Impacto dos Lançamentos de Filmes e Séries Tendências de Mercado da Netflix

**Desenvolvimento e Implementação do Sistema de Análise de Dados Distribuído**

1. **Introdução**

A aquisição de dados sempre representou um desafio significativo para a sociedade, especialmente no que diz respeito ao seu armazenamento. Nos dias de hoje, qualquer pessoa tem a possibilidade de adquirir vários gigabytes ou até ter acesso a centenas de gigabytes de armazenamento. No entanto, a quantidade de dados que geramos atualmente é tão vasta que, mesmo as máquinas mais poderosas, enfrentam dificuldades para processá-los de forma eficiente. Este fenómeno é denominado de **Big Data**, e a sua análise não pode ser realizada por uma única máquina. Para resolver este problema, a carga de trabalho e os dados são distribuídos por várias máquinas, permitindo que estas os armazenem e analisem de forma paralela sempre que necessário. Este conceito é conhecido por **sistemas de base de dados distribuídos**, onde rede de máquinas interligadas possibilita o armazenamento e a consulta de grandes volumes de dados que seriam inconcebíveis numa única máquina.

Neste trabalho, embora não trabalharemos com uma quantidade excessiva de dados, o nosso objetivo é demonstrar como tais sistemas são capazes de armazenar e disponibilizar dados aos seus utilizadores, mediante solicitações. Pretendemos também evidenciar como estes sistemas asseguram a redundância dos dados, mantendo a sua integridade mesmo durante transações entre as diversas máquinas na rede.

Iremos importar para o sistema de base de dados informações relativas às **quotas de mercado** e aos **lançamentos de filmes e séries** nas plataformas de entretenimento. O intuito é relacionar essas informações de modo a compreender o impacto dos lançamentos de filmes ou séries nas quotas de mercado dessas plataformas.

Este relatório tem, ainda, como objetivo servir de guia para explicar a forma como o sistema foi concebido, como os dados foram adquiridos e tratados, bem como as decisões tomadas na escolha dos componentes do sistema. Serão descritas, de forma sucinta, as camadas do sistema (HDFS, YARN, MapReduce, HBase, Phoenix, Hive, Hue) e o papel de cada uma delas no funcionamento geral da solução.

1. **Objetivos do Projeto**

O principal objetivo deste projeto centra-se no desenvolvimento de uma solução de análise de dados utilizando o stack Hadoop, com especial incidência em sistemas distribuídos. Pretende-se demonstrar metodologias eficientes e escaláveis para o armazenamento, processamento e análise de dados numa infraestrutura distribuída.

Os objetivos específicos contemplam:

A implementação de uma arquitetura distribuída suportada pelas tecnologias do ecossistema Hadoop (HDFS, YARN, MapReduce, HBase, Phoenix, Hive e Hue); a execução de processos de ingestão e preparação de dados sobre lançamentos de filmes e séries e quotas de mercado das plataformas de streaming; a identificação de padrões e correlações entre os dados de lançamentos e o comportamento do mercado; a garantia de integridade e redundância dos dados num sistema distribuído; o desenvolvimento de visualizações dos resultados e a criação de um produto final que possa suportar decisões estratégicas nas plataformas de entretenimento, através do Streamlit, executado dentro de um container Docker, permitindo a visualização de forma acessível.

O projeto contemplará ainda a documentação detalhada do processo desenvolvido, bem como uma análise técnica exaustiva da utilização das diversas ferramentas e componentes do stack Hadoop.

1. **Execução e Configuração do Ambiente  
   3.1 Configuração do Ambiente**

Neste projeto, foi criada uma arquitetura distribuída utilizando Docker para configurar os componentes necessários ao processamento e análise de dados com o stack Hadoop. A escolha do Docker permite isolar e replicar facilmente o ambiente, garantindo a integração de todas as dependências.

