

Apache Flume基础及使用案例

• objective(本课目标)

- ☐ 介绍Apache Flume架构
- ☒ 了解Apache Flume工作流程
- ☒ 掌握Apache Flume 与 Apache Kafka的集成
- ☒ 了解Apache Flume的常用案例



• What is Apache Flume?

- Apache Flume is
 - A service for streaming logs into Hadoop and other targets from multiple sources. (一种用于将日志从多个源流式传输到Hadoop和其他目标的服务)
 - A distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of streaming data into the Hadoop Distributed File System (HDFS). (分布式，可靠且可用的服务，用于有效地收集，聚合大量流数据并将其移动到Hadoop分布式文件系统HDFS)
- Apache Flume has a simple and flexible architecture based on streaming data flows; and is robust and fault tolerant with tunable reliability mechanisms for failover and recovery. (Apache Flume具有基于流数据流的简单灵活的体系结构。并且具有可调节的可靠性机制，用于故障转移和恢复，具有强大的功能和容错能力)
- User Guide: <https://flume.apache.org/FlumeUserGuide.html>

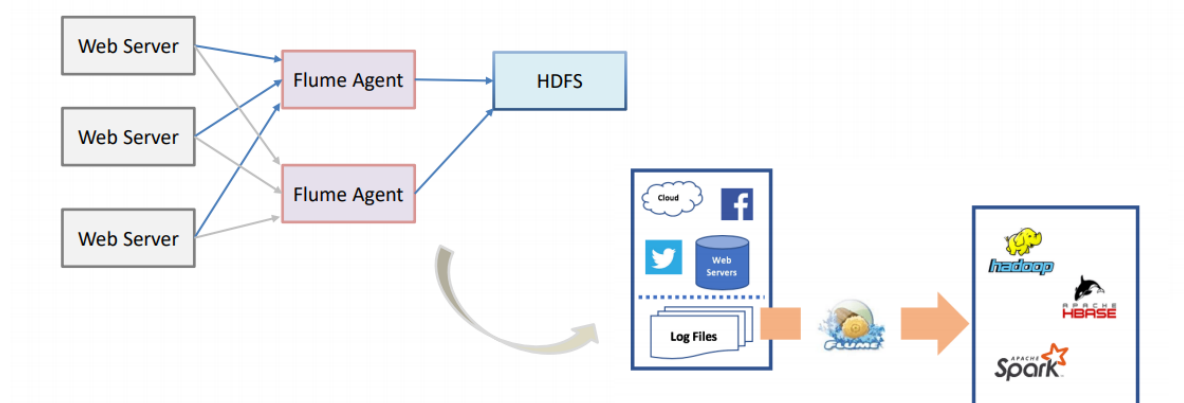
• What Apache Flume does?(Apache Flume是做什么的?)

- Stream Data(流式数据)

- Ingest streaming data from multiple sources into Hadoop for storage and analysis(将来自多个源的流数据集成到Hadoop储存与分析)
- Insulate Systems(缓冲区)
 - Buffer storage platform from transient spikes, when the rate of incoming data exceeds the rate at which data can be written to the destination (当传入数据的速率超过数据写入的速率时，缓冲存储平台将免受瞬时峰值的影响)
- Guarantee Data Delivery (保证数据传递)
 - Uses channel-based transactions to guarantee reliable message delivery. (使用基于channel的事务来确保可靠的消息传递)
- Scale horizontally (横向扩展)
 - Ingest new data streams and additional volume as needed (根据需要集成新的数据流和额外的数据量)

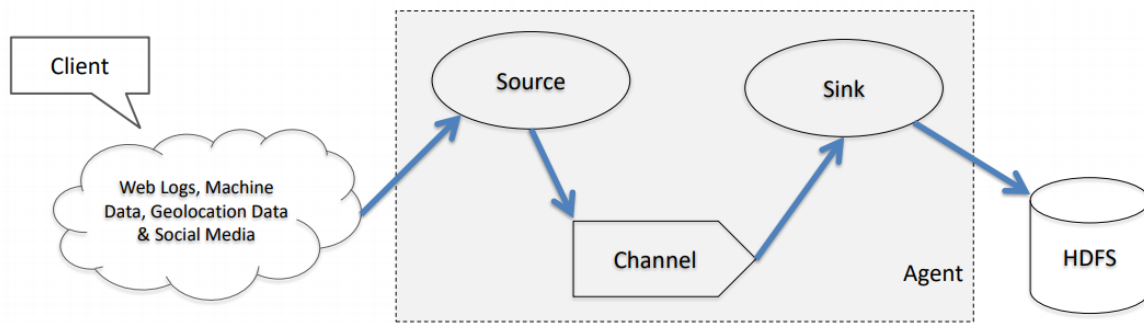
• Simple Flume Data-Flow(简单Flume数据流)

