

# Controle automatizado de recursos computacionais em cluster com ESP32 e Naive Bayes para melhorar eficiência energética

Elias J. Miranda<sup>1</sup>, Marluce R. Pereira<sup>1</sup>

<sup>1</sup>Departamento de Computação Aplicada – ICET – Universidade Federal de Lavras  
Trevo Rotatório Prof. Edmir Sá Santos, Lavras - MG, 37203-202

elias.miranda@solahic.com.br, marluce@ufla.br

**Abstract.** This paper presents an approach to improve the energy efficiency of a low-cost cluster by automatically turning on and off machines. The cluster infrastructure used to test the solution includes four machines with Intel Celeron and wired network, Alpine Linux operating system and an ESP32 microcontroller connected to the motherboard of each node. A Naive Bayes algorithm evaluates the trend of CPU and memory usage activities on each node, allowing ESP32 to decide whether to turn on or off cluster nodes, ensuring high availability. The results were promising for the set of machines used.

**Resumo.** Este trabalho apresenta uma abordagem para melhorar a eficiência energética de um cluster de baixo custo através do acionamento automático para ligar e desligar as máquinas. A infraestrutura do cluster utilizado para testar a solução inclui quatro máquinas com processadores Intel Celeron e rede cabeada, sistema operacional Alpine Linux e um microcontrolador ESP32 conectado à placa-mãe de cada nó. Um algoritmo Naive Bayes avalia a tendência das atividades de uso de CPU e memória em cada nó, permitindo ao ESP32 decidir ligar ou desligar nós do cluster, garantindo alta disponibilidade. Os resultados foram promissores para o conjunto de máquinas utilizado.

## 1. Introdução

Sistemas computacionais demandam cada vez mais recursos para armazenamento e processamento dos dados gerados, exigindo a garantia de alta disponibilidade. Os *data centers* apresentam um elevado consumo de energia para manter todas as máquinas operantes. No entanto, pode ocorrer que algumas máquinas estejam ligadas, sem realizar computações, desperdiçando energia, reduzindo seu tempo de vida útil e agravando o problema do lixo eletrônico.

O uso de clusters de baixo custo, composto por computadores que não são top de linha mas que em conjunto permitem a resolução de problemas computacionais de forma mais rápida ou atendimento de muitas requisições, pode ser a solução para pesquisadores e até mesmo pequenas empresas que não podem adquirir máquinas top de linha. Esses clusters também precisam ser construídos visando o baixo consumo de energia.

Segundo [Katal 2023], as principais formas de reduzir o consumo de energia são: virtualização de software, virtualização de sistema operacional, uso de contêineres e estratégias de gerência de energia. Uma revisão da literatura sobre consolidação de máquinas virtuais é apresentada por [Dias et al. 2021].

A redução do consumo energético por meio de hipervisores e virtualização tipo 1, ou *bare metal*, proporciona o gerenciamento de máquinas virtuais e contêineres para prover a disponibilidade e eficiência energética, de forma manual, mas não promove o gerenciamento

automático. Ainda existem lacunas significativas de ociosidade em operações em cluster, com nós, contêineres e máquinas virtuais subutilizados [de Oliveira et al. 2023]. O uso de um algoritmo probabilístico associado a microcontroladores pode permitir o gerenciamento automático destes recursos.

O Naive Bayes é um algoritmo de classificação probabilística que se baseia no "Teorema de Bayes" [Wickramasinghe and Kalutarage 2021]. O trabalho de [Yang et al. 2017] utilizou esse algoritmo e alcançou uma precisão de 88%, sendo satisfatória em um sistema de monitoramento de carga.

Os microcontroladores têm sido utilizados para atuar de forma analógica [AlMadhoun 2023]. Eles gerenciam sinais detectados pelos sensores, executam programas e processam sinais para obter os resultados desejados e podem ser utilizados em diversas soluções visando eficiência energética.

Diante disso, este trabalho tem como objetivo apresentar uma abordagem para melhorar a eficiência energética em cluster de computadores de baixo custo utilizando o algoritmo Naive Bayes e o microcontrolador ESP32.

## 2. Metodologia

A solução proposta para melhorar a eficiência energética de um cluster de baixo custo consiste no ligamento e desligamento automático dos nós do cluster de acordo com a demanda. Para a tomada de decisão foi necessário armazenar logs de uso de CPU e memória de cada máquina e a execução do algoritmo Naive Bayes para tomada de decisão.

