Exercício-Programa
Análise de Dados com Apache Spark

Prof. Dr. Daniel Cordeiro
Monitor: Miguel F. S. Vasconcelos
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo

Entrega: **3 de julho de 2022**

Descrição Geral

O Apache Spark[1] é um arcabouço de desenvolvimento de software distribuído para análise de dados que facilita (e muito!) análises de **grandes** volumes de dados. Graças a este projeto de código aberto desenvolvido sob a tutela da Fundação Apache, qualquer desenvolvedor pode escrever aplicações distribuídas escaláveis que podem utilizar milhares de máquinas simultaneamente.

O Spark introduziu o conceito de *Resilient Distributed Dataset* (RDD), uma coleção distribuída e imutável de elementos de um conjunto de dados, que é particionada entre nós de um aglomerado de computadores (*cluster*) e que podem ser manipulados em paralelo. O Apache Spark implementa uma API de baixo nível que oferece transformações e ações que podem ser realizadas nos elementos de um RDD em paralelo, usando os vários computadores do aglomerado.

A documentação oficial do Apache Spark pode ser encontrada em `https://spark.apache.org/docs/latest/`, que inclui um breve tutorial para aprender a utilizá-lo rapidamente em `https://spark.apache.org/docs/latest/quick-start.html`.

Neste Exercício-Programa, utilizaremos o Apache Spark para extrair estatísticas sobre os traços de execução de um grande aglomerado de computadores

[1] Apache Spark™ - Unified Engine for large-scale data analytics: `https://spark.apache.org/`

utilizado para executar aplicações da indústria. Usaremos dados de traços de execução de um sistema de computação distribuída proposto pelo Google: o Borg[2], sistema responsável por gerenciar os *clusters* do Google — compostos por dezenas de milhares de máquinas — e executar centenas de milhares de tarefas de milhares de aplicações diferentes.

O objetivo deste EP é desenvolver um sistema distribuído de análise de informações de desempenho de uma plataforma de computação e que seja escrito usando o paradigma de programação introduzido pelo Spark. Neste EP vocês deverão processar os dados e fazer algumas análises sobre eles. Essas podem se limitar a estatísticas descritivas simples (como médias, medianas, desvios-padrão, dentre outras). Além do código-fonte utilizado para a realização das análises, o grupo deve elaborar um relatório detalhado que descreva quais análises foram realizadas e quais resultados foram obtidos.

Traço de execução

O *traço de execução* de uma aplicação é um conjunto de informações registradas sobre a execução de uma aplicação. As informações do traço de execução de programas são utilizados para diagnosticar problemas na aplicação (depuração), mas também pode ser usada por administradores de sistemas para monitorar a utilização dos recursos computacionais disponíveis para os programadores.

O Google disponibiliza o traço de execução de tarefas lançadas por suas aplicações em oito de seus aglomerados de computadores. Essas informações compreendem as informações coletadas das tarefas executadas nesses aglomerados durante o mês de maio de 2019. As informações sobre os traços disponibilizados e como obtê-los estão em `https://github.com/google/cluster-data`.

Os dados completos do traço coletado em 2019 (referido na documentação como "versão 3") totalizam 9 terabytes quando compactado. Por isso, no contexto deste EP, iremos analisar um recorte (mais especificamente, são disponibilizadas as tabelas `collection_events` e `instance_events`) dos dados referente a primeira semana de execução das tarefas em um único aglomerado (o *cluster* G). Esse recorte dos dados foi disponibilizado

[2]Verma, Abhishek, et al. "Large-scale cluster management at Google with Borg." Proceedings of the Tenth European Conference on Computer Systems. 2015. `https://research.google/pubs/pub43438/`

em `https://drive.google.com/file/d/1hqB9w-48vRXy9KFHxMmgaXFQKfpv2qyK/view?usp=sharing` e possui 1,3 GB quando compactado e 23 GB quando descompactado.

A versão 3 do formato dos dados, usada nos arquivos disponibilizados, é descrita em `https://github.com/google/cluster-data/blob/master/ClusterData2019.md`. Para entender o formato, dois documentos são importantes: a descrição técnica do formato dos dados, disponível em `https://github.com/google/cluster-data/blob/master/clusterdata_trace_format_v3.proto`, e a descrição detalhada dos traços de execução, disponível em `https://drive.google.com/file/d/10r6cnJ5cJ89fPWCgj7j4LtLBqYN9RiI9/view`.

