Containers in Science (at NERSC)

Shane Canon
Data and Analytics Group, NERSC

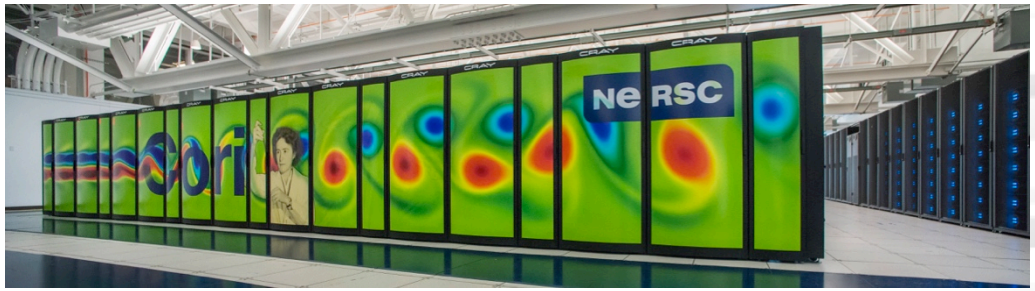
ISC19

Focus on Science

- NERSC supports the broad mission needs of the six DOE Office of Science program offices
- 6,000 users and 750 projects
- 2,078 referred publications in 2015
- 2015 Nobel prize in physics supported by NERSC systems and archive

HPC is Awesome

- Cori Cray XC40
  - Data-intensive (32-core Haswells, 128GB) partition
  - Compute-intensive (68-core KNLs, 90GB) partition
  - ~10k nodes, ~700k cores
- High speed parallel file system
  - >10 PB project file system (GPFS)
  - >28 PB scratch file system (Lustre)
  - >1.5 PB Burst Buffer (flash)
- High Speed Aires interconnect
  - 8 GB/s MPI bandwidth

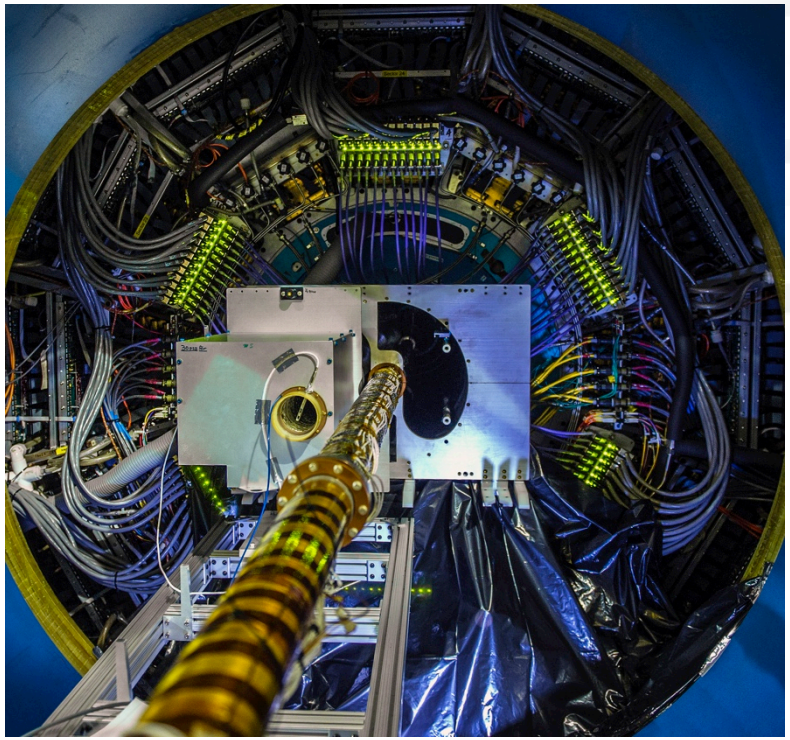


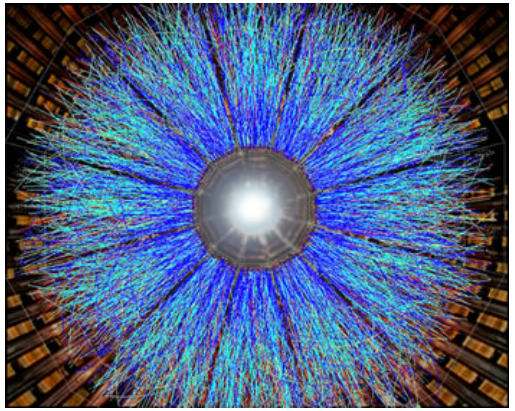
HPC is Awkward

- No local disk
  - Breaks a lot of standard Linux work flows
- Minimal OS
  - Designed to accelerate parallel software
  - Many “expected” Linux tools are absent
  - Runs SUSE, and doesn’t upgrade often
- Different file systems have different responses
  - Sometimes unclear to users where is the best place to put their software and data
- Many groups have turned to Shifter to over come these obstacles