Apache Flume was initially developed by Cloudera to provide a way to quickly and reliably stream large volumes of log files generated by web servers into Hadoop. (Apache Flume最初是由Cloudera开发的，旨在提供一种将Web服务器生成的大量日志文件快速可靠地流式传输到Hadoop的方法)

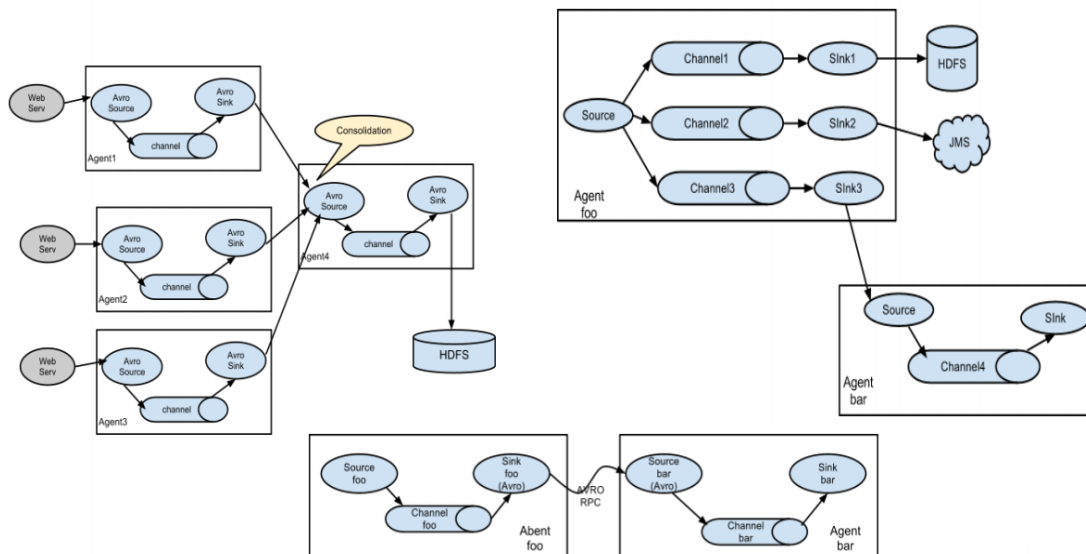


• Apache Flume Architecture(Apache Flume架构)

- Flume deploys as one or more agents;(Flume作为一个或多个代理部署)
- A Flume agent is a JVM process that hosts the components through which events flow from an external source to the next destination.(Flume agent是一个JVM进程，承载了组件，事件通过这些组件从外部源流到下一个目标)
- Each Agent contains three component(每个Agent包含三个组成部分)
 - Source(s), Channel(s) and Sink

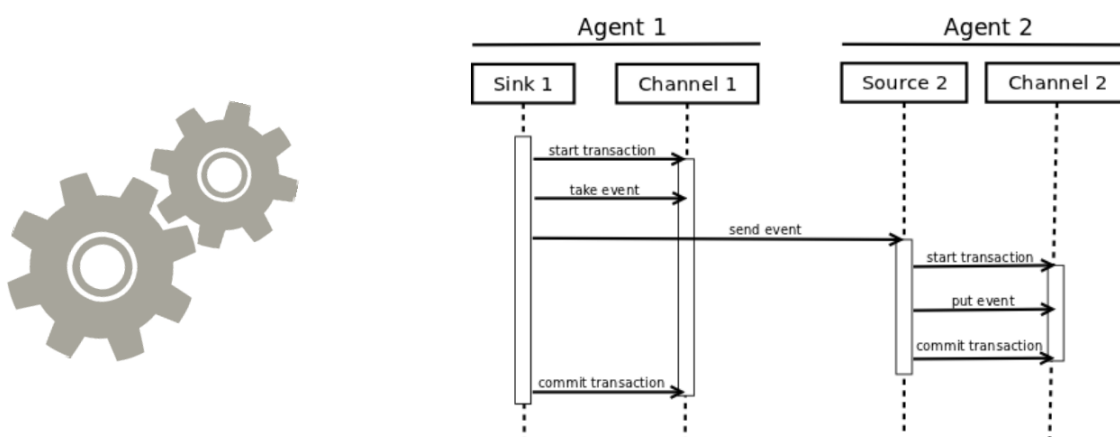


• Multiple Agents Architecture(多Agent架构)



