Mineração de Dados em Rede Social Baseada em uma Arquitetura em Nuvem

Lucas Gabriel Rezende de Jesus[1,4], **Alana Oliveira**[2,4], **Mario Meireles Teixeira**[3,4]

[1] Mestrado em Ciência da Computação – PPGCC

[2] Doutorado em Ciência da Computação – DCCMAPI
Coordenação de Engenharia da Computação

[3]Departamento de Informática – DEINF

[4]Laboratory of Advanced Web Systems – LAWS
Universidade Federal do Maranhão
Av. dos Portugueses, 1966 - Campus do Bacanga
CEP: 65080-805 – São Luís - MA

lgabrielrezende@gmail.com, alana@ecp.ufma.br, mario@deinf.ufma.br

Abstract. *Social networks have been called attention because they are an environment where more and more users are willing to express their opinions on the determining issue. Thus, it has become beneficial to process this information to generate applicable knowledge. Associated with this, so-called cloud computing has been touted as a solution to performance and scalability issues in systems that need to process large amounts of data. Given this scenario the purpose of this paper is to propose a methodology for social data mining based on a cloud architecture, using cloud provider services and techniques for data processing and sentiment analysis.*

Resumo. *As redes sociais vem chamado atenção por serem um ambiente em que cada vez mais usuários estão dispostos a expressar suas opiniões sobre determinando assunto. Dessa forma, tornou-se benéfico processar essas informações a fim de gerar conhecimento aplicável. Associado a isso, a chamada computação em nuvem tem sido apontada como solução para problemas de desempenho e escalabilidade em sistemas que necessitam processar grandes quantidades de dados. Diante desse cenário, o objetivo deste trabalho é propor uma metodologia para mineração de dados de redes socais baseada em uma arquitetura em nuvem, por meio da utilização de serviços de provedores de nuvem e de técnicas para processamento de dados e análise de sentimento.*

1. Introdução

Segundo estudos de 2016 da *Business Software Alliance* (BSA) aproximadamente 2,5 quintilhões de byte são gerados diariamente no mundo. Essa enorme quantidade de dados caracterizadas principalmente pelo o seu grande volume e diversidade é denominada *"Big Data"* e podem ser oriundas das mais variadas fontes, tais como sensores, publicações em sites, imagens digitais, vídeos, sinal de GPS e das chamadas redes sociais [Sri and Anusha 2016].

Dentre os mais diversos serviços disponíveis o *Twitter* chama atenção por sua simplicidade e objetividade, o usuário pode enviar e receber mensagens com até 280 caracteres denominados *tweets*. Seu recente crescimento aponta a necessidade de pessoas utilizarem a internet para expressar sua opinião de forma rápida sobre diversos assuntos. A exemplo disso, em 2018 o *Twitter* foi amplamente usado na vinculação de notícias sobre a Copa do Mundo [Henriques 2018] e em 2015 e 2016, nas manifestações sobre o impeachment da ex-presidente Dilma Roussef [de Camargo Penteado and Guerbali 2018]. Dessa forma, esses espaços virtuais estão sendo cada vez mais importantes fontes de informações, de modo que estudos e investimentos estão continuamente sendo direcionados na tentativa de encontrar valores nesse montante de dados [Bothos et al. 2010]. Nesse contexto, se tornou desafio coletar, armazenar e finalmente processar essas informações.

Uma arquitetura proposta por [Marz and Warren 2015] visa atender tanto o processamento de dados em lote quanto em velocidade. Segundo o autor, a arquitetura dividida em camadas, visa além de suprir as necessidades de um sistema robusto que possa compreender uma ampla carga de trabalho, responder com rapidez a fluxo de dados com baixa latência.

Tendo em vista a mineração de dados eficiente no contexto da análise de sentimento, neste trabalho propõe-se uma metodologia de mineração de dados baseada em ambiente de nuvem para viabilizar e favorecer o uso de sistemas complexos de processamento de dados. Com isso, objetiva-se diminuir o gargalo gerado pelo uso de arquiteturas e sistemas de armazenamento e processamento de dados comuns.

