Atividade extraclasse

A) Objetivo: Atualmente o conceito Big Data tem estado em evidência porque lida com grandes e variados volumes de dados, formando um ambiente propício para a construção de aplicações que favorecem a tomada de decisões. Nesse contexto, os *frameworks* Apache Hadoop e Apache Spark podem ser vistos como plataformas alinhadas com os requisitos do BigData, além de serem abertas e de uso facilitado. O objetivo desse laboratório é criar um contexto para a experimentação dessas plataformas a fim de identificar suas características principais.

B) Metodologia, especificação e detalhamento de atividades

Para o alcance do objetivo desejado, a metodologia envolve *(i)* estudo do Hadoop, *(ii)* estudo do Spark. As subseções a seguir detalham cada um desses itens.

B1) Conhecendo o Apache Hadoop

No caso do Hadoop, os alunos devem realizar as atividades seguintes, com base no notebook disponível em https://colab.research.google.com/drive/160BLmEgto57pch16XLYqWeW493HYfqLh#scrollTo=f1UpCifppjNB . A solução final pode ser entregue na forma de um notebook (Google Colab ou jupyter) ou em configurações descritas em ambiente Linux. De qualquer modo, essa entrega deve conter o seguinte conteúdo:

Montagem de um cluster Hadoop básico (configuração básica) – Os alunos devem fazer uma instalação Hadoop em modo cluster, composto por um nó mestre e pelo menos dois nós escravos (workers), se possível com interface web de monitoramento do framework e dos serviços submetidos no ambiente. Neste caso, é importante anotar e documentar todos os arquivos de configuração utilizados nos nós *master* e *slaves*. O ideal é que o cluster seja formado por nós distintos em hosts distintos. No entanto, caso os alunos não consigam, podem entregar uma versão mais simples, por exemplo, com os nós instanciados em contêineres, desde que sejam tomados os cuidados necessários para não comprometer os testes de performance solicitados a seguir, especialmente se esses contêineres estiverem no mesmo *host*.

Teste de comportamento do *framework* Hadoop – A partir da configuração básica, os alunos devem promover algumas alterações no cluster (pelo menos 5 mudanças), de modo a gerar algum impacto no escalonador de processos (Yarn), no sistema de arquivos HDFS e no funcionamento geral de aplicações. Essas alterações podem ser feitas nos arquivos de configuração do Hadoop e os efeitos são relativos, por exemplo, ao modo como o *framework* escalona os serviços, como distribui recursos para as aplicações (memória, disco, ...), como o HDFS funciona, entre outros.

Teste de tolerância a faltas e performance de aplicações Hadoop – Os alunos devem criar uma aplicação que leia uma grande massa de dados a ponto de a aplicação ficar em execução por um tempo razoável no cluster com configuração completa (*master* e todos os *slaves* ativos). A partir daí, deve-se monitorar o comportamento do Hadoop e da aplicação, considerando *(i)* o tempo de resposta (quanto demora para executar), e *(ii)* a saúde da aplicação (se mantém o funcionamento normal) sob condições adversas. Essas condições adversas devem ser provocadas em experimentos controlados e monitorados (via interface web do Hadoop). São cenários onde os nós *master* e *slaves* são inseridos e retirados de formas variadas (simulando situações de falhas/faltas) enquanto a aplicação estiver em produção. Para cada condição adversa, documentar o cenário e os resultados obtidos. As conclusões sobre os testes devem estar no relatório de entrega e devem relatar se é mesmo possível melhorar o desempenho da aplicação pelo acréscimo de nós, e qual o nível de tolerância a faltas suportado pela aplicação no Hadoop. Além disso, deve-

se tecer comentários sobre eventuais vantagens/desvantagens de uso desse tipo de ambiente computacional.

A aplicação sugerida para os testes de tolerância a falhas, é o contador de palavras – aplicação exemplo utilizada na apresentação do Hadoop – (*wordcount*) de um ou mais arquivos em formato texto, usando o paradigma MapReduce, desde que a massa de dados de entrada (uma biblioteca livros, por exemplo; ou um gerador de textos automático) seja suficiente para manter a aplicação executando por um tempo razoável (em torno de 3 a 4 minutos ou mais), para que seja possível monitorá-la.

Obs.: Os alunos podem variar o experimento, de modo a testar outros elementos que julgarem importantes. Por exemplo, podem envolver mais de uma aplicação wordcount rodando ao mesmo tempo, dentre outras possibilidades.

B2) Conhecendo o Apache Spark

No caso do Apache Spark, os alunos devem criar uma nova versão do Notebook disponível em https://colab.research.google.com/drive/1BHFbFP7Bs38APEhYOsOxqrEYFNbXbBSt?usp=drive_link, de modo a substituir as entradas e saídas processadas, da seguinte forma:

Substituir o esquema atual de entrada por um método de coleta de palavras a partir de alguma rede social, como o Discord (https://discord.com/), ou outra rede que suporte o consumo via canais kafka. Nesse caso, todo o processo de configuração e consumo das palavras da rede social, bem como os comandos de recuperação dos dados de entrada devem estar documentados no Notebook Google Colab (ou em documento à parte, em último caso). Caso o uso de rede social como entrada não seja possível, os alunos devem escrever texto justificando os problemas que tiveram e o que os motivaram a mudar de estratégia para viabilizar a entrega do notebook.

