Processamento Distribuído na Borda: Um Estudo de Desafios e Oportunidades Utilizando Aparelhos de Tv Box Apreendidos pela Receita Federal

DANIEL L. SILVA[1], SAMUEL P. BARBOSA [2], FRANCISCO D. G. SILVA[3]

[1] Graduando em Engenharia de Controle e Automação, IFSP, Câmpus Salto, lourenco.daniel@aluno.ifsp.edu.br.
[2] Graduando em Engenharia de Controle e Automação, IFSP, Câmpus Salto, b.samuel@aluno.ifsp.edu.br.
[3] Prof.º Me. em Engenharia da Computação, IFSP, Câmpus Salto, diego@ifsp.edu.br.

Área de conhecimento: Computabilidade e Modelos de Computação – 1.03.01.01-1

RESUMO: Com o avanço de tendências como Internet das Coisas (*IoT*), computação em nuvem e dispositivos inteligentes, surge uma enorme geração de dados, ou big data. Isso traz desafios no processamento eficiente desses dados, principalmente em situações que exigem respostas ágeis e uso otimizado de recursos. A computação em nuvem tradicional tem limitações em latência e largura de banda, especialmente em aplicações *IoT* de tempo real. Este artigo aborda o processamento distribuído na borda para enfrentar esses desafios, processando dados mais perto da fonte. Usando TV Boxes apreendidos, associados a hardware de baixa capacidade, este estudo implementa um cluster com Apache Spark e CIFS. O foco é avaliar a eficácia do processamento distribuído em dispositivos menos potentes em cenários de *edge computing*. Os resultados mostram que, apesar das limitações, a estratégia reduziu em até 48% o tempo de execução ao distribuir a carga. O estudo também relaciona carga de trabalho, número de nós e tempo de execução nos testes.

PALAVRAS-CHAVE: processamento distribuído; computação na borda; internet das coisas; Apache Spark; e *big data*.

1 INTRODUÇÃO

A crescente interconexão da sociedade e a proliferação de dispositivos, como sensores e *smartphones*, têm gerado um aumento expressivo no volume e variedade de dados, impulsionado, por exemplo, por tendências como Internet das Coisas (IoT) e a expansão da computação em nuvem. Esse volume massivo de dados, muitas vezes referido como "*big data*", apresenta desafios substanciais em termos de infraestrutura, especialmente em transferência, armazenamento e processamento de dados, ocasionando em um consomo significativo de energia e largura de banda (LI et al., 2021; OUSSOUS et al., 2018).

A computação na borda emerge como uma solução viável, descentralizando o processamento de dados e permitindo que ocorra mais próximo à sua origem, otimizando a latência e uso da largura de banda (CAO et al., 2020). Essa descentralização, pode ser facilitada pelo processamento distribuído de dados, onde ferramentas como o Hadoop e

Apache Spark oferecem soluções eficientes para processar e analisar grandes conjuntos de dados (AHMED et al., 2020).

Em 2021, a Receita Federal do Brasil deu início ao programa “Além do Horizonte”, visando a destinação de receptores de sinal de TV pirata (popularmente conhecidos como TV Box), fruto de apreensões, para instituições de ensino realizarem a descaracterização do equipamento e sua respectiva transformação em minicomputadores. Dando segmento a tais parcerias, em 2023 foi firmado um protocolo de intenções com o Instituto Federal de São Paulo, visando o recebimento de tais materiais apreendidos para utilização em projetos e programas de ensino, pesquisa, extensão e inovação. Tais equipamentos, uma vez descaracterizados, tornam-se potenciais candidatos para integrar aplicações IoT e atuarem no processamento de dados na borda da rede.

Neste trabalho, são exploradas as oportunidades e desafios do processamento distribuído de dados no contexto de *edge computing* aplicado a computadores com baixo poder de processamento (TV Box descaracterizadas), destacando como essas tecnologias podem trabalhar em conjunto para potencializar aplicações de *big data*.

