Extração de métricas sobre candidatos do Enem usando Apache Spark

Bruna Santos dos Santos[1], **Caue Gomes de Oliveira**[1],
Odorico Machado Mendizabal[2], **Regina Barwaldt**[1]

[1] Centro de Ciências Computacionais
Fundação Universidade Federal do Rio Grande (FURG) – Rio Grande – RS – Brasil

[2]Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

bruna.ecomp@gmail.com, caue@furg.br

odorico.mendizabal@ufsc.br, reginabarwaldt@furg.br

Resumo. *Com a popularização da Internet e a larga quantidade de dados gerados por serviços online, o processo de extração de informações destes serviços torna-se desafiadora, pois o grande volume de informação normalmente demanda um elevado custo de processamento. Este artigo faz uma análise do perfil dos candidatos do ENEM, do ano de 2016, explorando a ferramenta Apache Spark como forma de otimizar o desempenho na análise dos dados.*

1. Introdução

Diversas aplicações da Internet são caracterizadas por acesso contínuo de um número diversificado de usuários, com características e comportamentos específicos. Desta forma, um conhecimento sobre hábitos e preferências dos usuários torna-se um diferencial na busca por adequar serviços aos seus usuários. Este conhecimento normalmente é obtido pela análise de *logs*, que contém registros sobre o uso do serviço.

Arquivos *log* fornecem um histórico com dados e ações que ocorrem em um sistema. Estes registros geralmente são armazenados em arquivos de texto simples. Por exemplo, pode-se extrair informações sobre os usuários, como gênero, faixa etária e nacionalidade, ou ainda caracterizar o perfil destes usuários enquanto acessam o serviço, como por exemplo, identificar por quanto tempo permaneceram no site, quais funcionalidades foram acessadas e como eles navegam pela aplicação.

Lidar com uma alta quantidade de dados é um desafio para desenvolvedores e cientistas de dados, uma vez que a alta dimensionalidade de parâmetros, combinada com o tamanho elevado da amostra aumenta o custo computacional [Fan et al. 2014]. Ao invés de reduzir a amostra, soluções para tratar este alto volume de dados visam aumentar a capacidade de processamento, visto que quanto maior for o número de dados em um arquivo de *log*, melhores e mais precisos serão os resultados obtidos na análise.

Este trabalho faz uma análise do perfil dos candidatos inscritos no Enem (Exame Nacional do Ensino Médio) no ano de 2016 [INEP 2018]. Como forma de otimizar o tempo de processamento para extração de dados dos *logs* do Enem, foi implementado um analisador distribuído, usando o Apache Spark [Spark 2018] .

2. Perfil dos candidatos do Enem

O Exame Nacional do Ensino Médio (Enem)[1] é uma prova nacional utilizada para avaliar a qualidade do ensino médio no país. Além disso, a classificação dos candidatos lhes permitirão acesso ao ensino superior em universidades públicas brasileiras.

Para analisar o perfil dos candidatos foi utilizado o *log* com informações dos inscritos no ano de 2016, disponibilizado pelo INEP [INEP 2018] e obtido através do repositório de dados Kaggle [Kaggle 2018]. Com base nas informações obtidas no *log*, foram selecionados os seguintes atributos: distinção de candidatos por gênero, por região e por faixa etária.

O *log* do Enem classifica aproximadamente 170 atributos por candidato e contém mais de 8,6 milhões de linhas, que correspondem à 5.7 GB de informação armazenada. Para extração e contabilização de ocorrências de cada atributo, foi implementado um analisador distribuído, utilizando a ferramenta Apache Spark [Spark 2018]. O Apache Spark é uma ferramenta distribuída para processamento de alto desempenho, de uso geral com APIs em Scala, Java e Python e bibliotecas para streaming, processamento gráfico e aprendizado de máquina [Armbrust et al. 2015].

