



Data Science Academy

www.datascienceacademy.com.br

Data Lake – Design, Projeto e Integração

Executando Aplicação de Processamento
em Real-Time com Apache Storm



Nós passamos pelos principais detalhes técnicos do Apache Storm e agora é hora de codificar um cenário simples, usando linguagem Java.

Não se preocupe se você não souber programar Java. Embora esse skill seja importante, os exemplos aqui fornecidos podem ser executados sem maiores Dificuldades, seguindo o passo a passo a seguir.

Cenário - Analisador de Registros de Chamadas Móveis

A chamada móvel e sua duração serão fornecidas como entrada para o Apache Storm e o Storm processará e agrupará as chamadas entre o mesmo chamador e o receptor e o número total de chamadas. Em exemplo típico de processamento de dados em tempo real.

O bico é um componente usado para geração de dados. Basicamente, um bico irá implementar uma interface `IRichSpout`. A interface “`IRichSpout`” possui os seguintes métodos importantes -

Em nosso cenário, precisamos coletar os detalhes do registro de chamadas. A informação do registro de chamadas contém.

- número do chamador
- número do receptor
- duração

Como não temos informações em tempo real dos registros de chamadas, geraremos registros de chamadas fictícios, usando a classe `Random` em Java. O código completo do programa você encontra no arquivo **FakeCallLogReaderSpout.java** em anexo (o arquivo pode ser aberto em um editor de texto como o Sublime Text). Esse script será usado como nosso Spout.

Bolt é um componente que recebe tuplas como entrada, processa cada tupla e produz novas tuplas como saída. Os Bolts irão implementar a interface `IRichBolt`. Neste programa, duas classes de Bolts, `CallLogCreatorBolt` e `CallLogCounterBolt`, são usadas para executar as operações. Você encontra o



código dos Bolts nos arquivos em anexo: CallLogCreatorBolt.java e CallLogCounterBolt.java.

Com Spout e Bolts definidos, precisamos agora criar a topologia. A topologia do Storm é basicamente uma estrutura do Thrift. A classe TopologyBuilder fornece métodos simples e fáceis para criar topologias complexas. A classe TopologyBuilder tem métodos para definir spout (setSpout) e set bolt (setBolt). Finalmente, o TopologyBuilder cria a topologia. E para isso temos o programa LogAnalyserStorm.java.

Para executar a aplicação, siga os passos abaixo:

- 1- Crie a pasta /home/hadoop/app no node master do cluster Storm e copie os 4 arquivos Java em anexo para esta pasta.
- 2- Compile a aplicação executando o comando abaixo:

```
javac -cp "/opt/storm/lib/*" *.java
```

Obs: Não copie e cole o comando. Digite-o.

- 3- Execute a aplicação com o comando abaixo:

```
java -cp "/opt/storm/lib/*":. LogAnalyserStorm
```

Depois que a aplicação for iniciada, ela exibirá os detalhes completos sobre o processo de inicialização do cluster, o processamento do Spout e do Bolt e, finalmente, o processo de encerramento do cluster, além da contagem de registros, nosso objetivo final.