A configuração inclui os seguintes serviços:

* Namenode e Datanode: Gere e armazena os dados no HDFS, respetivamente.
* Resource manager: Gestão e alocação de recursos no cluster Hadoop.
* Hive Server e Hive Metastore: Permitem consultas HiveQL sobre os dados armazenados no HDFS. Neste modelo os dados são visualizados através de queries de HiveQL, projetamos que no futuro iremos possuir um grande volume de dados de outras empresas de streaming, devido ao Hive ser eficiente com altos volumes de dados iremos então priorizar o seu uso.
* Hue: Fornece uma interface web para interagir com o Hive e outras ferramentas do Hadoop. A ingestão dos dados foi realizada através do Hue devido a sua grande simplicidade, não sendo necessário utilizar linhas de comando, apenas ficheiros csv. A ingestão predefinida é realizada para o Hive, pelas razões previamente mencionadas como o volume de dados.
* Streamlit: Foi configurado para criar visualizações interativas dos dados extraídos do Hive.

A configuração foi orquestrada com um arquivo docker-compose.yml, permitindo a comunicação entre os serviços e garantindo a persistência de dados através de volumes. O ambiente foi iniciado com o comando docker-compose up, criando uma infraestrutura funcional para a análise e visualização dos dados.

**3.2 Integração e Preparação dos Dados**

Os dados utilizados foram obtidos na plataforma Kaggle, encontramos dois datasets um com a informação relativa aos filmes e series lançadas na plataforma Netflix (entre 2008 e 2021) e outro com quotas diárias no mercado de ações.

O dataset que contém toda informação relativa à ação da Netflix desde 2014 até 2023, 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 em que foi adicionado 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 mencionadas nos dois datasets foram realizadas em Python, com o auxílio da biblioteca pandas. A partir dos arquivos CSV originais, aplicaram-se as transformações necessárias e, em seguida, geraram-se três novos arquivos CSV contendo os dados prontos para serem ingeridos no sistema distribuído. Após a execução do código, os seguintes arquivos foram gerados: "moviesAltered", "genres" e "sharesAltered".  
  
Estando todas as transformações necessárias nos nossos dados podemos então passar à ingestão dos dados para o nosso sistema distribuído pelo HUE.

**3.3 Análise**

A análise dos dados centrou-se em avaliar o impacto dos lançamentos de filmes no preço das ações da Netflix. Para tal, foram examinados diferentes aspetos dos dados recolhidos, com o objetivo de identificar padrões e tendências que pudessem ilustrar a relação entre a produção de conteúdo e o desempenho financeiro da empresa.

Em primeiro lugar, foi analisada a evolução histórica do preço das ações ao longo do tempo. Este gráfico permitiu observar as flutuações nos preços, especialmente nos períodos em torno dos lançamentos de filmes. A análise procurou identificar períodos de maior variação no preço das ações, relacionados com lançamentos de conteúdos específicos.

Além disso, foi realizada uma análise sobre o número de filmes lançados por ano, com o intuito de compreender a relação entre a quantidade de conteúdo produzido e o comportamento das ações da empresa. Esta análise procurou identificar se o volume de lançamentos de filmes tinha alguma correlação com variações no preço das ações.

A análise incluiu também o estudo do impacto dos géneros de filmes nas variações do preço das ações. Foram examinados os géneros de filmes que tiveram maior impacto nas flutuações percentuais do preço das ações, com o objetivo de entender como diferentes tipos de conteúdo poderiam afetar a perceção do mercado e dos investidores.

Por fim, foi realizada uma análise sobre os filmes com maior impacto nas ações, focando nos lançamentos que causaram as maiores variações percentuais no preço das ações da Netflix. Esta análise teve como objetivo identificar os filmes que mais influenciaram as flutuações no mercado financeiro da empresa.

Estas análises permitiram compreender como a produção de filmes da Netflix pode estar relacionada com o comportamento do mercado financeiro, explorando os efeitos dos lançamentos e dos diferentes géneros no preço das ações da empresa.

**3.4 Visualização e Apresentação**

Para a visualização dos resultados, foi utilizado o **Streamlit**, que possibilitou a criação de um dashboard. Este painel apresenta gráficos e tabelas que facilitam a análise da relação entre os lançamentos de filmes e as flutuações no preço das ações da Netflix.

As visualizações incluem a evolução histórica do preço das ações, o número de filmes lançados por ano, o impacto dos géneros nos preços das ações e os filmes com maior influência no mercado.

