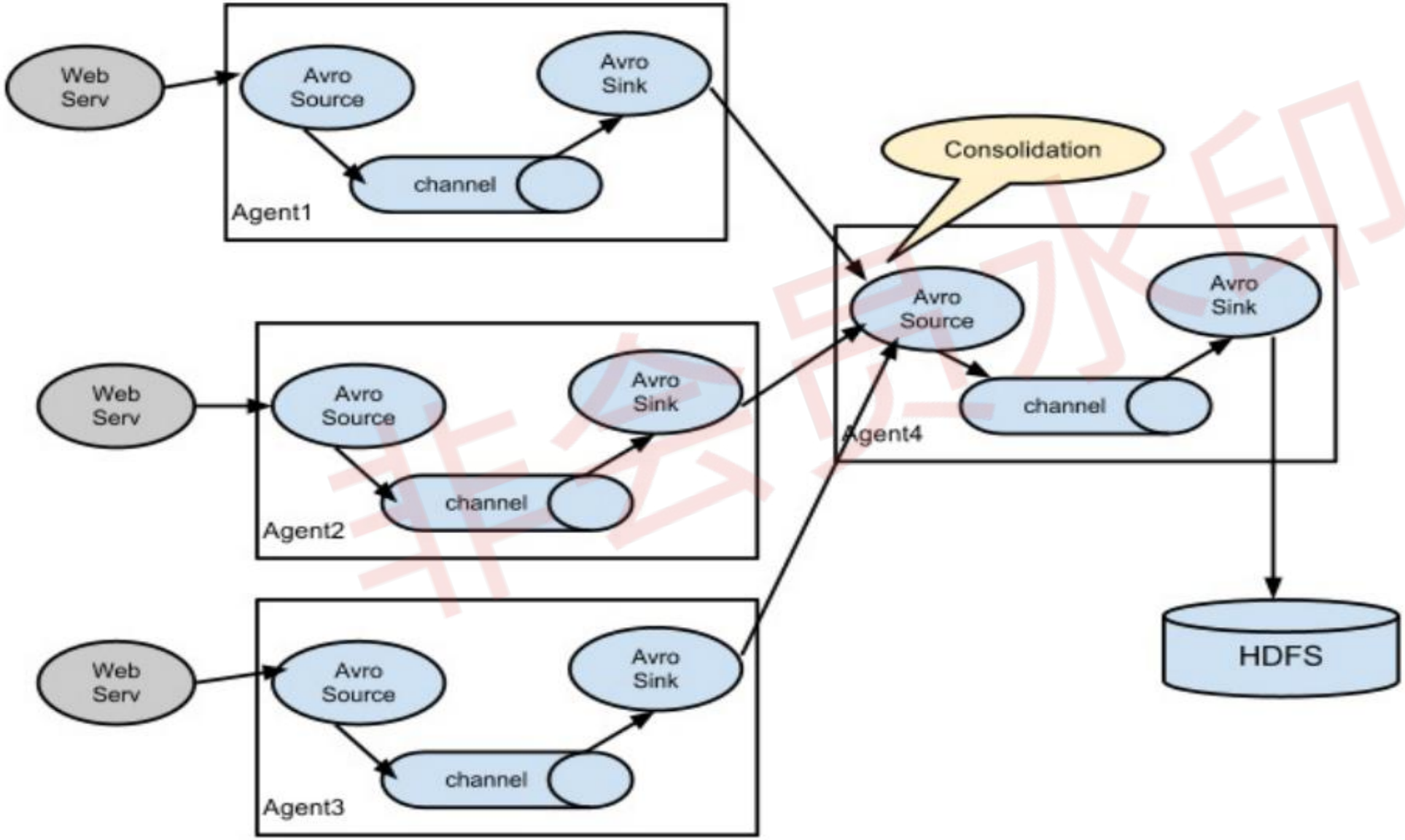
→FLUM OG 有三种角色的节点：代理节点（agent）、收集节点（collector）、主节点（master）。

→agent 从各个数据源收集日志数据，将收集到的数据集中到 Collector，然后由收集节点汇总存入 HDFS。master 负责管理 agent，collector 的活动。

