

Consolidação de uma ambiente computacional para a simulação de ondas eletromagnéticas

Consolidation of an computational environment for electromagnetic waves simulation

RESUMO

Luís Guilherme Medeiros de Farias
luisfarias@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

Cassio Henrique dos Santos Amador
cassioamador@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

O trabalho "Preparação de um ambiente para estudo e simulação de ondas eletromagnéticas em uma antena" é parte de um projeto de Iniciação Científica e Trabalho de Conclusão de curso, cujo objetivo é utilizar ferramentas de simulação para investigar os parâmetros da reflectometria de um plasma e com os presentes dados, utilizar-se de técnicas de redes neurais para se chegar ao perfil de densidade. O presente trabalho investiga a arquitetura utilizada para para a criação do ambiente de simulação e todos os pacotes necessários para que o projeto seja realizado computacionalmente. Apresenta-se neste trabalho o ambiente em funcionamento. Finalmente, são abordadas as conclusões sobre os resultados obtidos, identificando também os próximos passos a serem realizados.

PALAVRAS-CHAVE: Python, Compreensão de Dados, Sistemas Operacionais

Recebido: 19 ago. 2020.

Aprovado: 01 out. 2020.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



ABSTRACT

The project "Preparation of an environment for the study and simulation of electromagnetic waves in an antenna" is part of a Scientific Initiation project and Course Conclusion Project, whose objective is to use simulation tools to investigate the parameters of plasma reflectometry and with the present data, use neural network techniques to reach the plasma density profile. The present work investigates the architecture used to create the simulation environment and all the packages necessary for the project to be carried out computationally. This work presents the operating environment. Finally, conclusions about the results obtained are addressed, also identifying the next steps to be taken.

KEYWORDS: Python, Data Comprehension, Operational Systems



INTRODUÇÃO

A aplicação por meio de softwares para a investigação de dados científicos que demandam de alto poder computacional é muito comum atualmente. Nota-se que a computação mais inserida na sociedade, deu aos cientistas uma maior acessibilidade para a apuração e previsão de seus estudos.

Existe um impasse ainda vigente no meio do caminho, onde os ambientes de desenvolvimento não se tornam globais, ou seja, o funcionamento do sistema criado pode ser comprometido por problemas de compatibilidade, versionamento de pacotes ou sistemas operacionais.

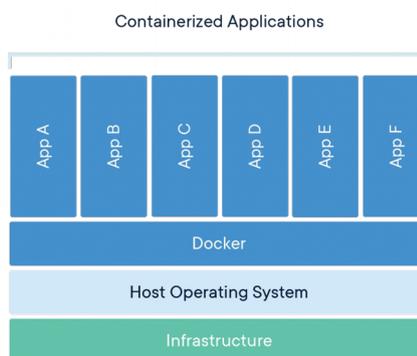
Nesse cenário, será demonstrado uma arquitetura de ferramentas utilizadas para o desenvolvimento de um projeto de simulação de um guia de onda, com uma fonte de onda eletromagnética sendo emitida. Sendo assim, objetivo será apresentar e investigar sobre os métodos utilizados para a criação do ambiente até a simulação do guia de onda.

MATERIAL E MÉTODOS

É essencial a compreensão da ferramenta Docker. É um plataforma que utiliza do conceito de virtualização para a entrega um pacote chamado Container, onde serve de base para a montagem de um software. Sendo assim, esse pacote unitário de software carrega as suas dependências, independentemente do sistema operacional (Docker Inc., 2020).

O contexto de contêineres em Docker é ilustrado da seguinte forma:

Figura 1 - Modelo conceitual de um Docker



Fonte:

<https://www.docker.com/resources/what-container>

Em conjunto, é configurado um ambiente Anaconda, nele contém um compilado de bibliotecas abordadas principalmente no contexto de ciência de

dados. Este é o local onde será criado um ecossistema com base no PyMeep. Vale destacar, que a linguagem utilizada será Python, a mesma permite uma melhor interação com pacote Meep.

