**Task1:- FlumeNG**

(a) Flume NG is a refactoring of Flume and was originally tracked in FLUME-728.

(b) To solve certain known issues and limitations, Flume requires a refactoring of some core classes and systems. This bug is a parent issue to track the development of a "Flume NG" - a poorly named, but necessary refactoring.

(c) This is a large and far reaching set of tasks. The intent is to perform this work in a branch as to not disrupt immediate releases or short term forthcoming releases while still allowing open development in the community.

(d) Flume NG's high level architecture solidifies a few concepts from Flume OG and drastically simplifies others. Flume NG retains Flume OG's general approach to data transfer and handling.

**Task2:- Can Flume provides 100 % reliability to the data flow:-**

Yes, Apache Flume provides end to end reliability because of its transactional approach in data flow.

**Task3:- Can Flume distribute data to multiple destinations;-**

Yes, it support multiplexing flow . The event flows from one sources to multiple channels and multiple destinations .

It’s achieved by defining a flow multiplexer.

**Task4:- different channel types in Flume. and the faster channel type :-**

The 3 different built in channel types available in Flume are-

1) MEMORY Channel – Events are read from the source into memory and passed to the sink.

2) JDBC Channel – JDBC Channel stores the events in an embedded Derby database.

3) FILE Channel –File Channel writes the contents to a file on the file system after reading the event from a

source. The file is deleted only after the contents are successfully delivered to the sink.

Memory Channel is the fastest channel among the three however has the risk of data loss. The channel that you choose

completely depends on the nature of the big data application and the value of each event.