2 TEORIA

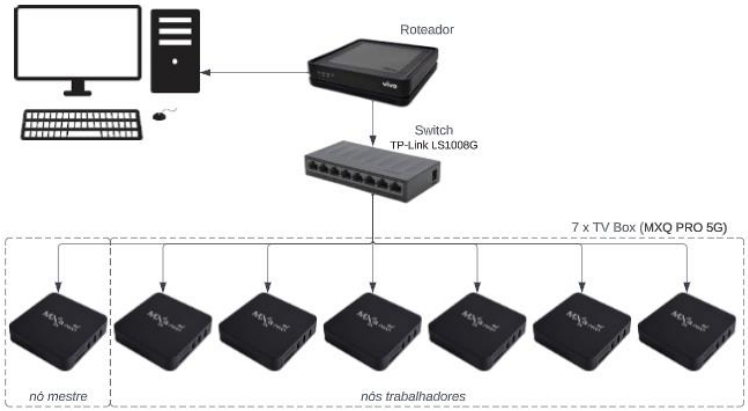
Diferente da computação em nuvem, a computação na borda move o processamento para dispositivos locais, economizando largura de banda e diminuindo a latência, crucial para aplicações em tempo real (CAO et al., 2020). Nesse cenário, para processar grandes volumes de dados, utiliza-se o processamento distribuído. *Frameworks* como Apache Spark distribuem tarefas em um cluster, garantindo paralelismo e eficiência (AHMED et al., 2020; LU et al., 2018). A eficácia desse método depende de acesso confiável aos dados no cluster. Sistemas como o CIFS permitem compartilhamento de arquivos em rede, garantindo disponibilidade de dados para todos os nós, enquanto o Spark gerencia a distribuição e execução de tarefas, criando uma infraestrutura unificada para análise distribuída.

3 MATERIAL E MÉTODOS

Para a realização dos experimentos deste estudo, sete TV Boxes modelo MXQ PRO 5G (processador Allwinner H3 ARMv7 e 1GB de RAM), foram utilizados e preparados através de um processo de descaracterização usando a imagem Sunvell-r69 do Armbian, através do processo de *boot* com cartões SD de 16GB e o software Etcher.

O *cluster* (Figura 1) foi montado com um TV Box como nó mestre e seis como nós trabalhadores, interconectados via switch TP-Link LS1008G e cabos de rede Cat.5E. Um computador desktop foi utilizado para acesso aos nós via SSH.

FIGURA 1 – Diagrama do cluster montado com sete receptores TV Box.



Fonte: elaborado pelo autor.

Após a montagem física do cluster, procedeu-se com a instalação e configuração do Apache Spark 3.4.1 em cada um dos equipamentos. O Apache Spark foi utilizado como principal ferramenta para os experimentos de processamento distribuído, assim como a configuração do cluster em modo *standalone*, um modo de cluster nativo do próprio Spark, não dependente de gerenciadores de cluster externos, como o YARN ou Apache Mesos.

Utilizando o CIFS para estabelecer um ponto de montagem comum e configurando um *bucket* no *Google Cloud Storage*, assegurou-se o acesso a um sistema de arquivos distribuído e uma entrega eficiente dos resultados processados, facilitando a integração com outras ferramentas e plataformas de nuvem.