→agent、collector 都称为 node，node 的角色根据配置的不同分为 logical node（逻辑节点）、physical node（物理节点）。

agent、collector 由 source、sink 组成，代表在当前节点数据是从 source 传送到 sink。

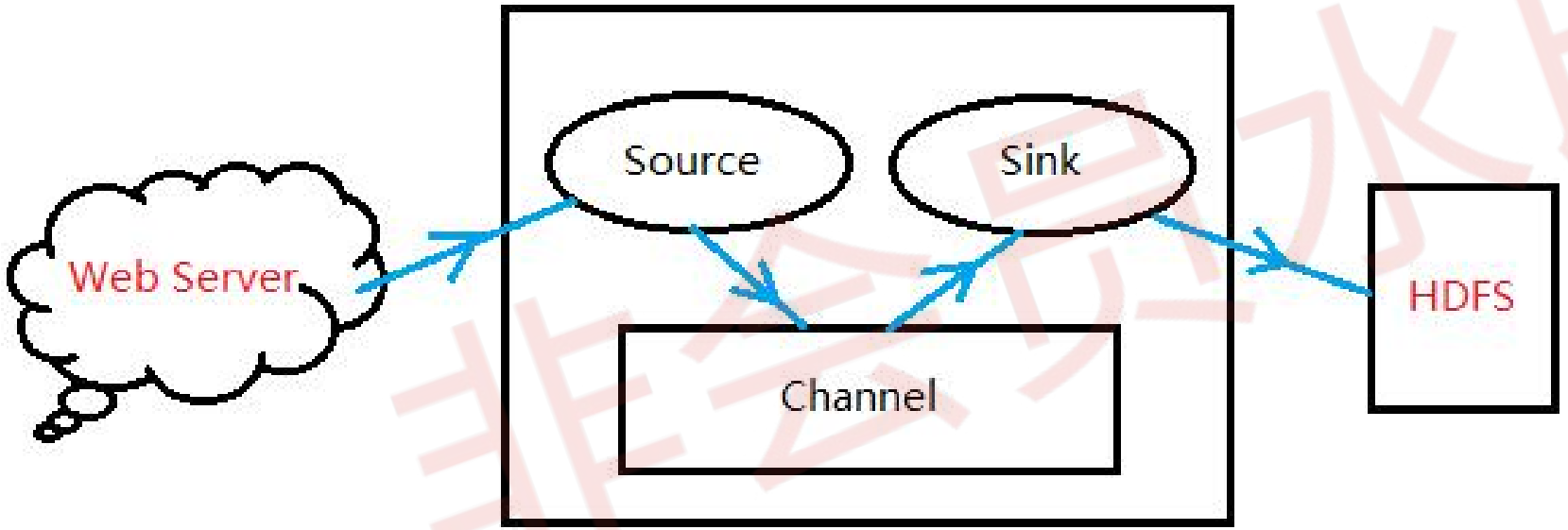
Flume NG架构


Web Serv
Avro Source
Avro Sink
channel
Agent1
Web Serv
Avro Source
Avro Sink
channel
Agent2
Web Serv
Avro Source
Avro Sink
channel
Agent3
Consolidation
Avro Source
Avro Sink
channel
Agent4
HDFS

