Projeto: Criação de um Data Lake na nuvem utilizando a AWS

O presente projeto decorre de uma atividade proposta no curso Engenheiro de Dados da Data Science Academy. A primeira parte consiste na criação e teste de um Data Lake utilizando a AWS. Antes de entrarmos nas especificações técnicas se faz necessárias algumas definições.

Data Lake

É um vasto repositório de uma variedade de dados brutos de toda uma empresa que podem ser adquiridos, processados e analisados. Um Data Lake é uma ferramenta de negócios e resulta de um modelo business – to – business. Podemos citar algumas plataformas famosas para a construção de um Data Lake, como o Hadoop, Hadoop/Spark, bancos de dados NoSQL e DataStores.

Neste projeto utilizaremos o Hadoop/Spark.

AWS

A Amazon Web Services é uma plataforma de serviço de computação em nuvem. Mas o que é um serviço de computação em nuvem?

Para responder essa questão vamos imaginar uma empresa que decide utilizar os dados produzidos por ela mesma e por outras fontes. Essa empresa irá precisar de muitos computadores (servidores) para armazenar os dados. Agora imagine o custo de comprar várias máquinas, preparar o local onde essas máquinas ficarão, segurança física dessas máquinas, controle de temperatura da sala dos servidores, entre outros problemas. Uma alternativa é utilizar um serviço de computação em nuvem onde a empresa terá acesso a máquinas de configurações desejadas (instâncias) sem ter que se preocupar com os custos discutidos anteriormente. Neste caso uma empresa de serviço de computação em nuvem é contratada, por exemplo a AWS, e ela irá disponibilizar a segurança, integridade e disponibilidade dos dados armazenados nas instâncias oferecidas.

Para saber mais sobre a AWS:

https://aws.amazon.com/pt/free/

Cluster

A maneira mais simples de se entender o conceito de cluster é dizer que é a integração de dois ou mais computadores que irão trabalhar juntos para

terminar uma tarefa. Vamos imaginar uma tarefa simples de agrupar e contar as palavras em um texto. Se o texto tiver 30 linhas essa tarefa não parece ser complicada. Agora e se ao invés de um texto nós quiséssemos agrupar e contar as palavras contidas em um livro de mais de mil páginas? Isso exigiria uma máquina com alto poder de processamento e mais tempo do que o desejado. E se fosse possível agrupar várias máquinas de configurações medianas e que essas mil páginas fossem divididas em cada máquina. Então cada computador contaria menos palavras e ao final da operação todos os resultados parciais são reunidos para formar o resultado final.

Essa é a vantagem de se trabalhar com um cluster! Pensando nesse processo todo aquele ditado "dividir para conquistar" nunca fez tanto sentido, não é? Neste projeto iremos criar um cluster utilizando a AWS, isto é, ao invés de usarmos máquinas físicas (on-premises), utilizaremos instâncias (máquinas virtuais) na nuvem.

Tudo isso parece muito interessante, mas como essa divisão das tarefas é feita dentro de um cluster? Como é feita essa coordenação? Veremos isso e um pouco mais logo adiante.

Hadoop

O Apache Hadoop é um framework open-source que nos permite trabalhar com armazenamento e processamento de grandes quantidades de dados através de clusters.

Podemos dividir o Hadoop em

HDFS:

É o Hadoop Distributed File System que faz a gestão e distribuição dos dados dentro do cluster. Outra forma de entendermos o HDFS é como um sistema de arquivos distribuídos. Mas o que seria um sistema de arquivos?

Um sistema de arquivos é utilizado para armazenar, organizar e acessar dados em um computador de forma efetiva. Esse sistema de arquivos é comum a sistemas operacionais também como Windows e Linux.

Vamos agora nos aprofundar um pouco mais no HDFS detalhando suas principais características.

- Tolerante a falhas: Se um node (uma máquina) falhar o processo não para.
- Integridade: Garante a segurança e a integridade dos dados.

- Segurança: Garante que o arquivo não será modificado durante a transferência. Garante também a segurança no acesso aos dados.
- Desempenho: Possui alto desempenho.
- Consistência: Garante que todos os usuários devem ter a mesma visão dos arquivos.

Agora podemos olhar para a arquitetura do HDFS:

Namenode (nodemaster): É o componente central em uma arquitetura HDFS. Ele armazena os metadados e também gerencia os datanodes.