Probing The Nucleus

- STAR at Brookhaven, NY
  - smashing nuclei into each other to understand their component parts
- Data analysis and simulation
- Why Shifter?
  - Difficult software dependencies (32-bit libraries)





How'd They Do It?

```
# Build STAR environment image from tarballs
FROM ringo/scientific:6.4
MAINTAINER Mustafa Mustafa <mmustafa@lbl.gov>

# RPMs
RUN yum -y install libxml2 tcsh libXpm.i686 libc.i686 libXext.i686 libXrender.i
686 libstdc++.i686 fontconfig.i686 zlib.i686 libgfortran.i686 libSM.i686 mysql-
libs.i686 gcc-c++ gcc-gfortran glibc-devel.i686 xorg-x11-xauth wget make libxml
2.so.2 gdb libXtst.{i686,x86_64} libXt.{i686,x86_64} glibc glibc-devel gcc-c++

# Dev Tools
RUN wget -O /etc/yum.repos.d/slc6-devtoolset.repo http://linuxsoft.cern.ch/cern
/devtoolset/slc6-devtoolset.repo && \
    yum -y install devtoolset-2-toolchain
COPY enable_scl /usr/local/star/group/templates/

# untar STAR OPT
COPY optstar.sl64_gcc482.tar.gz /opt/star/
COPY installstar /
RUN python installstar SL16c && \
     rm -f installstar && \
     rm -f optstar.sl64_gcc482.tar.gz

# untar ROOT
COPY rootdeb-5.34.30.sl64_gcc482.tar.gz /usr/local/star/
COPY installstar /
RUN python installstar SL16c && \
     rm -f installstar && \
     rm -f rootdeb-5.34.30.sl64_gcc482.tar.gz

# DB load balancer
COPY dbLoadBalancerLocalConfig_generic.xml /usr/local/star/packages/SL16d/StDb/
servers/

# production pipeline utility macros
COPY Hadd.C /usr/local/star/packages/SL16d/StRoot/macros/
COPY lMuDst.C /usr/local/star/packages/SL16d/StRoot/macros/
COPY checkProduction.C /usr/local/star//packages/SL16d/StRoot/macros/
```

Publically available SL6.4 image

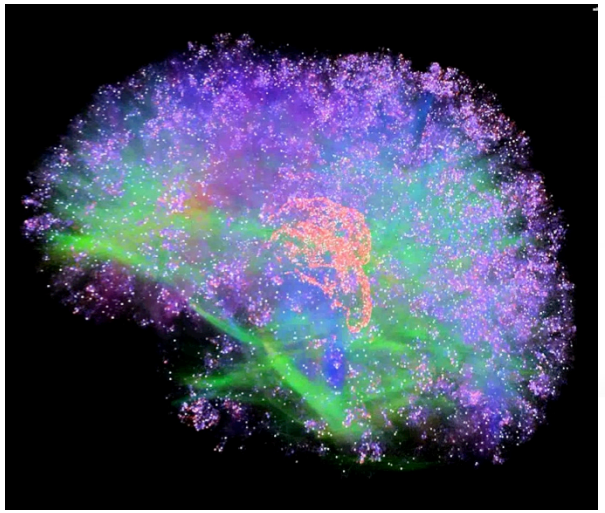
Custom STAR software:
Compiled on **another** system