O dashboard permite uma exploração dinâmica dos dados, proporcionando uma compreensão mais clara dos padrões identificados.

A seguir, é apresentada uma captura do dashboard desenvolvido, destacando as principais visualizações.

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

Descrição gerada automaticamente

* 1. **Desafios do projeto**

Durante o projeto foram identificados diversos desafios ao longo das etapas de desenvolvimento, o nosso objetivo neste paragrafo é perceber os problemas e as soluções que encontramos.

* **Limpeza de dados incorreta**

Durante a limpeza e a construção do nosso dataset, diversas alterações foram necessárias para que fosse possível conectar o calendário de filmes com o mercado de ações, as datas tiveram de ser alteradas para que houve um consenso.   
No campo de géneros, a divisão por virgula causou uma ingestão incorreta dos dados para tal foi necessário que a divisão realizasse por “;”.

* Hue

A interface do Hue, embora seja bastante simples para a ingestão de dados, esta complica a componente visual do dashboard, não consigo de alguma forma apresentar os dados nos moldes que pretendíamos, como gráficos de barras ou tabelas. Para solucionarmos este problema, encontramos uma biblioteca capaz de conectar ao Hive e executar as nossas queries e por fim apresentá-las em gráficos e tabelas através de python.

* Streamlit

Tivemos dificuldade no setup desta componente dentro do próprio Docker, ocorreu diversos erros entre versões das dependências utilizados no script utilizado para correr o dashboard, foi necessário instalar todas as bibliotecas individualmente e algumas vezes em outras versões para que o script corre se sem erros.

1. **Análise dos Componentes do Hadoop**

Um sistema de base de dados distribuído é composto por um conjunto de bases de dados interligadas por uma rede, a rede é essencial para ser considerado um sistema distribuído. O principal objetivo destes sistemas é aumentar a capacidade de processamento e de armazenamento além daquilo que uma única máquina poderia alcançar. No entanto, este tipo de sistema cria vários desafios, como a redundância dos dados, a sua integridade e a forma como as informações são partilhadas quando necessárias para responder a queries em que os dados estão distribuídos entre diferentes sites.

Os benefícios de um sistema de base de dados distribuída são evidentes, pois permite um armazenamento de grandes volumes de dados e um processamento em larga escala. Empresas de grande dimensão, como Spotify e Netflix, utilizam arquiteturas semelhantes em junção com algoritmos de recomendação baseados nos dados de consumo dos seus clientes, e aproveitam as suas vastas bibliotecas de conteúdos para alcançar altos níveis de rentabilidade.

O funcionamento de sistemas distribuídos, como o nome sugere, depende da capacidade de fragmentar e distribuir dados de forma que possam ser reconstruídos posteriormente por um operador relacional. Esse processo levanta desafios adicionais, como o gerenciamento dos sites e da informação neles contida, e como processar e apresentar essa informação de forma eficaz.

A fragmentação de dados deve seguir três princípios fundamentais:

* Completude: Ao fragmentar uma tabela em múltiplos fragmentos (R1, R2, …), cada pedaço de dados deve ser encontrado em algum desses fragmentos.
* Reconstrução: Se uma relação R for fragmentada em múltiplos fragmentos, deve existir um operador relacional capaz de reconstruir a relação na totalidade.
* Disjunção: Os dados de um fragmento (RI) devem ser encontrados apenas nesse fragmento e em nenhum outro.

Existem dois tipos principais de fragmentação: vertical e horizontal. A fragmentação vertical separa as colunas de uma tabela, enquanto a horizontal separa as linhas. Essas abordagens são úteis quando se pretende agrupar certas colunas em um site ou filtrar uma tabela para armazenar apenas dados relevantes em um site específico.

A construção de sistemas distribuídos de bases de dados pode variar de acordo com os requisitos da aplicação, seja priorizando redundância em detrimento da velocidade ou de maior disponibilidade com menos replicação.

No decorrer deste trabalho, utilizamos diversas componentes para a construção de um dashboard. Iremos apresenta las de seguida e o seu impacto no nosso sistema.

