Arquitetura de Edge AI para Processamento de Big Data em Cooperativas Agroindustriais

Edge AI Architecture for Big Data Processing in Agro-Industrial Cooperatives

Hugo José Teixeira de Freitas *, Thiago França Naves†, Arlete Teresinha Beuren‡

RESUMO

Utilizar uma arquitetura de *Big Data* para coleta de dados e melhoria do processo de tomada de decisão é um dos grandes desafios do setor industrial. É preciso preciso considerar fatores como escalabilidade, disponibilidade, gerenciamento e segurança. Em setores como o das cooperativas agroindustriais esse desafio é ainda maior uma vez que os dados estão em diferentes formatos e possuem pontos de coletas físicos espalhados por diferentes regiões. Este trabalho apresenta uma arquitetura para gestão e processamento de grandes volumes de dados com uso de algoritmos de Aprendizado de Máquina, baseada no conceito de *Edge AI*. São utilizadas ferramentas da Apache como o Hadoop, Hive, Scoop e Spark que são configuradas para representar a arquitetura, além do Docker Swarm para virtualizar e distribuir o processamento de dados da mesma. São realizados experimentos para mensurar o *speedup* da arquitetura, que mostram a capacidade da mesma em realizar tarefas de aquisição e processamento de *Big Data*.

Palavras-chave: Big Data. Aprendizado de Máquina. Processamento de Dados. Cooperativa. Agroindustrial.

ABSTRACT

Using a Big Data architecture to collect data and improve the decision-making process is one of the major challenges in the industrial sector. It is necessary to consider factors such as scalability, availability, management and security. In sectors such as agro-industrial cooperatives, this challenge is even greater since the data are in different formats and have physical collection points spread across different regions. This work presents an architecture for managing and processing large volumes of data using Machine Learning algorithms, based on the concept of Edge AI. Apache tools such as Hadoop, Hive, Scoop and Spark are configured to represent the architecture, and Docker Swarm to virtualize and distribute the architecture's data processing. Experiments are performed to measure the architecture's Speedup, which show its ability to perform big data acquisition and processing tasks.

Keywords: Big Data. Machine Learning. Data Processing. Cooperative. Agroindustrial.

1 INTRODUÇÃO

A indústria brasileira gera um grande volume de dados dentro de suas atividades e organizações, que são artefatos valiosos no processo de análise para tomada de decisão (CODA et al., 2020). A coleta de dados em larga escala e sua organização e preparação para uso com algoritmos de Aprendizado de Máquina (AM) formam a base do *Big Data*, sendo a construção de uma arquitetura capaz de integrar estas atividades um dos maiores

* Coordenação de Ciência de Computação; hugofreitas@alunos.utfpr.edu.br; https://orcid.org/0000-0000-0000-0001.
† Coordenação de Ciência de Computação; naves@utfpr.edu.br; https://orcid.org/0000-0002-3152-1197.
‡ Coordenação de Ciência de Computação; arletebeuren@utfpr.edu.br; https://orcid.org/0000-0001-7565-6184 .

desafios para as indústrias (MOHAMMADPOOR; TORABI, 2020). Com isso, aplicações com armazenamento e processamento de dados em nuvem estão sendo exploradas, como uma alternativa capaz de automatizar parte do gerenciamento do *Big Data* (NAGY et al., 2021).

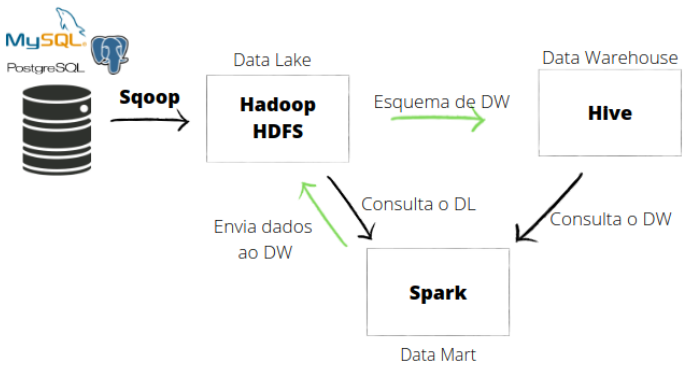
No setor de cooperativas agroindustriais os volumes de dados são grandes e advindos de diferentes fontes e locais físicos (GIAGNOCAVO; CÁCERES, 2019), elevando o custos da nuvem sem uma solução para consolidar a aquisição dos dados espalhados. Com isso, as chamadas arquiteturas de *Edge Computing* são uma alternativa para gestão de grandes quantidades de dados e uso de AM (SUN; LIU, J.; YUE, 2019). Essas são construídas utilizando técnicas e algoritmos integrados e utilizam recursos de *hardware* dentro das organizações. Fisicamente próximo as fontes de dados e com processamento direto, os tempos de resposta das aplicações são melhores (KHAN et al., 2019), oferecendo redução de custos e gestão customizada na coleta de dados.