Objetivo

O objetivo deste trabalho é analisar as características do ambiente de execução e das tarefas executadas no aglomerado do Google. Você deve desenvolver um programa distribuído usando o Apache Spark que permitirá analisar os dados dos traços e elaborar um relatório detalhado com as diferentes análises realizadas e as suas descobertas. Seu relatório deve apresentar ao menos as análises para as seguintes perguntas:

- Como é a requisição de recursos computacionais (memória e CPU) do cluster durante o tempo?
- As diversas categorias de *jobs* possuem características diferentes (requisição de recursos computacionais, frequência de submissão, etc.)?
- Quantos *jobs* são submetidos por hora?
- Quantas tarefas são submetidas por hora?
- Quanto tempo demora para a primeira tarefa de um *job* começar a ser executada?

Outras análises — que podem inclusive utilizar outras tabelas dos traços de execução disponibilizados pelo Google — também podem ser realizadas, e serão recompensadas. ☺

Inclua em seu relatório análises que mostrem a descrição das estatísticas obtidas e também análises em função do tempo. Por exemplo, o número de *jobs*

submetidos por hora pode ser apresentado de forma agregada (média, desvio padrão, etc.), mas também em um gráfico que mostre a série temporal (ou seja, como o número total de *jobs* evolui ao longo do tempo).

Instruções

Seu programa deve ser implementado usando apenas o Apache Spark, ou seja, não é permitida a utilização de outros projetos da Apache para auxiliar o desenvolvimento.

O EP será testado em um computador equipado apenas com o sistema operacional GNU/Linux e com a distribuição Ubuntu 20.04.4 LTS.

A execução dos experimentos pode ser realizada tanto localmente, em uma instalação do Apache Spark no(s) seu(s) computador(es), quanto em um provedor de Computação em Nuvem. Apesar de não ser um requisito para o EP, aproveite a oportunidade para configurar uma conta em uma plataforma de Computação em Nuvem e experimente suas possibilidades.

Vários provedores de Computação em Nuvem dão créditos para novos usuários e definem alguns recursos que podem ser utilizados de graça, desde que dentro de certos limites (o chamado *free tier*). Dentre eles, Amazon e Google também dão acesso a uma plataforma pré-configurada para computação usando Spark. Veja os sites `https://aws.amazon.com/pt/emr/features/spark/` e `https://cloud.google.com/dataproc/` para mais informações.

Note que o uso indiscriminado de uma plataforma de computação em nuvem **pode gerar custos financeiros** (que serão debitados do cartão de crédito associado à conta quando os créditos acabarem). Os possíveis custos adicionais incorridos da má utilização da plataforma são de inteira responsabilidade dos grupos. Se estiver em dúvida, **utilize uma instalação local**.

Observações

Para este trabalho, vocês devem se organizar em **grupos de 4 (quatro) ou 5 (cinco) pessoas**.

Dúvidas em relação ao EP devem ser discutidas no fórum da disciplina no e-Disciplinas: `https://edisciplinas.usp.br/`. Todos são fortemente encorajados a participar das discussões e ajudar seus colegas.

Entregue junto com o código-fonte do programa um **relatório detalhado** que:

1. Explique em detalhes todos os passos necessários para a execução do programa;
2. Descreva em detalhes as análises que foram realizadas com o uso do programa escrito para o Apache Spark;
3. Descreva como as análises foram implementadas no Apache Spark (ou seja, a ideia geral do algoritmo que processa os dados).

A entrega será feita única e exclusivamente via e-Disciplinas, até a data final indicada na tarefa correspondente. Um (e apenas um) dos integrantes do grupo deve fazer a postagem de um arquivo `.zip` contendo o código-fonte do programa, além do relatório final da entrega. Não esqueça de indicar claramente todos os integrantes do grupo (e seus respectivos números USP) no relatório.

A responsabilidade de postagem é exclusivamente do grupo. Por isso, submeta e certifique-se de que o arquivo submetido é o correto (fazendo seu download, por exemplo). Problemas referentes ao uso do sistema devem ser resolvidos *com antecedência*.