Universidade de Brasília — Faculdade UnB Gama

Programação p Sistemas Paralelos e Distribuídos - Turma A Período: 2o./2021
Prof.: Fernando W. Cruz

Projeto de pesquisa - Programação de *Streaming* em clusters

A) Objetivos do projeto

O objetivo deste projeto é permitir que o aluno avance seus conhecimentos sobre arquitetura de clusters de processamento de *Streaming* e programação de aplicações para consumo e tratamento de eventos, em tempo real. Esses conhecimentos devem ser adquiridos por meio da configuração de um processador de *streaming* aliado a um servidor de mensageria para classificação dos eventos. Para atendimento a essa atividade, o projeto foi dividido em duas partes, descritas a seguir:

Parte 1 - Spark Streaming contabilizando palavras de entrada via socket

Nesse caso, os alunos devem instalar a API Apache Spark Streaming (https://spark.apache.org/streaming/) em um *cluster* (envolvendo mais de um *host*) e fazer uma aplicação que *(i)* consiga ler palavras enviadas a esse servidor à partir de um socket TCP ou UDP, *(ii)* contabilize o total de palavras recebidas pelo *socket* e o número de ocorrências de cada palavra, durante o tempo de atividade do servidor e *(iii)* apresente o resultado dessa contabilização em arquivo ou console, de modo que seja possível perceber a dinâmica de leitura/contabilização das entradas.

Observações:

- Para construção dessa aplicação, os alunos devem fazer uso da API Apache Spark Streaming e programação usando linguagem Python
- As palavras devem ser provenientes de um ou mais arquivos, em quantidade suficiente para que seja possível perceber a *streaming* de palavras chegando no servidor
- Para implementação do *socket* de leitura das palavras, sugere-se o uso da biblioteca Netcat (http://netcat.sourceforge.net), mas outras opções como websocket (https://pt.wikipedia.org/wiki/WebSocket) também podem ser utilizadas

Parte 2 - Spark Streaming contabilizando palavras via Apache Kafka

Nessa parte do projeto, os alunos devem montar uma infraestrutura de servidor (*frameworks* + *scripts* de contabilização) de modo que consiga ler a *streaming* de palavras e gerar as seguintes saídas:

a) Número total de palavras recebidas, considerando o intervalo de tempo que a aplicação se manteve ativa

b) Número total de ocorrências de cada palavra, durante o tempo de coleta e processamento da *streaming* de entrada

c) Número de ocorrências de palavras iniciadas por cada uma das seguintes letras: S, P e R, a cada intervalo de tempo. Exemplo: No último intervalo de 3 segundos, foram coletadas 500 palavras com a letra S, 400 com a letra P e 200 com a letra R. Esse resultado deve ser atualizado à medida que as palavras vão chegando no servidor.

d) Número de ocorrências de palavras contendo cada uma das seguintes quantidades de caracteres: 6, 8 e 11 a cada intervalo de tempo. Por exemplo, nos últimos 3 segundos 20 palavras de 6 caracteres, 30 palavras de 8 caracteres e 40 palavras de 11 caracteres. Esse resultado deve ser atualizado à medida que as palavras vão chegando no servidor.

Para atender a essa parte do projeto, o serviço de contabilização de palavras agora é formado por um Servidor de Mensageria Apache Kafka (https://kafka.apache.org) associado a um *framework* de processamento genérico, no caso o Apache Spark, além dos *scripts* (programas descritos em Python) para contabilizar e gerar os resultados solicitados. Algumas observações:

- A alimentação do Apache Kafka Server com a *streaming* de palavras pode ser feita à partir do *socket* montado para a Parte 1 deste projeto. As considerações sobre o volume de palavras ao longo do tempo, descritas na Parte 1, valem aqui também

- Diferente do que ocorreu na Parte 1, as informações processadas pela API Apache Spark Streaming agora devem ser lidas à partir do servidor Kafka montado para este projeto de pesquisa
- Sugere-se que os alunos façam separação das palavras em tópicos no Kafka, de modo a facilitar a contabilização e apresentação dos resultados solicitados pela aplicação
- A leitura dos dados do Kafka deve ser feita via API Spark Streaming. Além disso, a contabilização e apresentação de resultados deve ser feita também via Apache Spark.
- Os resultados da aplicação devem ser apresentados de modo que seja possível perceber, no tempo, a dinâmica de atualização dos quantitativos de palavras, de acordo com as classes descritas nos itens (c) e (d). Nesse caso, sugere-se a criação de gráficos para facilitar a demonstração das saídas obtidas.

C) Questões de Ordem

- O projeto pode ser feito por grupos de até 3 alunos
- O projeto deve ter os artefatos entregues no Moodle até 02/5/2022 e apresentado ao professor em data estabelecida
- A entrega deve ser composta por: *(i) slides* de apresentação, *(ii)* relatório do projeto (descrito adiante) e, *(iii)* código criado, instruções de uso e todas as informações necessárias para esclarecimento e uso da aplicação feita (postados no Moodle ou disponibilizados no GitHub)
- O relatório do projeto deve ter a seguinte estrutura:
  i) Introdução - Descrever contexto associado a uma descrição do problema e uma visão geral da solução apresentada no relatório
  ii) Metodologia utilizada - Como cada grupo se organizou para realizar o atividade, incluindo um roteiro sobre os encontros realizados e o que ficou resolvido em cada encontro).
  iii) Descrição da solução - Essa seção pode ser organizada de modo que fique claro os módulos (frameworks + scripts) usados em cada uma das partes do projeto, envolvendo os sockets, o Apache Kafka e o Apache Streaming (devidamente documentados quanto à instalação, configuração e ativação etc.). Deve ainda conter uma subseção que mostre a descrição dos *scripts* construídos para a conclusão dos requisitos desse projeto de pesquisa
  iv) Conclusão - Aqui deve constar resultados alcançados e limitações da solução final. Além disso, deve conter subseções relativas a cada membro do grupo para que possam se manifestar *(i)* sobre o projeto (aprendizado, sugestões de melhoria, comentários, etc.), *(ii)* sobre como participaram, e *(iii)* sobre autoavaliação - atribuição de a si uma nota de avaliação, em função da participação pessoal e do atendimento aos requisitos do projeto.
- A nota é individual e o valor máximo será dado ao aluno que demonstrar conhecimento sobre a solução e as plataformas usadas (Kafka e Spark, preferencialmente em modo cluster) e aprendizado satisfatório com o projeto. Outros requisitos como cumprimento das datas e nível de colaboração e balanceamento de atividades entre os membros do grupo também serão consideradas para emissão da nota final.
- Além dos elementos citados no tópico anterior, a nota desse projeto será calculada em função dos seguintes itens: *(i)* atendimento aos requisitos definidos, *(ii)* qualidade do relatório, *(iii)* qualidade da apresentação oral, *(iv)* nível de participação do aluno no projeto, e *(v)* funcionalidades extras no projeto, exemplificadas a seguir
- Funcionalidades extras são admitidas no projeto, em adição ao que foi solicitado. Por exemplo, gráficos dinâmicos como saída, são interessantes para mostrar a evolução de quantitativos ao longo do tempo (um gráfico apontando a média de palavras iniciadas com a letra S recebidas a cada intervalo de 3 segundos). Nesse caso, é importante documentar todo o ferramental usado para esse tipo de saída (biblioteca Spark utilizada, por exemplo)