

Sciflow-Bridge: Uma Ferramenta para Criação e Distribuição de Imagens de Containers para Workflows Científicos

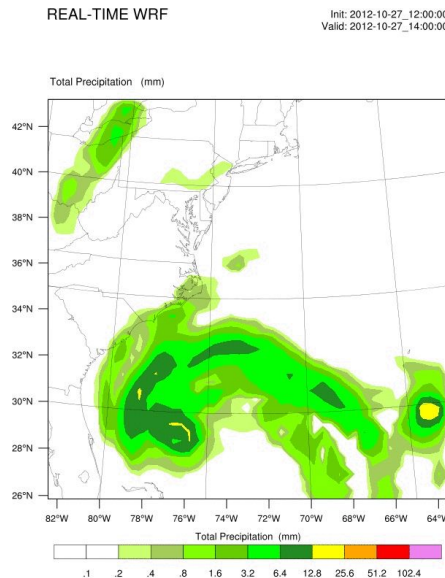
Bruno da Silva Alves
Orientadora: Andrea S. Charão

16 de outubro de 2019

Motivação

WRF: Sistema para previsão de tempo feito voltado tanto para pesquisa atmosférica quanto para aplicações de previsão.

**Contêineres &
Workflow &
Pesquisa &
Reprodutibilidade**



ARTICLES

A CONTAINERIZED MESOSCALE MODEL AND ANALYSIS TOOLKIT TO ACCELERATE CLASSROOM LEARNING, COLLABORATIVE RESEARCH, AND UNCERTAINTY QUANTIFICATION

JOSHUA P. HACKER, JOHN EXBY, DAVID GILL, IVO JIMENEZ, CARLOS MALTZAHN, TIMOTHY SEE, GRETCHEN MULLENDORE, AND KATHRYN FOSSELL

Software containers can revolutionize research and education with numerical weather prediction models by easing use and guaranteeing reproducibility.

Numerical models are a cornerstone of weather prediction today, and they support a broad range of weather research. Studies quantifying the ability of numerical models to predict the atmospheric state, and to simulate atmospheric phenomena, form two key lines of inquiry. By establishing model fidelity to the atmosphere, those studies also provide a basis for broader scientific inquiry with models. The use of numerical models in atmospheric research has become ubiquitous over the past few decades. Areas of research include physical process identification and analysis, atmospheric predictability, and predictions of future climates, among others.

Extensive use of numerical models in research demands educating students for diligent application of these large and complex codes. Beyond basic theory and best practices in numerical methods, the opportunities for mistakes are vast. While mistakes can provide positive learning experiences, they can

also produce misleading results and consequently incorrect interpretation.

The purpose of this article is to describe the implementation of software containers for numerical weather prediction (NWP) research and education. Container technology has profound implications for education and research in numerical weather prediction. Containers not only enable reproducibility, but they greatly lower barriers for accessing cutting-edge NWP codes. The tools discussed herein, such as source code repositories and containers, may be relatively new for many readers. These tools are presented in detail to fully describe the procedures used. But a deep understanding is not needed. The containers presented here have already been tested successfully by multiple users, primarily from universities. The technical details can remain mostly transparent for new users; the power of these tools can then be further realized by users (including students and educators) who become more advanced.

AMERICAN METEOROLOGICAL SOCIETY

JUNE 2017 | MJS | 1029

Uma barreira para a pesquisa

- A instalação de dependências é um processo demorado.
- Alguns pesquisadores não possuem conhecimentos técnicos para tal.
- Reprodutibilidade é comprometida.
- A barreira inicial pode gerar desinteresse.

