Checkpoint 1

Matheus Carvalho Raimundo[1]

[1]Instituto De Ciências Matemáticas e de Computação – Universidade de São Paulo
São Carlos – SP – Brasil

mcarvalhor@usp.br

Abstract. *This article defines the project for the class of Cloud Computing and Service-Oriented Architecture, the technologies used and its architecture.*

Resumo. *Este artigo define o projeto da disciplina de Computação em Nuvem e Arquitetura Orientadas a Serviços, as tecnologias usadas e sua arquitetura.*

1. Projeto

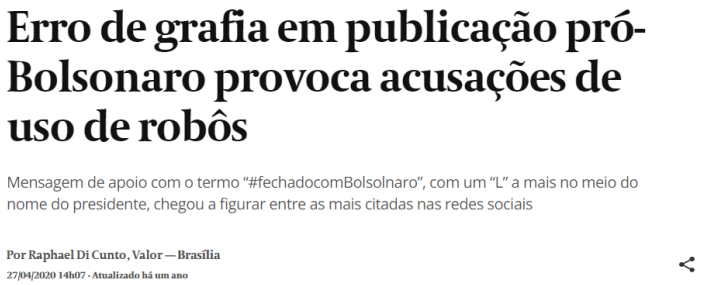
Nosso projeto consistirá de um sistema para analisar mensagens (chamados '*tweets*') publicadas na rede social Twitter.

É fornecido para o projeto uma *query* ou critério de busca. Tal critério de busca pode consistir de diversos fatores: uma frase, uma palavra, um tópico (chamado de *hashtag*, e sempre começa com '#' no Twitter) ou um usuário (sempre começa com '@' no Twitter). O sistema irá baixar todos os tweets que satisfazem tal busca usando da API do Twitter, e criar uma cópia local no banco de dados. Tal cópia consistirá apenas dos metadados - como conta que tweetou, geolocalização, data, entre outros. Após isso, é possível fazer análises destes metadados para determinar algum padrão entre eles.

Como exemplo específico, imagine que a *query* seja "#CPIdaCovid", um assunto muito comentado nas últimas semanas. O sistema irá primeiramente salvar todos os metadados de todos os tweets que satisfazem esta query. Após isso, nós vamos iniciar a análise para determinar, por exemplo, se a maioria dos usuários que tweetou sobre "#CPIdaCovid" são usuários que se cadastraram recentemente na rede social ou não, ou se são usuários que se concentram numa determinada geo-localização, etc. Com isso podemos determinar se há um padrão entre todos estes tweets.

1.1 Motivação e Objetivo

A motivação deste projeto vem de uma reportagem publicada com a seguinte manchete:



Erro de grafia em publicação pró-Bolsonaro provoca acusações de uso de robôs

Mensagem de apoio com o termo "#fechadocomBolsolnaro", com um "L" a mais no meio do nome do presidente, chegou a figurar entre as mais citadas nas redes sociais

Por Raphael Di Cunto, Valor — Brasília
27/04/2020 14h07 - Atualizado há um ano

Figura 1. Manchete publicada no jornal "Valor Econômico - Globo".

O objetivo é descobrir se nosso sistema será capaz de reunir informações o suficiente para ser possível determinar se uma "hashtag" é autêntica (impulsionada por usuários autênticos do Twitter) ou não (impulsionada através do uso de robôs), tendo como fonte apenas os metadados dos tweets. Não saberemos se este objetivo será cumprido - pois isto só será possível saber ao final do semestre quando todo o projeto estiver pronto e funcional. Mas o sistema se enquadra nos requisitos do projeto da disciplina, pois será necessário um servidor de banco de dados, um programa que se comunique com a API do Twitter e vários programas rodando de maneira paralela que fazem o processamento dos dados. Além disso, será usado o Apache Kafka, para que os dados provenientes da API do Twitter sejam enfileirados e, posteriormente, organizados no banco de dados.

2. Tecnologias Selecionadas

Nesta seção será detalhada as tecnologias que serão usadas no desenvolvimento do projeto.

2.1 Tweepy

A biblioteca do Python Tweepy será usada para se comunicar com a API do Twitter. Esta biblioteca é *open-source* e contém todas as chamadas que serão necessárias para o projeto. A biblioteca foi escolhida por ter uma extensa documentação e ser de simples implementação.

