

ARTIGO CURTO/SHORT PAPER

# Uma plataforma em nuvem para suporte à execução de MPI na educação

## A cloud-based platform for MPI execution support in education

**Felipe Barbedo** • ✉ fdbarbedo@inf.ufpel.edu.br  
Universidade Federal de Pelotas (UFPel)

**João Antonio Soares** • ✉ jansoares@inf.ufpel.edu.br  
Universidade Federal de Pelotas (UFPel)

**Gerson Cavalheiro** • ✉ gerson.cavalheiro@inf.ufpel.edu.br  
Universidade Federal de Pelotas (UFPel)

**Rafael Burlamaqui Amaral** • ✉ rafael.amaral@inf.ufpel.edu.br  
Universidade Federal de Pelotas (UFPel)

**RESUMO.** Computação em nuvem é um paradigma que oferece soluções para armazenamento e processamento de informações. No contexto educacional, a nuvem apresenta um enorme potencial para melhorar o aprendizado, especialmente em áreas como programação paralela e distribuída. Este trabalho descreve o desenvolvimento e a implementação de uma plataforma de programação paralela baseada em nuvem, projetada para fins educacionais, com foco no ensino de programação paralela e distribuída. Esta plataforma tem como objetivo simplificar a execução de programas *Message Passing Interface* (MPI) em um *cluster* de máquinas virtuais em uma nuvem privada, oferecendo uma interface que abstrai os detalhes complexos da configuração de ambientes distribuídos. Seu propósito é oferecer aos estudantes de ciência da computação um aprendizado mais direto, focando no desenvolvimento de algoritmos paralelos sem exigir conhecimento técnico da infraestrutura.

**PALAVRAS-CHAVE:** computação em nuvem • computação paralela e distribuída • interface web

## 1 Introdução

A computação em nuvem é um paradigma de computação distribuída que possibilita o acesso a recursos virtualizados, incluindo computadores, redes, armazenamento, plataformas de desenvolvimento ou aplicativos [1]. Na educação, a computação em nuvem oferece soluções para atender a necessidades específicas e fornecer serviços de aprendizado *online*, especialmente em cenários em que esses serviços são intensivos em computação (mundos virtuais, simulações, etc.) [2].

No entanto, emergem diversos desafios ao implementar serviços baseados em nuvem para a educação. Na Universidade Federal de Pelotas, um dos principais desafios está na disponibilidade dinâmica de recursos computacionais para atender às disciplinas de programação nos currículos da graduação. Além disso, a configuração de ambientes paralelos e distribuídos apresenta uma complexidade significativa para estudantes e educadores, exigindo conhecimento especializado que pode ser uma barreira para o aprendizado. Portanto, torna-se necessário desenvolver soluções que possam fornecer esses recursos de maneira acessível e amigável, abstraindo as dificuldades técnicas e permitindo que o foco dos estudantes esteja na implementação e

exploração dos conceitos de programação paralela.

As **contribuições** presentes neste trabalho são listadas a seguir:

- Implementação de uma plataforma de programação paralela em nuvem voltada para fins educacionais;
- Desenvolvimento de uma aplicação *web* com interface amigável para facilitar a execução de programas MPI em um *cluster* de máquinas virtuais;
- Abstração da complexidade de configuração de ambientes paralelos e distribuídos para estudantes;
- Criação de uma infraestrutura de nuvem privada utilizando *OpenStack* para suporte educacional.

O restante do texto está estruturado da seguinte forma: A Seção 2 descreve a arquitetura da nuvem privada e sua implementação. A Seção 3 apresenta a metodologia de avaliação e os resultados da pesquisa de usabilidade. Por último, a Seção 4 traz as conclusões e direções futuras para aprimorar a plataforma.

## 2 A Arquitetura Proposta de Nuvem Privada para Atividades Educacionais

A arquitetura da interface *web* proposta é modular, integrando um *front-end* amigável para submissão de

códigos MPI pelos estudantes e um *back-end* que gerencia as requisições e executa os códigos em um servidor dedicado. O *back-end* interage com um servidor configurado para compilar e distribuir as tarefas paralelas entre os nós de processamento, proporcionando uma experiência prática e intuitiva de processamento paralelo para os alunos.

A interface permite que os alunos submetam códigos e acompanhem os resultados em tempo real. O fluxo de execução envolve a submissão de arquivos pelo *front-end*, o processamento no *back-end*, e a comunicação com o servidor, retornando os resultados ao usuário. A integração do *back-end* com o sistema de nuvem privada garante uma gestão eficiente e segura do processamento MPI.