Passos para configuração do WRF*

Passo da configuração	Nº de Comandos
Configuração de ambiente	20
Compilação das bibliotecas	47
Testes de compatibilidade	12
Compilação do WRF e WPS	23
Obtenção dos dados	10
Total	112

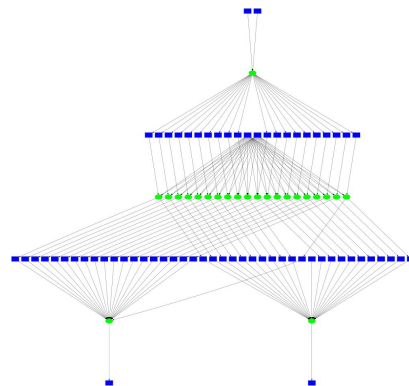
*Fonte: http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compilation_tutorial.php

Workflows Científicos

- Workflows científicos permitem a descrição de experimentos constituídos por várias tarefas computacionais.
- O componente básico de workflows científicos são processos:

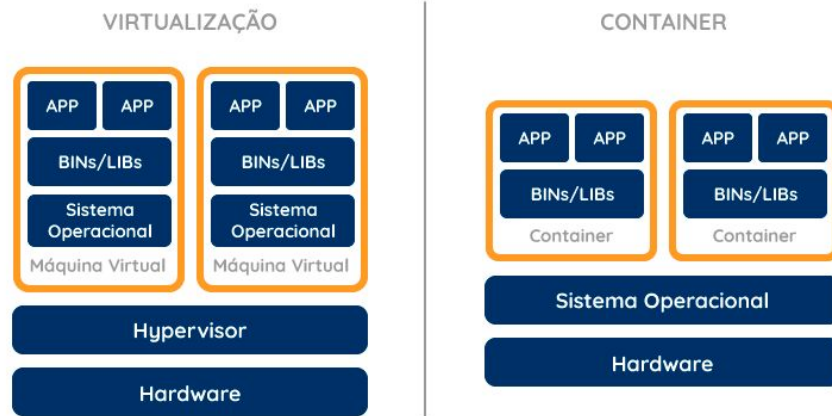


- A execução de workflows é orientada ao fluxo de dados.
- Sistemas de Workflow oferecem um ambiente para que os fluxos sejam modelados e executados.



Contêineres

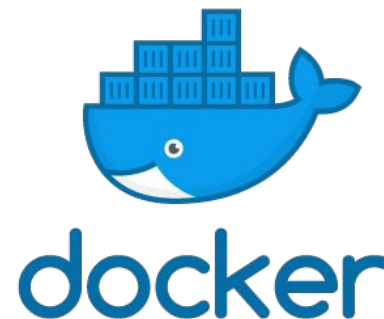
- É um tipo de virtualização em nível de SO.
- Utilizam algumas funcionalidades do Kernel, como Namespaces, CGroups, CHroot.
- Possuem níveis de performance próximos aos nativos.
- Acesso a disco mais rápido do que VMs.
- Possui questões de segurança e isolamento.



Fonte Imagem: <https://www.funcao.com.br/wp-content/uploads/2019/01/VirtualizacaoxContainer.png>

Docker e Docker Swarm

- Docker é uma das ferramentas mais famosas para criação de contêineres.
- Docker:
 - Possui um repositório com mais de 2.000.000 de imagens de contêineres.
 - Suporte ativo dos desenvolvedores.
 - Possui uma comunidade de usuários ativos.
- Docker Swarm:
 - Orquestrador
 - Gerencia todo o ciclo de vida dos contêineres.



A ferramenta Sciflow-Bridge

- Visa integrar workflows científicos e contêineres.
 - Contêineres oferecem uma solução no provisionamento de ambientes.
 - Os contêineres podem encapsular todas as bibliotecas, executáveis e dependências de entrada de cada tarefa do workflow.
- A integração deve ser eficiente.
 - Contêineres não podem ser tratados simplesmente como um local onde todas as dependências são despejadas.
 - As dependências devem ser distribuídas entre os hosts.
 - As tarefas devem ser distribuídas entre os hosts.

Implementação

Analisar as dependências de cada tarefa.

Gerenciar as dependências de cada tarefa.

Transferir as dependências entre os nós do sistema.

Executar um workflow baseado em um modelo.

Definir uma estratégia para executar as tarefas do workflow.

