Flume

Fernando Anselmo
http://fernandoanselmo.orgfree.com/wordpress/

Versão 1.0 em 19 de dezembro de 2022

Resumo

Flume[1] ou "Apache Flume"(aqui chamarei apenas de Flume) é parte do Ecosistema Hadoop criado para ser um sistema confiável e distribuído para coletar, agregar e mover grandes quantidades de dados de registros (Log), eventos, dados de redes sociais ou sensores para um armazenamento de dados centralizados como o **HBase** ou **HDFS** para posterior análise. Possui uma arquitetura simples, porém flexível, baseada em fluxos de dados de streaming.

1 Parte inicial

Flume é uma ferramenta para alimentação de dados em HDFS. Acumula, integra e transporta grandes quantidades de dados de *streaming*, como arquivos de log, eventos de diferentes origens, como tráfego de rede, mensagens de e-mail para o **HDFS**. Flume é altamente confiável e distribuído.

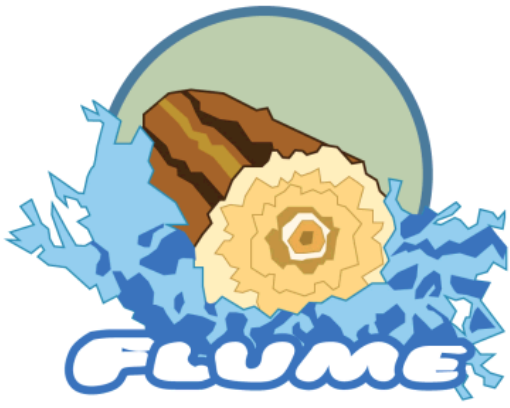

Figura 1: *Logo do Apache Flume*

Flume tem um design flexível baseado em fluxos de dados de *streaming*. É tolerante a falhas e robusto com vários *failovers* (tratamento de exceção) e mecanismos de recuperação. O **Flume Big Data** tem diferentes níveis de confiabilidade para oferecer, que incluem *best-effort delivery* (entrega com o melhor esforço) e *end-to-end delivery* (entrega de ponta a ponta). *Best-effort delivery* não tolera nenhuma falha de nó do Flume, enquanto o modo *end-to-end delivery* garante a entrega mesmo no caso de várias falhas de nó.

Flume transporta dados entre fontes e coletores. Essa coleta de dados pode ser agendada ou orientada a eventos. O Flume tem seu próprio mecanismo de processamento de consultas que facilita a transformação de cada novo lote de dados antes de serem movidos para o coletor pretendido.

Possíveis coletores Flume incluem **HDFS** e **HBase**. O Flume Hadoop também pode ser usado para transportar dados de eventos, incluindo, entre outros, dados de tráfego de rede, dados gerados por sites de mídia social e mensagens de e-mail.

Estes são os benefícios que o tornam uma alternativa melhor em relação aos outros:

- É tolerante a falhas, confiável e escalável.
- Pode armazenar dados em locais centralizados como HBase e HDFS.
- Escalável horizontalmente.
- Se a taxa de leitura ultrapassar a taxa de gravação, o Flume fornecerá um fluxo constante de dados entre a leitura e a gravação de dados.
- A entrega de mensagens é confiável. As transações são baseadas em canal; para cada mensagem, duas transações são mantidas.
- Sustenta um conjunto abrangente de tipos de origens e destinos.
- Dados de várias fontes podem ser ingeridos no Hadoop.

2 Arquitetura do Flume

Um agente Flume é um processo independente que roda por trás de uma JVM (*Java Virtual Machine*), possui 3 componentes: **Flume Source**, **Flume Channel** e **Flume Sink**. Através dos quais os eventos se propagam após serem iniciados em uma fonte externa:

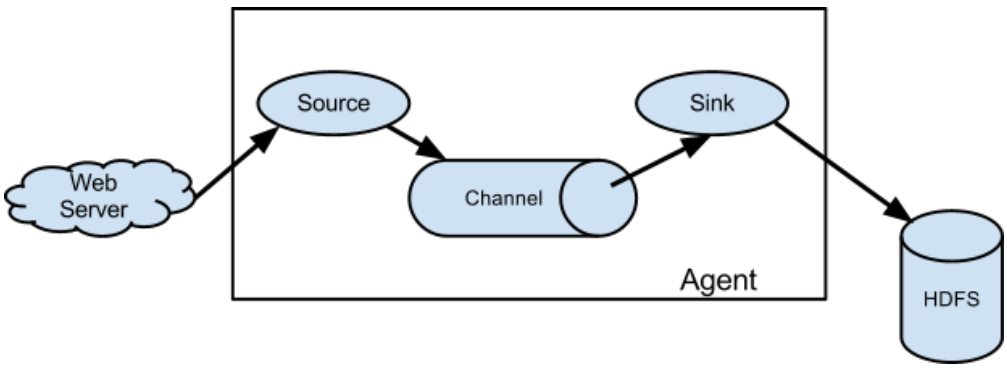

Figura 2: *Arquitetura do Apache Flume*

Na imagem vemos como os componentes se comportam:

- Os eventos gerados pela fonte externa (*WebServer*) são consumidos pelo **Flume Source**. A fonte externa envia eventos para a fonte Flume em um formato que é reconhecido pela fonte de destino.
- **Flume Source** recebe um evento e o armazena em um ou mais canais através do **Flume Channel**. Atua como um armazenamento transitório e mantém o evento até que seja

consumido pelo **Flume Sink**. Este também pode utilizar um sistema de arquivos local para armazenar esses eventos.

- **Flume Sink** remove o evento de um canal e o armazena em um repositório externo como, por exemplo, **HDFS**. Pode haver vários agentes do Flume, caso no qual **Flume Sink** encaminha este evento para o próximo agente no fluxo.

Definições do Flume:

- **Event** (evento) - É a menor unidade de dados que transita no Flume.
- **Header** (cabeçalho) - É a parte opcional de *Event* ou do dado em si (denominado de *Payload*).

Figura 3: *Definições da Arquitetura do Flume*

Exemplos de soluções com o Flume:

- **Facebook Scribe** - **Scribe** é uma ferramenta usada para agregar e transmitir dados de log. Projetado para dimensionar um número muito grande de nós e extremamente robusto a falhas de nós e de rede.
- Apache Kafka - desenvolvido pela **ASF** (*Apache Software Foundation*) como um agente de mensagens de código aberto. Com o Kafka podemos lidar com *feeds* através de alta transferência (*high-throughput*) e baixa latência.

3 Hadoop no Docker

O modo mais simples de se conseguir trabalhar com o Hadoop é utilizando o Docker, para baixar a imagem do Hadoop:

```
$ docker pull suhothayan/hadoop-spark-pig-hive:2.9.2
```

Nessa imagem temos outros produtos do Ecossistema Hadoop: Spark, Pig e Hive. Para criar e executar a primeira vez o contêiner (a pasta que este comando for executado será associada a uma pasta interna chamada **/home/tsthadoop**):

```
$ docker run -it -d --name meu-hadoop -v $(pwd):/home/tsthadoop
suhothayan/hadoop-spark-pig-hive:2.9.2
```

Uma vez interrompido o contêiner:

```
$ docker stop meu-hadoop
```

Podemos executá-lo novamente com os seguintes comandos:

```
$ docker start meu-hadoop
$ docker exec -it meu-hadoop /etc/bootstrap.sh bash
```

3.1 Erro de Permissão

Na primeira vez que entramos é dado um erro na execução do script "bootstrap.sh"de permissão negada para executar o script "spark-env.sh", vamos corrigir isso com o comando:
`# chmod 777 /usr/local/spark/conf/spark-env.sh`

Vamos sair do bash:
`# exit`

Podemos executá-lo novamente:
`$ docker exec -it meu-hadoop /etc/bootstrap.sh bash`

E o erro desapareceu.

E podemos verificar a versão do Hadoop está instalado:
`$ hadoop version`

3.2 Instalar o Flume

Se leu corretamente essa imagem NÃO POSSUI o **Flume** e agora realmente começa nossa instalação, optei por utilizar essa imagem por já possuir todos os componentes necessários para posterior tratamento dos dados. Desta forma ao invés de realizar um trabalho de obter uma imagem somente com o Flume (que existem várias por aí e ter que instalar todos os outros componentes), vamos utilizar uma imagem que estamos acostumados e que já possui todas as outras ferramentas necessárias e nos preocuparmos apenas com o **Flume**.

