

Tutorial Hadoop - HBase

O que é o Hadoop?

O Hadoop é uma plataforma utilizada para realização de projetos distribuídos, que tratam de grandes quantidades de dados. É uma plataforma voltada para cluster desenvolvida com a linguagem de programação Java.

A arquitetura do Hadoop consiste de um pacote chamado Hadoop common, que contém todas as bibliotecas (.jar) necessários para realizar o acesso ao sistema de arquivos do Hadoop (HDFS), que é um sistema de arquivos distribuído, escalável e portátil.

Um Cluster do Hadoop é formado por um nó mestre(master) e vários nós escravos(slaves).

Nesse tutorial estaremos explicando como criar um cluster com 1 master e 2 slaves.

O que é o HBase?

O HBase é o banco de dados desenvolvido para o Hadoop. Ele é utilizado principalmente quando se tem grande quantidade de dados e tabelas extensas com muitos atributos e muitos dados armazenados.

Instalando o Hadoop

Recomenda-se iniciar a instalação do Hadoop para que este seja executado em apenas uma máquina, e só após essa etapa ser finalizada, realizar a configuração para que o Hadoop execute em várias máquinas.

No site da IBM, existe um tutorial que auxilia na instalação e configuração da distribuição Cloudera. Esse tutorial terá suas etapas mais importantes descritas no presente documento.

Inicialmente, deve-se possuir a tecnologia Java (JDK 1.6 ou mais recente) e o aplicativo cURL instalados. Para instalar essas aplicações pode-se utilizar as referências encontradas ao final desse documento, na seção Bibliografia. O tutorial segue sua instalação na plataforma Linux, na distribuição Ubuntu.

Download e Instalação

Crie um arquivo `/etc/apt/sources.list.d/cloudera.list` e inclua os seguintes comandos nesse arquivo:

```
deb http://archive.cloudera.com/debian intrepid-cdh3 contrib
deb-src http://archive.cloudera.com/debian intrepid-cdh3 contrib
```

Logo após, abra o terminal e instale a distribuição Cloudera, digitando o seguinte comando:

```
sudo apt-get install hadoop-0.20-conf-pseudo
```

Configure também a opção para utilizar o ssh sem senha:

```
sudo su -  
ssh-keygen -t dsa -P "" -f ~/.ssh/id_dsa  
cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

Iniciando o Hadoop

Deve-se formatar o sistema de arquivos do Hadoop (HDFS), utilizando o comando:

```
hadoop-0.20 namenode -format
```

Após formatar, deve-se iniciar os daemons, que são interfaces simples que auxiliam o usuário a utilizar qualquer programa sobre o Hadoop. Na configuração pseudo-distribuída (simulando várias máquinas), são inicializados 5 daemons: namenode, secondarynamenode, datanode, jobtracker e tasktracker. A inicialização desses daemons pode ser realizada executando os seguintes scripts:

```
/usr/lib/hadoop-0.20/bin/start-dfs.sh  
/usr/lib/hadoop-0.20/bin/start-mapred.sh
```

Aplicação Distribuída

Ao executar todos os daemons numa só máquina, não é possível visualizar o paralelismo que o Hadoop oferece. Para que isso seja possível, deve-se executar os daemons namenode, secondary namenode e jobtracker na máquina (nó) principal. Esses daemons proporcionam gerenciamento e coordenação do conjunto de computadores (cluster) onde o Hadoop será executado. Os nós escravos devem executar os daemons tasktracker e datanode, que implementam as funções de armazenamento do sistema de arquivos do Hadoop (HDFS) e da funcionalidade MapReduce (processamento de dados). Para isso, deve-se criar um novo arquivo de configuração, baseada no anterior:

```
sudo cp -r /etc/hadoop-0.20/conf.empty /etc/hadoop-0.20/conf.dist
```

O arquivo criado, conf.dist, será editado para criar a nova configuração distribuída. Para ativar a nova configuração (que ainda não foi modificada), digite o comando:

```
sudo update-alternatives --install /etc/hadoop-0.20/conf hadoop-0.20-conf \  
/etc/hadoop-0.20/conf.dist 40
```

Se for utilizado um ambiente criado por máquinas virtuais, esse é o ponto correto para que o disco rígido virtual seja copiado e transferido para as outras máquinas. Em todos os nós que fazem parte do cluster, deve-se editar o arquivo /etc/hosts indicando os endereços IP dos nós que fazem parte do cluster do Hadoop da seguinte forma:

```
master 192.168.108.133  
slave1 192.168.108.134  
slave2 192.168.108.135
```

No nó principal, deve-se adicionar no arquivo /etc/hadoop-0.20/conf.dist/masters o identificador que diz que esse será o nó principal:

master

E em seguida, adicione da mesma forma os identificadores para os nós escravos, em suas respectivas máquinas:

slave1

slave2

Configuração Específica

Agora, deve ser realizada a configuração específica do Hadoop no diretório /etc/hadoop-0.20/conf.dist, em todas as máquinas que fazem parte do cluster. Inicialmente, deve-se identificar o HDFS principal no arquivo core-site.xml, onde define-se a porta e host do namenode:

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://master:54310</value>
    <description>The name and URI of the default FS.</description>
  </property>
</configuration>
```

Em seguida, deve-se identificar o jobtracker MapReduce, que deve ser colocado apenas no nó principal. Deve-se editar o arquivo mapred-site.xml da seguinte forma:

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>master:54311</value>
    <description>Map Reduce jobtracker</description>
  </property>
</configuration>
```

Por fim, deve-se definir o valor de replicação dos dados, que, normalmente, não ultrapassa o valor 3. No caso do exemplo, o melhor valor seria 2, já que existem apenas 2 nós escravos. Esse valor é definido em hdfs-site.xml:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
    <description>Default block replication</description>
  </property>
</configuration>
```

Existem mais configurações disponíveis, porém, as que foram realizadas acima são suficientes para que essa aplicação distribuída funcione corretamente.

Iniciando o Hadoop

O primeiro passo é formatar o namenode, que é o nó principal do HDFS. Em seguida, deve-se iniciar os daemons do Hadoop, como na configuração pseudo-distribuída, discutida anteriormente. Esses comandos devem ser executados no nó principal:

```
sudo su -  
hadoop-0.20 namenode -format  
/usr/lib/hadoop-0.20/bin/start-dfs.sh
```

Ao utilizar esses comandos, os daemons namenode, secondary namenode e datanode serão iniciados corretamente no nó principal e em seus escravos. Em seguida, deve-se iniciar os daemons jobtracker e tasktracker, utilizando o seguinte comando no nó principal:

```
/usr/lib/hadoop-0.20/bin/start-mapred.sh
```

Assim, em poucos comandos, é possível instalar e configurar o Hadoop no Ubuntu.

Instalando o HBase

O HBase possui 3 modos de funcionamento: standalone, pseudo-distributed e fully-distributed. Para o nosso caso utilizaremos o modo fully-distributed.

Qualquer que seja o modo que você esteja utilizando, defina `${HBASE_HOME}` para ser o local da raiz da instalação do HBase, ou seja, em `/user/local/hbase`. Edite `${HBASE_HOME}/conf/hbase-env.sh`. Neste arquivo você pode definir a heapsize do HBase e essas coisas. No mínimo, defina `JAVA_HOME` para apontar para a raiz de sua instalação JAVA.

Como vamos utilizar o modo fully-distributed, precisamos de uma instância do Hadoop Distributed File System (DFS) que é necessária para os modos distribuídos.

Para rodar uma operação fully-distributed em mais de um host, precisamos realizar as configurações para o modo pseudo-distributed e mais algumas configurações adicionais descritas abaixo.

Para configurar o HBase para o modo pseudo-distributed modifique o `${HBASE_HOME}/conf/hbase-site.xml` que precisa ser apontado para a instancia do Hadoop DFS que esta sendo executada. Utilize o `hbase-site.xml` para sobrescrever as propriedades definidas em `${HBASE_HOME}/conf/hbase-default.xml` (o `hbase-default.xml` em si não deve ser modificado).

O `hbase.rootdir` property precisa ser redefinido no `hbase-site.xml` para apontar o HBase ao Hadoop filesystem. Por exemplo, adicionando a propriedade abaixo para o seu `hbase-site.xml` indica que o HBase deve usar o diretório `/hbase` no HDFS na qual o namenode está na porta 9000 em sua maquina local:

```
<configuration>  
...  
<property>  
  <name>hbase.rootdir</name>  
  <value>hdfs://localhost:9000/hbase</value>
```

```

    <description>The directory shared by region servers.
  </description>
</property>
...
</configuration>

```

Observação: Deixe o HBase criar o diretório. Se você não deixar, você receberá um aviso dizendo que o HBase precisa executar uma migração por faltar arquivos esperados pelo HBase no diretório (se você deixar ele os criará para você).

Para executar uma operação fully-distributed em mais de um host, as seguintes configurações adicionais devem ser seguidas.

No hbase-site.xml, modifique o hbase.cluster.distributed para true.

```

<configuration>
...
  <property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
    <description>The mode the cluster will be in. Possible values are
      false: standalone and pseudo-distributed setups with managed Zookeeper
      true: fully-distributed with unmanaged Zookeeper Quorum (see hbase-env.sh)
    </description>
  </property>
...
</configuration>

```

No modo fully-distributed, você provavelmente precisará mudar o seu hbase.rootdir do localhost para o nome do nó que está executando o HDFS NameNode. Também precisará modificar o \${HBASE_HOME}/conf/regionserver. O regionserver file lista todos os hosts rodando na HRegionServers, um host por linha.