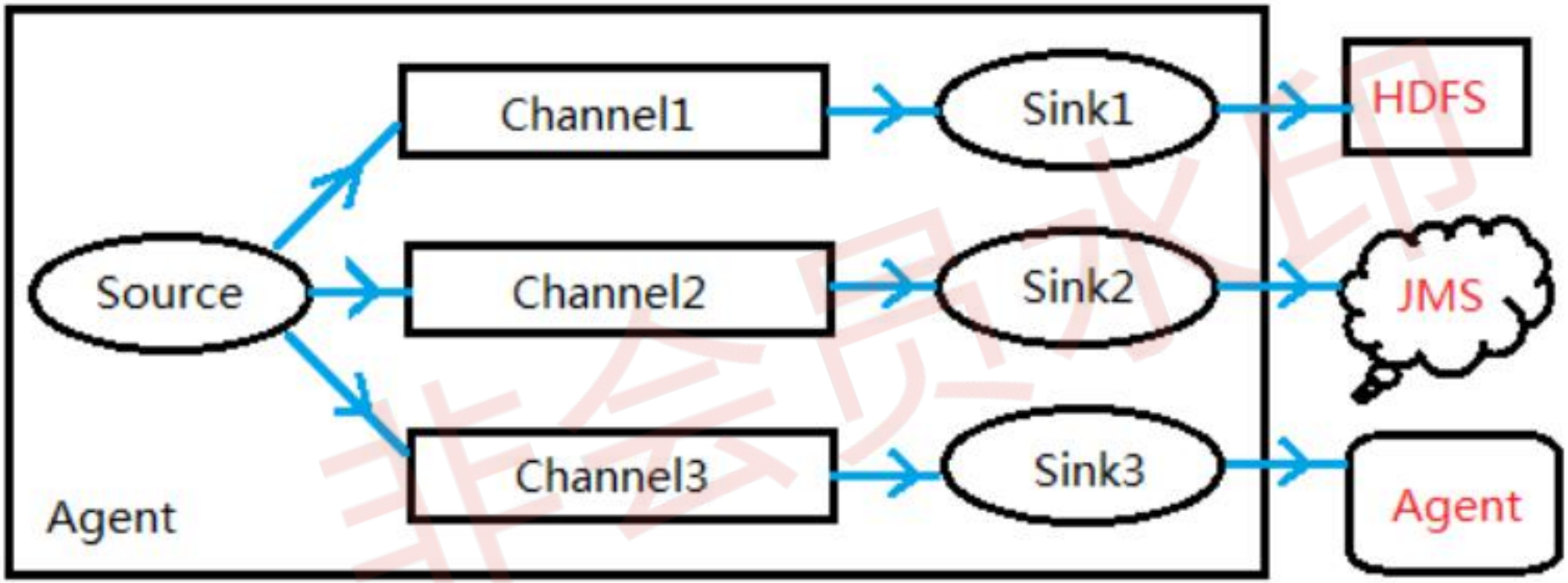
Flume之Agent

Flume以Agent为最小的独立运行单位。Agent是Flume中产生数据流的地方，一个Agent就是一个JVM。

单Agent由Source、Sink和Channel三大组件构成，如下图：



Flume组件之Sink



Flume相关面试题汇总

1、请给Flume下一个定义。

一个高可用的，高可靠的，分布式的海量日志采集、聚合和传输的系统

2、请简要阐述Flume的结构。

→ Source

默认的有Avro(监视端口)、Thrift、Exec(执行linux命令)、JMS、Spooling Directory(监视目录)、TailDirSource(1.7新增类似tail功能，支持断点续传)，第三方插件有kafka

→ 拦截器

所有events,增加头,类似json格式里的"headers":{" key":" value"}

时间戳（头部插入时间戳）、主机（头部插入主机名和IP）、静态（头部插入指定KV）、正则过滤（留下符合条件的）、自定义

→ Channel

Memory、JDBC、Kafka、File、Custom

→ 拦截器

→ Sink

HDFS、Hive、Logger、Avro、File Roll Sink(本地文件存储)、HBase、ElasticSearch、Kafka

Flume相关面试题汇总

3、请阐述Flume HA机制。

→ 负载均衡

flume NG通过设置sinkgroups将多个沉潜节点分到一组中，然后设置该组启用负载均衡，沉潜时会自动轮流选择节点，如果节点宕机则选择其它节点。

→ 事务机制

flume基于事务传输event（批量传输），使用两个独立的事务分别处理source到channel和channel到sink，失败时会将所有数据都回滚到source或channel进行重试。

该事务机制遵循'最少一次'语义，因此数据绝不会丢失，但有可能重复。

source-channel之间的重复可以靠TailDirSource自带的断点续传功能解决;

channel-sink之间的重复，可以延长等待时间，或者设置UUID拦截器，然后在redis里维护一个布隆表来使下游实时应用去重。（UUID拦截器:在Event Header中添加全局唯一UUID；自定义的sink：public class RedisSink extends AbstractSink implements Configurable，）