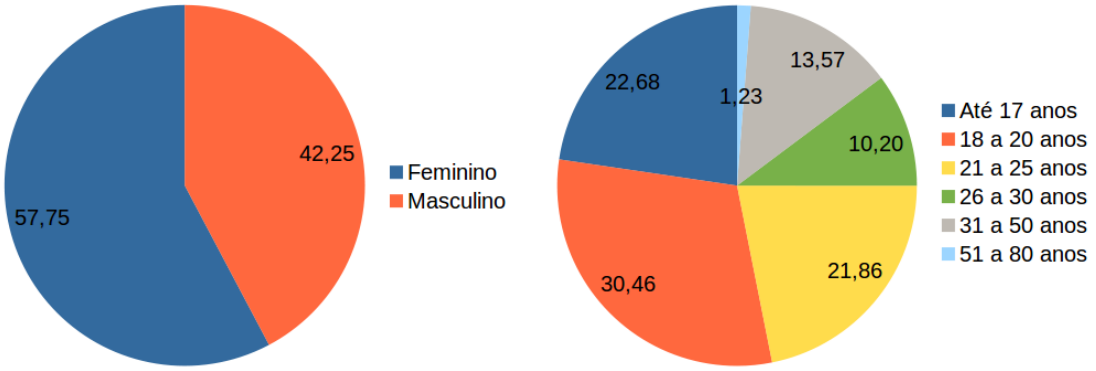
O analisador distribuído implementa o modelo *map-reduce*, um modelo de programação criado para processar grandes coleções de dados, dividindo o processamento em um grupo de serviços independentes. Primeiramente é feito um mapeamento (*mapping*) dos registros em função dos atributos sob análise. A fase de *mapping* é responsável por fazer a separação do *log* em blocos, quebrando as linhas em palavras e, como saída, é gerado um par *chave/valor* para cada uma destas palavras, contendo a palavra e a contagem por bloco. Por fim, a fase *Reduce* percorre os valores que estão associados com a chave e produz saídas com o número total de ocorrências para cada palavra. Em nossa análise, a função *Reduce* toma os valores do *log* do Enem, resume-os e gera uma única saída contendo o número total de ocorrências de cada atributo.

Para comparação de desempenho, foi implementada uma versão sequencial do analisador em Python. A versão sequencial trabalha com a manipulação de arquivo csv (arquivo em texto plano, com atributos separados por vírgulas), lendo cada linha e obtendo os dados das colunas de interesse.

A Figura 1 mostra os resultados obtidos. Foram contabilizados um total de 8.627.265 candidatos inscritos. Na distribuição por gênero, $57,75\%$ dos candidatos são do sexo feminino, enquanto $42,25\%$ são do sexo masculino. Na distribuição por faixa etária, a maioria dos candidatos têm entre 18 e 20 anos, contabilizando 30% dos candidatos. Foram identificados 37 candidatos com mais de 80 anos, sendo que o percentual de candidatos acima dos 30 anos atinge em torno de 15%.

A Tabela 1 apresenta a distribuição de candidatos por região. Pode-se observar que a maioria de candidatos encontra-se na região sudeste, contabilizando $35,72\%$ da população total, sendo São Paulo o estado com mais candidatos, contando com $16,3\%$ do total. O estado de Roraima foi o que apresentou menos participantes, com apenas $0,28\%$ do total.

[1] https://enem.inep.gov.br/



(a) Separação por gênero (b) Separação por faixa etária

Figura 1. Características dos candidatos do Enem (ano 2016).

Tabela 1. Distribuição de candidatos por região.

Estados	Total Candidatos	Percentual	Estados	Total Candidatos	Percentual
BA	665192	7,71	MG	954219	11,06
AM	193830	2,25	MA	326960	3,79
AL	146277	1,70	MT	163250	1,89
PR	420641	4,88	DF	166007	1,92
AP	60064	0,70	MS	140082	1,62
PB	222920	2,58	CE	516359	5,99
PA	441188	5,11	RR	24111	0,28
PE	441345	5,12	RS	419580	4,86
SP	1406003	16,30	RN	194740	2,26
PI	186808	2,17	RO	100101	1,16
ES	172858	2,00	GO	286980	3,33
TO	76601	0,89	RJ	548687	6,36
SC	176891	2,05	AC	60220	0,70
SE	115453	1,34			

3. Avaliação de desempenho

Para avaliar o desempenho do analisador de *log* distribuído, foram utilizados 6 computadores com sistema operacional Ubuntu 16.04 LTS-64 bits, com 4 GB de memória RAM e processador Intel core i5-4460 com 4 núcleos de processamento (3.2 GHz). Foi utilizado o Apache Spark, versão 2.4.0, Python 2.7.12, Java 8 e Scala 2.12.6. Um dos nodos executa de forma dedicada, como mestre, e os demais como nodos de trabalho.