A infraestrutura de cluster utilizada para realizar os experimentos em um ambiente de rede local está ilustrada na Figura 1, sendo composto por 4 máquinas equipadas com processadores da família Intel Celeron, 2.66GHz, memória DDR2 de 2GB. A interface de rede cabeada utilizada foi RTL8201EL 10/100 e Wi-fi 802.11 b/g/n de 2.4 GHz (antena integrada).

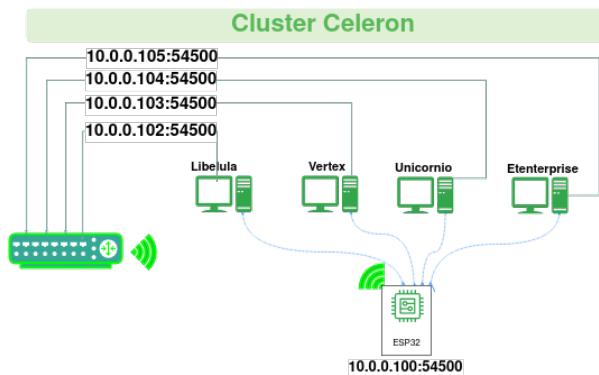


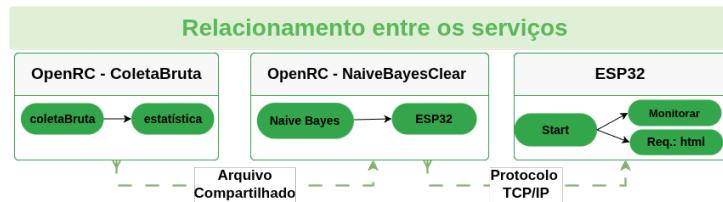
Figura 1. Representação do Cluster Celeron com 4 máquinas. Fonte: Do autor.

### 2.1. Instalação e adequação dos softwares

Nas máquinas que compõem o cluster foram instalados o Sistema Operacional Alpine Linux (versão 3.20.1), um Servidor Web Nginx para gerar requisições do cliente e os serviços *ColetaBruta* e *NaiveBayesClear*, e o script *StressServer*.

A Figura 2 apresenta o relacionamento entre os serviços **ColetaBruta**, **NaiveBayesClear** e o **ESP32** (processador Xtensa® Dual-Core de 32-bits) durante a operação do cluster.

**ColetaBruta** é um serviço (*daemon*) que possibilita a coleta contínua e sistemática de dados de uso da CPU, útil para o monitoramento de desempenho e análise de carga do sistema.



**Figura 2. Comunicação realizada entre os serviços. Fonte: Do autor.**

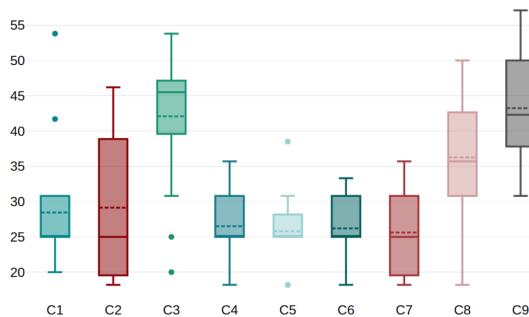
A coleta das atividades da CPU em tempo real gera dados para que o **NaiveBayesClear** possa prever o estado futuro da máquina e enviar para o **ESP32** para que haja uma decisão se a máquina precisará ser desligada. O **NaiveBayesClear** é um serviço que prediz e classifica o estado de trabalho das máquinas (**stress**, **ocioso** ou **ativo**) e envia para o **ESP32**, via socket.

O **ESP32** monitora as máquinas por meio de conexão serial, verificando se estão ligadas ou desligadas, para tomada de decisão. Os pinos 5 e 6 da placa mãe são ligados nos pinos do **ESP32** que envia para a placa mãe um sinal lógico alto por meio de uma conexão serial. Neste momento, o **ESP32** atua de forma paralela à função power (liga/desliga). Quando uma mensagem é recebida, o estado lógico do circuito é alterado proporcionando a ação desejada. Já o pino 1 e o pino 3 são conectados ao **ESP32** afim de monitorar o sinal lógico, informando o estado da máquina.

Neste trabalho, o cluster foi sobrecarregado por um serviço de estresse induzido. O **ESP32** detecta o estado atual das máquinas. O **ColetaBruta** obtém os dados do estresse e gera uma matriz de estatística. O **NaiveBayesClear** lê essa matriz de forma concorrente, classifica os dados e se comunica com o **ESP32** que, por sua vez, realiza a ação de ligar ou desligar nós do cluster.