Flume相关面试题汇总

4、请举例说明Flume配置多个Sink。

#配置源->两个通道

myagent.sources = r1

myagent.sources.r1.channels = c1 c2

#分别配置c1->k1;c2->k2

myagent.sinks = k1 k2

myagent.channels = c1 c2

myagent.sinks.k1.channel = c1

myagent.sinks.k2.channel = c2

同理也可以配置多个source

5、请介绍Flume 的 Channel 。

Channel 被设计为 Event 中转临时缓冲区，存储 Source 收集并且没有被 Sink 读取的 Event，为平衡 Source 收集和 Sink 读取的速度，可视为 Flume 内部的消息队列。

Channel 线程安全并且具有事务性，支持 Source 写失败写，和 Sink 读失败重复读的操作。常见的类型包括 Memory Channel，File Channel，Kafka Channel。

6、介绍一下Flume的 Memory Channel。

读写速度快，但是存储数据量小，Flume 进程挂掉、服务器停机或者重启都会导致数据丢失。资源充足、不关心数据丢失的场景下可以用。

7、说说你对Flume中的File Channel的认知。

将 event 写入磁盘文件，与 Memory Channel 相比存储容量大，无数据丢失风险。File Channel 数据存储路径可以配置多磁盘文件路径，通过磁盘并行写入提高 File Channel 性能。Flume 将 Event 顺序写入到 File Channel 文件的末尾。可以在配置文件中通过设置 maxFileSize 参数配置数据文件大小，当被写入的文件大小达到上限的时候，Flume 会重新创建新的文件存储写入 Event。当一个已经关闭的只读数据文件的 Event 被读取完成，并且 Sink 已经提交读取完成的事务，则 Flume 把存储该数据的文件删除。

Flume相关面试题汇总

8、说说 Kafka Channel。

Memory Channel 有很大的丢数据风险，而且容量一般，File Channel 虽然能缓存更多的消息，但如果缓存下来的消息还没写入 Sink，此时 Agent 出现故障则 File Channel 中的消息一样不能被继续使用，直到该 Agent 恢复。而 Kafka Channel 容量大，容错能力强。

有了 Kafka Channel 可以在日志收集层只配置 Source 组件和 Kafka 组件，不需要再配置 Sink 组件，减少了日志收集层启动的进程数，有效降低服务器内存、磁盘等资源的使用率。而日志汇聚层，可以只配置 Kafka Channel 和 Sink，不需要再配置 Source。

kafka.consumer.auto.offset.reset，当 Kafka 中没有 Consumer 消费的初始偏移量或者当前偏移量在 Kafka 中不存在（比如数据已经被删除）情况下 Consumer 选择从 Kafka 拉取消息的方式，earliest 表示从最早的偏移量开始拉取，latest 表示从最新的偏移量开始拉取，none 表示如果没有发现该 Consumer 组之前拉取的偏移量则抛出异常。

9、介绍一下 Kafka常用的几种 Sink。

→ HDFS Sink: 将 Event 写入 HDFS 文件存储，能够有效长期存储大量数据。

→ Kafka Sink: Flume 通过 Kafka Sink 将 Event 写入到 Kafka 中的主题，其他应用通过订阅主题消费数据。kafka.producer.acks 可以设置 Producer 端发送消息到 Broker 之后不需要等待 Broker 返回成功送达的信号。0表示 Producer 发送消息到 Broker 之后不需要等待 Broker 返回成功送达的信号，这种方式吞吐量高，但存在丢失数据的风险。1表示 Broker 接收到消息成功写入本地 log 文件后向 Producer 返回成功 接收的信号，不需要等待所有的 Follower 全部同步完消息后再做回应，这种方式在数据丢失风险和吞吐量之间做了平衡。-1表示 Broker 接收到 Producer 的消息成功写入本地 log 并且等待所有的 Follower 成功写入本地 log 后向 Producer 返回成功接收的信号，这种方式能够保证消息不丢失，但是性能最差（层层递进）。

Flume相关面试题汇总

10、说说你对Flume拦截器的认知。

Source 将 Event 写入到 Channel 之前可以使用拦截器对 Event 进行各种形式的处理，Source 和 Channel 之间可以有多个拦截器，不同拦截器使用不同的规则处理 Event，包括时间、主机、UUID、正则表达式等多种形式的拦截器。

