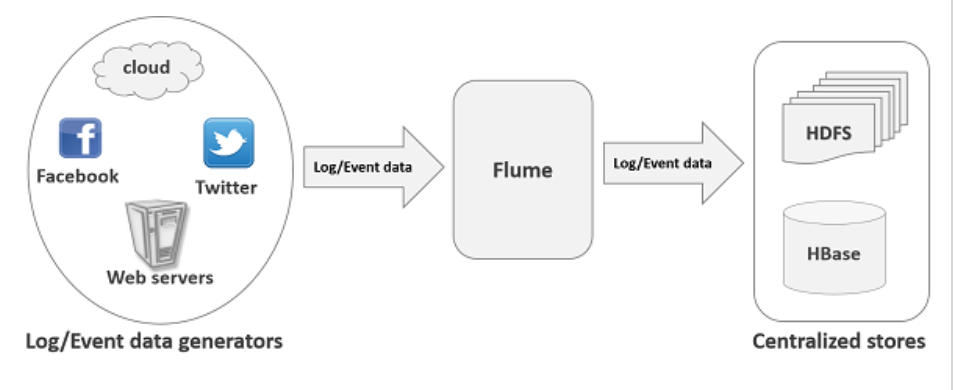
**Flume: Data ingestion tool**

1. **What is Flume?**

Data ingestion tool used to collect, aggregate and transport large amount of streaming data from various data sources to a centralized data source.

Here data can be log or some events.

For Example: Inserting vguid events into Hadoop directly from kafka to hdfs



1. **Where we can use flume give example?Flume applications?**

Telecom: to dump huge events from HDFS to kafka or from kafka to HDFS.

E-commerce: Web application wants to analyze the customer behavior from a particular region. To do so, they would need to move the available log data in to Hadoop for analysis at higher speed

1. **What are the main advantages of the flume?**
2. *Helps to write the data at steady flow* to the destination (centralized store). Even though the rate of incoming data exceed the rate at which data can be written
3. The data transfer between source and destination is *channel based.* There will be 2 transactions maintained for each message one for receiving and one for sending in this way flume guarantees reliable message delivery.
4. *Reliable, scalable, fault tolerant, customizable and manageable*
5. *Contextual routing (As per situation).*
6. We can store huge web server log data or data from any other source to centralized data storage like HDFS or HBASE.
7. **Features of flume?**
8. Flume supports large set of data sources and destinations.
9. We can get the data from multiple servers into Hadoop immediately.
10. Using flume we can also import events data produced by social networking sites like Facebook, twitter and e commerce websites like Flipkart and amazon.
11. Flume supports multi-hop flows, fan-in fan-out flows, contextual routing, etc.
12. Flume can be scaled horizontally.
13. **Why in D Project we decided to use flume when initial decision was to use batch or streaming application?**
14. Initially it was suggested to use a spark batch job which will read data from Kafka at some intervals and put it to HDFS.

Problem: To Kafka there will be 10K events every second writing batch application is not suitable as there will be continuous flow.

1. Then it is decided to use spark streaming application but streaming application can shutdown at any time so we need to write monitor scripts if some issue we may lose some event data and in case of long shutdown we need to read old data. As it is spark streaming application it consumes some resources in the form of driver and executor.
2. So considering reliability, implementation time,resources and maintenance it was decided to use flume.Which is ready made and stable solution to ingest huge amount of data from one source to other source. I can say I suggested this.
3. **What is the traditional way to store the data to HDFS and what is its draw back?**

Using HDFS put command.

Drawbacks:

1. Using put command we can transfer only one file at a time but the data generators generate the data at much higher rate.

So if we transfer one by one then the data is not moved in real time.In that case the data which we analyze is on old data and it is not accurate.

1. If we use put command the data is needs to be packaged and should be ready for the upload. Since webservers generate data at high speed it is very difficult task

Also there is one drawback of HDFS.

We know that in HDFS the file stored as directory entry and the length of the file is store as zero till the file is closed.It means if the source is writing data into hdfs and if in between if network is interrupted in the middle of the data transfer without closing the file then the data written to the file is completely lost.

What is the meaning of POSIX file system?