O sistema foi implementado em uma nuvem privada baseada no *OpenStack* [3], que permite a criação de um *cluster* de máquinas virtuais. A escolha do *OpenStack* se deve à sua flexibilidade, permitindo a escalabilidade necessária para o projeto. O uso do *DevStack* [4] facilitou a instalação e configuração da infraestrutura, proporcionando uma solução prática para testes e desenvolvimento.

O *hardware* utilizado incluiu três máquinas físicas com processadores Intel i5 e 16 GB de RAM, conectadas por um *switch* e um roteador. Cada máquina foi configurada com IPs estáticos, permitindo uma comunicação estável entre os nós do *cluster*. O *OpenStack* foi instalado em uma das máquinas como nó principal, enquanto as outras duas serviram como nós de trabalho, formando uma infraestrutura funcional para a execução dos códigos MPI.

Na Figura 1, é ilustrada a arquitetura simplificada

da plataforma, que envolve um *cluster* de máquinas virtuais gerenciado pelo *OpenStack*. Com a infraestrutura de nuvem configurada, o acesso ao *OpenStack* é feito via Horizon, sua interface gráfica. A rede foi configurada para que cada usuário tenha sua própria rede isolada dentro da nuvem, permitindo a execução de múltiplas requisições simultâneas de máquinas virtuais, assegurando o isolamento e a eficiência no processamento.

### 3 Metodologia de Avaliação

A ferramenta foi testada no semestre de 2023/2, em uma turma de Introdução à Programação Paralela e Distribuída (IPPD) da UFPel. Os alunos realizaram um trabalho final sobre programação paralela e distribuída com MPI. Vinte e cinco estudantes usaram o sistema e participaram de uma pesquisa de usabilidade. A experiência foi avaliada com o *System Usability Scale* (SUS) [5], utilizado para medir a usabilidade de tecnologias.

Os gráficos das Figuras 2 e 3 mostram as respostas dos 25 alunos, divididas para melhor visualização. A Figura 2 apresenta as respostas para as perguntas ímpares (formuladas positivamente), enquanto a Figura 3 mostra as respostas para as perguntas pares (formuladas negativamente). Cada barra no gráfico representa 100% da amostra, com as respostas indicadas por diferentes cores. O escore do SUS é calculado somando as contribuições de cada item (variando de 0 a 4, com itens ímpares avaliados positivamente e pares inversamente), multiplicando o total por 2,5, o que resultou em um escore final de 74,5, com base nas respostas de 25 usuários.

Em uma pesquisa, [6] investigou o significado das pontuações da Escala de Usabilidade de Sistemas (SUS)

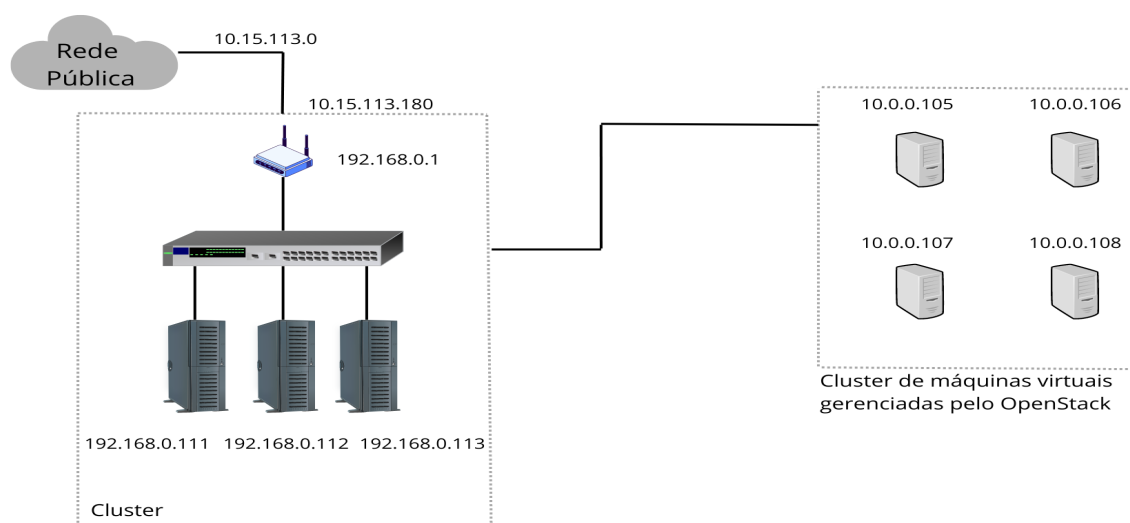


Figura 1. Arquitetura simplificada do Projeto com *cluster* de máquinas virtuais.

