

Nimbus: uma arquitetura *serverless* em nuvem para correção automática de códigos

Gabriel R. Scheidt Paulino¹, Rafael Carvalho J. Araújo¹,
Johan M. G. da Rocha¹, Daniel Sundfeld Lima¹

¹Faculdade do Gama – Universidade de Brasília (UNB)
Campus Gama, Setor Leste - 72.444-240 - Gama - DF - Brasil

`gabrielscheidt2000@gmail.com, rrafaelcarvalhoj@gmail.com`

`johan.rocha@aluno.unb.br, daniel.sundfeld@unb.br`

Abstract. *Online code judging tools are important for programming competitions and computer science education, as they allow for the automatic evaluation of submitted solutions. Among them, CD-MOJ stands out as one of the most widely used, currently available in a local infrastructure. This work proposes the development of a new code judge, named Nimbus, which will be available in the cloud and based on CD-MOJ. Comparative tests were conducted between CD-MOJ and Nimbus. The results show that migration to a serverless environment is a viable for implementing online judges.*

Resumo. *As ferramentas de julgadores de código online são muito importantes para as competições de programação e ensino de computação pois permitem a avaliação automática das soluções submetidas. Dentre elas, o CD-MOJ se destaca como um dos mais utilizados, estando atualmente disponível em uma infraestrutura local. Este trabalho propõe o desenvolvimento de um novo julgador, denominado Nimbus, que será disponibilizado na nuvem e terá como base o CD-MOJ. Foram realizados testes comparativos entre o CD-MOJ e o Nimbus. Os resultados mostram que a migração para um ambiente serverless é uma alternativa viável para a implementação de julgadores online.*

1. Introdução

Os juízes online são sistemas automatizados essenciais em competições de programação e em ambientes acadêmicos, sendo amplamente utilizados para avaliar a correção de algoritmos submetidos por usuários. Esses sistemas executam os códigos em ambientes controlados, garantindo uma avaliação precisa com base em critérios rigorosos, como tempo de execução, uso de memória e segurança. A confiabilidade destes sistemas reside em sua infraestrutura homogênea, que proporciona uma avaliação consistente e precisa [Wasik et al. 2018].

Com o aumento da popularidade de competições como o International Collegiate Programming Contest (ICPC) [ICPC 2024], a necessidade de julgadores online escaláveis e eficientes tornou-se ainda mais evidente [Wasik et al. 2018]. Além disso, em ambientes acadêmicos o uso de plataformas de julgamento como o Contest-Driven Meta Online Judge (CD-MOJ) [Bruno C. Ribas 2024] facilita a avaliação do desempenho dos alunos e proporciona julgamento em tempo real das atividades.

A migração para uma infraestrutura de computação em nuvem surge como uma solução promissora. A computação em nuvem oferece escalabilidade, flexibilidade e economia, eliminando a necessidade de investimentos pesados em hardware [Veras 2017].

Neste artigo, propomos um corretor de código em nuvem, denominado Nimbus, tem como objetivo realizar todas as funções de correção automática de questões e código fonte na nuvem. Para isso, é utilizada uma infraestrutura da AWS. Ao empregar serviços como AWS Lambda [AWS Lambda 2024], S3 [Amazon S3 2024], DynamoDB [Amazon DynamoDB 2024] e API Gateway [Amazon API Gateway 2024], espera-se alcançar uma arquitetura escalável e eficiente, alinhada aos princípios da arquitetura *serverless*. Essa abordagem permite a execução sob demanda, melhorando a eficiência dos recursos e reduzindo custos, sendo particularmente adequada para sistemas que, como os juízes online, precisam lidar com cargas variáveis e alta demanda em tempo real [Cavalheiro and Schepke 2023].

2. Nimbus

Com os diversos benefícios oferecidos pela computação em nuvem, foi desenvolvido um novo julgador, hospedado em um container no AWS Lambda. Esse julgador, chamado Nimbus, é baseado nos scripts do CD-MOJ [Bruno C. Ribas 2024], que passaram por algumas modificações para viabilizar o julgamento de questões em um ambiente *serverless*.