1. **What are the available solutions to overcome above drawbacks?**

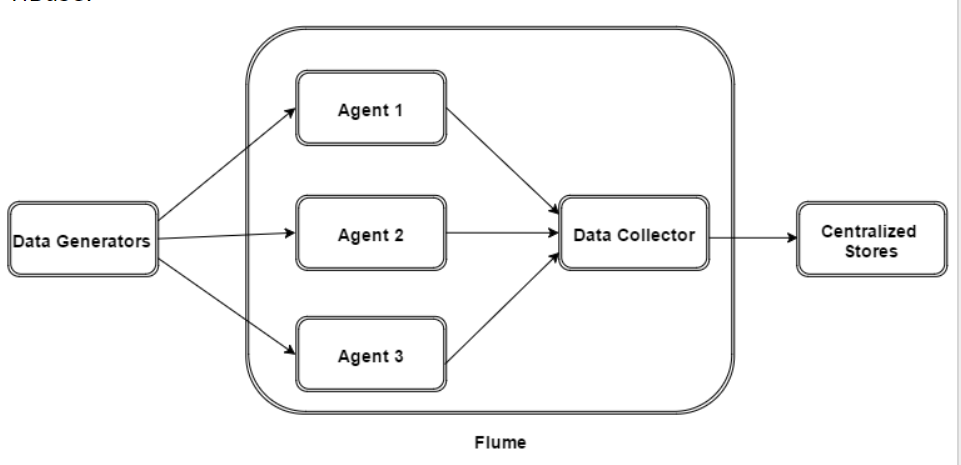
In order to overcome this drawbacks below are the solutions available in market

Facebook Scribe

Apache Kafka

Apache Flume

1. **Apache Flume Architecture?**



Data generators (such as Facebook, Twitter) generate data which gets collected by individual Flume agents running on them.

Thereafter, a data collector (which is also an agent) collects the data from the agents which is aggregated and pushed into a centralized store such as HDFS or HBase.

Below are the components in the flume architecture.

1. ***Flume Event***:



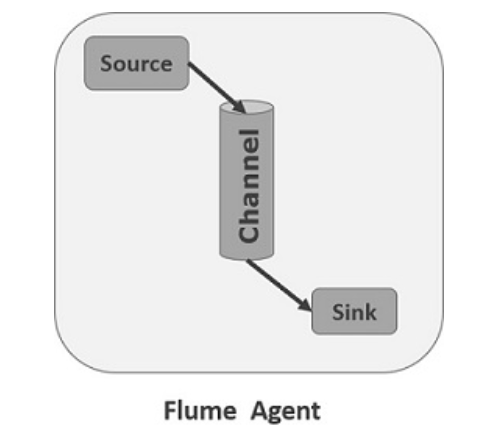
It is the data transported inside flume. Flume event contains payload byte array that is to be transported from to destination along with optional headers.

1. ***Flume Agent***

It is an independent daemon process(JVM) in flume.

It receives the data (events) from clients or other agents and forwards it to its next destination (sink or agent).

Flume can have multiple agent



As shown in the diagram a Flume Agent contains three main components namely, source, channel, and sink.

Which is the agent machine running in D Project?

**Source:**

A source is the component of an Agent which receives data from the data generators and transfers it to one or more channels in the form of Flume events.

Apache Flume supports several types of sources and each source receives events from a specified data generator.

Example − Avro source, Thrift source, twitter 1% source etc.

Which is the source used in D project?

**Channel:**

A channel is a transient store which receives the events from the source and buffers them till they are consumed by sinks. It acts as a bridge between the sources and the sinks.

These channels are fully transactional and they can work with any number of sources and sinks.

Example − JDBC channel, File system channel, Memory channel, etc.

Which is the channel used in D Project?

**Sink**

It consumes the data (events) from the channels and delivers it to the destination. The destination of the sink might be another agent or the central stores.

Example − HDFS sink

Note − A flume agent can have multiple sources, sinks and channels

Which is the sink used in D project?

Below are the additional components of the

**Interceptors**

Interceptors are used to alter/inspect flume events which are transferred between source and channel.

We have written interceptor for D Project understand it.

**Channel Selectors**

In case of multiple channels available channel selectors helps to choose which channel to be used to transfer the data

There are two types of channel selectors

*Default channel selectors (Replicating channel selectors)*- replicates all the events in each channel.

*Multiplexing channel selectors:*

Decides the channel through which data is to be transferred when there are multiple channels based on the address in the events.

Learn how to configure channel selectors?

