目录

[一． Kafka 2](#_Toc417974534)

[1 概述 2](#_Toc417974535)

[2 架构 2](#_Toc417974536)

[3 系统组件 3](#_Toc417974537)

[二． Scribe 3](#_Toc417974538)

[1 概述 3](#_Toc417974539)

[2 架构 3](#_Toc417974540)

[3 系统组件 4](#_Toc417974541)

[三． Flume 4](#_Toc417974542)

[1 概述 4](#_Toc417974543)

[2 架构 5](#_Toc417974544)

[3 系统组件 5](#_Toc417974545)

非结构化数据产生于许多应用领域，包括互联网实时数据、企业内部数据等，这些数据的容量也正在迅猛的增长。如何及时、准确的获取到所关心的数据并进行必要的预处理已经成为企业必须解决的课题。

企业中常用的数据采集技术有Kafka、Scribe和Flume。下面对这三种技术框架做一个简单的描述。

### 一． Kafka

# 概述

Apache Kafka是一个以高吞吐量为设计目标的分布式消息系统。它具有如下特性：

1). 即便是在TB级的消息存储情况下，Kafka仍可提供稳定的响应时间。

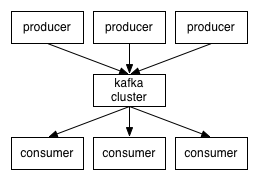
2). 高吞吐量。即便是运行在很普通的机器上，Kafka仍可支持每秒几十万条的消息处理。

3). 在保证分区有序的前提下，支持Kafka服务器间的消息分区和分布式消息提取。

4). 支持Hadoop数据并行加载。

# 架构

Kafka是发布订阅式的消息传输系统，基本的传输单元是一条消息。当与生产者交互时，它没有日志文件的概念；当发送数据给消费者时，它也没有数据库的概念。这解耦了生产者和它的机器及应用，消除了一系列的数据序列化问题。所有的消息都归类为“topic”，这个“topic”用于将消息路由到消费者。



# 系统组件

系统主要包括三类组件：

1). 生产者：产生消息。每一个产生日志数据的应用都可以是一个生产者。

2). Kafka服务器：从生产者接收消息，处理消息持久化，发送消息到合适的“topic”队列。

3).消费者：从“topic”队列中处理消息。

ZooKeeper负责协调组件间的相关通讯和集群的状态

### 二． Scribe

# 概述

Scribe是Facebook使用Apache Thrift实现的日志收集系统。它具有如下特性：

1). 多语言支持。使用非阻塞C++服务器实现Thrift服务。

2). 方便扩展。Scribe服务器部署在一个有向图里，每一台服务器只知道图中的下一台服务器。

3). 可靠性高。当Scribe服务器无法向下一台Scribe服务器发送消息时（因为机器或者网络故障），这些消息将存储到本地磁盘，等待故障消除后重新发送。

4). 不提供事务保障。有可能造成少部分数据丢失。

# 架构

Scribe系统由一台或多台服务器构成，可从大量的日志服务器上汇集数据。基于有向拓扑图设计使得系统可以扩展到很大规模，具有很强的容灾性能。系统中每个节点上都运行一个Scribe服务器，负责汇集、改善消息到中央服务器。如果中央服务器或者下一层服务器因为网络或者机器故障暂时不可用，则消息会保存到本地磁盘的文件里，等待故障消除后重新发送。这个原则同样适用于中央服务器。中央服务器最终将消息写到NFS或者分布式文件系统。



# 系统组件

系统主要包括四类组件：

1). 应用服务器：产生日志信息。每一台应用服务器都通过Thrift接口与一个Scribe客户端连接，通过这个连接向Scribe发送消息。

2). Scribe客户端：从应用服务器接收消息，如果中央服务器或下一层服务器可用，则发送消息；否则保存到本地磁盘，后续可再次发送。在系统负载较轻的情况下，客户端也可以不存在，应用服务器可直接通过Thrift接口发送消息到中央服务器。

3). Scribe中央服务器：从客户端或者应用服务器接收消息，如果目标服务器可用，则写数据到目标服务器；否则保存到本地磁盘，后续可再次发送。

4).目标服务器：消息的最终存储位置。目标服务器通常是NFS或者分布式文件系统，如比较常用的HDFS。

### 三． Flume

# 概述

Flume是一种可靠的分布式日志收集服务。它具有如下特性：

1). 与HDFS充分整合。简单配置后即可向HDFS导入数据。

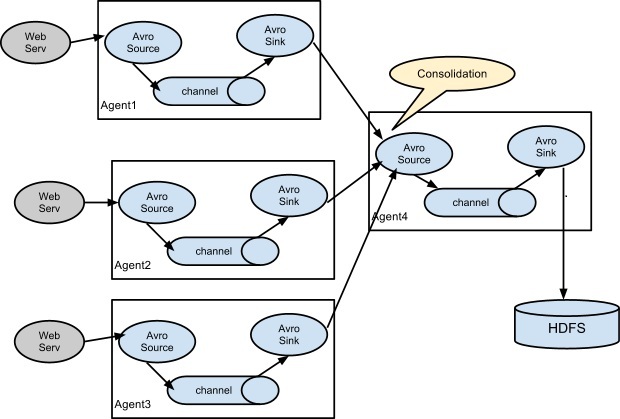
2). 用途广泛。不仅用于日志汇集，还可用于传输网络流量数据、社交媒体数据、电子邮件等很多数据源。

3). 可靠性高。只有事件到达下一个Flume组件之后，这个事件才会从通道中删除。

4). 容灾性好。通道可设置为本地文件，保证了Flume组件失败之后事件仍可恢复。

# 架构

Flume用于从多种数据源高效的汇集并传输大量数据到一个中央数据存储，基本的传输单元是事件。一个事件包含数据和一组可选属性。当Flume组件接收到事件时，会将该事件存储到一个或多个通道中，等待下一个Flume组件从通道中取走事件。事件最终会存储到外部存储媒介，如HDFS。事件的接收和取走是异步进行的，事件会暂时驻留在通道里。用户可定义多个组件依次完成事件的传送，也可定义fan-in和fan-out模式的数据流、基于上下文的路由功能和备份路由（容灾），等等。



# 系统组件

系统主要包括三类组件：

1). 应用服务器：产生源数据。每一台应用服务器都通过Flume Source与一个Flume组件连接，通过这个连接向Flume组件发送消息。

2). Flume组件：从数据源接收事件，放入通道里，等待Flume Sink将事件取走。每一个Flume组件都由Flume Source、Flume Sink和通道三部分组成。其中，Flume Source负责从外部接收并保存事件到通道，通道里会暂时驻留未取走的事件数据，而Flume Sink则从通道中取走事件并存储到外部存储媒介或者传递给下一个Flume组件。

3）目标服务器：事件的最终存储位置。