**4.1 HDFS**

Permite a divisão de ficheiros em blocos menores durante o upload, facilitando a replicação dos dados através dos sites. Esta replicação pode ser total, o que consome mais memória, ou parcial, oferecendo um equilíbrio entre eficiência e tolerância a falhas. A necessidade de replicação pode ser determinada pela relação entre o número de queries de leitura e o número de queries de atualização.

**4.2 YARN**

É o componente responsável pelo gerenciamento de recursos no cluster, coordenando a distribuição de recursos como CPU e memória, e alocando dados para sites com maior capacidade disponível.

**4.3 MapReduce**

Permite o processamento paralelo dos dados no cluster, dividindo tarefas entre os nós e consolidando os resultados de forma eficiente.

**4.4 HBase**

O HBase é uma base de dados NoSQL, projetada para armazenar dados não relacionais com alta disponibilidade e desempenho. Usa o modelo "schema on read", o que significa que os dados são armazenados sem uma estrutura predefinida e só são interpretados quando lidos. O HBase é composto por três componentes principais:

* HMaster: Monitora o cluster HBase e distribui os Region Servers.
* Region Servers: Armazenam e gerenciam as regiões (fragmentos horizontais das tabelas), realizando operações de leitura e escrita.
* ZooKeeper: Coordena os serviços de distribuição no cluster e mantém o controle sobre a localização de cada região e seu conteúdo**.**

**4.5 Phoenix**

O Phoenix é uma camada SQL que interage diretamente com o HBase, permitindo que dados NoSQL sejam manipulados através de queries SQL (como SELECT, UPDATE, DELETE, INSERT). Esta integração é vantajosa para desenvolvedores e analistas familiarizados com SQL, facilitando a adoção do HBase sem a necessidade de aprender novas APIs.

**4.6 Hive**

O Hive é uma camada complementar ao Hadoop, utilizada para análise de grandes volumes de dados em sistemas distribuídos. Ele oferece uma interface que permite a escrita de queries semelhantes ao SQL (HiveQL), suportando operações como JOIN, GROUP BY, entre outras.

O Hive é preferível para quem possui conhecimento prévio em SQL, evitando o uso de Java ou MapReduce, facilitando a análise de dados distribuídos.

**4.7 Hue**

O Hue é uma interface gráfica que facilita a interação com o ambiente Hadoop, permitindo o acesso a múltiplos componentes do sistema através de uma interface intuitiva. Sua utilização é benéfica para quem tem pouca experiência com as camadas inferiores do sistema, como o HBase, Hive, simplificando o processo de manipulação de dados e interação com os componentes do cluster.

1. **Conclusão**

Os sistemas de base de dados distribuídos representam uma evolução significativa na gestão e processamento de dados, permitindo uma escalabilidade que seria impossível de alcançar com sistemas centralizados. Estes sistemas têm um papel crucial em empresas modernas, como Spotify e Netflix, ao viabilizar a gestão de grandes volumes de dados e o processamento em larga escala, potencializando as capacidades de análise e recomendação aos seus utilizadores.

Apesar das suas vantagens, os sistemas distribuídos enfrentam desafios complexos relacionados à redundância, integridade e fragmentação de dados. A implementação destes sistemas requer a aplicação rigorosa de princípios como completude, reconstrução e disjunção, garantindo que os dados sejam devidamente organizados e possam ser eficazmente utilizados em consultas.

Componentes como HDFS, YARN, MapReduce e HBase desempenham papéis essenciais na arquitetura de sistemas distribuídos. Eles asseguram a replicação eficiente dos dados, a gestão de recursos e o processamento paralelo, enquanto suportam dados não estruturados com alta disponibilidade e desempenho.

O dashboard criado exemplifica a aplicação prática destas tecnologias, destacando o impacto positivo da combinação entre arquitetura distribuída e ferramentas avançadas de processamento.

Por fim, a utilização de sistemas de bases de dados distribuídas exige um equilíbrio cuidadoso entre eficiência, disponibilidade e redundância, conforme os requisitos específicos de cada aplicação. Este balanço, associado a um entendimento profundo das tecnologias subjacentes, é fundamental para o sucesso em cenários de grande escala e alta complexidade.

1. **Referências**