### 3. Resultados

O cluster de baixo custo foi configurado em uma rede local conectando 4 máquinas por meio de cabeamento estruturado e um roteador. Foi realizado um experimento considerando o cluster desligado, em seguida as máquinas foram ligadas uma a uma de acordo com a sobrecarga de trabalho e desligadas quando estavam ociosas.



**Figura 3. Dados de percentual de uso da memória gerados pelo ColetaBruta. Fonte: Do autor.**

O experimento iniciou com o cluster desligado. No momento em que o **ESP32** recebeu a requisição por parte do cliente ele acionou a primeira máquina do cluster. O **StressServer** simulou a sobrecarga no sistema de 5 em 5 segundos até atingir a capacidade máxima da memória. O stress inicial induzido na memória foi de 128MB e foi sendo incrementando de 128MB em 128MB até atingir a capacidade máxima da memória, gerando o gráfico de coleta da Figura 3,

onde no eixo x encontra-se as coletas realizadas (C1 a C9) e no eixo y o percentual de uso da memória.

O *ColetaBruta* obteve cada  $C_i$  ( $1 \leq i \leq 9$ ) referente à atividade do processador e enviou para o *NaiveBayesClear* que classificou o estado da máquina em **ocioso** (C5), **ativo** (C1, C2, C4, C6, C7) ou **stress** (C3, C8, C9). Quando o estado gerado foi **stress**, foi enviada uma mensagem para o ESP32 e este ligou outra máquina. À medida em que foi chegando mensagens de *stress*, outras máquinas foram ligadas (após C3, C8 e C9). Quando, a atividade das máquinas foi reduzindo e o *NaiveBayesClear* enviou a mensagem **ocioso**, o ESP32 desligou uma das máquinas do cluster. E à medida em que a mensagem **ocioso** foi enviada novamente ao ESP32, outras máquinas foram desligadas. Quando o estado recebido foi ativo, não ocorreu nenhuma ação pelo ESP32, não ligando ou desligando máquinas sem analisar a carga das máquinas. Desta forma foi possível observar que a automação atingiu os resultados esperados.

#### 4. Conclusão

Apesar da virtualização em data center e cluster de computadores promover maior eficiência energética, ainda existem recursos subutilizados. Há uma preocupação com o melhor aproveitamento dos componentes de tecnologia, periféricos, redução da emissão de carbono e eficiência energética.

Este trabalho apresenta uma solução que promove a eficiência energética através do racionamento dos recursos computacionais em cluster com acionamento automático das máquinas. A implementação do cluster com 4 máquinas, combinada com os serviços *ColetaBruta*, *NaiveBayesClear* e ESP32, permitiu um controle flexível na automação do sistema. Esse sistema automatizado demonstrou ser uma solução sustentável para a gestão de cargas de trabalho variáveis, promovendo eficiência energética, além de aumentar a vida útil dos componentes. Como trabalhos futuros pretende-se realizar testes com mais nós e com workloads reais. Pretende-se também medir a eficiência energética utilizando biblioteca do sistema operacional e com medidor de energia analógico ou digital com isolamento da rede elétrica para comparação do consumo. Outra possibilidade é realizar a comparação do NaiveBayes com outros algoritmos.

#### Referências

- AlMadhoun, A. S. A. (2023). Microcontrolador. In *Guia de Início Rápido de Design e Simulação de Circuitos*, Série Maker Innovations, chapter 1. Apress.
- de Oliveira, A. C. A., Spohn, M. A., Fetzer, C., Do, L. Q., and Martin, A. (2023). Cost-based virtual machine scheduling for data-as-a-service. *Journal of Universal Computer Science*, 29(12):1461–1481.
- Dias, A. H. T., Correia, L. H. A., and Malheiros, N. (2021). A systematic literature review on virtual machine consolidation. *ACM Comput. Surv.*, 54(8).
- Katal, A., D. S. . C. T. (2023). Energy efficiency in cloud computing data centers: a survey on software technologies. *Cluster Computing*, 26:1845–1875.
- Wickramasinghe, I. and Kalutarage, H. (2021). Naive bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. *Soft Computing*, 25(3):2277–2293.
- Yang, C. C., Soh, C. S., and Yap, V. V. (2017). A non-intrusive appliance load monitoring for efficient energy consumption based on naive bayes classifier. *Sustainable Computing: Informatics and Systems*, 14:34–42.