• Transaction(事物)

- The Transaction Interface is the basis of reliability for Flume(事务接口是Flume可靠性的基础)
- A Transaction is implemented within a Channel. (transaction 是在Channel内实现的)
 - Each Source and Sink that is connected to a Channel must obtain a Transaction Object (连接到channel的每个source和接收器都必须获得一个事务对象)



• Apache Flume Event(事件)

- An event is a single packet of data passed through a system (Source → Channel → Sink); (event是通过系统传递的单个数据包)
- In log files terminology, an event is a line of text followed by a new line character. (在日志文件术语中, event是一行文本, 后跟一个换行符)

• Apache Flume Source(源)

- A Flume Source
 - Listens for events from an external application, such as web server, geolocation collector, sensor: (监听来自外部应用程序的events, 如web服务器、地理定位收集器、传感器:)
 - Reads data; (读取数据)
 - Translates events; (变换为events)
 - Handles failure (处理失败)
 - Doesn't store events (不存储事件)
 - Sends events to the Channel (将事件发送到Channel)
- Sources with built-in APIs: (内置的API)
 - Avro source, Spooling-Directory Source, Syslog source, HTTP source, Twitter source, Netcat source, etc.
 - Exec source – an Unix command to produce data on STDOUT (在STDOUT上生成数据的Unix命令)

• Spooling Directory Source(目录监控源)

- Spooling Directory Source:
 - Watches a specified directory for new files; (监控指定目录产生的文件)
 - Parses events out of new files as they appear; (从出现的新文件中解析为事件)
 - After a file has been fully processed, it is renamed to indicate the completion (or optionally deleted) (完全处理文件后, 将其重命名以指示完成, 或选择删除)

• Apache Flume Channel(管道)

- A Channel
 - Is communication bridge between Sources and Sinks within an agent. (是agent中的Sources和Sinks之间的通信桥接器)
 - Stores events until they are consumed by a Flume Sink (存储events, 直到它们被Flume Sink消耗)
- Channels with built-in support: (内置支持)
 - Memory Channel
 - File Channel (with Checkpoint)

- JDBC Channel
- Spill-able Memory Channel (内存和disk)
- Pseudo Transaction Channel (伪事务channel仅用于单元测试目的,不适用于生产用途)

• Apache Flume Sink

- A Sink
 - Removes the event from the channel (从channel中删除event)
 - Puts it into an external repository like HDFS or forwards it to the Flume source of the next Flume agent in the flow; (将其放入HDFS等外部存储库中, 或将其转发到流程中下一个Flume Agent的 Flume Sources)
- A Flume source and sink within a given agent run asynchronously with the events staged in the channel (source和sink在agent中是异步执行的)
- Sinks with built-in support: (Sink的内置支持)
 - HDFS Sink, Logger Sink, Avro Sink, Thrift Sink, File-Roll Sink, Null Sink, HBase Sink, ElasticSearch Sink, IRC Sink, MorphlineSolr Sink, etc.

• HDFS Sink

- HDFS Sink:
 - Writes events to HDFS (将events写入HDFS)
 - Supports multiple file formats - Text, Avro, etc. (支持多种文件格式)
- Rollover properties: (特性)

attribute	value	description
hdfs.rollInterval	30	Number of seconds to wait before rolling a file (0 deactivates this feature)(在滚动文件之前等待的秒数)
hdfs.rollSize	1024	File size, in bytes, to trigger roll of a file (0 deactivates this feature)(触发文件滚动的文件大小)
hdfs.rollCount	10	Number of events written to file before rolling a file (0 deactivates this feature)(在滚动文件之前写入文件的事件数)
hdfs.idleTimeout	0	Timeout after which inactive files get closed (0 deactivates this feature)(超时后无效文件关闭)
gdbs.batchSize	100	Number of events written to file before they are flushed to HDFS(在刷新到HDFS之前写入文件的事件数)