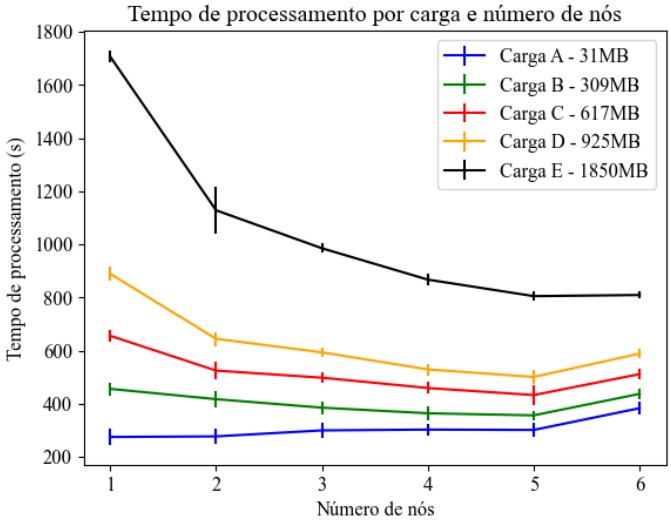
Para realização dos experimentos, foram preparadas cinco cargas de trabalho de diferentes tamanhos, sendo elas: carga A de 31MB; carga B de 309MB; carga C de 617MB; carga D de 925MB; e carga E de 1850MB. Todas as cargas contêm quatorze arquivos no formato CSV (*Comma-Separated Values*), conhecido por sua ampla aplicabilidade em variados contextos, além de ser de fácil interpretação e acesso. Para todos as cargas de trabalho, as mesmas tarefas de processamento foram realizadas, sendo estas: leitura dos arquivos; junção entre tabelas (*join*); renomeação de coluna; transformação de uma coluna inteira; e a escrita da carga processada no *bucket*. Foram

realizados cinco testes para cada cenário experimental com um tamanho de carga X e um número N de nós.

4 RESULTADOS E DISCUSSÃO

Através dos dados apresentados no gráfico da Figura 2, duas observações podem ser apontadas. A primeira observação, é que a eficiência do aumento do número de nós de um cluster depende fortemente do tamanho da carga. Em cargas maiores, como no caso da carga D e E, é inegável dizer que o tempo de processamento reduziu com o aumento do número de nós, enquanto para cargas menores, como a carga A, o aumento do número de nós representou um aumento do tempo de processamento. Esse fenômeno pode ser explicado considerando dois fatores principais: a sobrecarga de coordenação; e a inicialização e comunicação dos nós. Em sistemas de processamento distribuído, como o Apache Spark ou Hadoop, a coordenação entre nós irá introduzir uma sobrecarga, assim para cargas de trabalho menores, os processos operacionais de cada nó associados a sobrecarga de dividir o trabalho e coordenar tarefas do mestre quase sempre superarão os benefícios de distribuir a tarefa.

FIGURA 2 – Tempo de processamento das cargas para as diferentes quantidades de nós.



Fonte: elaborado pelo autor.

A fim de determinar a eficiência do cluster durante o experimento, e como forma de estabelecer uma relação entre as variáveis tamanho da carga e número de nós, se fez o uso de uma técnica estatística de regressão não-linear. Essa regressão (Equação 1) vem de modo a abstrair através de inferência estatística variáveis de perturbação relacionadas ao meio físico pelo qual se propaga a informação, planejamento e execução da

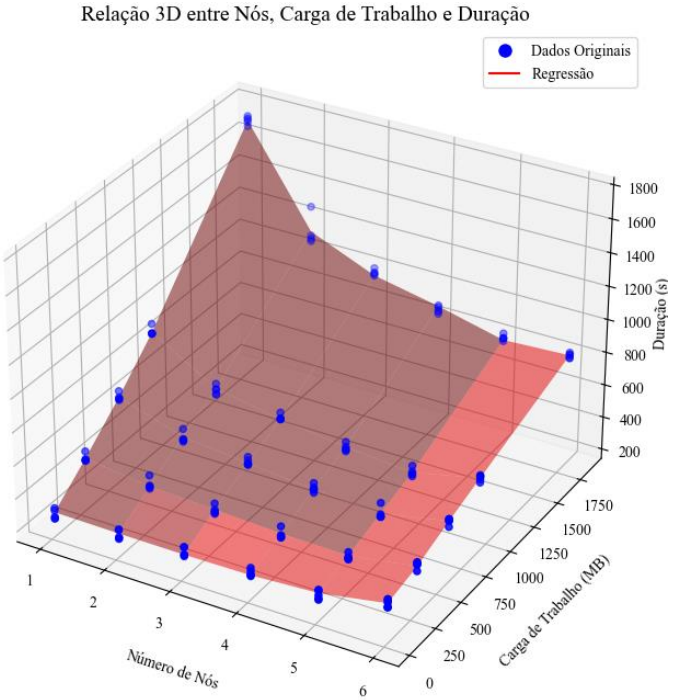
coordenação, dentre outras. A escolha da técnica se deu pela abordagem de GAVILANES (2020) que corrobora o uso de modelos de regressão para conjuntos de testes reduzidos, como o apresentado nesse experimento.