2. Trabalhos Relacionados

[Di Capua et al. 2015] propõe uma implementação da modificada arquitetura *Lambda* [Marz and Warren 2015] para realizar técnicas de aprendizado de máquina. A fim de executar a análise de sentimento em grandes fluxos de dados fornecidos pelo *Twitter*, o autor compara métodos de *Deep Learning* e NLP (*Natural Language Processing*). Ainda são apresentados algumas ferramentas utilizadas durante na metodologia como o *Apache Hadoop* e o *Apache Spark*. [Kiran et al. 2015] desenvolve uma proposta de implantação da arquitetura Lambda utilizando serviços de provedores em nuvem.

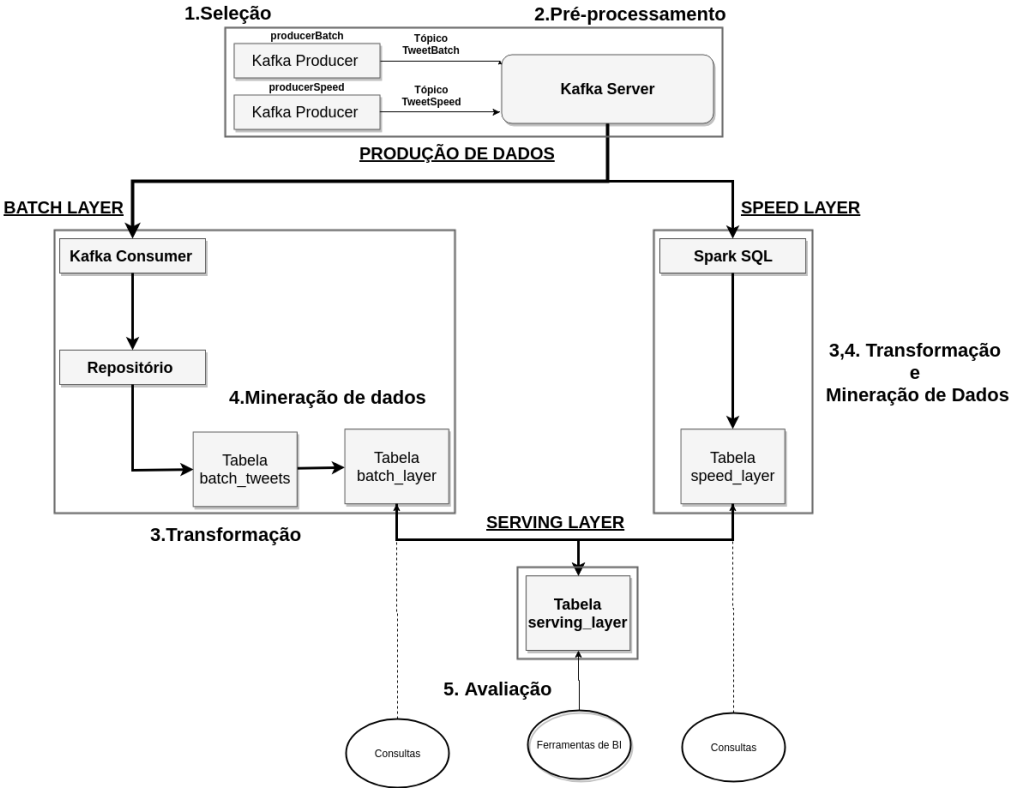
O trabalho de [Di Capua et al. 2015] apresenta um modelo para predição de bolsa de valores baseado em mineração de dados que também faz uso de NLP porém somado à métodos de aprendizagem supervisionada, como máquinas de vetor de suporte. Não obstante, o caso de estudo presente em [Ozdemir 2016] encoraja o uso de técnicas de ciência de dados para tentar produzir um modelo capaz de auxiliar na previsão de preços das ações baseada em mídias sociais.

[Corrêa et al. 2017] usa diferentes abordagens para a análise de sentimentos expressos por usuários do *Twitter* em relação aos filmes indicados à categoria de Melhor Filme do Oscar 2017, após avaliar as diversas abordagens foi escolhido o *Naive Bayes Multinomial* como algoritmo de classificação. [de Matos Galante and de Oliveira 2008] realizam um estudo sobre mineração de dados em redes sociais, dando enfoque no processo de descoberta de conhecimento. É feito um comparativo de técnicas, métodos e algoritmos adotados por artigos que serviram de base para seu trabalho.