11、介绍一下什么是选择器。

Source 发送的 Event 通过 Channel 选择器来选择以哪种方式写入到 Channel 中，Flume 提供三种类型 Channel 选择器，分别是复制、复用和自定义选择器。

→ 复制选择器: 一个 Source 以复制的方式将一个 Event 同时写入到多个 Channel 中，不同的 Sink 可以从不同的 Channel 中获取相同的 Event，比如一份日志数据同时写 Kafka 和 HDFS，一个 Event 同时写入两个 Channel，然后不同类型的 Sink 发送到不同的外部存储。

→ 复用选择器: 需要和拦截器配合使用，根据 Event 的头信息中不同键值数据来判断 Event 应该写入哪个 Channel 中。

12、介绍一下Flume中 的负载均衡和故障转移。

目的是为了提高整个系统的容错能力和稳定性。简单配置就可以轻松实现，首先需要设置 Sink 组，同一个 Sink 组内有多个子 Sink，不同 Sink 之间可以配置成负载均衡或者故障转移。

Flume相关面试题汇总

13、Flume的内存一般配置为多少？FileChannel如何优化？HDFS Sink小文件如何处理？（续）

1)Flume内存配置为4G（flume-env.sh修改）

2)FileChannel优化

通过配置dataDirs指向多个路径，每个路径对应不同的硬盘，增大Flume吞吐量。

checkpointDir和backupCheckpointDir也尽量配置在不同硬盘对应的目录中，保证checkpoint坏掉后，可以快速使用backupCheckpointDir恢复数据

3)Sink：HDFS Sink小文件处理

这三个参数配置写入HDFS后会产生小文件，hdfs.rollInterval、hdfs.rollSize、hdfs.rollCount

14、说说Flume中常用的Source和Channel，并阐述Put事务，Take事务的用法。

→常用的Source:

Taildir Source：断点续传、多目录。Flume1.6以前需要自己自定义Source记录每次读取文件位置，实现断点续传。

→常用的Channel:

File Channel：数据存储在磁盘，宕机数据可以保存。但是传输速率慢。适合对数据传输可靠性要求高的场景，比如，金融行业。

Memory Channel：数据存储在内存中，宕机数据丢失。传输速率快。适合对数据传输可靠性要求不高的场景，比如，普通的日志数据。

Kafka Channel：减少了Flume的Sink阶段，提高了传输效率。

→Put事务、Take事务：

Source到Channel是Put事务

Channel到Sink是Take事务

Flume相关面试题汇总

15、Flume中拦截器如何使用？

（1）拦截器注意事项

项目中自定义了：ETL拦截器和区分类型拦截器。

采用两个拦截器的优缺点：优点 → 模块化开发和可移植性；缺点 → 性能会低一些

（2）自定义拦截器步骤

a）实现 Interceptor

b）重写四个方法

initialize 初始化

public Event intercept(Event event) 处理单个Event

public List<Event> intercept(List<Event> events) 处理多个Event，在这个方法中调用Event intercept(Event event)