O objetivo deste trabalho é a construção e validação de uma arquitetura de *Big Data* para cooperativas Agroindustriais baseada em *Edge AI* (LEE; TSUNG; WU, 2018). Esse termo deriva do *Edge Computing* e indica o foco no uso de algoritmos de AM na abordagem. A arquitetura proposta utiliza de computação distribuída para coleta, armazenamento e processamento dos dados, além de facilitar o uso de algoritmos de AM na análise destes. Uma base de dados com informações de abate e processamento de animais de corte é utilizada para testes com a arquitetura, além de experimentos que analisam a capacidade de processamento da mesma utilizando módulos distribuídos.

2 CONFIGURAÇÕES DA ARQUITETURA DE BIG DATA

A coleta de dados é a primeira atividade em uma arquitetura de *Big Data*. Em ambientes de cooperativas as fontes de dados podem estar em documentos físicos, em bancos de dados, e em dispositivos físicos de coletas como sensores e câmeras. A Figura 1 mostra um resumo dos principais componentes da arquitetura de *Big Data*. Para garantir robustez no processo de aquisição e ingestão de dados heterogêneos é utilizado o Hadoop Distributed File System (HDFS) (SUN; LIU, J.; YUE, 2019), que atua como *Data Lake* (DL) da arquitetura. Esse consegue escalar o uso de memória dependendo do formato e complexidade do dado, além de distribuir todos os pré-processamentos que venham a ser feitos com os dados armazenados.

Figura 1 – Arquitetura de *Big Data* e seu fluxo de operação.



Fonte: Autoria própria (2021).

No HDFS para lidar com importação de dados que venham de forma direta por outros SGBDs ou exportar aqueles já armazenados é utilizada a ferramenta Sqoop (CODA et al., 2020). Essa foca no transporte dos dados

que passaram por processamento de Extração, Transformação e Carregamento (ETL), entre Hadoop e o *Data Warehouse* (DW). Nesse contexto, o DW é construído utilizando a ferramenta Hive (WILLNER; GOWTHAM, 2020), que representa o banco onde os dados processados do HDFS estão organizados e podem ser consumidos por algoritmos de AM ou apresentados de plotagens gráficas. O Hive utiliza o HDFS como base em suas operações com os dados, garantindo um sistema distribuído com foco no desempenho.

Todos os processamentos de dados, sejam eles no DL com o Hadoop para os tratamentos de ETL ou utilizando os dados do DW no HIVE para análise com algoritmos de AM, são executados pelo Apache Spark (MENG et al., 2016). Esse torna-se o *Data Mart* (DM) da arquitetura e consegue trabalhar de forma distribuída criando *clusters* de nós que dividem a memória e processamento operando diretamente com os dados independente do seu volume. O Spark baseia-se em utilizar dois tipos de nós, o Spark Master e o Worker, onde o Master interage com o gerenciador de *clusters* para agendar trabalhos e realizar tarefas, e Worker para execução de tarefas agendadas.

Na arquitetura, o Hadoop é capaz de armazenar frações de dados em diferentes locais, composto pelo *NameNode* (*Master* do *Cluster*) e os *DataNodes* (*Slaves* do *Cluster*). O *NameNode* é responsável por armazenar informações a respeito dos dados, como metadados de armazenamento, numero de blocos utilizados e informações sobre os *DataNodes*, designando tarefas à eles. O *DataNodes* armazena os dados indica onde são realizadas operações de leitura, escrita e processamento. O *NameNode* executa tarefas de ingestão de dados através do Sqoop, por exemplo, e os armazena de maneira distribuída entre os *Data Nodes*. O Spark processa os dados armazenados nos *DataNodes*, podendo realizar tarefas de pré-processamento, Aprendizagem de Máquina e armazenamento em *Data Warehouses*.