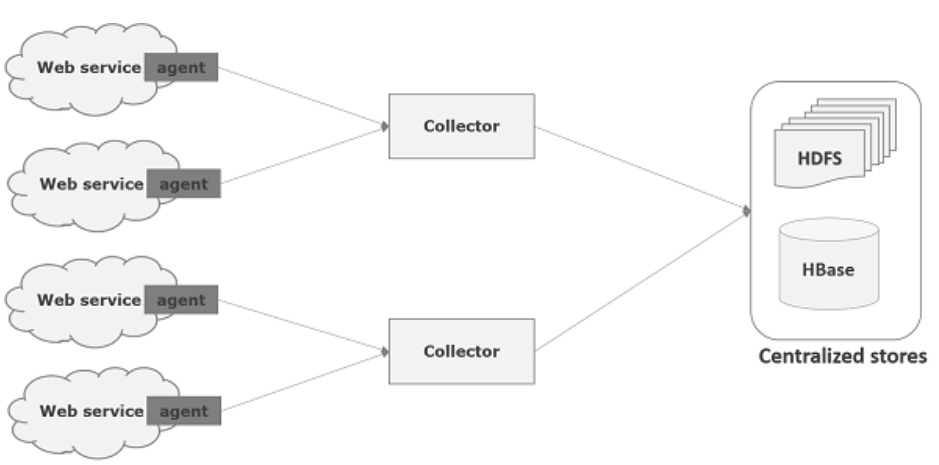
**Sink Processors:**

If there are group of sinks available then sink processor helps to invoke a particular sink.

Also helps to create failover paths for sinks.

Helps for load balancing the events across multiple sinks from a channel.

1. **What is the version of the flume which is used in D Project?**
2. **How Data Flows in Flume? Flume Data Flow?**

****

We know that flume is used to move log data into HDFS these events and log data are generated by the log servers and on each of these servers Flume agents will be running. These agents are responsible for receiving the data from data generators.

The data in these agents will be collected by an intermediate node known as Collector. Just like agents, there can be multiple collectors in Flume.

Finally, the data from all these collectors will be aggregated and pushed to a centralized store such as HBase or HDFS.

1. **What are the different type of data flows in the flume?**

*Multi-hop Flow:*

In the flume before reaching the final destination, an event may travel through more than one agent. This is known as multi-hop flow.

*Fan-out Flow*

The dataflow from one *source* to multiple *channels*

There are 2 types

Replicating − data will be replicated in all the configured channels.

Multiplexing − data will be sent to a selected channel which is mentioned in the header of the event.

*Fan-in Flow*

The data will be transferred from many sources to one channel is known as fan-in flow.

1. **How failure are handled in flume?**

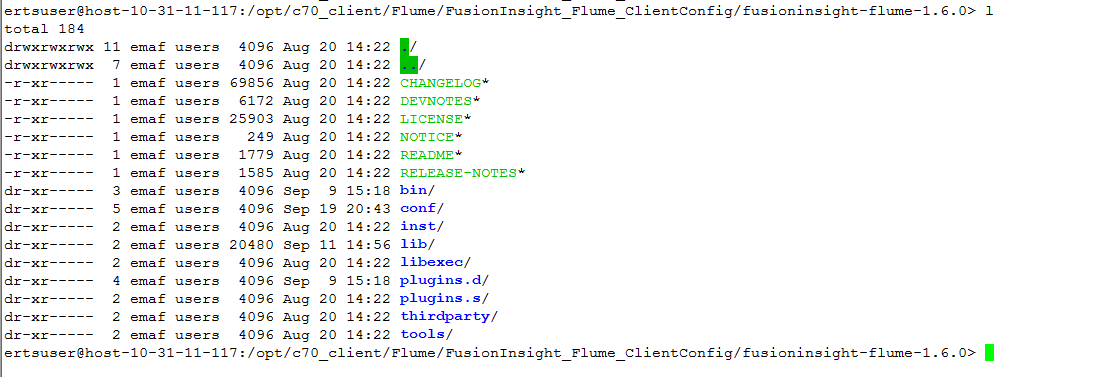
In Flume, for each event, two transactions take place. One at the sender and one at the receiver.

Sender sends an event to the receiver. Once the event is received the receiver commits its own transaction and sends a received signal to the sender. After receiving the signal the sender commits the transaction.

Sender will not commit its transaction till it receives a signal from the receiver.

1. **Apache Flume – Environment:**

<https://www.tutorialspoint.com/apache_flume/apache_flume_environment.htm>



1. **Apache Flume Configuration?**

Say we need to sink data from Kafka to HDFS in that case we need to prepare our own .properties file where we configure details about the sources, channel and Sinks.

The property file content will be something like below.

D:\Technology\02-BigData\Flume\Softwares\Sample.properties.

<https://www.tutorialspoint.com/apache_flume/apache_flume_configuration.htm>

1. What are apache flume interceptors?

<https://data-flair.training/blogs/flume-interceptors/>

Using this feature we can modify or we can drop the events before writing to sync.

We can write our own interceptor by implementing org.apache.flume.interceptor.Interceptor.

Not only one interceptors we can also do chaining of interceptors.The flume interceptors are specified in source configuration as white space separated list. The order in which interceptors are executed is the order in which they are specified in configuration file.

1. What are the types of interceptors?

<https://data-flair.training/blogs/flume-interceptors/>