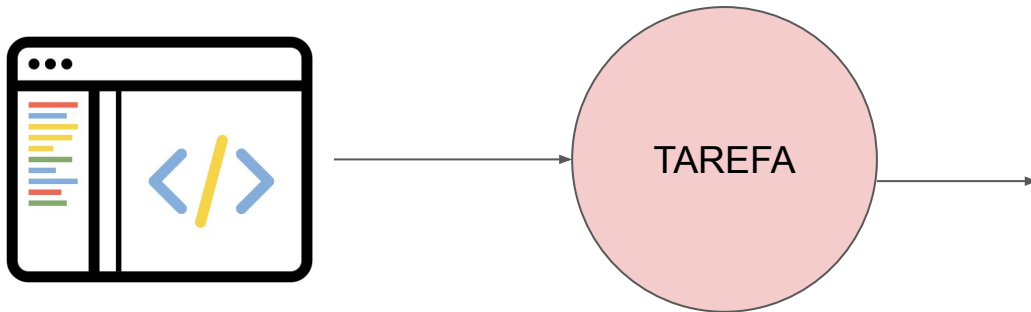
Gerenciar as tarefas do workflow.

Gerenciar os contêineres.

Utilizar padrões existentes.

Análise de Dependências

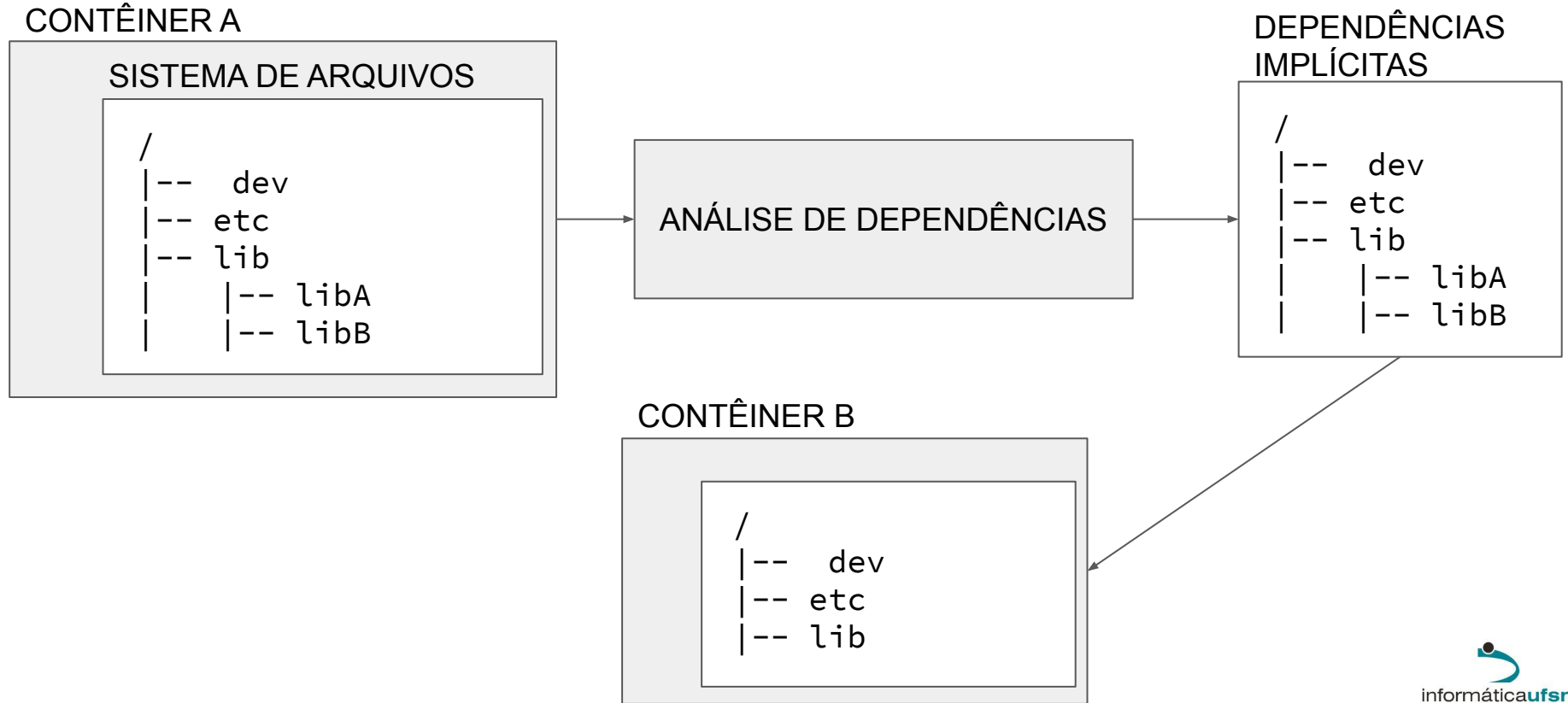
- Existem dois tipos de dependências: dependências de entrada/saída e dependências implícitas.
- Para que a tarefa execute corretamente, é necessário prover um ambiente que contenha os dois tipos de dependências.



```
root@container: $tree dep
dep
|-- blastall
|-- etc
|   |-- ld.so.cache
|   |-- nsswitch.conf
|   \-- passwd
|-- lib
|   \-- x86_64-linux-gnu
|       |-- libc.so.6
|       |-- libm.so.6
|       |-- libnsl.so.1
|       |-- libnss_compat.so.2
|       |-- libnss_files.so.2
|       |-- libnss_nis.so.2
|       \-- libpthread.so.0
|-- nt
|   |-- nt.44.nhr
|   |-- nt.44.nin
|   |-- nt.44.nnd
|   |-- nt.44.nni
|   |-- nt.44.nsd
|   |-- nt.44.nsi
|   |-- nt.44.nsq
|   \-- nt.nal
\-- small.fasta.0
```