3. Proposta de Arquitetura para Mineração de Dados

Nesta seção são apresentados os métodos desenvolvidos para a implementação da arquitetura proposta. O sistema foi construído utilizando soluções computacionais em ambiente nuvem e a API da rede social *Twitter* como fonte de dados. Cabe destacar as escolhas do Pyhton e Scala como linguagens de programação, dos *frameworks Apache Kafka* para a distribuição *streaming* de mensagens e *Spark* para o processamento de dados distribuído. A Figura 1 apresenta um esquema geral do modelo proposto.

Figura 1. Esquema da Proposta



3.1. Produção dos Dados

Esta camada atua como a fonte de dados do resto da arquitetura. Na metodologia presente neste trabalho, ela é responsável por fazer a seleção das variáveis de interesse da rede social, pelo pré-processamento destes dados e transmissão. Nesta etapa é utilizado o *Kafka* que permite a criação de fluxos distribuídos para as diferentes camadas da arquitetura. Diante desse cenário foi criado um fluxo de registros para a camada *batch* e outro para a camada *speed*.

Uma vantagem da implementação é a possibilidade de formular mensagens diferentes provenientes da mesma fonte de dados. Diante desse fato, é optado pelo envio do campo texto somente para a camada *batch* por seu propósito de permitir uma análise mais completa sobre o conjunto de dados. Em contrapartida, na camada de velocidade a baixa latência e alta taxa de atualização dos dados impediria a leitura do texto, sendo mais objetivo o envio dos outros dados da mensagem. Foram escolhidos 7 valores que influenciam diretamente no processo de análise de opinião e na mineração dos dados.

- *Id*: identificador único de cada *tweet*.
- *Text*: o texto propriamente dito.
- *Created_at*: data de criação do *tweet*.
- *User.followers_count*: número de seguidores do usuário que publicou a mensagem.
- *User.location*: localização geográfica do usuário.
- *Favorite_count*: quantidade de vezes que o *tweet* recebeu marcação de favorito.
- *Retweet_count*: quantidade de vezes que o tweet foi repostado.

No intuito de permitir a eficiência na etapa da mineração de dados e a fim de contornar as dificuldades de se realizar o mesmo procedimento de *machine learning* em camadas diferentes, na etapa de pré-processamento é realizada a análise de sentimento do texto dos *tweets*. Isso permite que os valores de polaridade das sentenças sejam enviadas no escopo da mensagem para as camadas que realizam a mineração dos dados (*Batch Layer* e *Speed Layer*), garantindo integridade desses dados.

3.2. Batch Layer

Nesta camada os dados de entrada são armazenados de forma permanente a fim de permitir uma precisão perfeita do processamento e a possibilidade de recálculo dos resultados. Na metodologia, todos os dados ficam contidos em repositórios de objetos na nuvem e são posteriormente transferidos para um banco de dados relacional, onde são estruturados, processados e fornecem visões completas sobre os dados provindos das etapas anteriores.

O fluxo de mensagens é consumido por uma classe consumidora do *Kafka* associado ao tópico *batch layer*. Todo esse tráfego é decodificado e armazenado em formato CSV em um repositório imutável e de alta disponibilidade de uma provedora nuvem organizado hierarquicamente por data. Logo após, os dados são enviados à uma instância de um banco de dados e carregados em uma tabela chamada *lote_tweets*. A nomenclatura das colunas segue o mesmo padrão dos campos dos arquivos de origem.

Posteriormente são executadas consultas em formato SQL sobre os valores da tabela por meio de cálculos sobre a totalidade dos dados a fim de produzir visualizações em lote (*batch views*) e associações entre os valores. A alta precisão é traduzida na possibilidade de correção de erros por meio do recálculo sobre o conjunto completo de dados e das atualizações das exibições. Segue abaixo uma lista das operações:

- Média Aritmética dos valores de sentimentos.
- Contagem dos id's.
- Soma dos valores de seguidores dos usuários.
- Soma das vezes favoritadas.
- Soma das quantidade de *retweet's*.