É importante salientar que é recomendado implementar o Namenode em um node exclusivo.

A máquina em que o namenode será implementado deverá ser mais robusta que as outras (datanodes), principalmente em memória.

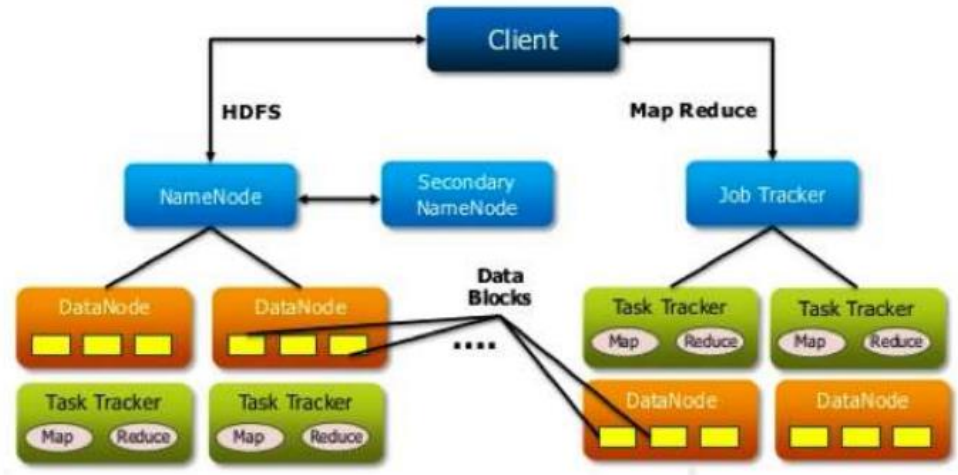
Componentes do namenode:

- FSImage: É responsável por armazenar informações estruturais dos blocos (mapeamento e namespace de arquivos)
- Editlog: É um arquivo de log que armazena todas as alterações ocorridas nos metadados dos arquivos.
- Secundary Namenode: Realiza a junção entre o FSImage e o Editlog, criando assim pontos de checagens de modo a limpar os arquivos de logs.

Datanode: Responsável pelo armazenamento físico dos dados e também executam as tarefas.

Essa dinâmica entre o namenode e os datanodes é conhecido como master/slave.

A figura abaixo ilustra a arquitetura do HDFS.



Na figura os retângulos amarelos são os dados e o task tracker é o serviço responsável pelo processamento dos dados. Convém agora falarmos um pouco sobre a replicação dos blocos de dados. Primeiro o HDFS divide os dados em blocos (128Mb por padrão), depois replica esses arquivos para aumentar a segurança. Isto é ele não vai apenas dividir o nosso arquivo e sim dividir e criar copias para o caso de falhas. Por padrão um bloco no HDFS possui 3 réplicas em diferentes nós.

Podemos agora então resumir todo o funcionamento de um cluster Hadoop:

1ª) O cliente faz uma requisição (dados + processamento). Isto é namenode (HDFS) e o jobtracker (mapreduce).

2ª) Os dados são divididos em blocos e os Jobs em partes menores (tarefas).

3ª) Alocar as partes menores para cada node. Blocos e tarefas são distribuídos pelo cluster.

4ª) Armazenamento/processamento paralelo, o jobtracker aciona o tasktracker.

5ª) Unir os resultados das tarefas, conhecido como etapa de shuffle.

6ª) Temos então o resultado final, onde o jobtracker informa o resultado para o cliente.

Neste ponto é importante salientar que os hardwares das máquinas que compõe um cluster devem ser semelhantes em sua configuração. Caso contrário a performance do cluster será a performance do node mais lento.

Para finalizar essa etapa é importante falarmos um pouco sobre o armazenamento. Os dados são armazenados nos datanodes e os metadados no namenode. Mas o que são metadados?

Metadados são informações gerais sobre o cluster, como localização dos blocos, configurações etc). Portanto os datanodes armazenam e recuperam os dados. E o tasktracker executa os Jobs de mapreduce.

Para encerrarmos essa "breve" introdução ao Apache Hadoop vamos falar um pouco sobre o YARN (yet another resource negotiator). É a camada de gerenciamento de recursos do Hadoop e também é usado para o agendamento de jobs. Pode ser considerado como o sistema operacional de dados do Hadoop 2.x. E possui algumas distribuições conhecidas como a Cloudera, Hortonworks e MAPR.