O Meep é a biblioteca que aplica a técnica de Diferenças Finitas no Domínio do Tempo (FDTD, do acrônimo em Inglês de *Finite Difference Time Domain*), método utilizado para uma melhor visualização da atividade eletromagnética em decorrência do tempo. Portanto, é a ferramenta que será utilizada para a simulação eletromagnética.

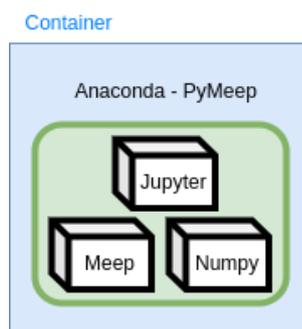
Futuramente, no desenvolvimento da simulação, será preciso converter o sinal para uma representação pela frequência, utilizando um algoritmo de Transformada Rápida de Fourier (FFT). Sendo assim, é necessário o pacote Numpy, um dos diversos pacotes disponíveis que contem a função correspondente para a transformada.

De início, será utilizado o Jupyter Notebook para o desenvolvimento do projeto, onde é possível executar um código interativo, neste caso, um encima de um interpretador Python 3 (IPython). Em conjunto, pode-se documentar o código utilizando markdown (linguagem de marcação de texto).

RESULTADOS E DISCUSSÃO

Com todos os ambientes já preparados, o modelo final é composto pela ilustração:

Figura 2 - Modelo final da arquitetura



Fonte: Desenvolvido pelo Autor

O Container neste caso é responsável por manter um ambiente debian virtualizado. Internamente, o ambiente anaconda é criado com base no PyMeep, este que se utiliza da versão 3.6.8 do Python. O ecossistema do PyMeep já instala todas as dependências para a execução da biblioteca Meep.

Um notebook Jupyter é montado na rede local, os pacotes Meep e Numpy também já preparados para a importação dentro da rede local:

Figura 3 - Ambiente jupyter inicializado no localhost

```
To access the notebook, open this file in a browser:  
file:///root/.local/share/jupyter/runtime/nbserver-1-open.html  
Or copy and paste one of these URLs:  
http://e293c79040a6:8888/?token=99af71f920c58be694fcbeaf1c811b02aa630a325a03f0c0  
or http://127.0.0.1:8888/?token=99af71f920c58be694fcbeaf1c811b02aa630a325a03f0c0
```

Fonte: Desenvolvido pelo Autor

Vale destacar que o Docker Compose esta sendo executado na rede local, assim como, internamente, o Notebook do Jupyter.

Nota-se que por ser um ambiente virtualizado, foi solucionado o impasse de incompatibilidade para desenvolver o projeto em outros ambientes, com diferentes sistemas operacionais ou versões.

CONCLUSÃO

O pacote principal, Meep, responsável pela simulação, requer diversos módulos, estes que necessitam de um ecossistema consistente para evitar qualquer tipo de conflito de versão ou compatibilidade de sistemas operacionais. Logo, com base na arquitetura desenvolvida e dos resultados obtidos após a configuração do Container, é evidente que o ambiente do projeto agora pode ser montado em qualquer ambiente ou sistema operacional.

Este projeto por completo fundamenta-se no estudo de métodos para facilitar a reconstrução do perfil de plasma com base nos resultados obtidos da técnica de reflectometria gerada na simulação. Logo, os próximos passos do projeto consistem em investigar e desenvolver métodos de redes neurais para melhor investigar os resultados do comportamento do plasma em um tokamak.

AGRADECIMENTOS

O autor agradece o orientador Cassio Henrique dos Santos Amador, que vem oferecendo um apoio intelectual para a realização do projeto. Também compartilhar um sentimento de gratidão ao colega de trabalho, Vinícius Mesel por apresentar e elucidar o conhecimento sobre algumas das técnicas que foram utilizadas.

REFERÊNCIAS

GOMES, C. Docker + Jupyter for Machine Learning in 1 Minute. 2019 Disponível em: <https://towardsdatascience.com/docker-jupyter-for-machine-learning-in-1-minute-30e1df969d09>. Acesso em: 2 set. 2020.

DOCKER. **What is a Container?**. 2020. Disponível em: <https://www.docker.com/resources/what-container>. Acesso em: 2 set. 2020.