4 directories , 20 files

Ambientes containerizados



Transferência de Dependências

- Algumas opções para transferência de imagens de contêineres:
 - Tar file.
 - Dockerfile.
 - Via DockerHub.
- Solução adotada:
 - **Transferir somente os arquivos utilizados.**
 - Reutilização de contêineres.
- Implementação:
 - Cache -> Hashes recursivas.
 - Notificação -> MQTT.

```
/
|-- dev
|-- etc
|-- lib
|   |-- libA
|   |-- libB
```

Makeflow

- Makeflow é um sistema de workflows científicos.
- Maneira simples para descrever workflows baseado na sintaxe de Makefiles.

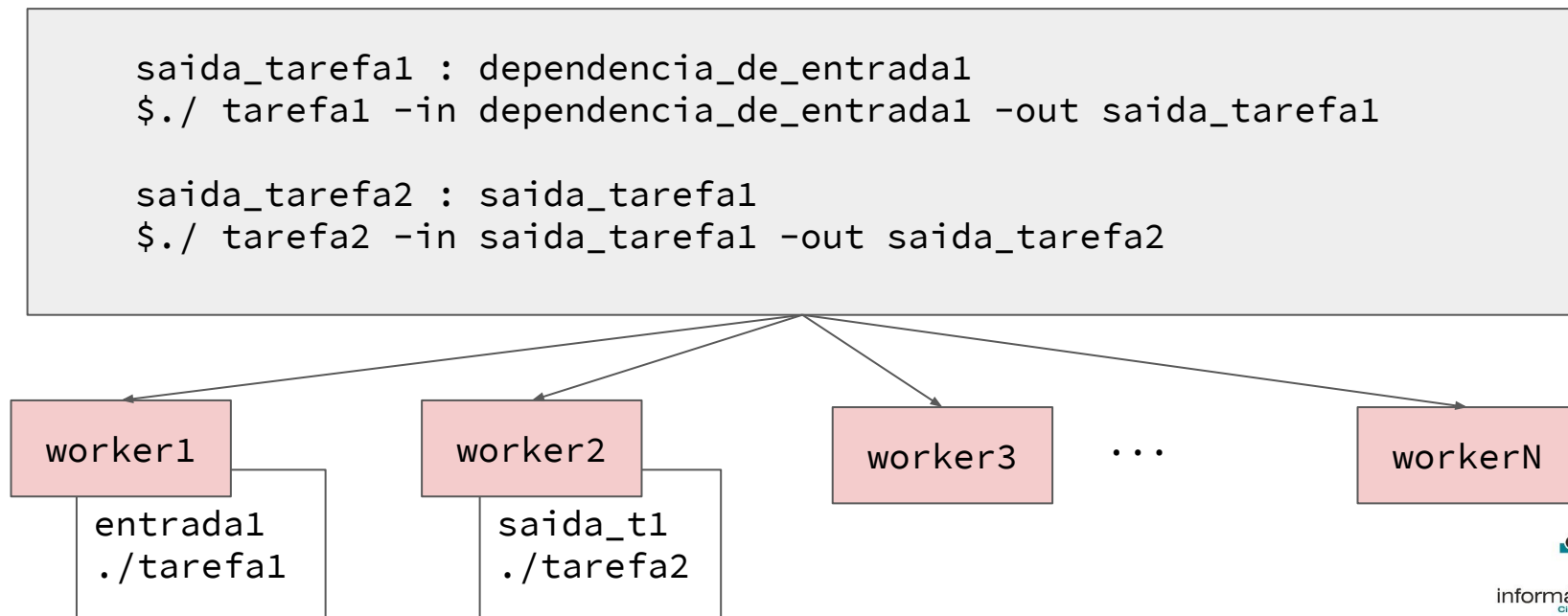
```
saida_tarefa1 : dependencia_de_entrada1
$./ tarefa1 -in dependencia_de_entrada1 -out saida_tarefa1

saida_tarefa2 : saida_tarefa1
$./ tarefa2 -in saida_tarefa1 -out saida_tarefa2
```