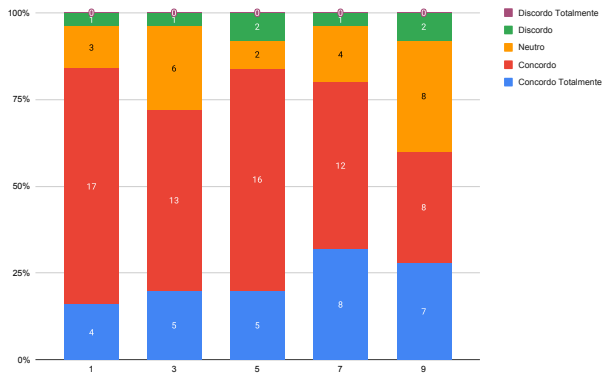


Figura 2. Resultado da aplicação da escala, perguntas com índices ímpares.

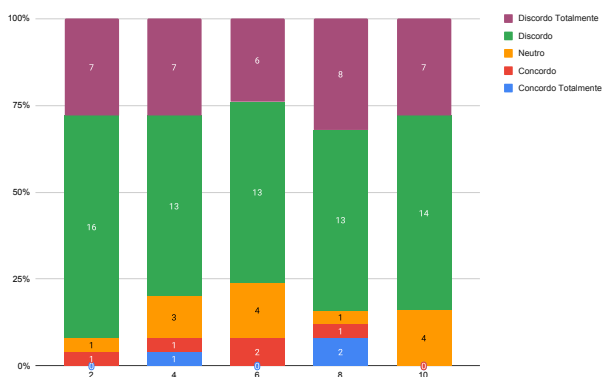


Figura 3. Resultado da aplicação da escala, perguntas com índices pares.

e propôs uma interpretação qualitativa. Quase 1.000 pesquisas SUS foram analisadas, correlacionando pontuações numéricas a adjetivos. Uma pontuação de 74,5 é considerada boa, indicando que o sistema é intuitivo e satisfatório. O estudo sugere que sistemas com essa pontuação tendem a ter maior aceitação e retenção de usuários, o que é relevante para disciplinas como IPPD.

#### 4 Conclusão e Direções Futuras

Neste trabalho foi apresentada uma plataforma de programação paralela em nuvem voltada para fins educacionais, utilizando tecnologias como Java, Spring Boot e Vaadin para criar uma interface *web* intuitiva e acessí-

vel. Os resultados preliminares sugerem que a aplicação desenvolvida satisfaz a necessidade de um ambiente escalável, padronizado e pronto para uso, superando as limitações do método tradicional de execução local. A complexidade de gerenciar e configurar o ambiente foi mitigada pela abstração das etapas de configuração de *clusters* e execução de códigos MPI, permitindo aos estudantes focar no desenvolvimento e otimização de algoritmos paralelos. Trabalhos futuros incluem o desenvolvimento de uma interface gráfica mais completa, com suporte a diferentes tipos de usuários e permissões variadas, além da criação de imagens de máquinas virtuais com diferentes configurações para melhor aproveitamento de *clusters*.

#### Declarações complementares

##### Disponibilidade de dados e materiais adicionais

A implementação proposta está disponível publicamente no repositório online: <https://github.com/FelipeBarbedo/cluster-webInterface-MPIExeApp>.

#### Referências

- 1 Mell, P. M.; Grance, T. *SP 800-145. The NIST Definition of Cloud Computing*. Gaithersburg, MD, USA, 2011.
- 2 González-Martínez, J. A. *et al.* Cloud computing and education: A state-of-the-art survey. *Computers & Education*, Elsevier, v. 80, p. 132–151, 2015.
- 3 OpenStack Foundation. *OpenStack Documentation, Version 2024.1*. 2024. Accessed: 2024-08-17. Disponível em: <https://docs.openstack.org/2024.1/>.
- 4 OpenStack Foundation. *DevStack Documentation*. 2024. Accessed: 2024-08-17. Disponível em: <https://docs.openstack.org/devstack/latest/>.
- 5 Brooke, J. *et al.* SUS-A quick and dirty usability scale. *Usability evaluation in industry*, London, England, v. 189, n. 194, p. 4–7, 1996.
- 6 Bangor, A.; Kortum, P.; Miller, J. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies*, Usability Professionals' Association Bloomingdale, IL, v. 4, n. 3, p. 114–123, 2009.