Journal of Research in Science and Engineering (JRSE)
ISSN: 1656-1996 Volume-7, Issue-9, September 2025
DOI: 10.53469/jrse.2025.07(09).06

Real - Time Data Ingestion Techniques

Sangoju Rajeswari

Data Engineering & Cloud Technologies, USA

Abstract: ***Real-time data ingestion from diverse sources became an essential for businesses seeking to capture and analyze large volumes of data for immediate insights. Apache Flume, an open-source, distributed service designed for collecting, aggregating, and transporting streaming data, provides an efficient solution for real-time data ingestion in big data processing environments. This article explores the role of Apache Flume in enabling businesses to efficiently capture data from various sources, and Ingest it to big data processing systems.***

Keywords: Data Ingestion, Big Data, Apache Flume, Hadoop Eco system

1. Introduction

Apache Flume has emerged as one of the most widely used open-source solutions for real-time data ingestion in big data ecosystems. Designed to be highly reliable, scalable, and fault-tolerant, Flume is well-suited for collecting, aggregating, and transporting data from various sources into distributed storage and processing systems. It provides a flexible architecture that supports integration with other big data frameworks like Apache Hadoop, Apache Kafka, and Apache Spark, enabling seamless data flow and real-time analytics.

2. Hadoop Ecosystem Overview

The Hadoop ecosystem [1] is a collection of open-source tools and frameworks designed for scalable storage, processing, and analysis of large datasets. It is centered around Hadoop Distributed File System (HDFS) and MapReduce for data storage and processing, and is highly extensible, allowing integration with other components to manage, access, process, and analyze big data.

3. Key Components of the Hadoop Ecosystem

The key components are provided in the journal [2]. These components work together in the Hadoop ecosystem to create a powerful and flexible environment for handling big data.

4. Apache Flume Architecture

Flume [3] is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and fault tolerant with tunable reliability mechanisms and many failover and recovery mechanisms. It uses a simple extensible data model that allows for online analytic application.

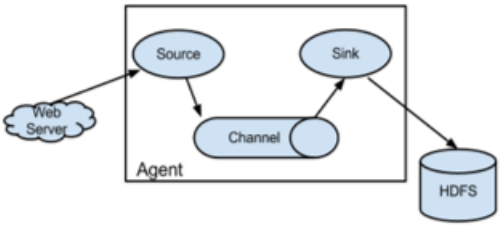

Figure 1: Apache Flume Architecture

An Event is a unit of data that flows through a Flume agent. The Event flows from Source to Channel to Sink, and is represented by an implementation of the Event interface. An Event carries a payload (byte array) that is accompanied by an optional set of headers (string attributes). A Flume agent is a process (JVM) that hosts the components that allow Events to flow from an external source to an external destination.

A Source [4] consumes Events having a specific format, and those Events are delivered to the Source by an external source like a web server. For example, an AvroSource can be used to receive Avro Events from clients or from other Flume agents in the flow. When a Source receives an Event, it stores it into one or more Channels. The Channel is a passive store that holds the Event until that Event is consumed by a Sink. One type of Channel available in Flume is the FileChannel which uses the local file system as its backing store. A Sink is responsible for removing an Event from the Channel and putting it into an external repository like HDFS (in the case of an HDFSEventSink) or forwarding it to the Source at the next hop of the flow. The Source and Sink within the given agent run asynchronously with the Events staged in the Channel.

5. Implementation of Real time Data ingestion using Apache Flume

As discussed in section 4 (Apache Flume Architecture), there are different types of data sources but based on the need and use case respective source can be used for Data ingestion in Big Data Environments. We will be using access logs as source for this implementation and this log can be streamed from other online sources.

```
-rw-r--r-- 1 cloudera cloudera 34505 Aug  1  2014 /opt/gen_logs/logs/access.log
```

Sample log file is available in local file system.