Substituir o canal de saída atual por um gráfico de nuvens usando ElasticSearch e Kibana integrados a um canal kafka de saída de modo que tudo o que for escrito no canal de saída seja visualizado em gráficos (nuvem de palavras ou similar), no Kibana. Todos os passos de instalação e configuração necessários para viabilizar esse tipo de saída devem ser documentados no notebook. Da mesma forma, se a utilização do Elastic/Kibana não for possível como dashboard, os alunos devem escrever texto justificativo e adotar alguma outra alternativa gráfica compatível.

Observações: O notebook a ser entregue pode ser feito como Google Colab ou Jupyter Notebook e deve conter todos os comandos que garantam o alcance do objetivo proposto, sem a necessidade de uso de comandos externos ao próprio notebook (valem apenas os comandos do notebook). Ou seja, deve-se manter o mesmo roteiro do Google Colab usado como referência, alterando apenas as partes relacionadas às interfaces de entrada e saída, seguindo mais ou menos a seguinte sequência:

1. Inicializações para o laboratório Google Colab
2. Configurações de mecanismos para garantir o acesso aos resultados do notebook (visualização dos resultados da contagem de palavras na forma gráfica)
3. Instalação/configuração do Spark – versão cluster, preferencialmente
4. Instalação/configurando do kafka e canais de entrada e saída
5. Instalação/configuração da API de consumo de rede social
6. Instalação/configuração de saída gráfica
7. Contabilização de palavras no Spark
8. Apresentação de resultados em um dashboard gráfico no Kibana/ElasticSearch

Obs.: Opcionalmente os alunos podem substituir o contador de palavras sugerido no notebook Spark por uma versão que faça uso de modelos de rede neural, como análise de sentimentos

(https://en.wikipedia.org/wiki/Sentiment_analysis) ou outro similar (não necessariamente envolvendo textos). Nesse caso, se houver uso de materiais de terceiros, é importante referenciar e destacar os diferenciais promovidos pelos alunos em relação ao que foi utilizado de terceiros.

C) Questões de ordem

- O experimento pode ser feito por grupos de 4 ou 5 alunos e todos devem trabalhar nos dois *frameworks*. Nesse caso, basta que um dos alunos faça a postagem das entregas no Moodle da disciplina.
- A entrega é composta por *(i)* um relatório, cuja estrutura e conteúdo está descrito a seguir, *(ii)* informações (arquivos) com a configuração e teste dos *frameworks*, incluindo informações necessárias para replicação do laboratório pelo professor, *(iii)* um vídeo gravado pelos membros participantes, com apresentação do experimento. Nesse caso, considerar uma média de 5 minutos por aluno para que possam demonstrar como participaram e conhecimentos adquiridos em cada *framework (iv)* uma documentação sobre os códigos e configurações entregues (se houver alguma configuração extra, incluir à parte) – essa documentação deve fazer parte do relatório a ser entregue.
- O relatório a ser entregue deve conter os seguintes pontos:
    - Título da atividade extraclasse, dados do curso, da disciplina/turma e identificação dos alunos participantes, data.
    - Introdução – pequena descrição da solicitação feita e uma visão geral sobre o conteúdo do relatório.
    - Uma seção descritiva sobre o experimento feito com *framework* Hadoop, conforme solicitado no item B1. Incluir informações sobre arquitetura e configurações adotadas para a configuração entregue; associar informações do grupo sobre os experimentos de performance e tolerância a falhas (descrição de cenários e resultados alcançados).
    - Uma seção descritiva sobre o experimento feito com *framework* Spark, conforme solicitado no item B2. Incluir informações sobre arquitetura e configurações adotadas para a configuração entregue; comentar as principais dificuldades e aprendizados com os experimentos realizados até chegar à versão final de notebook entregue.
    - Conclusão – iniciar com um texto conclusivo sobre os experimentos (tecer comentários e conclusões sobre os resultados alcançados em cada *framework*) e subseções para que cada aluno possa manifestar sua opinião e aprendizados específicos sobre o que foi feito, em função do grau de envolvimento com essa atividade extraclasse.
    - Apêndice/Anexo (seção opcional) – com eventuais informações não apresentadas anteriormente, tais como arquivos de definição de interface, comentários sobre os códigos construídos, instruções de execução, dentre outros.
- Além das entregas, os alunos devem estar preparados para uma apresentação em sala, conforme definido pelo professor em data oportuna (nesse caso, trazer slides para facilitar a palestra).