2.2 Apache Kafka

A plataforma *open-source* do Apache Kafka será usada para evitar atrasos por conta de congestionamentos do banco de dados. Durante a análise dos *tweets*, o produtor será o Gerenciador de Processamento, que enviará os *tweets* ao Kafka. Os consumidores serão os nós Processadores, que irão analisar os dados.

2.3 MySQL Server

O servidor de banco de dados MySQL Server será usado para armazenar todos os dados e as análises geradas por eles. O MySQL foi escolhido por ser um servidor de banco de dados relacional com potencial, eficiente e de rápida instalação e inicialização.

2.4 Flask

Será criada uma interface gráfica simples *web* para visualizar e trabalhar com os dados. O *framework* utilizado para esta interface *web* será o Flask, e o servidor *backend* utilizará do protocolo REST. O Flask será usado por ser um *framework* que simplifica a implementação do *backend* do servidor *web*, ao mesmo tempo em que tem alta escalabilidade.

3. Arquitetura do Projeto

Primeiramente, será requisitado uma busca por todos os tweets que satisfazem uma determinada *query*. O usuário acessa a interface *web* - que se comunica com o servidor *web* - e cadastra uma nova base de dados. O servidor *web* então vai criar a base no servidor de banco de dados e notificar o nosso programa responsável por se comunicar com a API do Twitter para que ele comece a coletar os *tweets* e salvar no banco de dados. Veja o diagrama de comunicação abaixo:

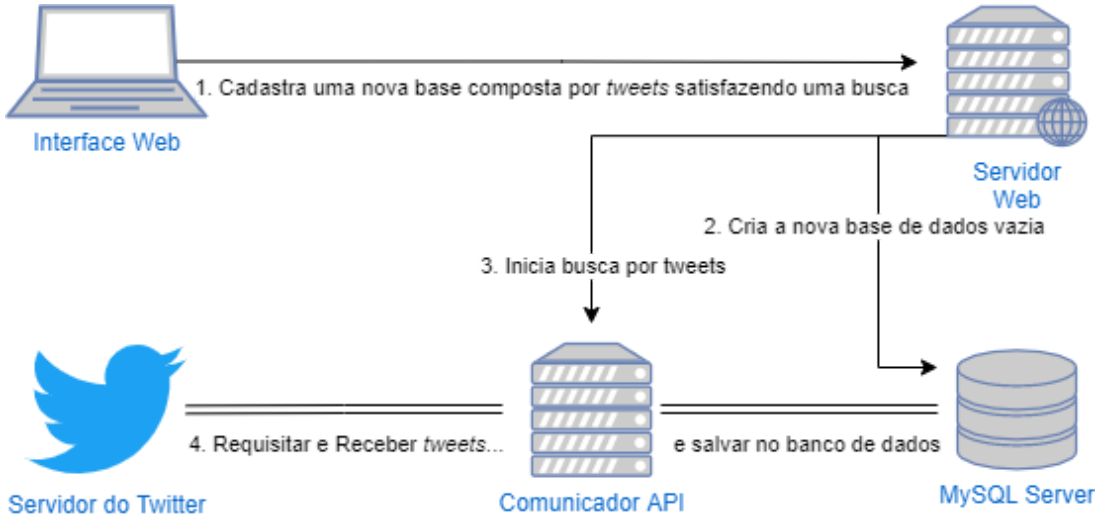


Figura 2. Diagrama de comunicação na criação de uma base de dados.

Os dados serão coletados até que o usuário deseje parar - ou seja, o Comunicador API vai continuar coletando *tweets* até o usuário ficar satisfeito, acessar a interface *web* e parar a coleta. Neste momento o Comunicador API vai parar de trabalhar.

