Impacto dos Lançamentos de Filmes e Séries nas Quotas de Mercado

ISCTE-Instituto de Lisboa

Bruno Santos nº 125712

Eduardo Bagulho nº

Gonçalo Rosa nº 125684

# Introdução

A aquisição de dados sempre gerou um problema para a sociedade, o seu armazenamento. Nos dias de hoje qualquer pessoa pode comprar dezenas de centenas de gigas de armazenamento. Mesmo assim a quantidade de dados existente atualmente que mesmo as melhores máquinas teriam dificuldades em processar tais dados, a este fenómeno de dados chamamos de BIG DATA, a sua análise não é suportada apenas numa só máquina. Para tal dividimos a carga e os dados por diversas máquinas onde iram armazenar e analisar sempre que necessário, este conceito é chamado de sistemas distribuídos, uma rede de máquinas interligada entre si para que possamos questioná-la ou armazenar nela, uma quantidade de dados que não seria concebível numa única máquina.

Neste trabalho não iremos ter uma quantidade exorbitante de dados, mas pretendemos demonstrar como os sistemas distribuídos armazenam dados e os disponibilizam aos seus utilizadores, quando questionados, pretendemos mostrar como estes sistemas possibilitam redundância de dados com a capacidade de manter a integridade dos dados ao efetuarem transações entre si.

Iremos ingerir no sistema de base de dados com informações relativas as quotas de mercado e aos lançamentos de filmes/series nas plataformas de entretenimento. Pretendemos relacionar esta informação entre si para que possamos tirar conclusões sobre o impacto de lançamentos de filmes ou series na plataforma com a quota de mercado.  
  
Este relatório também terá como finalidade de guia para como o sistema foi montado, dados adquiridos e limpos e claro as escolhas para os componentes dentro do nosso sistema, iremos descrever brevemente cada camadas (HDFS, YARN, MapReduce, HBase, Phoenix, Hive, Hue) e a sua parte no funcionamento do sistema em geral.

# Datasets

Os dados que utilizamos no nosso trabalho foram obtidos na plataforma kaggle, encontramos dois datasets um com a informação relativa aos filmes e series lançadas na plataforma Netflix (por o intervalo de datas do data set) e outro com quotas diárias no mercado de ações. O objetivo será observar se o lançamento na plataforma impacta de alguma forma as ações da empresa.

## Shares

O dataset Shares contem toda informação relativa à ação da Netflix desde (datas), juntamente com diversos indicadores que possibilitam uma análise mais aprofundada do comportamento da ação. Para o nosso relatório final e para ser ingerido pelo nosso sistema removemos algumas colunas, sendo as seguintes colunas as informações que iremos utilizar.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Date | Open | Close | Volume | sma\_50 | sma\_100 | ema\_50 | ema\_100 | Next day  close |
| Data do mercado | Preço de abertura do dia | Preço de fecho do dia | Quantidade de ações vendidas | *Simple moving*  *Average* de 50 dias | *Simple moving*  *Average* de 100 dias | *Exponential moving*  *Average* de 50 dias | *Exponential moving*  *Average* de 100 dias | Valor do fecho do dia seguinte |

Visto que pretendemos avaliar a performance da quota e como esta se comporta ao longo do tempo, foi necessário utilizar a variável “*Next day close*” que tem o valor do fecho da ação no dia seguinte, juntamente com a variável *close* que possui o valor do fecho do dia ao aplicar a fórmula 1 é possível retirar o crescimento da ação relativamente ao dia seguinte (**Growth)**.

## Netflix movie/series releases

Neste dataset iremos encontrar informações relativas aos filmes/series da Netflix, iremos usar este dataset para realizar perguntas sobre a evolução da quota de mercado relativamente aos filmes/series que são lançados na plataforma.

Os campos que utilizamos na nossa base de dados serão os seguintes:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Type | Title | Country | Date\_added | Release\_year | Durantion | Genre |
| Tipo de filme | Título  Da peça | País | Data de adicionamento à plataforma | Ano de lançamento do filme | Duração | Todos os géneros que a peça foi listada a ter |

Realizamos algumas transformações neste dataset para que fosse possível criar uma relação na base de dados com os diversos dias de lançamento na plataforma com a data de mercado, para tal alteramos a estrutura da Date\_added para que coincidisse com coluna Date de shares.  
Ao analisarmos o dataset da Netflix, descobrimos que podíamos também criar uma tabela de referência com todos os tipos de géneros de peças existentes, possibilitando uma análise futura do mercado pelo género de peça em si.