close 方法

c）静态内部类，实现Interceptor.Builder

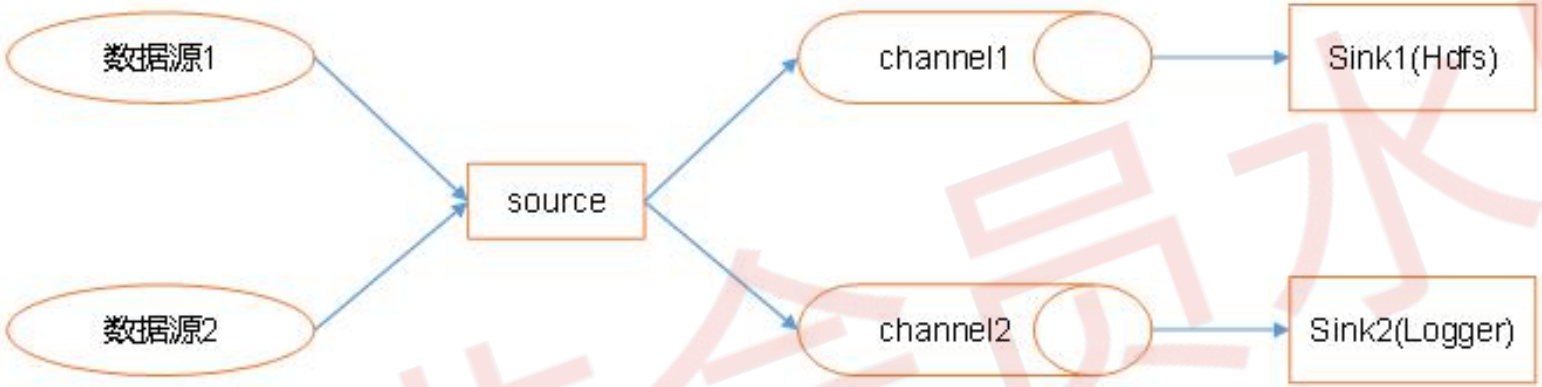
d）将自定义拦截器打包，上传到flume的lib目录下

e）修改flume的核心配置文件：

```
#source1-interceptor
customInterceptor.sources.r1.interceptors=i1
customInterceptor.sources.r1.interceptors.i1.type=com.qf.interceptor.CustomInterceptor$Builder
customInterceptor.sources.r1.interceptors.i1.param=parameter
```

Flume相关面试题汇总

16、请对Flume Channel选择器进行深度剖析。



Channel Selectors，可以让不同的项目日志通过不同的Channel到不同的Sink中去。

官方文档上Channel Selectors 有两种类型:Replicating Channel Selector (default)和 Multiplexing Channel Selector

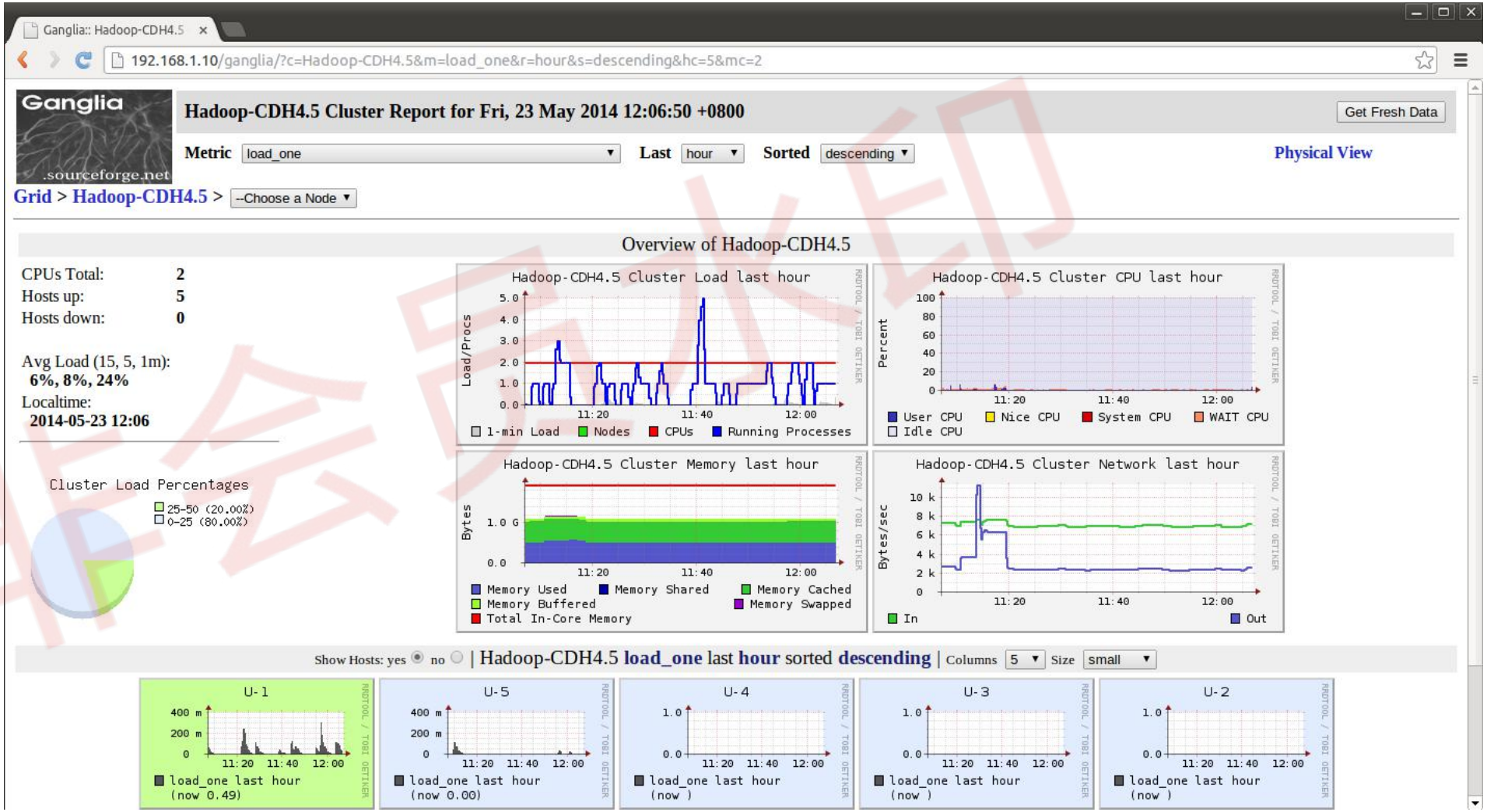
这两种Selector的区别是:Replicating 会将source过来的events发往所有channel,而Multiplexing可以选择该发往哪些Channel。