Agora que temos uma base cheia de *tweets*, podemos iniciar o processamento deles. Neste caso, o usuário acessa a interface *web* e cadastra uma nova análise sob os dados coletados. O servidor *web* vai então criar a análise no banco de dados e notificar o Gerenciador de Processamento para que ele comece a distribuir o processamento dos dados entre cada um dos nós - chamados Processadores. Essa distribuição dos *tweets* entre os nós é dada pelo Apache Kafka. Veja o diagrama de comunicação abaixo:

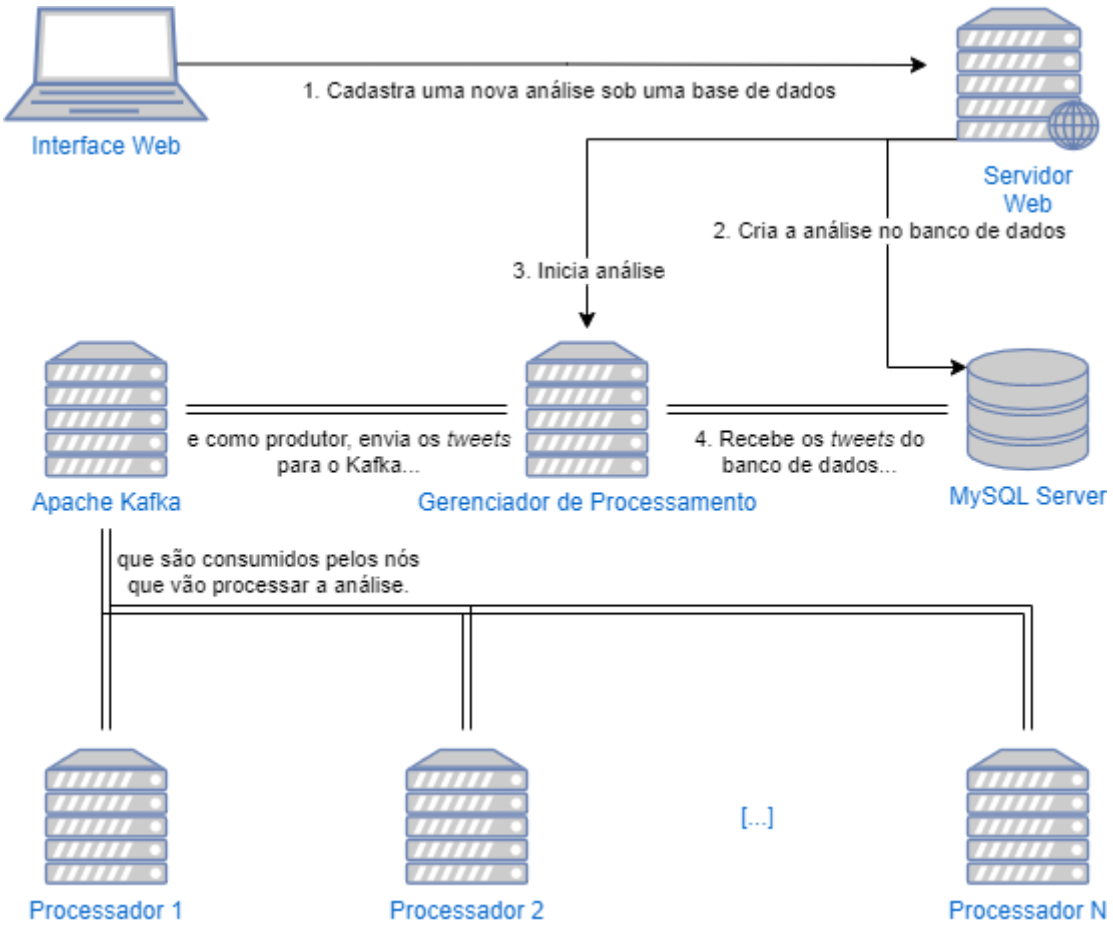


Figura 3. Diagrama de comunicação na análise dos *tweets*.

Quando a análise termina, o Gerenciador de Processamento apenas salva o resultado final no banco de dados MySQL.

Referências Bibliográficas

Universidade de São Paulo (2021). “Aulas de SSC0158 Computação em Nuvem e Arquitetura Orientadas a Serviços”, maio.

Raphael Di Cunto (2020) “Erro de grafia em publicação pró-Bolsonaro provoca acusações de uso de robôs”, https://valor.globo.com/politica/noticia/2020/04/27/erro-de-grafia-em-publicacao-pro-bolsonaro-levanta-acusacoes-de-uso-de-robos.ghtml, maio.

Twitter API v1.1 (2021) “Standard search API”, https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets, maio.