O *handler* da função Lambda foi implementado em Python 3.12, sendo responsável por receber as submissões, processar as submissões e executar os scripts modificados. Para acionar o julgador, as execuções do Lambda são disparadas a partir de um *trigger* do API Gateway. Além disso, o sistema foi configurado para realizar até 1000 execuções simultâneas no Lambda. Com o objetivo de ter uma análise mais abrangente, exploramos uma funcionalidade do CD-MOJ que executa os casos de teste de uma submissão em modo paralelo. Essa funcionalidade não estava em uso no CD-MOJ, e por isso optamos por criar duas versões do Nimbus, uma em modo paralelo, e outra em modo sequencial.

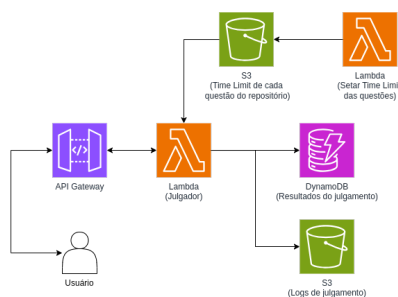


Figura 1. Fluxo de funcionamento do Nimbus na AWS.

A Figura 1 ilustra a arquitetura do fluxo de submissão no Nimbus, desde a interação inicial do usuário até o armazenamento dos resultados. O diagrama apresenta o caminho seguido pelos dados, começando no envio da submissão pelo usuário para o API Gateway, que aciona a execução da função Lambda. Em seguida, os resultados processados são armazenados de forma persistente no DynamoDB e no S3, compondo assim a solução final *serverless*.

Na arquitetura proposta, foram realizadas as submissões de 500 questões simultâneas. O CD-MOJ obteve uma média de tempo de 3.56s para o julgamento das

questões, enquanto o Nimbus judge Sequencial apresentou 5.66s e o Nimbus Paralelo apresentou 6.61s. O aumento do tempo pode ser explicado pela maior distância dos servidores locais e os da nuvem. Em todos os casos, os resultados produzidos pelo Nimbus e CD-MOJ foram iguais.

3. Conclusão

Este trabalho propôs uma arquitetura em nuvem sem servidor para o julgamento de questões. A implementação e análise do Nimbus demonstraram que os serviços *serverless* são uma alternativa viável para julgadores de código. Embora não tenha havido ganhos significativas nos tempos de execução para os casos que resultaram em aceite.

A migração para um ambiente *serverless* oferece benefícios como escalabilidade e alta disponibilidade, sem comprometer a eficiência do julgamento de códigos. O uso de funções Lambda impõe certas limitações, como restrições de tempo de execução e a necessidade de adaptar scripts para esse ambiente, o que até certo ponto, pode afetar a execução dos casos de teste.

Como trabalhos futuros, pretende-se otimizar o Nimbus para melhorar ainda mais seu desempenho, explorar diferentes configurações de paralelismo e avaliar seu comportamento com uma variedade maior de problemas. Além disso, a integração do Nimbus com plataformas educacionais e de competições pode ampliar seu uso e contribuir para a evolução dos julgadores online.

4. Agradecimentos

Este trabalho foi parcialmente financiado pelo CNPq (Processo 400038/2023-4) e UnB/FAPDF (EDITAL 05/2022 – PIBITI).

Referências

- Amazon API Gateway (2024). Disponível em: <https://aws.amazon.com/pt/api-gateway/>. Acesso em: 27 set. 2024.
- Amazon DynamoDB (2024). Disponível em: <https://aws.amazon.com/pt/dynamodb/>. Acesso em: 27 set. 2024.
- Amazon S3 (2024). Disponível em: <https://aws.amazon.com/pt/s3/>. Acesso em: 27 set. 2024.
- AWS Lambda (2024). Disponível em: <https://aws.amazon.com/pt/lambda/>. Acesso em: 27 set. 2024.
- Bruno C. Ribas (2024). Contest-Driven Meta Online Judge (CD-MOJ). Disponível em: <https://cd-moj.github.io/cd-moj.docs>. Acesso em: 21 set. 2024.
- Cavalheiro, A. P. and Schepke, C. (2023). Exploring the serverless first strategy in cloud application development. *SBAC-PADW*.
- ICPC (2024). International collegiate programming contest. Disponível em: <https://icpc.global/>. Acesso em: 27 set. 2024.
- Veras, M. (2017). *Computação em nuvem*. Brasport.
- Wasik, S., Antczak, M., Badura, J., Laskowski, A., and Sternal, T. (2018). A survey on online judge systems and their applications. *ACM Comput. Surv.*, 51(1):34.