Apache Sqoop

Apache Sqoop

- A command line interface (web in Sqoop 2)
- A tool designed to transfer data b/w Hadoop and relational databases.
  - Transform data in Hadoop with MapReduce or Hive
  - Export data back to RDB.
- Written in Java
- Licensed by Apache

Sqoop Under the Hood

- The dataset being transferred is sliced up into different partitions
  - A map only job is launched with individual mappers responsible for transferring a slice into dataset
- Each record of data is handled in a type safe manner since Sqoop uses the database metadata to infer the data types.

Apache Flume

Apache Flume

- Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data.
- Has a simple and flexible architecture based on streaming data flows.
- Open Source, scalable. Manageable, Fault Tolerant

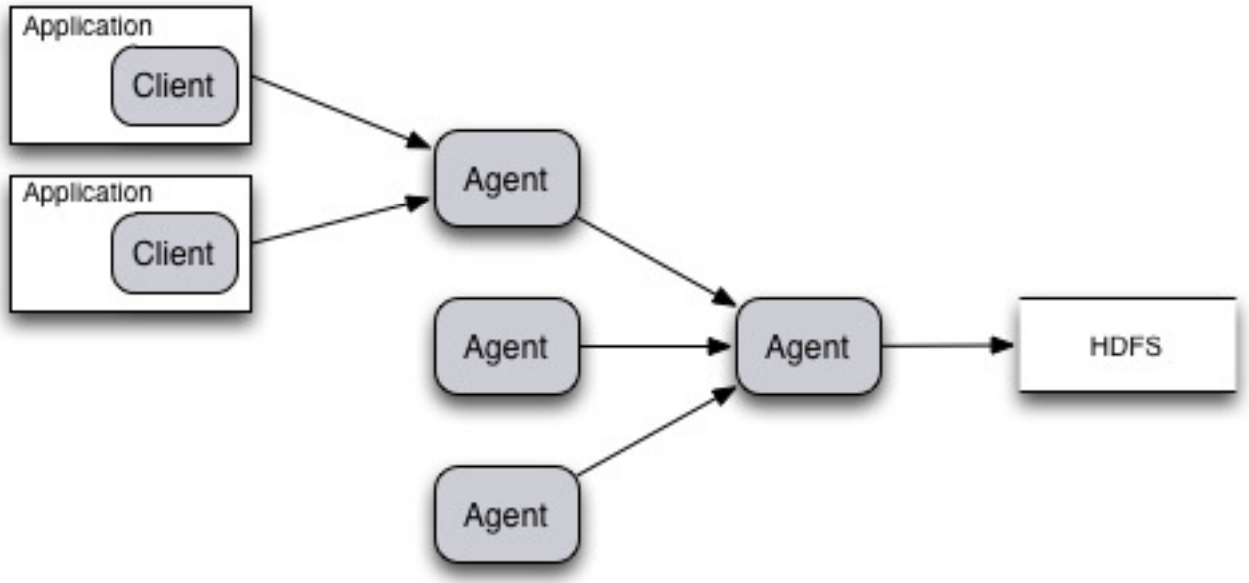
Core Concepts

- Event -- *An Event is the fundamental unit of data transported by Flume from its point of origination to its final destination. Event is a byte array payload accompanied by optional headers.*

- ***Client --*** *An entity that generates events and sends them to one or more Agents.*

- ***Agent --*** *A container for hosting Sources, Channels, Sinks and other components that enable the transportation of events from one place to another.*

Typical Aggregation Flow



Source

An active component that receives events from a specialized location or mechanism and places it on one or Channels.

- Different Source types:
  - Specialized sources for integrating with well-known systems. Example: Syslog, Netcat
  - Auto-Generating Sources: Exec, SEQ
  - IPC sources for Agent-to-Agent communication: Avro
- Require at least one channel to function

Channels

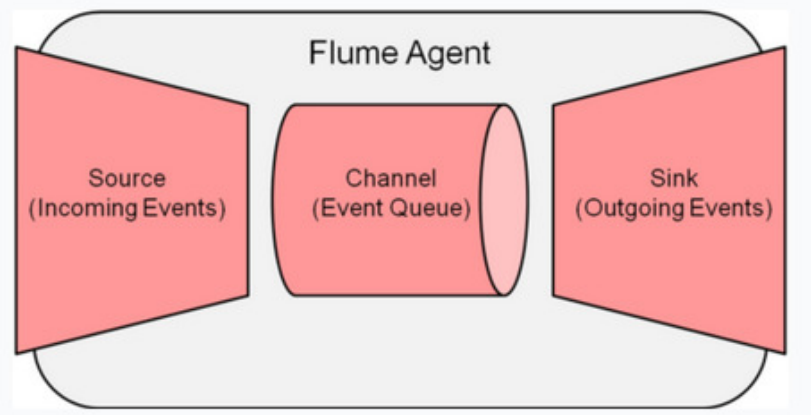
A passive component that buffers the incoming events until they are drained by Sinks.

- Different Channels offer different levels of persistence:
  - Memory Channel: volatile
  - File Channel: backed by WAL implementation
  - JDBC Channel: backed by embedded Database
- Channels are fully transactional
- Provide weak ordering guarantees
- Can work with any number of Sources and Sinks.

Sink

An active component that removes events from a Channel and transmits them to their next hop destination.

- Different types of Sinks:
  - Terminal sinks that deposit events to their final destination. For example: HDFS, HBase
  - Auto-Consuming sinks. For example: Null Sink
  - IPC sink for Agent-to-Agent communication: Avro
- Require exactly one channel to function



Configuration

agent1.properties:

```
# Active components
agent1.sources = src1
agent1.channels = ch1
agent1.sinks = sink1

# Define and configure src1
agent1.sources.src1.type = netcat
agent1.sources.src1.channels = ch1
agent1.sources.src1.bind = 127.0.0.1
agent1.sources.src1.port = 10112

# Define and configure sink1
agent1.sinks.sink1.type = logger
agent1.sinks.sink1.channel = ch1

# Define and configure ch1
agent1.channels.ch1.type = memory
```

Active Agent Components

(Sources, Channels, Sinks)

Individual Component Configuration

Configuration

- A configuration file can contain configuration information for many Agents
- Only the portion of configuration associated with the name of the Agent will be loaded
- Components defined in the configuration but not in the active list will be ignored
- Components that are misconfigured will be ignored
- Agent automatically reloads configuration if it changes on disk

Thank You