3 GERENCIAMENTO DE PROCESSAMENTO AUTOMATIZADO

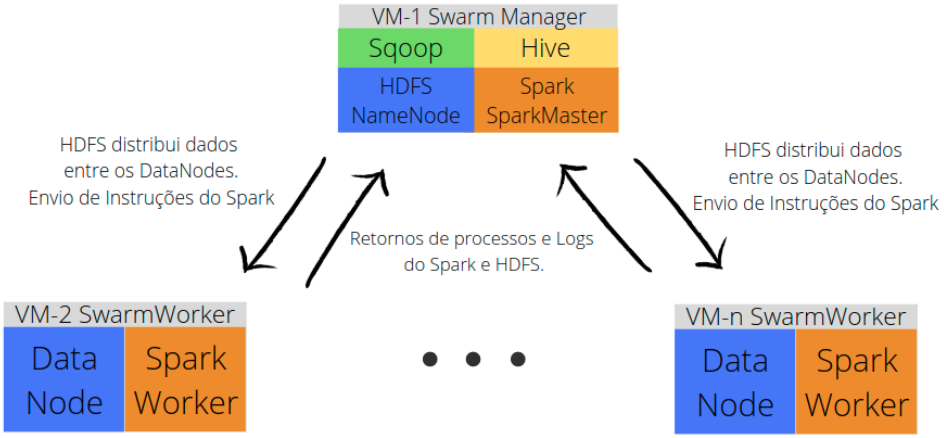
Com gerenciamento de dados distribuídos pelo HDFS e processamento paralelo nos algoritmos de AM pelo Spark, foi investigada uma forma de também distribuir o processamento da arquitetura completa de forma automatizada. Com isso, a arquitetura foi virtualizada utilizando o Docker Swarm (BARDEN, s.d.), que gerencia a mesma como um contêiner habilitando a configuração de *clusters* de processamento.

Com a virtualização o *Data Lake* gerenciado pelo HDFS precisa coletar e gerenciar volumes de dados que possam ser atribuídos a diferentes contêineres que venham a ser distribuídos. Para isso, foi criado um Docker Volume que separa volumes de discos exclusivos para diferentes aplicações, sendo automaticamente iniciado quando necessário tratar dados na arquitetura, onde recebe o endereço do volume e coloca este como seu disco. Assim, sempre que o serviço do HDFS é iniciado ele reconhece o volume e identifica os dados. A Figura 2 mostra a estrutura de virtualização do Swarm e como os *clusters* se comunicam quando a arquitetura está em execução.

Na Figura 2, um nó Manager (VM-1 Swarm Manager) sempre é criado para gerenciar a quantidade de Workers (VM-n Swarm Worker) desejável. Quando o Manager é iniciado automaticamente os Workers também o são, assim o Master do Hadoop que está contido no Manager abre comunicação com os Slaves, reconhecendo quem são os NameNode e DataNodes. Nesse ponto a comunicação entre os dados da arquitetura no HDFS e a distribuição de *clusters* para processamento distribuído está feita, sendo possível coletar e gerenciar dados no *Data Lake*.

Ao executar algum pré-processamento de ETL com os dados para armazenamento no *Data Warehouse*, o Spark Master é invocado e distribui entre os nós reconhecidos. Assim, os dados coletados do HDFS e pré-processados são enviados ao Hive, que os armazena e disponibiliza para que o Spark possa utilizá-los com

Figura 2 – Estrutura do Swarm e composição dos *clusters*.



Fonte: Autoria própria (2021).

algoritmos de AM de forma distribuída. Ao escolher algum algoritmo é possível utilizar os nós já construídos para executar o mesmo junto aos dados ou pode ser feito a gestão de novos *clusters* pelo Swarm Manager.

A arquitetura construída sob o Docker Swarm habilita as vantagens do uso de *Edge AI*, uma vez que todo o armazenamento/processamento é feito de forma local e distribuído. Todo a manipulação e movimentação é feita utilizando dados que podem estar em diferentes formatos e tamanhos, e a gestão destes entre as ferramentas e algoritmos são feitas utilizando volumes específicos de maneira rápida e controlada. O gerenciamento dos *clusters* que podem ser construídos tira proveito dos recursos computacionais que a organização possui, utilizando ao máximo esses de forma distribuída.