$$duration = 295.96 + (-0.01\ x\ nodes) + (-0.01\ x\ nodes^2) + (-0.65\ x\ size^2) + (0.01\ x\ nodes\ x\ size) \quad [1]$$

em que,
duration – Tempo de processamento, s;
nodes – número de nós;
size – tamanho da carga de trabalho, MB.

Vale salientar que o modelo apresentado (Figura 3) não tem como intenção representar exaustivamente todos os cenários, mas sim inferir o comportamento observado nos testes e oferecer uma forma de estimar possíveis relações entre carga, número de nós e tempo de processamento. Possuindo um erro quadrático médio de $743{,}63\ s^2$ pode se perceber que o modelo ainda sofre com *outliers* nos extremos do experimento, contudo, o modelo oferece uma boa capacidade de inferir sobre cenários que se distanciem da limitação do hardware ou sobrecarga de coordenação, exemplos de 1 nó e carga demasiada grande, e 6 nós e carga demasiado pequena respectivamente. Conclui-se que para melhorar a capacidade preditiva do modelo será necessário aumentar o universo amostral.

FIGURA 3 – Tempo de processamento pela carga de trabalho e número de nós.



Fonte: elaborado pelo autor.

5 CONSIDERAÇÕES FINAIS

Este trabalho destaca a viabilidade de usar dispositivos de baixo poder computacional, como TV Box, para *clusters* de processamento distribuído na borda da rede, com a implementação eficaz de tecnologias como Apache Spark e CIFS em hardware limitado, representando uma oportunidade no processamento de dados em cenários de aplicações de *big data*, como IoT, mitigando desafios de latência e largura de banda. As estratégias exploradas não só atendem às necessidades de processamento de dados, mas também possibilitam diversas aplicações práticas e futuras pesquisas, especialmente onde os recursos são limitados. Futuramente, este estudo comparará os resultados obtidos com o mesmo cenário utilizando computação em nuvem, avaliando aspectos como tempo de processamento e custos, além de explorar o processamento em tempo real de dados em streaming com Apache Spark, para expandir as capacidades do cluster e oferecer mais insights sobre as potencialidades e limitações do processamento distribuído em dispositivos de baixo poder computacional.

REFERÊNCIAS

AHMED, N. et al. A comprehensive performance analysis of Apache Hadoop and Apache Spark for large scale data sets using HiBench. **Journal of Big Data**, v. 7, n. 1, p. 110, 14 dez. 2020.

CAO, K. et al. An Overview on Edge Computing Research. **IEEE Access**, v. 8, p. 85714–85728, 2020.

GAVILANES, J. M. R. Low Sample Size and Regression: A Monte Carlo Approach. **Journal of Applied Economic Sciences (JAES)**, v. XV, n. 67, p. 22–44, 2020.

LI, W. et al. A Comprehensive Survey on Machine Learning-Based Big Data Analytics for IoT-Enabled Smart Healthcare System. **Mobile Networks and Applications**, v. 26, n. 1, p. 234–252, 1 fev. 2021.

LU, Z. et al. IoTDeM: An IoT Big Data-oriented MapReduce performance prediction extended model in multiple edge clouds. **Journal of Parallel and Distributed Computing**, Special Issue on Advanced Algorithms and Applications for IoT Cloud Computing Convergence. v. 118, p. 316–327, 1 ago. 2018.

OUSSOUS, A. et al. Big Data technologies: A survey. **Journal of King Saud University - Computer and Information Sciences**, v. 30, n. 4, p. 431–448, 1 out. 2018.