Figure 2: Access logs source data file

Figure 3: Access logs source content

The below code captures the data from local file system (source) and ingested into HDFS Big Data environment(sink) by means of a channel using the Apache architecture frame work.

```
# Describ/configure  sources r1
a1.sources = r1
a1.sources.r1.type = exec
a1.sources.r1.command = tail -F /opt/gen_logs/logs/access.log

a1.channels = c1

# Use a channel which buffers events to a file
# The component type name, needs to be FILE.
a1.channels.c1.type = FILE

# The maximum size of transaction supported by the channel
a1.channels.c1.capacity = 20000
a1.channels.c1.transactionCapacity = 1000

# Amount of time (in millis) between checkpoints
a1.channels.c1.checkpointInterval 3000

# Max size (in bytes) of a single log file
a1.channels.c1.maxFileSize = 2146435071

# Describe the sink
a1.sinks.k1.type = hdfs
a1.sinks.k1.channel = c1
a1.sinks.k1.hdfs.path = /user/cloudera/flume/%y-%m-%d
a1.sinks.k1.hdfs.filePrefix = flume-%y-%m-%d
a1.sinks.k1.hdfs.rollSize = 1048576
a1.sinks.k1.hdfs.rollCount = 100
a1.sinks.k1.hdfs.rollInterval = 120
a1.sinks.k1.hdfs.fileType = DataStream
a1.sinks.k1.hdfs.idleTimeout = 10
a1.sinks.k1.hdfs.useLocalTimeStamp = true

# Bind the source and sink to the channel
a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
a1.sinks = k1
```

Figure 4: Apache Flume Code

Flume process is executed by calling agent (a1) and respective configuration file.

```
[cloudera@quickstart conf]$ flume-ng agent --name a1 --conf /home/cloudera/flume/conf --conf-file /home/cloudera/flume/conf/reallogs.conf
```

Figure 5: Flume Execution

6. Results

Apache Flume created below directories (flume and run date) and as part of this HDFS file is created in Big Data Environment.

```
drwxr-xr-x   - cloudera cloudera          0 2025-02-07 23:50 /user/cloudera/flume/25-02-07
```

Figure 6: HDFS Directory/File

HDFS file contents are displayed below:

```
-rw-r--r--   1 cloudera cloudera       1981 2025-02-07 23:50 /user/cloudera/flume/25-02-07/flume-25-02-07.1739001005723
```

Figure 7: Output HDFS file content

7. Advantages of Apache Flume for Real-Time Data Ingestion

Flume has several advantages [5] for real-time data ingestion. Some of the key benefits of using Apache Flume are:

a) Compatibility: Apache Flume integrates seamlessly with most distributions of the Hadoop framework, allowing interaction with various technologies.
b) Fault Tolerance: In the event of detecting faulty components, Flume utilizes backup components that automatically replace them, preventing service interruptions.
c) Performance: Being a distributed solution, Flume achieves excellent levels of performance and scalability. Businesses with complex information systems that handle thousands of events per second can effectively utilize this tool.
d) Accessibility: Apache Flume is a SaaS (Software as a Service) tool, making it compatible with all operating systems, including Windows, Mac, and mobile OS. This accessibility enables users to access it from any web browser.

8. Conclusion

As businesses continue to generate increasing amounts of real-time data, Apache Flume's role in facilitating seamless data ingestion will grow. Integration of Apache Flume with machine learning and edge computing enhances stream processing Flume's in achieving scalability and efficiency in next-generation big data architectures. By leveraging

Apache Flume, organizations can unlock the full potential of their real-time data, gaining actionable insights and improving operational outcomes with speed and precision.

References

[1] Efficient Data Processing with Apache Spark, http://dx.doi.org/10.2139/ssrn.5023456
[2] Data Streaming with Apache Spark, doi.org/10.55041/IJSREM41137
[3] https://flume.apache.org/FlumeDeveloperGuide.html
[4] Flume Sources https://www.cloudduggu.com/flume/source/
[5] Flume Advantages https://datascientest.com/en/understanding-apache-flume-its-purpose-and-applications