Outros ganhos adicionais desse método, é ser excelente para aprendermos como adicionar uma ferramenta ao ecossistema do **Hadoop**, além de não ficarmos presos a uma determinada versão. E agora podemos "brincar"melhor com as outras ferramentas como o **Spark**, **Hive** e **Pig** que já estão nessa imagem (e se já utilizou o tutorial sobre **Sqoop** verá que este também já se encontra instalado).

No endereço oficial [1] existe o link para *download*, e ao acessar este temos as versões disponíveis do **Flume**. Utilizaremos a última até o momento, clicar no link "1.9.0"e baixar o arquivo **apache-flume-1.9.0-bin.tar.gz** e colocá-lo na pasta associativa da imagem.

Retornar a pasta raiz do root:
`# cd root/`

Descompactar o arquivo:
`# tar vfx /home/tsthadoop/apache-flume-1.11.0-bin.tar.gz`

Mover o pasta descompactada para o endereço local (aonde estão os outros produtos do Ecossistema):
`# mv apache-flume-1.11.0-bin /usr/local/.`

Criar um link simbólico para facilitar o acesso:
```
# cd /usr/local/
# ln -s /usr/local/apache-flume-1.11.0-bin flume
```

Modificar o arquivo "etc/bootstrap.sh": `# vim /etc/bootstrap.sh`

E adicionar as seguintes linhas abaixo dos comandos EXPORT:

```
export FLUME_HOME="/usr/local/flume"
export PATH=$PATH:/usr/local/sqoop/bin:/usr/local/flume/bin
```

Aproveitamos e inserimos o caminha do path para a pasta do Flume (considerando que já existe a do Sqoop). Pressionar "ESC"para sair do modo de edição e salvar o arquivo com ":wq". Sair da imagem, parar e iniciar o contêiner, entrar no bash novamente. Acessar a pasta associativa:
`# cd /home/tsthadoop`

Verificar se o Flume está correto:
`# flume-ng version`

3.3 Trazer os dados

Toda a brincadeira com o Flume se resume a um simples arquivo contendo os dados que ele irá levar diretamente para o HDFS, consideremos para este exemplo um simples arquivo texto contido na pasta associativa chamado "familia.txt", com o seguinte conteúdo:

```
Fernando,Anselmo
Jessica,Rabbit
Robin,Batman
Hugo,Anselmo
Roger,Rabbit
Fox,Rabbit
Bruce,Batman
```

Nosso primeiro passo é criar um agente de configuração que observe esses dados, para tanto, na pasta "/usr/local/flume/conf"criamos um arquivo chamado "familia.conf": `# vim familia.conf`

E definimos as seguintes características:

```
agent1.sources = tail
agent1.channels = Channel-2
agent1.sinks = sink-1

agent1.sources.tail.type = exec
agent1.sources.tail.command = cat /home/tsthadoop/familia.txt
agent1.sources.tail.channels = Channel-2

agent1.channels.Channel-2.type = memory

agent1.sinks.sink-1.channel = Channel-2
agent1.sinks.sink-1.type = hdfs
agent1.sinks.sink-1.hdfs.path = hdfs://localhost:9000/flume01
agent1.sinks.sink-1.hdfs.fileType = DataStream
agent1.sinks.sink-1.hdfs.useLocalTimeStamp = false
agent1.sinks.sink-1.hdfs.rollInterval = 60
agent1.sinks.sink-1.hdfs.round = true
agent1.sinks.sink-1.hdfs.roundValue = 10
agent1.sinks.sink-1.hdfs.roundUnit = second
```

Pressionar "ESC"para sair do modo de edição e salvar o arquivo com ":wq". Executamos o agente com o comando:

```
# flume-ng agent {conf /usr/local/flume/conf/ -f /usr/local/flume/conf/familia.conf
-n agent1 -Dflume.root.logger=DEBUG,console
```