3.3. Speed Layer

O propósito da camada de velocidade é permitir uma análise em baixa latência dos *tweets* de forma eficiente. A ingestão dos dados é feita somente sobre o dia atual e sobre esses são feitas as mesmas agregações da camada em *batch*, porém em um escopo de dados diferente.É iniciada uma sessão *Spark* com o *Kafka* como o formato de *stream* e feito a conexão com o tópico *speed* criado. A configuração permite com que o ponto inicial de consulta seja os registros mais recentes.

Durante o processamento são feitas extrações dos valores dos *tweets* por meio de expressões SQL. Os dados são mapeados para os tipos de dados correspondentes e assim transformadas em RDD's, que são coleções de dados tolerante a falhas e podem ser operados em paralelo. No cenário, são feitos as mesmas operações de agregações e agrupamento da camada em lote, ou seja, contagem, soma e média agrupados por localização. Dessa forma, o formato da saída será o mesmo, o que possibilita a comparação dos resultados das camadas. A pesquisa por novos dados é por meio de uma *trigger*. Os registros são exportados para a tabela *speed layer* em um banco de dados instanciado na nuvem, o que permite a visualização diária dos dados.

3.4. Serving Layer

Nesta etapa, a saída corresponde a mesclagem entre os resultados das camada em lote e da camada de velocidade. No cenário isso possibilita agregações de dados dos *tweets* diários e históricos, servindo como uma espécie de camada intermediária entre as consultas e os dados. Sua composição é bem simples, uma tabela de dados somente leitura com uma tabela temporária preenchida pela *batch view* e outra atualizável a cada chamada provinda da *speed view*. Como exemplo foi usada a soma dos valores dos campos de cada camada agregados pela localização, ficando da seguinte forma:

- Soma das médias dos valores de sentimento.
- Soma da contagens dos id's.
- Soma dos números de seguidores.
- Soma da contagem de favoritos.
- Soma das quantidade de retweet's

3.5. Avaliação

Na metodologia proposta, a etapa de avaliação é servida pelos resultados dos processos das camadas *batch*, *speed* e *serving*. Diante dos dados, podem ser utilizadas ferramentas de *BI(business intelligence)* para a análise das informações permitindo o auxílio de decisões e melhorias de desempenho ou simples consultas estruturadas às tabelas de resultado das camadas.

4. Implantação em Ambiente de Nuvem

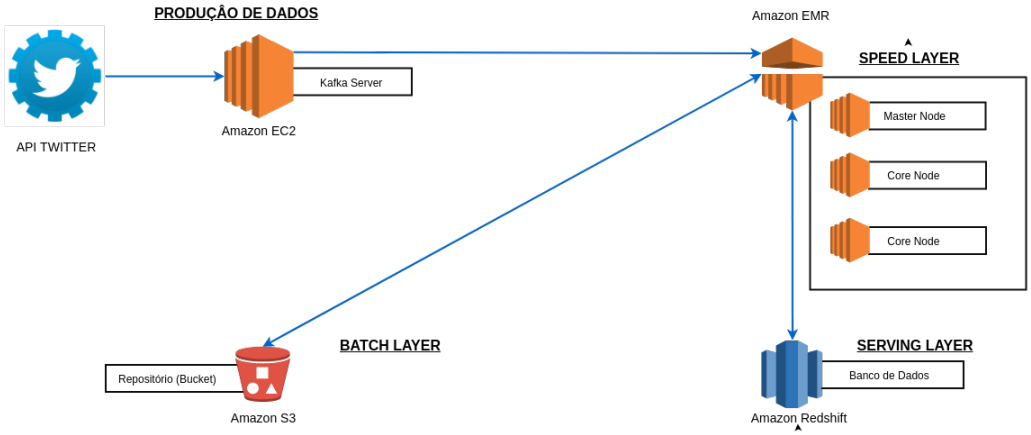
A implantação em ambiente nuvem visa atender os requisitos de escalabilidade e eficiência do sistema. São utilizados quatro serviços diferentes do provedor *Amazon Web Services* (AWS). A Figura 2 apresenta a disposição desses serviços na arquitetura.

Uma máquina virtual EC2 é instanciada a fim de exercer a produção dos dados. O *Kafka Server* permanece fazendo requisições a API do *Twitter*. Neste momento, um programa Pyhton realiza a seleção das variáveis e o pré-processamento.

