Sirepo: Containerized HPC Engineering in the Cloud

Robert Nagler Paul Moeller David Bruhwiler

Chris Hall Nathan Cook

rsl.link/sea19

radiasoft

SEA Scientific Software Conference 2019

10 April 2019 – Boulder

This work is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Award #DE-SC0011340.

U.S. DEPARTMENT OF ENERGY

Office of Science

Overview

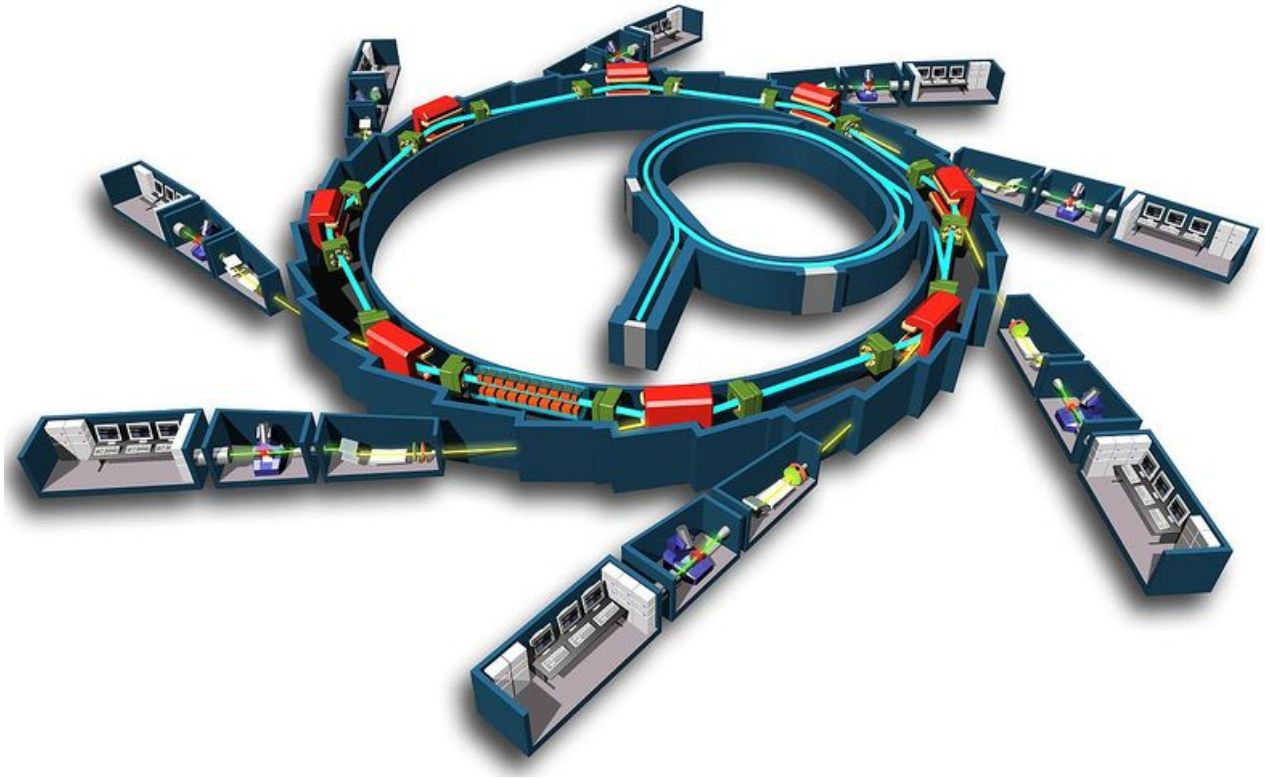
- *Intro to Sirepo*
- *Deployment Challenge*
- *Build/Install Tool*
- *Conclusion*

What is Sirepo?

- *Goal: improve accessibility of particle accelerator and beam physics software (science gateway)*
- *Introduce codes in layers for novices: usable at first click*
- *Experts import and share existing simulations instantly*
- *No builds necessary, just visit sirepo.com*
- *Can be installed locally to use your own resources*
- *Simplify simulations on clusters and supercomputers*