Um HBase distribuído depende de um cluster ZooKeeper em execução. Todos os nós e clientes participantes precisam estar aptos a obter o cluster ZooKeeper em execução. O HBase por padrão gerencia um cluster ZooKeeper para você, ou você pode gerencia-lo e apontar o HBase para ele. Para alternar o gerenciamento do ZooKeeper, utilize a variável HBASE_MANAGES_ZK em \${HBASE_HOME}/conf/hbase-env.sh. Essa variável, que é por padrão true, indica ao HBase para parar ou iniciar os ZooKeeper quorum servers junto com o resto dos servers.

Enquanto o HBase gerencia o ZooKeeper cluster, você pode especificar as configurações do ZooKeeper usando sua canônica zoo.cfg file, ou apenas especificar as opções do zooKeeper direto em \${HBASE_HOME}/conf/hbase-site.xml. Todas as opções de configuração do ZooKeeper tem uma propriedade correspondente no HBase hbase-site.xml que é um arquivo XML de configuração chamado hbase.zookeeper.property.OPTION. Por exemplo a configuração do clientPort no

ZooKeeper pode ser mudado pela propriedade `hbase.zookeeper.property.clientPort`. Para visualizar uma lista dessas propriedades olhe o `zoo.cfg` do ZooKeeper. Para os valores padrão do HBase olhe em `${HBASE_HOME}/conf/hbase-default.xml`.

Você também precisa listar os servers que você deseja que o ZooKeeper rode usando o `hbase.zookeeper.quorum` property. Recomenda-se rodar o ZooKeeper quorum de 3, 5 ou 7 máquinas, e dar para cada ZooKeeper server por volta de 1GB de RAM, e se possível, seu próprio disco dedicado.

Para apontar o HBase para um cluster ZooKeeper existente, adicione um `zoo.cfg` devidamente configurado ao CLASSPATH. O HBase irá reconhecer esse arquivo e usa-lo para saber a localização do ZooKeeper. Também defina o `HBASE_MANAGES_ZK` em `${HBASE_HOME}/conf/hbase-env.sh` para `false` então o HBase não irá modificar a configuração do ZooKeeper:

```
...
# Diz ao HBase se ele deve ou não gerenciar uma própria instancia do ZooKeeper.

export HBASE_MANAGES_ZK=false
```

Como um exemplo, para que o HBase gerencie um ZooKeeper quorum nos nós `rs{1,2,3,4,5}.example.com`, delimite a porta 2222 , usando:

`${HBASE_HOME}/conf/hbase-env.sh`:

```
...

# Diz se o HBase deve ou não gerenciar uma própria instancia do Zookeeper.
export HBASE_MANAGES_ZK=true
```

`${HBASE_HOME}/conf/hbase-site.xml`:

```
<configuration>
...
<property>
  <name>hbase.zookeeper.property.clientPort</name>
  <value>2222</value>
  <description>Property from ZooKeeper's config zoo.cfg.
  The port at which the clients will connect.
  </description>
</property>
...
<property>
  <name>hbase.zookeeper.quorum</name>
```

```
<value>rs1.example.com,rs2.example.com,rs3.example.com,rs4.example.com,rs5.example.com</value>
  <description>Comma separated list of servers in the ZooKeeper Quorum.
```

Por exemplo, "host1.mydomain.com,host2.mydomain.com,host3.mydomain.com". Para uma configuração fully-distributed, ele deve ser definido para uma lista inteira de ZooKeeper quorum servers. Se o HBASE_MANAGES_ZK está definido no hbase-env.sh esta é a lista dos servers que irão iniciar e/ou parar no ZooKeeper.

```
</description>
</property>
...
</configuration>
```

Quando o HBase gerencia o ZooKeeper, ele irá iniciar e/ou parar os ZooKeeper servers. Se você quiser fazer isso você mesmo, pode digitar:

```
${HBASE_HOME}/bin/hbase-daemons.sh {start,stop} zookeeper
```

Se você deixou o HBase gerenciar o ZooKeeper para voce, tenha certeza de que você configurou onde os dados são armazenado. Por padrão, eles são armazenados em /tmp que é limpo algumas vezes em live systems. Modifique esta configuração:

```
<property>
  <name>hbase.zookeeper.property.dataDir</name>
  <value>${hbase.tmp.dir}/zookeeper</value>
  <description>Property from ZooKeeper's config zoo.cfg.
  The directory where the snapshot is stored.
</description>
</property>
```

Note que você pode utilizar o HBase desta maneira, para rodar um cluster ZooKeeper, não relacionado com o HBase. Apenas tenha certeza de definir o HBASE_MANAGES_ZK como false se você quer que ele fique ligado, então quando o HBase desligar não vai desligar o ZooKeeper também.