Todas as alterações que foram mencionadas em ambos datasets foram realizadas em python com ajuda do pandas, extraindo do csv original, aplicamos as transformações e criamos 3 ficheiros csv contendo os dados que pretendemos ingerir no nosso sistema distribuído.   
  
Retirando os seguintes ficheiros, após corremos o nosso código “Netflix\_Clean”, “moviesAltered”, “genres”, “sharesAltered”.   
  
Estando todas as transformações necessárias nos nossos dados podemos então passar à ingestão dos dados para o nosso sistema distribuído pelo HUE.

# Sistema distribuído e suas camadas

Um sistema de base de dados distribuído é composto por um array de base de dados interligadas entre si através de uma rede. Não podemos considerar um sistema de base de dados distribuída caso não exista uma rede que a interligue.  
O objetivo destes sistemas é aumentar a capacidade de processamento e de armazenamento para o um nível acima daquilo que uma só máquina alcançaria. Tendo diversos desafios associados, como a redundância dos dados, sua integridade e a forma como estes sistemas partilham informação quando necessário responder a queries onde a informação não se encontra apenas num site.

O benefício destes sistemas de base de dados é que proporcionam capacidades de armazenamento nunca visto, como também um alto processamento de dados. Diversas empresas de alto calibre utilizam arquiteturas semelhantes para rentabilizar os seus negócios ao mais alto nível. Tanto o spotify como a Netflix usam algoritmos de recomendação usando os dados de consumo dos seus clientes como input, agregando ao facto de possuírem bibliotecas gigantes de entretinimento, não é de admirar que seus negócios sejam muito lucrativos

O funcionamento de sistemas distribuído como o nome o aponta, vem da sua capacidade de fragmentar e distribuir dados de forma que estas mais tardes possam ser reconstruídas através de um operador relacional. Levantando outros desafios, como por exemplo, gerenciamento dos sites e da sua informação, e como podemos processar toda a informação na nossa rede e apresenta la de forma concisa.

Ao fragmentar dados temos de ter em conta os seguintes conceitos para que a fragmentação ocorra de forma correta.

* Completude

Ao decompormos uma tabele em diversos fragmentos, R1, R2, …, o pedaço de dados tem de ser encontrado em algum RI.

* Reconstrução

Se a relação R é decomposta em múltiplos RI, então tem de haver um operador relacional que seja capaz de reconstruir a relação na totalidade.

* Disjunção

O dado dentro de um RI, só podem ser encontrados dentro desse RI e em mais nenhum outro RI.  
  
As tabelas, podemos aplicar dois tipos de fragmentação, vertical e horizontal. Capacitando a separação de colunas ou de linhas respetivamente, beneficiando quando pretendemos agrupar certos grupos de colunas em um site, ou então filtrar a tabela inicial e guardando apenas esses dados em um site específico.

A construção de sistemas distribuídos de bases de dados, pode variar de projeto para projeto devido à necessidade requerida da aplicação, seja para que haja mais redundância que velocidade, mais disponibilidade, mas menos replicação. Sendo o objetivo deste trabalho o melhoramento do funcionamento destas componentes e o seu contributo para o sistema em geral de seguida temos um breve descrição de cada componente e a sua utilidade na nossa aplicação.

## HADOOP

O Hadoop é um framework de código aberto que permite o armazenamento e processamento de grandes volumes de dados em diversos nós conectados por uma rede. Ele é tolerante a falhas devido à capacidade de replicação dos fragmentos de dados e altamente escalável, possibilitando a interligação de um número significativo de nós. O Hadoop é composto principalmente por três pilares que sustentam suas funcionalidades: o HDFS, o YARN e o MapReduce.

### HDFS

Proporciona a capacidade de dividir ficheiros por diferentes blocos de tamanho no momento do upload.

Facilitando a capacidade de replicação dos fragmentos por os diferentes sites, esta replicação pode ser completamente partição, ou seja, todos os nos contem todos os fragmentos dos dados criando uma redundância total, mas consome muita memória e desperdiçando o principal aspeto do sistema de base de dados distribuído, pois basicamente cada site será uma replica da base de dados no geral.

Outra possibilidade seria a parcialmente replicação, este modelo oferece um equilíbrio entre tolerância a falhas e eficiência, pois permite maior disponibilidade de recursos e capacidade de processamento. Entretanto, a tolerância a falhas é reduzida em comparação com a replicação total, já que a perda de múltiplos nós pode comprometer os dados.

Um bom indicador de necessidade de replicação ou não de replicação é a seguinte:

Caso haja muitas escritas no sistema do que leituras então a replicação é necessária.