Um *Cluster EMR* com três nós, sendo um master e dois core, realiza o processamento das outras três camadas. Cabe destacar que os nodes são máquinas virtuais, portanto são instâncias de EC2. Uma aplicação Pyhton é responsável pelo envio do *tweets* ao *bucket S3* (repositório) e pela comunicação com o banco de dados *RedShift*. Este recebe as consultas que são executadas e geram as visões em lote.

O *Spark* instancia um master node e dois core node na estrutura do *cluster*. Este conjunto de máquinas virtuais realiza o processamento em velocidade que em um intervalo definido repassa os resultados a uma tabela do banco de dados.

Figura 2. Implementação com serviços da AWS



O ponto central do *serving layer* é o banco de dados *RedShift*. Desse modo, uma aplicação pode realizar mesclagens dos dados de forma a obter uma visão diferente e mais completa sobre o escopo tratado. No cenário, um código Pyhton realiza esse procedimento como exemplo.

4.1. Estudo de Caso

Como estudo de caso, foram pesquisados por termos relacionados a grandes empresas presentes na bolsa de valores de Nova York (*New York Stock Exchange*). Pelo fato do algoritmo utilizado Naive Bayes ter sido previamente treinado com uma base de língua inglesa para a classificação, os *tweets* são nesse idioma.

Foram rodados experimentos no ambiente da AWS com instancias de máquinas virtuais seguindo configurações de *hardware* sugeridas pelo próprio provedor. A Tabela 1 apresenta as configurações utilizadas.

Tabela 1. Configurações de instâncias EC2

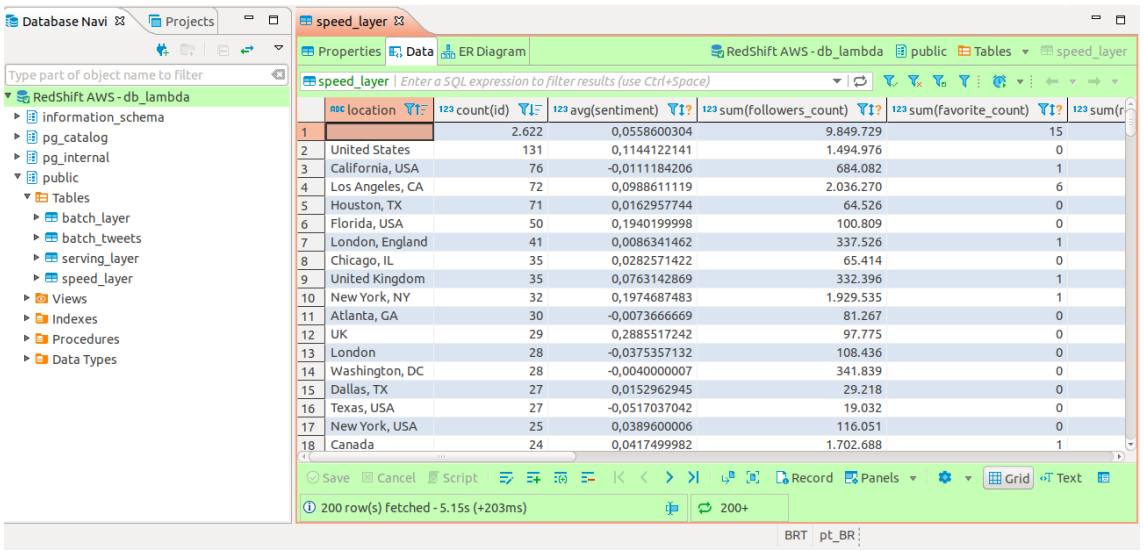
Tipo de instância	Configuração	Quantidade
m3.xlarge	8 vCore, 15 GiB memória, 80 SSD GB	3
t2.small	1 vCore, 2Gib memória	1

A Figura 3 apresenta a disposição dos primeiros registros, ordenados por localização e quantidade, em um experimento de duração de 7,2 horas. Nesse intervalo foram processados 8847 *tweets*. No exemplo, a pesquisa foi feita pelo termo *"Apple"*. Os dados podem então ser consultados por meio de ferramentas de consultas, permitindo associações sobre os resultados e assim auxiliar em processos de tomada de decisão.

