**#Assignment\_12.3**

**Que.1) what is meant by FlumeNG?**

Flume NG is a refactoring of Flume and was originally tracked in [FLUME-728](https://issues.apache.org/jira/browse/FLUME-728).

To have end-to-end reliability for the system at a high-level, Flume NG uses a single-hop message delivery guarantee semantics. The purpose of Flume is to achieve a reliable, distributed and available system for efficiently storing, aggregating and moving large amounts of log data from many different sources to a centralized data store.

While passing from one to another agent when there occurs some failure at any leg of the flow, the events affected by the failure of another begin to buffer at unaffected agent in flow which was unaffected. If the issue is not resolved within the time, it may lead to the failure of last unaffected event then it would force the agent before him to start buffering. The failure can be reported if the client tries to transmit event to its first hop destination and necessary actions could be taken on time, but if failure is resolved before first hop agent fails, the buffering events present in downstream agents will start to drain towards their destination, though flow is restored to its original levels.

**Que.2) Can Flume provides 100 % reliability to the data flow?**

Yes, Flume definitely provide 100% reliability to the data flow by means of transaction mechanism.

For every event, two mechanism takes place in flume. One at the sending and while other at receiving end. The events are sent by sender, when complete data is received by receiver, the receiver commits its own transaction by sending Received signal, Only On reaching received signal sender commits its transaction.

For some reasons if events are not logged, then transaction will roll back again and events will be there in channel for redelivery. The channel we are using is a file channel, which has the property of being durable: once an event has been written to the channel, it will not be lost, even if the agent restarts.

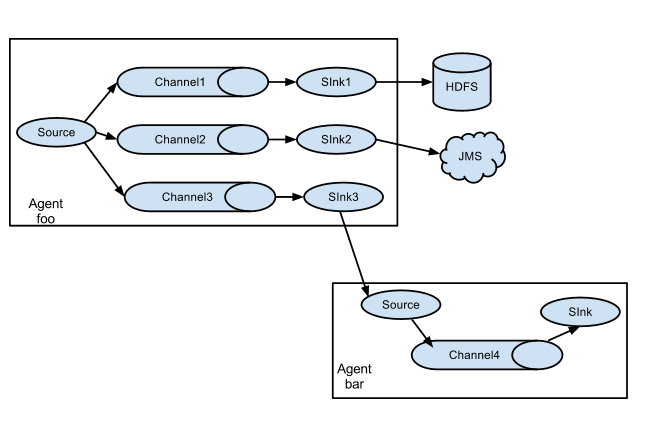
Hence 100% reliability is achieved.

**Que.3) Can Flume can distributes data to multiple destinations?**

Yes, the event flows from one sources to multiple channels and multiple destinations hence it can be achieved using fan-outs which defines the flow of the multiplexer. There exists two types of Fan-Outs:

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

**Multiplexin**g − the data flow where the data will be sent to a selected channel which is mentioned in the header of the events only.



In figure above, we can see that the data read on channel 1 is transmitted to the HDFS while from channel 2 is transmitted to JMS and channel 3 is aligned to channel 4 using sink and source of another event . This was made possible because of multiplexing and replication of data in cluster.

**Que.4) Explain about the different channel types in Flume. And which channel type is faster?**

There are three different types of channels in Flume, **Memory Channel, JDBC channel and File Channel.**

1. **Memory Channel**

Memory Channel is an in-memory channel that stores events written to it. The sources write to its tail and sinks reads its head. The Memory Channel supports very high throughput, as it holds all data in memory. There exists no data on disk hence should not be used when data loss recovery is not an objective.

**2. JDBC Channel**

JDBC Channel stores the events in an embedded Derby database.

**3. File Channel**

The File Channel is Flume’s persistent channel. It writes out all events to disk and thus does not lose data on process or machine shutdown or crash like memory channel. It is used to handle quite a few sources and sinks at the same time.

**The Fastest of them:**

MEMORY Channel is the fastest channel among the three. The channel that you choose completely depends on the nature of the big data application and the value of each event.