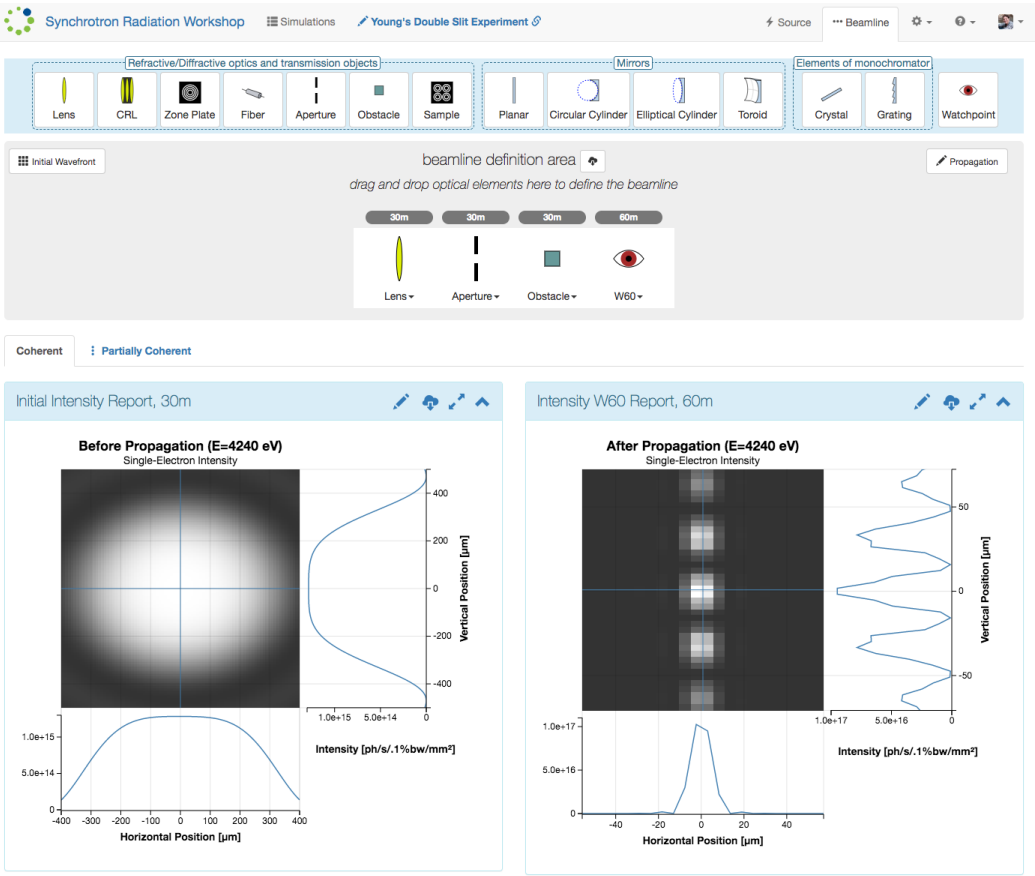
Beam Physics 0.1

- *Synchrotron directs electrons around a storage ring*
- *Electrons diverted to generates x-rays (light)*
- *Beamline directs light to sample and detector*



Demo

- *Thomas Young's Double Slit Experiment*
- *Demonstrated wave theory of light in 1801*
- *Code: Synchrotron Radiation Workshop (SRW)*



Deploying Sirepo

- *Codes are large, slow to build, many dependencies*
- *Docker simplifies builds/installs but is not composable*
- *RPMs used to compose codes into Docker & Vbox*
- *Vagrant/VirtualBox more convenient for development*
- *Git-centric glue furthers composability/flexibility*
- *Some challenges:*
  - *Same codes also need to run in Jupyter/Hub*
  - *All codes need to run in same shell (preferable same virtualenv)*
  - *Updates to codes happen frequently and (sometimes) urgently*
  - *Custom build scripts for most codes need wrapping*
  - *New environments, e.g. NERSC (SHIFTER) or BNL (Debian)*
- *Design is policy-rich to simplify implementation*

Git-centric Glue?

- *Rely on GitHub, Nginx, Curl, Bash, Yum, etc.*
- *Reusable, composable build/install tool (1,000 LOC)*
- *Standalone but not a config manager (part of one)*
- *radiasoft/beamsim image gets built with:*
  `curl radia.run | bash -s container-build`
- *Which runs repo's* `container-conf/build.sh`

```
build_image_base=radiasoft/fedora
build_is_public=1
build_as_run_user() {
    build_run_user_home_chmod_public
    install_repo_eval beamsim-codes
}
```

Vagrant Sirepo Dev Installer

radia_run vagrant-sirepo-dev

```
vagrant_sirepo_dev_main() {
    install_repo_eval vagrant-dev fedora "$@"
    vagrant ssh <<EOF
$(install_vars_export)
source ~/.bashrc
radia_run sirepo-dev
EOF
}
```

- *`vagrant-dev` sets up VirtualBox with development environment*
- *`sirepo-dev` installs beamsim codes and sirepo for development*
- *`install_vars_export` supports test/dev of curl installer API*
- *Installer is a directory with `radiasoft-download.sh` with a `main`*

Typical Dockerfile

```
FROM python:3.7
WORKDIR /root/
ARG BRANCH="master"
ARG NUM_CORES=2

RUN echo "deb http://ftp.us.debian.org/debian unstable main
contrib non-free" >> /etc/apt/sources.list.d/unstable.list
&&\
    apt-get update && apt-get install -y \
    gcc \
    g++ \
    git \
    cmake \
    libgmp-dev \
    libmpfr-dev \
    libgmpxx4ldbl \
    libboost-dev \
    libboost-thread-dev && \
    apt-get clean && \
    git clone --single-branch -b $BRANCH
https://github.com/PyMesh/PyMesh.git

ENV PYMESH_PATH /root/PyMesh
ENV NUM_CORES $NUM_CORES
WORKDIR $PYMESH_PATH

RUN git submodule update --init && \
    pip install -r $PYMESH_PATH/python/requirements.txt &&
\
    ./setup.py bdist_wheel && \
    rm -rf build third_party/build && \
    pip install dist/pymesh2*.whl && \
    python -c "import pymesh; pymesh.test()" && \
    python $PYMESH_PATH/docker/patches/patch_wheel.py
```

Building Code RPMs

radia_run rpm-code pymesh

```
codes_dependencies common
codes_yum_dependencies mpfr-devel gmp-devel
codes_download https://github.com/radiasoft/PyMesh.git
git submodule update --init
NUM_CORES=$(codes_num_cores) codes_python_install
# run tests outside build directory
cd ..
python -c 'import pymesh; pymesh.test()'
```

- *Allows dependencies to other code RPMs (common is a "code")*
- *Providers wrappers for yum (dnf), git, python install (pip), etc.*
- *Flexibility to do shell-type things (commands, functions, & variables)*
- *`fpm` is used create RPMs; Build runs inside Docker for reproducibility*
- *RPM documents source repo and version*
- *Test will cause build to fail; Fail Fast is implicit policy*

Takeaways

- *Docker+RPMs is Sirepo's build/install:*
  - *Docker supports reproducibility*
  - *RPMs allows composability*
- *A little Git+Curl+Bash is needed to put them together*
- *By fixing policies early, we reduced the code required.*

Thank You!

Questions?

rsl.link/sea19