A configuração do *Cluster* e máquinas adotada suportou todo processamento durante o intervalo de experimentação. A utilização de CPU do nó master responsável pelo gerenciamento das tarefas permaneceu em 10% e dos 16 vCores (núcleos de máquinas virtuais) foram utilizados 4.

As operações em disco foram minimizadas por conta do processamento em memória principal adotado pelo modelo distribuído do *Spark*. Este cenário ainda se

Figura 3. Disposição dos registros na tabela *speed_layer*



justifica quando é observado a quantidade de memória utilizada e disponível durante o experimento. De 22,50 Gb disponíveis foram utilizados 17,38 Gb.

5. Conclusão

Este trabalho apresentou uma proposta de mineração de dados em redes socais baseado em ambiente nuvem. A metodologia desenvolvida realizou as etapas do processo de descoberta de conhecimento mediante uma abordagem da arquitetura lambda, para tal foram utilizadas ferramentas de análise e distribuição de dados por meio de recursos computacionais em nuvem. A proposta mostrou-se eficiente tanto no que diz respeito à contemplar os estágios da mineração quanto no desempenho do processamento dos dados.

A seleção e pré-processamento realizados na camada de produção de dados em conjunto com a análise de sentimentos dos textos foram essenciais para atenuar a dificuldade de se implementar o mesmo algoritmo de classificação em fluxos diferentes. A transformação e mineração dos dados realizadas em camadas separadas e por ferramentas diferentes permitiu comparar o processamento em lote com o de velocidade.

Sobre a ótica do processamento dos dados, o emprego de serviços hospedados em ambiente nuvem foram cruciais para o desempenho e sucesso do sistema. A alocação e elasticidade dos recursos otimizaram o funcionamento do *Cluster Spark* que apesar de consumir um grande volume de memória principal, mostrou ser eficiente nas tarefas de processamento de dados. O amplo acesso à rede permite que a aplicação e a plataforma da AWS, fossem acessíveis pela *internet* e por meio desta realizar o monitoramento e medição dos serviços alocados.

Por fim, tendo em vista o processo de mineração em redes sociais e a computação em ambiente nuvem, este trabalho atinge o objetivo de contribuir com a literatura dessas áreas por meio da exposição de uma proposta que faz usos de técnicas e princípios relacionados tanto à analise de sentimento quanto a computação em nuvem.

É sugerido o aprimoramento das técnicas de análise de sentimento de forma que

possam realizar classificação textual em sentenças em português. Ainda é de grande-valia a realização de simulações com modelos diferentes da arquitetura lambda e serviços distintos de provedores nuvem.

Referências

Bothos, E., Apostolou, D., and Mentzas, G. (2010). Using social media to predict future events with agent-based markets. *IEEE Intelligent Systems*, (1).

Corrêa, I. T. et al. (2017). Análise dos sentimentos expressos na rede social twitter em relação aos filmes indicados ao oscar 2017.

de Camargo Penteado, C. L. and Guerbali, J. G. (2018). As manifestações do impeachment no twitter: uma análise sobre as manifestações de 2015. *Ponto-e-Vírgula: Revista de Ciências Sociais*, (19).

de Matos Galante, R. and de Oliveira, J. P. M. (2008). Um estudo sobre mineração de dados em redes sociais.

Di Capua, M., Di Nardo, E., and Petrosino, A. (2015). An architecture for sentiment analysis in twitter. In *Proceed. of Int. Conference on E-learning, Germany*, Germany. E-learning.

Henriques, G. (2018). Copa do mundo bomba no twitter com hashtags e jogadores; veja os mais mencionados. Acessado em : 2018-06-29.

Kiran, M., Murphy, P., Monga, I., Dugan, J., and Baveja, S. S. (2015). Lambda architecture for cost-effective batch and speed big data processing. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 2785–2792, Hong Kong. IEEE.

Marz, N. and Warren, J. (2015). *Big Data: Principles and best practices of scalable realtime data systems*. Manning Publications Co., New York.

Ozdemir, S. (2016). *Principles of Data Science*. Packt Publishing Ltd, Baltimore, Maryland.

Sri, P. A. and Anusha, M. (2016). Big data-survey. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, 4(1):74–80.