• Demo – Spooling Directory

- Source:
 - spoolDir
 - spoolDir = /tmp/events
- Channel:
 - memory
 - capacity = 10000 (容量)
 - transactioncapacity = 50000 (事务处理能力)
- Sink:
 - hdfs
 - fileType – SequenceFile, DataStream (output files not compressed), and Compressed Stream

```
# 案例1
# agent
bw1.sources=eventDir
bw1.channels=memroyChannel
bw1.sinks=eventHDFS

# source
bw1.sources.eventDir.type=spooldir
bw1.sources.eventDir.spoolDir=/tmp/users

# channels
bw1.channels.memroyChannel.type=memory
bw1.channels.memroyChannel.capacity=10000 #可以多少个event
bw1.channels.memroyChannel.transactionCapctiy=5000 #一个transaction多少个消息

# sink
bw1.sinks.eventHDFS.type=hdfs
bw1.sinks.eventHDFS.hdfs.fileType=DataStream
bw1.sinks.eventHDFS.hdfs.path=/tmp/users
bw1.sinks.eventHDFS.hdfs.batchSize=10000

# asosation
bw1.sources.eventDir.channels=memroyChannel
bw1.sinks.eventHDFS.channel=memroyChannel
```

```
# 案例2
# 准备数据 /var/log/eventlog_demo.log
# 查看日志 /var/log/flume/
sjsd1.sources=source_logs
sjsd1.channels=channel_logs
sjsd1.sinks=sink_logs
```

```

sjsd1.sources.source_logs.type=exec
sjsd1.sources.source_logs.command= tail -F --lines=500
/tmp/log/flume/userLog_2013.log
sjsd1.sources.source_logs.restart=true
sjsd1.sources.source_logs.batchSize=1000

sjsd1.sinks.sink_logs.type=hdfs
sjsd1.sinks.sink_logs.hdfs.fileType=DataStream
sjsd1.sinks.sink_logs.hdfs.path=/flume_stage/exec
sjsd1.sinks.sink_logs.hdfs.filePrefix=userLog
sjsd1.sinks.sink_logs.hdfs.fileSuffix=.log
sjsd1.sinks.sink_logs.hdfs.batchSize=5000

sjsd1.channels.channel_logs.type=file
sjsd1.channels.channel_logs.checkpointDir=/tmp/flume/checkpoint
sjsd1.channels.channel_logs.dataDirs=/tmp/flume/data

sjsd1.sources.source_logs.channels=channel_logs
sjsd1.sinks.sink_logs.channel=channel_logs

```

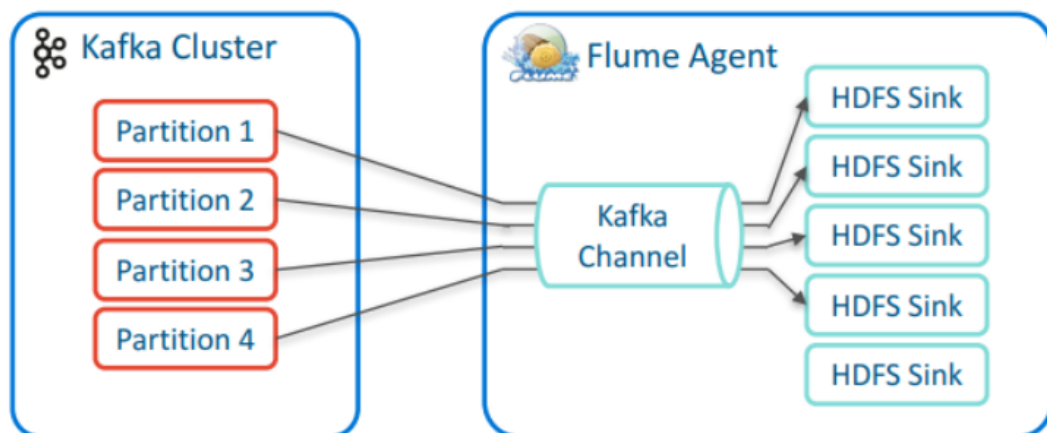
• kafka: Flume + Kafka (1)

• Kafka Channels

- In an Agent, Kafka (one topic) is used as a Channel(在Agent中, Kafka(one topic)用作Channel)



• Bottleneck: HDFS Sink



```

#创建Topic
#kafka-topics.sh --zookeeper sandbox-hdp.hortonworks.com:2181 --create --topic
kafkademio --partitions 2 --replication-factor 1