Esse comando executa o Flume para verificar o arquivo e trazer os dados contidos nele, ao recebermos uma mensagem tal como: `2022-12-19T19:44:05,631 INFO [hdfs-sink-1-call-runner-3] org.apache.flume.sink.hdfs.BucketWriter - Renaming hdfs://localhost:9000/flume01/FlumeData.1 to hdfs://localhost:9000/flume01/FlumeData.1671475952091`

Significa que os dados já foram importados, damos um CTRL+C para interromper. Agora vamos ver se está tudo correto com o seguinte comando para verificar as pastas do HDFS:
`# hdfs dfs -ls /`

Que mostra algo como:
```
Found 3 items
drwxr-xr-x - root supergroup 0 2022-12-19 18:53 /flume01
drwx-wx-wx - root supergroup 0 2019-07-21 16:09 /tmp
drwxr-xr-x - root supergroup 0 2019-07-21 16:09 /user
```

Se verificarmos a pasta criada pelo Flume:
`# hdfs dfs -ls /flume01`

Teremos como resposta (obviamente o número final no arquivo pode variar):
```
Found 1 items
-rw-r--r-- 1 root supergroup 95 2022-12-19 18:53 /flume01/FlumeData.1671475952091
```

E podemos verificar seu conteúdo:
`# hdfs dfs -cat /flume01/FlumeData.1671475952091`

Que é exatamente igual aos dados que temos no arquivo original. Mais detalhes sobre as configurações dos canais podem ser acessadas aqui: `https://flume.apache.org/releases/content/1.6.0/FlumeUserGuide.html`

Mais fontes de informação podem ser obtidas em diversos sites que apresenta tutoriais completos sobre o Apache Flume como a Tutorials Point[2].

4 Conclusão

Flume é uma ferramenta no ecossistema Hadoop que fornece recursos para coletar, agregar e trazer de forma eficiente grandes quantidades de dados para o Hadoop. Exemplos de grandes quantidades de dados são dados de log, dados de tráfego de rede, dados de mídia social, dados de geolocalização, dados de sensores e máquinas e dados de mensagens de e-mail.

Fornece vários recursos para gerenciar dados e permite que os usuários absorvam de várias fontes de dados no Hadoop. Protege os sistemas contra picos de dados quando a taxa de entrada de dados excede a taxa na qual os dados são gravados. O Flume NG garante a entrega de dados usando transações baseadas em canal. É dimensionado horizontalmente para processar mais fluxos de dados e volumes de dados.

Geralmente é usado para dados de log. O Flume pode obter dados de várias fontes como Avro, Syslog's e arquivos e os entrega para vários destinos como Hadoop HDFS ou HBase. Por ser basicamente um sistema distribuído usado para agregar os arquivos em um único local. Em termos

simples, é usado para mover os dados de um local para outro de maneira confiável e eficiente. Possui recursos integrados, como mecanismos de confiabilidade, failover e recuperação, e é tolerante a falhas. É um componente importante no ecossistema Hadoop.

Sou um entusiasta do mundo **Open Source** e novas tecnologias. Qual a diferença entre Livre e Open Source? Livre significa que esta apostila é gratuita e pode ser compartilhada a vontade. Open Source além de livre todos os arquivos que permitem a geração desta (chamados de arquivos fontes) devem ser disponibilizados para que qualquer pessoa possa modificar ao seu prazer, gerar novas, complementar ou fazer o que quiser. Os fontes da apostila (que foi produzida com o LaTex) está disponibilizado no GitHub [5]. Veja ainda outros artigos que publico sobre tecnologia através do meu Blog Oficial [3].

Referências

[1] Página do Apache Flume
`https://flume.apache.org`

[2] Tutorials Point sobre Flume
`https://www.tutorialspoint.com/apache_flume/index.htm`

[3] Fernando Anselmo - Blog Oficial de Tecnologia
`http://www.fernandoanselmo.blogspot.com.br/`

[4] Encontre essa e outras publicações em
`https://cetrex.academia.edu/FernandoAnselmo`

[5] Repositório para os fontes da apostila
`https://github.com/fernandoans/publicacoes`