- Tarefa: Arquivos de entrada, arquivos de saída e comando para execução da tarefa.

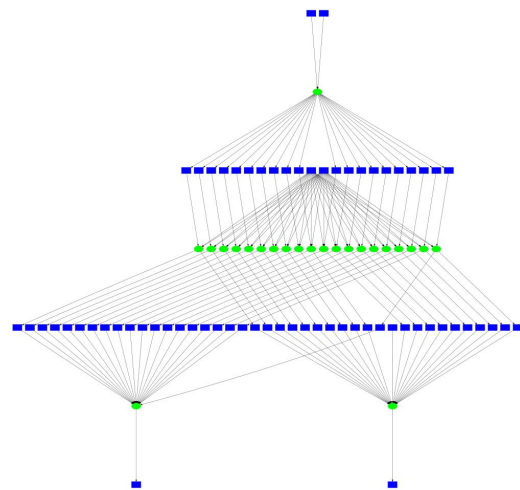
Makeflow & WorkQueue

- O WorkQueue é o software responsável por distribuir as tarefas entre os nós trabalhadores.
- O WorkQueue recebe as tarefas definidas pelo makeflow e as distribui.



Resumo da Execução

1. Escolha de uma imagem base.
2. Instalação das dependências.
3. Análise de dependências implícitas.
4. Configuração do Swarm.
5. Configuração do Sciflow-bridge.
6. Criação dos contêineres trabalhadores.
7. Execução do workflow científico.
8. Coleta dos resultados.



Revisão das Atividades

1. Revisão das tecnologias de contêiner.
2. Revisão e comparação das engines de Workflow Científico.
3. Revisão dos métodos utilizados para integração de contêineres e sistemas de Workflow.
4. Revisão de workflows científicos disponíveis para acesso.
5. Pesquisa por ferramentas para orquestração de contêineres.
6. Propor uma solução para a criação e distribuição de imagens de contêineres.
7. Implementar a solução proposta.
8. Avaliação da ferramenta.

Etapa	Agosto	Setembro	Outubro	Novembro	Dezembro
1	✓				
2	✓				
3	✓				
4		✓			
5		✓			
6		✓	✓		
7			✓	✓	
8				✓	✓

Próximas Etapas

- Utilizar o WorkQueue para realizar a transferência das dependências implícitas.
- Avaliar o desempenho de alguns cenários possíveis:
 - Um trabalhador por host.
 - Vários trabalhadores por host.
 - Arquitetura hierárquica.
- Executar a solução com outros workflows:
 - Blast
 - BWA
 - Shakespeare