```

```

#监控数据
# kafka-console-consumer.sh --bootstrap-server sandbox-hdp.hortonworks.com:6667 -
-topic

kafkaexe.sources = execSource
kafkaexe.channels = kafkaChannel
kafkaexe.sinks = hdfsSink

kafkaexe.sources.execSource.type = exec
kafkaexe.sources.execSource.command = tail -F --lines=500
/tmp/log/flume/userLog_2013.log
kafkaexe.sources.execSource.channels = kafkaChannel

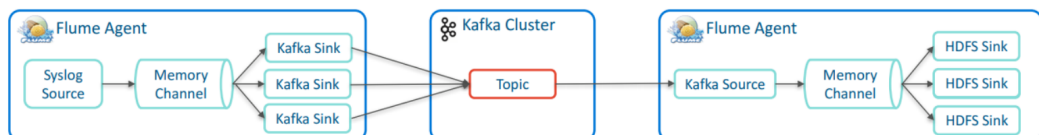
kafkaexe.channels.kafkaChannel.type = org.apache.flume.channel.kafka.KafkaChannel
kafkaexe.channels.kafkaChannel.capacity = 10000
kafkaexe.channels.kafkaChannel.transactionCapacity = 1000
kafkaexe.channels.kafkaChannel.brokerList = sandbox-hdp.hortonworks.com:6667
kafkaexe.channels.kafkaChannel.topic = kafkademio
kafkaexe.channels.kafkaChannel.zookeeperConnect = sandbox-
hdp.hortonworks.com:2181
kafkaexe.channels.kafkaChannel.parseAsFlumeEvent = true

kafkaexe.sinks.hdfsSink.type = hdfs
kafkaexe.sinks.hdfsSink.hdfs.path = /tmp/kafka/channel
kafkaexe.sinks.hdfsSink.hdfs.rollInterval = 5
kafkaexe.sinks.hdfsSink.hdfs.rollSize = 0
kafkaexe.sinks.hdfsSink.hdfs.rollCount = 0
kafkaexe.sinks.hdfsSink.hdfs.fileType = DataStream
kafkaexe.sinks.hdfsSink.channel = kafkaChannel

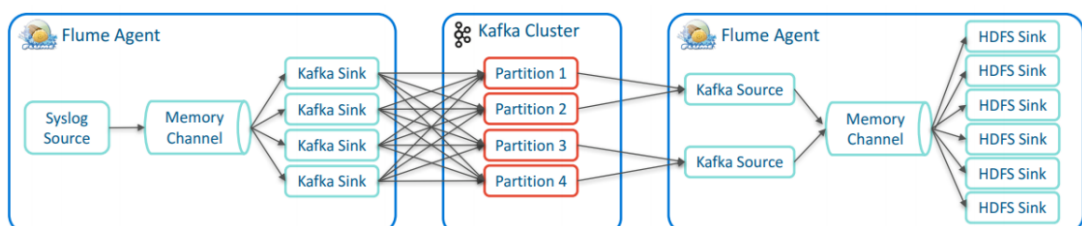
```

• kafka: Flume + Kafka (2)

- Kafka Source & Sink
 - In Flume agents, Kafka are used as Source and/or Sink (在Flume Agent中, Kafka用作Source和Sink)



- Bottleneck – Kafka Source & HDFS Sink.




```

kafkaexe2.sources = kafkaSource
kafkaexe2.channels = memoryChannel
kafkaexe2.sinks = hdfsSink

# kafka Source
kafkaexe2.sources.kafkaSource.type = org.apache.flume.source.kafka.KafkaSource
kafkaexe2.sources.kafkaSource.zookeeperConnect = sandbox-hdp.hortonworks.com:2181
kafkaexe2.sources.kafkaSource.topic = kafkademo
kafkaexe2.sources.kafkaSource.batchSize = 10
kafkaexe2.sources.kafkaSource.channels = memoryChannel

# memory Channel
kafkaexe2.channels.memoryChannel.type = memory
kafkaexe2.channels.memoryChannel.capacity = 500
kafkaexe2.channels.memoryChannel.transactioncapacity = 100

# hdfs Sink
kafkaexe2.sinks.hdfsSink.channel = memoryChannel
kafkaexe2.sinks.hdfsSink.type = hdfs
kafkaexe2.sinks.hdfsSink.hdfs.writeFormat = Text
kafkaexe2.sinks.hdfsSink.hdfs.fileType = DataStream
kafkaexe2.sinks.hdfsSink.hdfs.filePrefix = kafkaexe2Events
kafkaexe2.sinks.hdfsSink.hdfs.useLocalTimeStamp = true
kafkaexe2.sinks.hdfsSink.hdfs.path = /tmp/sink/%y-%m-%d
kafkaexe2.sinks.hdfsSink.hdfs.rollCount=100
kafkaexe2.sinks.hdfsSink.hdfs.rollSize=0

