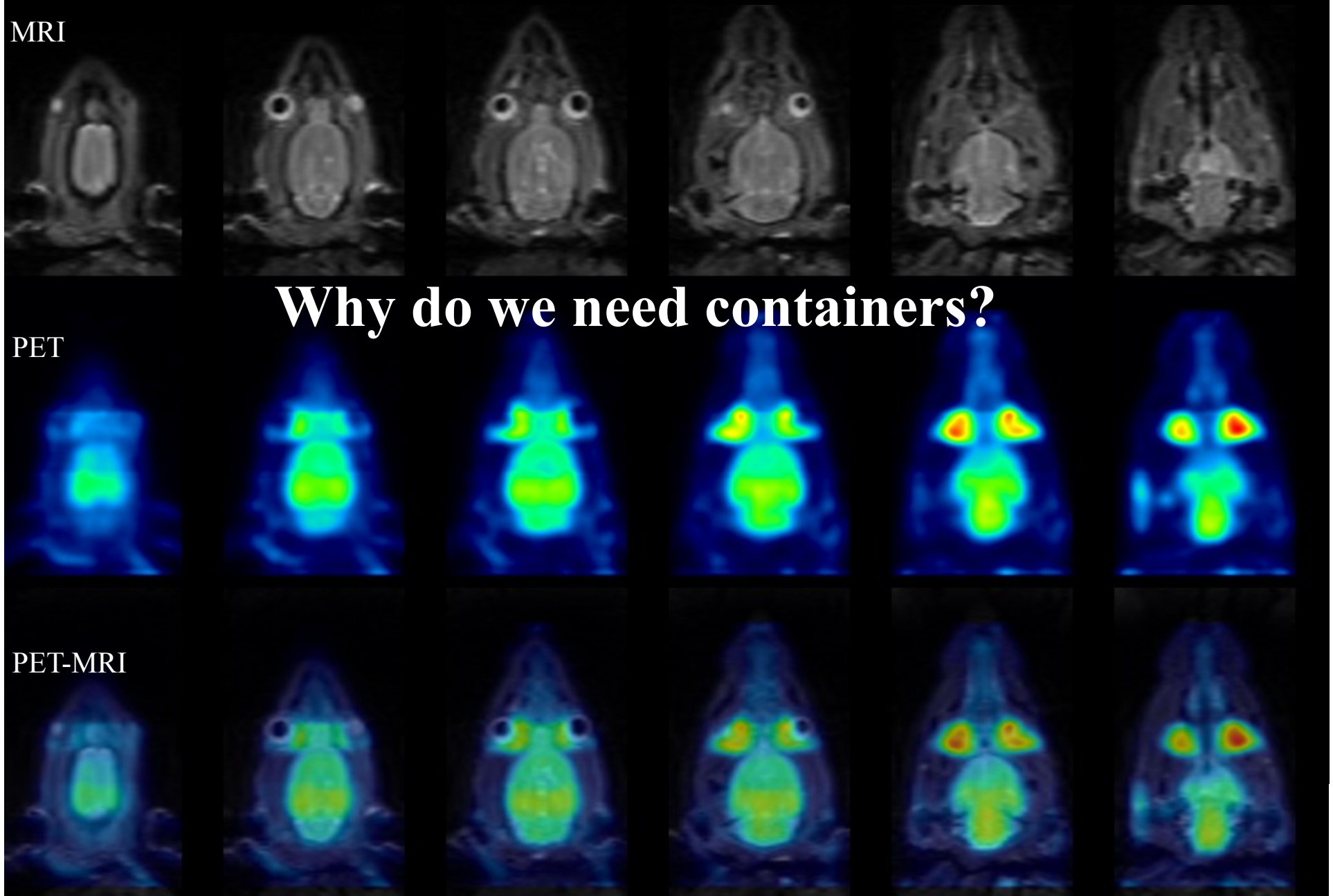
The Application of Containers in my Work

Parisa Khateri

Institute for Particle Physics and Astrophysics, ETH Zurich

SAFIR : Small Animal Fast Insert for MRI



Why do we need containers?

- Simulation → Gate (prerequisites: Geant4, Root, ...)
- Data analysis→ petaAnalysis (prerequisites: Root, ...)
- Reconstruction → STIR (standard prerequisites)

Why do we need containers?