Tendo os dois algoritmos implementados, foi executado o algoritmo distribuído, variando de 1 até 6 máquinas. A versão sequencial foi executada em uma máquina, e o tempo de execução ficou em torno de 3 minutos. A Figura 2 apresenta o *speed-up* obtido com o analisador distribuído. Pode-se observar que a versão distribuída chega a ser 7.2 vezes mais rápida que a sequencial. Ao adicionar mais de 6 nodos não se observou ganho expressivo de desempenho. Possivelmente o custo de comunicação e coordenação entre os nodos passa a não compensar para um número mais elevado de nodos.

4. Considerações Finais

Este trabalho apresentou uma análise do perfil dos candidatos do Enem no ano de 2016. Devido ao grande volume de dados e do *log* com registros dos candidatos estar represen-

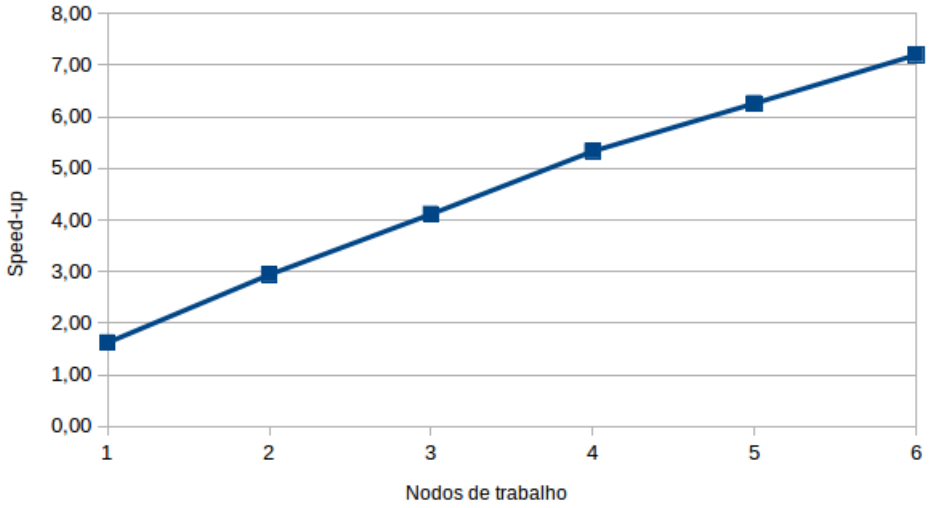


Figura 2. Speed-up.

tado em modo texto, o custo de processamento para extração de informações é elevado. Portanto, para esta análise de dados, foi implementado um analisador distribuído utilizando o Apache Spark. Pode-se observar um *speed-up* de 7,2 com a versão paralela.

Como trabalhos futuros pretende-se avaliar dados dos candidatos do Enem de outros anos. Utilizando estratégias de mineração de dados, espera-se, além de adquirir conhecimento sobre dados históricos, prever comportamentos futuros. Após adquirir conhecimento com este estudo de caso, pretende-se aplicar uma análise semelhante sobre *logs* da ferramenta Moodle na Universidade Federal do Rio Grande. Desta forma, será possível identificar perfis de uso dos alunos, tutores e professores, além de estimar e inferir a qualidade das respostas e interações entre estes agentes na plataforma.

Referências

Armbrust, M., Xin, R. S., Lian, C., Huai, Y., Liu, D., Bradley, J. K., Meng, X., Kaftan, T., Franklin, M. J., Ghodsi, A., and Zaharia, M. (2015). Spark sql: Relational data processing in spark. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 1383–1394, New York, NY, USA. ACM.

Fan, J., Han, F., and Liu, H. (2014). Challenges of big data analysis. *National Science Review*, 1(2):293–314.

INEP (2018). Departamento do ministério da educação do brasil. http://portal.inep.gov.br/microdados, acessado em 08/01/2019.

Kaggle (2018). The home of data science and machine learning. https://www.kaggle.com/gbonesso/enem-2016, acessado em 08/01/2019.

Spark (2018). Lightning-fast unified analytics engine. https://spark.apache.org/, acessado em 08/01/2019.