4 EXPERIMENTOS E ANÁLISE DOS RESULTADOS

Os experimentos foram realizados em uma máquina Linux com 16Gb de memória ram, processador Intel i7 2.60GHz e 8 núcleos. Foi configurado um ambiente em Docker na versão 20.10.8 onde foram conteinerizados o Sqoop 1.4.7, Hadoop HDFS 3.2.1, Spark 3.1.2 e Hive 3.1.2. São utilizados os algoritmos de AM Regressão Linear, Árvore de Decisão e Random Forest direto pelo Spark para a tarefa de predição de dados.Foram medidos os *speedups* de cada algoritmo para avaliar a eficácia no processamento distribuído de dados, utilizando 1 nó *Master* e variando os *Workers* de 1 a 8.

A base de dados é relativa ao abate de de animais de corte, contendo: tipo de inspeção; rebanho (Bois, Vacas, Novilhos, Novilhas, Vitelos e Vitelas, Suínos e Frangos); quantidade de informantes; cabeças abatidas; e quilogramas de carcaças. A base contém dados mensais de diversas regiões do país do período de 1997 a 2021. Os blocos de armazenamento do HDFS foram configurados em 128MB de tamanho com fator de replicação de 2, ou seja, entre os DataNodes haverá duas cópias dos dados.

A Tabela 1 apresenta os *speedups*. Nessa, a Regressão Linear apresenta um comportamento constante de aumento entre os nós de 2 a 6, ao utilizar 7 nós o algoritmo tem um aumento significativo em seu *speedup*. Esse comportamento pode está ligado a forma de processamento da curva de predição do algoritmo, que pode se beneficiar de um número maior de nós para construir mais amostras da mesma durante o processamento.

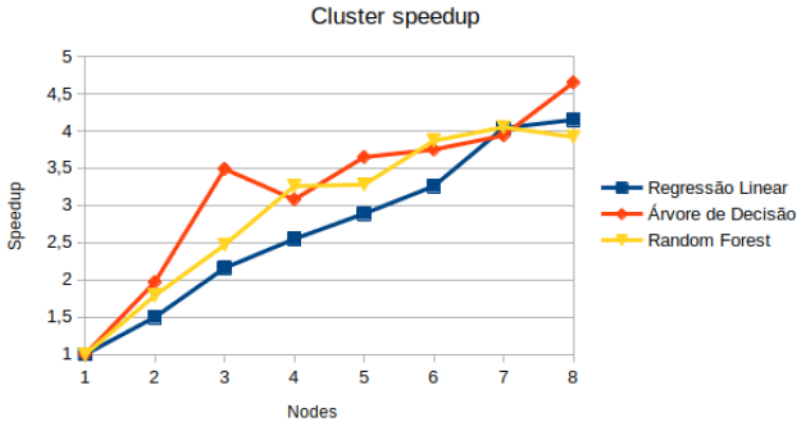
A Árvore de decisão apresenta um crescimento significativo já a partir do 3 nó, conseguindo o melhor *speedup* entre os algoritmos utilizando 8 nós. Esse é um exemplo de processamento que uma arquitetura distribuída traz

Tabela 1 – Speedup por algoritmo e quantidade de Workers.

Workers	Regressão Linear	Árvore de Decisão	Random Forest
1	1,00	1,00	1,00
2	1,50	1,97	1,79
3	2,16	3,49	2,47
4	2,55	3,08	3,26
5	2,89	3,65	3,28
6	3,26	3,75	3,87
7	4,04	3,94	4,05
8	4,15	4,65	3,92

Fonte: Autoria própria (2021).

Figura 3 – Grafico do speedup em função do número de workers.



Fonte: Autoria própria (2021).

benefícios mesmo com poucos nós a disposição, e pode melhorar a medida que novos recursos computacionais são adicionados para habilitar mais *clusters*. O Random Forest apresenta um crescimento constante entre os nós, com uma redução no seu *speedup* entre os nós 7 e 8. Esse pequeno decaimento pode ter ocorrido devido ao fato do algoritmo ter encontrado o resultado da regressão de forma mais rápida com 8 nós utilizando menos processamento de CPU.

