**Big Data e Apache Hadoop**

**Video no youtube:** **https://www.youtube.com/watch?v=iMo\_glMoDlY**

**O que é big data?**

Área que estuda como tratar, analisar e obter informações a partir de conjuntos de dados grandes demais para serem analisados por sistemas tradicionais. Conceito começou com 3Vs hoje é falado em5Vs:

Volume: conceito de volume é relativo a variável tempo, o que é grande hoje, pode não ser nada amanhã.

Velocidade: Analogia usada: Você cruzaria uma rua vendado se a última informação que tivesse fosse uma fotografia tirada do tráfego circulante de 5 minutos atrás?Provavelmente não, pois a fotografia de 5 minutos atrás é irrelevante, você precisa saber das condições atuais para poder cruzar a rua em segurança.

Variedade: informações diferentes e formas diferentes.

•    Dados estruturados: são armazenados em bancos de dados, sequenciados em tabelas;

•    Dados semi-estruturados: acompanham padrões heterogêneos, são mais difíceis de serem identificados pois podem seguir diversos padrões;

•    Dados não estruturados: são uma mistura de dados com fontes diversificadas como imagens, áudios e documentos online.

Estima-se que até 90% de todos os dados no mundo estão a forma de dados não estruturados. (ICD, 2011).

Veracidade: Dados que condizem com a realidade daquele momento, pois dados passados não podem ser considerados dados verídicos para o momento em que é analisado.

Valor:  o valor agregado de todo esse trabalho desenvolvido, coleta, armazenamento e análise de todos esses dados tem que compensar os custos financeiros envolvidos

**Hadoop:** framework open source para o armazenamento e processamento de dados em larga escala. MapReduce, responsável pelo processamento distribuído, e o Hadoop Distributed File System (HDFS), para armazenamento de grandes conjuntos de dados, também de forma distribuída.

**MAP REDUCE**

Map: Responsável por receber os dados de entrada, estruturados em uma coleção de pares chave/valor. Tal função map deve ser codificada pelo desenvolvedor, através de programas escritos em Java ou em linguagens suportadas pelo Hadoop;

•  Shuffle: A etapa de shuffle é responsável por organizar o retorno da função Map, atribuindo para a entrada de cada Reduce todos os valores associados a uma mesma chave. Esta etapa é realizada pela biblioteca do MapReduce;

•  Reduce: Por fim, ao receber os dados de entrada, a função Reduce retorna uma lista de chave/valor contendo zero ou mais registros, semelhante ao Map, que também deve ser codificada pelo desenvolvedor.

**HDFS:** sistema de arquivos distribuído nativo do Hadoop. Permite o armazenamento e transmissão de grandes conjuntos de dados em máquinas de baixo custo. Possui mecanismos que o caracteriza como um sistema altamente tolerante a falhas.

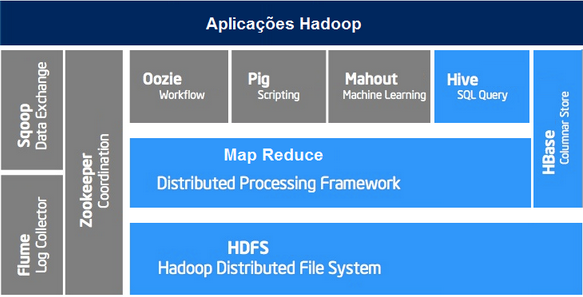
**Componentes adicionais**

Ambari: ferramenta baseada na Web para o suporte, gerenciamento e monitoramento de outros módulos Hadoop, como HDFS, MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig e Sqoop;

Cassandra: banco de dados escalável, com tolerância a falhas.

Mahout: sistema para desenvolvimento de aplicações de aprendizagem de máquina e biblioteca com funções de mineração de dados

**ARQUITETURA HANDOOP**

****

Para funcionar, uma aplicação Hadoop exige no mínimo a utilização das ferramentas da camada de armazenamento (HDFS) e processamento MapReduce. As demais camadas podem ser adicionadas conforme a necessidade, novos subprojetos, que são incorporados como componentes à arquitetura Hadoop, completam a infraestrutura do framework para resolver problemas específicos.

**Exemplo utilizado:**

**Deer Bear River**

**Car Car River**

**Deer Car Bear**





**Comando para instalar o hadoop**

wget http://apache.cs.utah.edu/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz

**Configs para o arquivo bashrc**

export JAVA\_HOME=/usr/lib/jvm/java-8-oracle/

export HADOOP\_INSTALL=/home/gregorioksteinke/hadoop2

export PATH=$PATH:$HADOOP\_INSTALL/bin

export PATH=$PATH:$HADOOP\_INSTALL/sbin

export HADOOP\_MAPRED\_HOME=$HADOOP\_INSTALL

export HADOOP\_COMMON\_HOME=$HADOOP\_INSTALL

export HADOOP\_HDFS\_HOME=$HADOOP\_INSTALL

export YARN\_HOME=$HADOOP\_INSTALL

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_INSTALL/lib/native

export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_INSTALL/lib"

export HADOOP\_CONF\_DIR=$HADOOP\_INSTALL/etc/hadoop

**Atualizar o bashrc**

exec bash

**Adicionar em hadoop-config.sh para configurar o java com o hadoop**

export JAVA\_HOME=/usr/lib/jvm/java-8-oracle/

**Configs para o yarn-env.sh**

export JAVA\_HOME=/usr/lib/jvm/java-8-oracle/

export HADOOP\_HOME=/home/gregorioksteinke/hadoop2

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

export YARN\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

export PATH=$PATH:$HADOOP\_HOME/bin

**Configs para o hadoop-env.sh**

export JAVA\_HOME=/usr/lib/jvm/java-8-oracle/

export HADOOP\_INSTALL=/home/gregorioksteinke/hadoop2

export PATH=$PATH:$HADOOP\_INSTALL/bin

export PATH=$PATH:$HADOOP\_INSTALL/sbin

export HADOOP\_MAPRED\_HOME=$HADOOP\_INSTALL

export HADOOP\_COMMON\_HOME=$HADOOP\_INSTALL

export HADOOP\_HDFS\_HOME=$HADOOP\_INSTALL

export YARN\_HOME=$HADOOP\_INSTALL

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_INSTALL/lib/native

export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_INSTALL/lib"

**Configs para o core-site.xml**

<property>

<name>fs.default.name</name>

<value>hdfs://vm0:8000</value>  <!----vm0 é o nome da minha vm—!>

<final>true</final>

</property>

 <property>

<name>hadoop.proxyuser.hadoop.hosts</name>

<value>localhost</value>

</property>

<property>

<name>hadoop.proxyuser.hadoop.groups</name>

<value>users</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>/ /home/gregorioksteinke/hdfsdrive value>

</property>