```

• Apache Flume Interceptor(Apache Flume拦截器)

- An Interceptor can modify or even drop events based on any criteria. (Interceptor可以根据条件修改甚至删除events)
- Flume supports chaining of interceptors.

```

#案例实现
# user header filter
users.sources.usersSource.interceptors = user_header_filter
users.sources.usersSource.interceptors.user_header_filter.type = regex_filter
users.sources.usersSource.interceptors.user_header_filter.regex =
^(\\s?)user_id(\\s?), (\\s?)locale(\\s?), (\\s?)birthyear(\\s?), (\\s?)gender(\\s?),
(\\s?)joinedAt(\\s?), (\\s?)location(\\s?), (\\s?)timezone(\\s?)$
users.sources.usersSource.interceptors.user_header_filter.excludeEvents =
true

```

- Interceptors are classes that implement org.apache.flume.interceptor.Interceptor (拦截器是实现org.apache.flum .interceptor. interceptor的类)

• Flume Custom Components(自定义组件)

- Apart from the built-in components, custom Sources, Channels and Sinks can be created in Java. (除了内置组件外，还可以用Java创建自定义Sources, Channels and Sinks)

```
package com.sjsd1learning.flume.source
public class sjsd1Source extends AbstractSource implements Configurable,
PollableSource { /* ... */ }
//config:
// sources.sjsd1.type=com.sjsd1learning.flume.source.sjsd1Source

package com.sjsd1learning.flume.interceptor
public class sjsd1Interceptor implements Interceptor { /* ... */ }
//config: sources.sjsd1.interceptors = i1 sources.sjsd1.interceptors.i1.type =
com.sjsd1learning.flume.interceptor.sjsd1Interceptor$Builde

public class sjsd1Sink extends AbstractSink { /* ... */ }
```

• Sink Processors(Sink处理器)

- Sink processors:
 - Sink groups allow users to group multiple sinks into one entity. (sink groups允许用户将多个sinks分组为一个实体)
 - Sink processors can be used to provide load balancing capabilities over all sinks inside the group or to achieve fail-over from one sink to another in case of temporal failure. (Sink processors可用于在组内所有sinks上提供负载平衡功能，或在临时故障时实现从一个sink到另一个sink的故障转移)

```
a1.sinkgroups = g1
a1.sinkgroups.g1.sinks = k1 k2
a1.sinkgroups.g1.processor.type = failover #用途
a1.sinkgroups.g1.processor.priority.k1 = 5 #优先级
a1.sinkgroups.g1.processor.priority.k2 = 10
a1.sinkgroups.g1.processor.maxpenalty = 10000 #10m的check时间

a1.sinkgroups = g1
a1.sinkgroups.g1.sinks = k1 k2
a1.sinkgroups.g1.processor.type = load_balance
a1.sinkgroups.g1.processor.backoff = true #从group中删除失败的节点
a1.sinkgroups.g1.processor.selector = random
```

• When Flume is not a GOOD fit?(Flume的优缺点)

- Very large Events (event比较大)

- An event cannot be larger than memory or a disk on an agent's machine (event不能大于Agent节点上的内存或磁盘)
- Infrequent bulks loads(不太适合数据批处理)
 - Other tools might be better, e.g. HDFS File Slurper (其他工具可能会更好, 例如 HDFS文件清理程序)

• Use Cases(实际用例)

- In factories, there are many machines, each machine produces lot of logs. Flume can be used for collecting logs for machine state, product quality analysis. (Flume可用于收集日志以进行机器状态, 产品质量分析)
- In e-Commerce, Flume can be used for collecting web logs to understand customers ' browsing / shopping behaviors. (在电子商务中, Flume可用于收集Web日志以了解客户的浏览/购物行为)
- In social-network, Flume can be used for collecting tweets, chats, messages for sentimental analysis. (在社交网络中, Flume可用于收集tweets, 聊天, 消息以进行情感分析)

• Summary(总结)

- 了解了Apache Flume架构
- 学习了Apache Flume工作流程
- 练习了Apache Flume 与 Apache Kafka的集成
- 了解了Apache Flume的常用案例