A Figura 3 mostra graficamente o comportamento dos algoritmos e seus *speedups*. A Árvore de Decisão tem crescimentos acentuados em relação aos demais algoritmos com 3 e 8 nós, contudo o Random Forest consegue superar seus valores com 4 e 6 nós. Já a Regressão Linear fica abaixo dos outros algoritmos na maioria dos nós, mas consegue igualar aos demais com 7 nós aumentando significativamente seu *speedup*. Com isso, a arquitetura se mostra apta a processar grandes quantidades de dados, com melhoria no desempenho e gestão destes através da distribuição do processamento entre *clusters* que se beneficiam dos recursos computacionais disponíveis.

5 CONCLUSÃO

Neste trabalho foi apresentada uma arquitetura de processamento de dados para cooperativas agroindustriais baseada em *Edge AI*. Utilizando ferramentas como Hadoop, Hive, Scoop e Spark e configurando a comunicação entre estas, foi possível criar um ambiente para coleta e gestão de dados de diferentes formatos e fontes de coleta.

Com a incorporação do Docker Swarm a arquitetura foi virtualiza em contêineres com capacidade de distribuir o seu processamento entre *cluster*, que automatizam e melhoram o uso dos recursos computacionais disponíveis.

Nos experimentos os resultados mostram uma boa performance na gestão e processamento de grande volumes de dados utilizando algoritmos de Aprendizado de Máquina. Com isso, a solução apresentada mostra-se uma ótima opção para gestão de *Big Data* no setor indústria, em especial o de cooperativas que podem se beneficiar do conceito de *Edge AI*. Para melhorias futuras sugere-se testar a arquitetura proposta com bases de dados ainda mais volumosas e utilizar outros orquestradores de contêineres, como o *Kubernetes* por exemplo.

AGRADECIMENTOS

Ao Laboratório de Aprendizado de Máquina e Imagens Aplicados a Indústria - LAMIA[1], por toda a orientação e suporte. A Universidade Tecnológica Federal do Paraná e a Fundação Araucária, pela concessão de bolsa de pesquisa.

REFERÊNCIAS

BARDEN, Carla de Olivera. **Configuração de um ambiente escalável usando Docker Swarm para aplicações de Big Data**. [S.l.: s.n.]. Monografia Bacharel em Ciência da Computação, (UFSM) Universidade Federal de Santa Maria, Santa Maria, Brasil, 2021.

CODA, Felipe A et al. Modelagem e análise de uma arquitetura do sistema de aquisição de big data no contexto da Indústria 4.0. In: 1. CONGRESSO Brasileiro de Automática-CBA. [S.l.: s.n.], 2020. v. 2.

GIAGNOCAVO, Cynthia; CÁCERES, Daniel Hernández. **Creación de un nuevo bien común para las cooperativas agrıcolas: Big data, TIC e intercambio de datos**. [S.l.]: CIRIEC International, Université de Liège, 2019.

KHAN, Wazir Zada et al. Edge computing: A survey. **Future Generation Computer Systems**, Elsevier, v. 97, p. 219–235, 2019.

LEE, Yen-Lin; TSUNG, Pei-Kuei; WU, Max. Techology trend of edge AI. In: IEEE. 2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT). [S.l.: s.n.], 2018. P. 1–2.

MENG, Xiangrui et al. Mllib: Machine learning in apache spark. **The Journal of Machine Learning Research**, JMLR. org, v. 17, n. 1, p. 1235–1241, 2016.

MOHAMMADPOOR, Mehdi; TORABI, Farshid. Big Data analytics in oil and gas industry: An emerging trend. **Petroleum**, Elsevier, v. 6, n. 4, p. 321–328, 2020.

NAGY, Enikő et al. Cloud-agnostic architectures for machine learning based on Apache Spark. **Advances in Engineering Software**, Elsevier, v. 159, p. 103029, 2021.

SUN, Wen; LIU, Jiajia; YUE, Yanlin. AI-enhanced offloading in edge computing: When machine learning meets industrial IoT. **IEEE Network**, IEEE, v. 33, n. 5, p. 68–74, 2019.

WILLNER, Alexander; GOWTHAM, Varun. Toward a Reference Architecture Model for Industrial Edge Computing. **IEEE Communications Standards Magazine**, IEEE, v. 4, n. 4, p. 42–48, 2020.

[1] https://www.lamia.sh.utfpr.edu.br/