Flume相关面试题汇总

17、使用什么对Flume的运行状态进行监控？

→ 使用Ganglia。

→介绍：Ganglia是UC Berkeley发起的一个开源集群监视项目，设计用于测量数以千计的节点。Ganglia的核心包含gmond、gmetad以及一个Web前端。主要是用来监控系统性能，如：cpu 、mem、硬盘利用率， I/O负载、网络流量情况等，通过曲线很容易见到每个节点的工作状态，对合理调整、分配系统资源，提高系统整体性能起到重要作用。



Flume相关面试题汇总

18、Flume采集数据会丢失吗?（防止数据丢失的机制）

不会，Channel存储可以存储在File中，数据传输自身有事务。

19、Flume内存如何进行设置？

开发中在flume-env.sh中设置JVM heap为4G或更高，部署在单独的服务器上（4核8线程16G内存）

-Xmx与-Xms最好设置一致，减少内存抖动带来的性能影响，如果设置不一致容易导致频繁full gc。

(Xms → 是指设定程序启动时占用内存大小；Xmx →是指设定程序运行期间最大可占用的内存大小)

20、FileChannel如何进行优化？请进行深度剖析。

通过配置dataDirs指向多个路径，每个路径对应不同的硬盘，增大Flume吞吐量。

官方说明如下：

Comma separated list of directories for storing log files. Using multiple directories on separate disks can improve file channel peformance

checkpointDir和backupCheckpointDir也尽量配置在不同硬盘对应的目录中，保证checkpoint坏掉后，可以快速使用backupCheckpointDir恢复数据

Flume相关面试题汇总

21、HDFS Sink小文件应该如何处理？请进行深度剖析。

→ HDFS存入大量小文件，有什么影响？

元数据层面：每个小文件都有一份元数据，其中包括文件路径，文件名，所有者，所属组，权限，创建时间等，这些信息都保存在Namenode内存中。所以小文件过多，会占用Namenode服务器大量内存，影响Namenode性能和使用寿命

计算层面：默认情况下MR会对每个小文件启用一个Map任务计算，非常影响计算性能。同时也影响磁盘寻址时间。

→ HDFS小文件处理

官方默认的这三个参数配置写入HDFS后会产生小文件，hdfs.rollInterval、hdfs.rollSize、hdfs.rollCount

基于以上hdfs.rollInterval=3600，hdfs.rollSize=134217728，hdfs.rollCount =0，hdfs.roundValue=10，hdfs.roundUnit= second几个参数综合作用，效果如下：

①tmp文件在达到128M时会滚动生成正式文件

②tmp文件创建超10秒时会滚动生成正式文件

举例：在2019-01-01 05:23的时侯sink接收到数据，那会产生如下tmp文件：

/wq/20190101/wq.201901010520.tmp

即使文件内容没有达到128M，也会在05:33时滚动生成正式文件