### YARN

Yet another resource manager é o módulo de gerenciamento de recursos do Hadoop, que coordena como os recursos do cluster (CPU, memória, etc.) é capaz de perceber sites que estejam próximos da capacidade máxima e advertir dados para outros sites com mais recursos, seja memoria ou capacidade de processamento.

### MapReduce

O Map reduce tem a funcionalidade de processar os dados em paralelo dentro do cluster dividindo tarefas entre os nós e consolidando os resultados em uma resposta concisa.

## HBase

O HBase é uma base de dados NoSQL projetada para armazenar dados de forma não relacional, com alta disponibilidade e desempenho. utiliza o método schema on read, o que significa que os dados podem ser armazenados sem uma estrutura predefinida e somente serão interpretados no momento da leitura. Ele permite reescrever linhas diretamente e armazena qualquer tipo de informação, preservando o histórico de alterações conforme necessário.

O Hbase é composto por três componentes:

* Hmaster

Responsável pela monotorização dos cluster Hbase, e pela distribuição dos regions servers dentro do cluster

* Region server  
   Armazena e gerência as regiões (fragmentos horizontais das tabelas), tem a função de escrita e de leitura dos clientes, garantindo a integridade dos dados nestas funções.
* ZooKeeper  
   Tem a visão geral do cluster, coordenando os serviços de distribuição do cluster, também possui uma localização de cada região e o seu conteúdo quando é necessário realizar queries as mesmas.

## Phoenix

O Phoenix é uma camada SQL com a função de interagir diretamente com o HBase, permitindo que bases de dados NoSQL sejam manipuladas por meio de queriesSQL (select, update, delete, insert). Esta integração é especialmente vantajosa para desenvolvedores e analistas já customizados com SQL, facilitando a adoção do Hbase não sendo necessário um novo estudo sobre diversas APIs.

## Hive

Outra camada complementar ao Hadoop, utilizado para analise de grande volume de dados em sistemas distribuídos, proporciona uma interface gráfica que possibilita a escrita de queries semelhantes ao SQL (hiveQL) permite realizar joins, group by, etc. Preferível visto que não pretendíamos utilizar Java ou usar o map reduce, e tendo mais conhecimento em SQL a utilização de hiveQL é ideal.

## Hue

Uma interface gráfica que facilita a interação com o ambiente Hadoop, permite aceder a múltiplos componentes através da sua interface gráfica. A sua utilização é nos benéfica devido a termos pouca experiência com a interação com as camadas inferiores no nosso sistema como HBase.

FALTA DIZER O PORQUE QUE UTILIZAMOS ESTES COMPONENTES

# Inserção de dados e analise

O passo seguinte terá com objetivo colocarmos todos os dados até agora trabalhados no nosso sistema de base de dados distribuído, podemos executar esta tarefa com ajuda da interface gráfica HUE, que nos permite comunicar com a linguagem SQL com as camadas mais abaixo no nosso sistema. Para tal é necessário seguir os seguintes passos.

1. Iniciar o Docker com o comando “docker-compose up -d”, que irá iniciar a imagem que se encontra no nossa pasta que previamente foi clonada de um repositório git.  
   Também podíamos através do Docker studio iniciar a mesma imagem.

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

Figura 1- ligar o docker

Com a imagem a correr, podemos então aceder ao seguinte url [*http://localhost:8888/*](http://localhost:8888/)para que possamos interagir com a interface gráfica. Relembrar que a primeira vez será necessário associar uma palavra-passe. Consequentemente sempre que acedam ao HUE será necessário utilizar a mesma.

1. Uma imagem com texto, software, computador, Sistema operativo

   Descrição gerada automaticamenteCriar uma base de dados com o nome project\_Netflix.

Figura 2 - Criar base de dados

Uma imagem com captura de ecrã, texto, file

Descrição gerada automaticamenteCom a base de dados criada, ao aceder podemos ver que esta se encontra vazia

Figura 3 – base de dados

1. Uma imagem com texto, captura de ecrã, software, número

   Descrição gerada automaticamenteCriação das tabelas com os ficheiros com os dados trabalhados

Importamos os ficheiros csv com os dados trabalhados e definimos os tipos de dados que os campos terão associados dentro da base de dados, este passo é essencial para garantir uma boa gestão de armazenamento e de performance.

1. Repetimos o processo para todos os ficheiro de dados que pretendemos inserir na nossa base de dados.

## Conclusão

Em conclusão as dinâmicas dos componentes que formam o nosso sistema de bases de dados proporcionam a capacidade de redundância e eficácia na resposta de informação mesmo esta estando distribuída. A construção deste trabalho permitiu nos a interação com ferramentas como o Docker, Hue, Phoenix refletindo a dinâmica real de sistemas em produção.