Executando e Confirmando a Instalação

Para executar um cluster distribuido você irá precisar do Hadoop DFS daemos e do ZooKeeper Quorum antes de iniciar o HBase e parar os daemons depois que o HBase for desligado.

Para iniciar ou fechar os daemons Hadoop DFS execute:

```
${HADOOP_HOME}/bin/start-dfs.sh.
```

Inicie seu ZooKeeper cluster e então inicie o HBase com o seguinte comando:

```
${HBASE_HOME}/bin/start-hbase.sh
```

Depois de iniciar o HBase, execute `${HBASE_HOME}/bin/hbase` Shell para obter um Shell onde você poderá executar os comandos. Digite 'help' no prompt do Shell para mostrar uma lista dos comandos. Teste a sua instalação criando tabelas, inserindo dados, exibindo dados e depois dando um drop em suas tabelas. Como por exemplo:

```
hbase> # Type "help" to see shell help screen
hbase> help
hbase> # To create a table named "mylittletable" with a column family of
"mylittletable", type
hbase> create "mylittletable", "mylittletable"
hbase> # To see the schema for you just created "mylittletable" table and its
single "mylittletable", type
hbase> describe "mylittletable"
hbase> # To add a row whose id is "myrow", to the column
"mylittletable:x" with a value of 'v', do
hbase> put "mylittletable", "myrow", "mylittletable:x", "v"
hbase> # To get the cell just added, do
hbase> get "mylittletable", "myrow"
hbase> # To scan you new table, do
hbase> scan "mylittletable"
```

Para parar o HBase, saia do Shell e execute:

```
${HBASE_HOME}/bin/stop-hbase.sh
```

Para operações distribuídas, antes de parar os daemons do Hadoop tenha certeza de que os HBase tenha desligado completamente.

A localização default dos logs é `${HBASE_HOME}/logs`.

O HBase também cria uma UI listing vital attributes. Por padrão ela é implantada no host máster na porta 60010.

Utilizando o Hadoop - HBase

Primeiro certifique-se de que todas as máquinas onde estão seu master e seus slaves estão ligadas. Para verificar se todas as máquinas estão vivas (ou seja, conectadas à rede), abra o terminal da sua máquina master e realize um ping nas máquinas slave.

```
$ ping slave1
```

```
$ ping slave2
```

Caso esteja tudo certo com a conexão das máquinas, você deve iniciar a montagem do sistema de arquivos do Hadoop o HDFS. Não se esqueça de entrar em modo supervisor(super usuário).

Para isso, formate a referência aos arquivos da máquina master, apenas para evitar conflitos.

```
$ hadoop namenode -format
```

Esse comando irá iniciar a formatação das referências do sistema de arquivo, ele irá perguntar se você deseja realmente realizar a operação. Digite "Y" para aceitar.

Feito isso, inicie o Hadoop:

```
$ usr/lib/hadoop/bin/start-dfs.sh
```


Esse comando irá montar todo HDFS no master e nos slaves.

Isso pode levar um tempo. Para verificar se o sistema foi montado execute o comando:

```
$ hadoop fsck /
```

Verifique se o valor de “number of data nodes” é igual a dois. Isso significa que os slaves já foram iniciados e está tudo pronto. Caso não esteja, espera mais algum tempo e verifique novamente.

Agora podemos iniciar o HBase. Para isso execute o comando:

```
$ /usr/lib/hbase/bin/start-hbase.sh
```

Aguarde a finalização da operação e então o banco já estará pronto para ser usado.

Para usá-lo entre em seu shell:

```
$ hbase shell
```

E então pode digitar os comandos para criar tabelas, acessar tabelas, realizar consultas ou qualquer operação que queira realizar no seu banco. Nesse link você encontrará alguns comandos de exemplo <http://brettwalenz.org/2009/08/23/using-the-hbase-shell/>.

Mas cuidado! Caso você digite algum comando errado, ocorrerá um erro no shell, e você precisará iniciar o HBase novamente.

Bibliografia

Instalando o JDK:

<https://docs.cloudera.com/display/DOC/Before+You+Install+CDH3+on+a+Single+Node+BeforeYouInstallCDH3onaSingleNode-InstalltheJavaDevelopmentKitBeforeInstallingCDH3>

Website do cURL:

<http://curl.haxx.se/>

Instalação do Hadoop:

<http://www.ibm.com/developerworks/br/linux/library/l-hadoop-2/index.html>