**2. Problem Statement**

**● Explain the need of Flume.**

**● Explain the working of Flume and its components in brief**

**Flume is required for the following:**

1. Stream data:

Ingest streaming data from multiple sources into Hadoop for storage and analysis

2. Insulate systems :

Buffer storage platform from transient spikes, when the rate of incoming data exceeds the rate at which data can be written to the destination

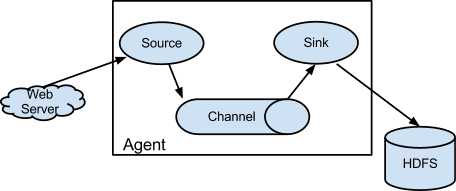
3. Guarantee data delivery:

Flume NG uses channel-based transactions to guarantee reliable message delivery. When a message moves from one agent to another, two transactions are started, one on the agent that delivers the event and the other on the agent that receives the event. This ensures guaranteed delivery semantics

4. Scale horizontally:

To ingest new data streams and additional volume as needed.

**FLUME ARCHITECTURE:**



Flume’s high-level architecture is built on a streamlined codebase that is easy to use and extend. The project is highly reliable, without the risk of data loss. Flume also supports dynamic reconfiguration without the need for a restart, which reduces downtime for its agents.

The following components make up Apache Flume:

1. Event:

A singular unit of data that is transported by Flume (typically a single log entry)

2. Source:

The entity through which data enters into Flume. Sources either actively poll for data or passively wait for data to be delivered to them. A variety of sources allow data to be collected, such as log4j logs and syslogs.

3. Sink:

The entity that delivers the data to the destination. A variety of sinks allow data to be streamed to a range of destinations. One example is the HDFS sink that writes events to HDFS.

4. Channel:

The conduit between the Source and the Sink. Sources ingest events into the channel and the sinks drain the channel.

5. Agent:

Any physical Java virtual machine running Flume. It is a collection of sources, sinks and channels.

6. Client:

The entity that produces and transmits the Event to the Source operating within the Agent.

Flume components interact in the following way:

1. A flow in Flume starts from the **Client**.
2. The **Client** transmits the **Event** to a **Source** operating within the **Agent**.
3. The **Source** receiving this **Event** then delivers it to one or more **Channels**.
4. One or more **Sinks** operating within the same **Agent** drains these **Channels**.
5. **Channels** decouple the ingestion rate from drain rate using the familiar producer-consumer model of data exchange.
6. When spikes in client side activity cause data to be generated faster than can be handled by the provisioned destination capacity can handle, the **Channel** size increases. This allows sources to continue normal operation for the duration of the spike.
7. The **Sink** of one **Agent** can be chained to the **Source** of another **Agent**. This chaining enables the creation of complex data flow topologies.