- Simulation → Gate (prerequisites: Geant4, Root, ...)
- Data analysis→ petaAnalysis (prerequisites: Root, ...)
- Reconstruction → STIR (standard prerequisites)

1) Use an already existing container in the Dockerhub https://hub.docker.com/
2) Build my own container

(1) Use an already existing container

- Create an account in the Docker hub : https://hub.docker.com/
- Install Docker
- Pull (download) a repository or an image
- Run the container

(1) Use an already existing container - on a local computer

usage	command
Install docker*	○ wget -qO- https://get.docker.com/ | sh
pull	○ docker pull ubuntu:latest ○ docker pull opengatecollaboration/gate:latest
List images	○ docker images / docker image ls
List containers	○ docker ps / docker container ls
Remove an image	○ docker image rmi <IMAGE ID>
Run	○ docker run ubuntu:latest cat /etc/os-release
Run interactively, inside the container	○ docker run **-it** ubuntu:latest bash ○ docker run **-it** opengatecollaboration/gate:latest bash ○ docker run **-v** /path/to/my/local/dir:/path/to/container/dir **-it** opengatecollaboration/gate:latest bash
Exit container	○ exit
Run from outside the container	○ docker run **-v** /path/to/my/local/files:/path/to/container/dir opengatecollaboration/gate:latest bash \ -c 'source /root/.bashrc; cd /path/to/container/dir; Gate my_macro.mac'

*https://docs.docker.com/install

(1) Use an already existing container - On Piz Daint

usage	command
Install docker	○ module load shifter-ng
pull	○ shifter pull ubuntu:latest ○ shifter pull opengatecollaboration/gate
List images	○ shifter images
List containers	○ shifter ps
Remove an image	○ shifter image rmi <IMAGE ID>
Run	○ srun -N1 -C mc --partition=2go shifter run ubuntu:latest cat /etc/os-release
Run interactively, inside the container	○ srun -N1 -C mc --partition=2go **--pty** shifter run ubuntu:latest bash ○ srun -N1 -C mc --partition=2go **--pty** shifter run opengatecollaboration/gate:latest bash
Exit the interactive mode	○ exit
Run from outside the container	○ shifter run opengatecollaboration/gate:latest bash -c 'source /root/.bashrc; cd /path/to/container/dir; Gate my_macro.mac'

(1) Use an already existing container - On Piz Daint run multiple programs using containers

The configuration file:

```
0 shifter run opengatecollaboration/gate bash -c 'source /root/.bashrc; cd
/path/to/container/dir; Gate macro100.mac’


1-99 shifter run opengatecollaboration/gate bash -c 'source /root/.bashrc; cd
/path/to/container/dir; Gate macro'"%t"'.mac'
```

(1) Use an already existing container - On Piz Daint run multiple containers

The sbatch file to submit the jobs:

```
#!/bin/bash -l
#SBATCH --job-name=gate
#SBATCH --time=20:00:00
#SBATCH --nodes=5
#SBATCH --ntasks-per-core=1
#SBATCH --ntasks-per-node=20
#SBATCH --cpus-per-task=1
#SBATCH --partition=2go
#SBATCH --output=/path/to/jobs_out/gate.%j.stdout.log
#SBATCH --error=/path/to/jobs_out/gate.%j.stderr.log
module load shifter-ng
srun --wait 0 --multi-prog multi_gate.config
```

(2) Build my own container

- Dockerfile
  - docker build -t <user/dockername:tag> path-to-dockerfile
- push image to docker-hub
  - docker login
  - docker push name:[tag]
- pull it on Daint or any other computer
  - Docker pull name:[tag]

- N.B. Location of the Dockerfile: the same level as the source or higher

(2) Build my own container – Dockerfile

- FROM <an image already exists on dockerhub>
- RUN executable param1 param2
- ENV <variable_name> <value>
- ENV <variable_name1>= <value1> <variable_name2>= <value2>
- WORKDIR <path>
- COPY <src> <dest>

https://docs.docker.com/engine/reference/builder

(2) Build my own container – Dockerfile example

```
FROM ubuntu:16.04
RUN apt-get update && apt-get install -y libboost-all-dev
RUN apt-get update && apt-get install -y build-essential cmake wget qt5-default vim
#     install ROOT       #
# install prerequisites
RUN apt-get update && apt-get install -y git dpkg-dev cmake g++ gcc binutils libx11-dev libxpm-dev libxft-dev
libxext-dev libgsl0-dev python-pip
# download and build ROOT
RUN wget -q https://root.cern.ch/download/root_v6.14.02.source.tar.gz && tar -xzf root_v6.14.02.source.tar.gz &&
cd root-6.14.02 && mkdir -p build && cd build && ldconfig && cmake ../ && make -j 8
ENV ROOTSYS /root-6.14.02/build
ENV PATH="${ROOTSYS}/bin:${PATH}" LD_LIBRARY_PATH="$ROOTSYS/lib:$LD_LIBRARY_PATH"
#      install petaAnalysis     #
WORKDIR /petaAnalysis
COPY . .   # Dockerfile is in my source directory
RUN mkdir -p build
WORKDIR build
RUN rm -r * && cmake .. && make -j 8
ENV PATH="/petaAnalysis/bin:$PATH"
```

Thank you for your attention