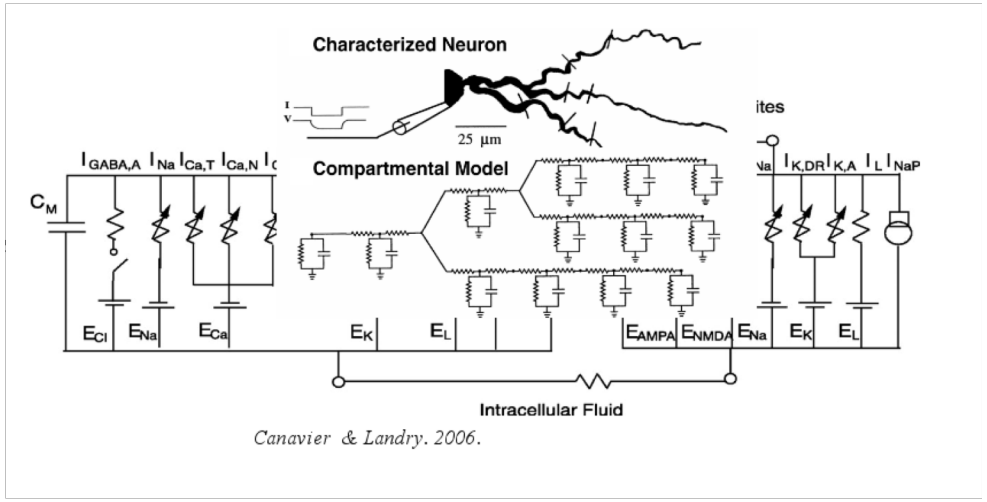
Leveraging Shifter for Easy Scalability

- Shifter has capability to loop mount an xfs file
  - Backed by Lustre, but all metadata actions are limited to a single node, so access is very fast
- STAR needs to read from a ~100 MB MySQL database
  - Running 32 individual jobs / node
- DB on Lustre, query timed out after 30 minutes
- Copied DB to Shifter's xfs
  - Initial copy ~5 minutes
  - DB Query was instantaneous
- Used this functionality to quickly scale up without re-engineering their workflow

Modeling the Mind

- Neuron
  - Simulation program to model neuron response to stimuli
- Simulation: 3000 points by 4000 time steps for 400 neurons
- Why Shifter?
  - Software developed in 1985, many older dependencies

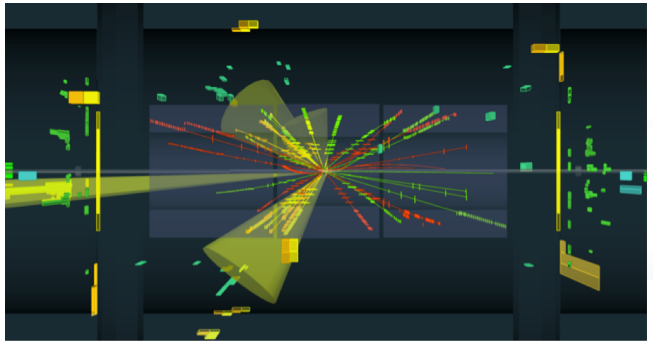


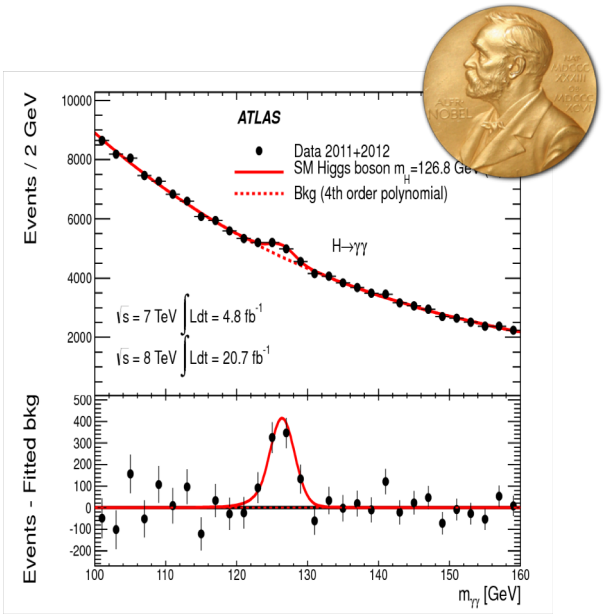


Canavier & Landry. 2006.

Probing the Fundamentals of Matter

- Large Hadron Collider (LHC)
  - 300 trillion proton-proton collisions and 30 PBs of data per year.
- Data analysis, simulation, multi-site data and computing pool
- Why Shifter?
  - Complicated software stack: Needs FUSE and elevated permissions to run
  - Integrated framework for running with images at all computing sites

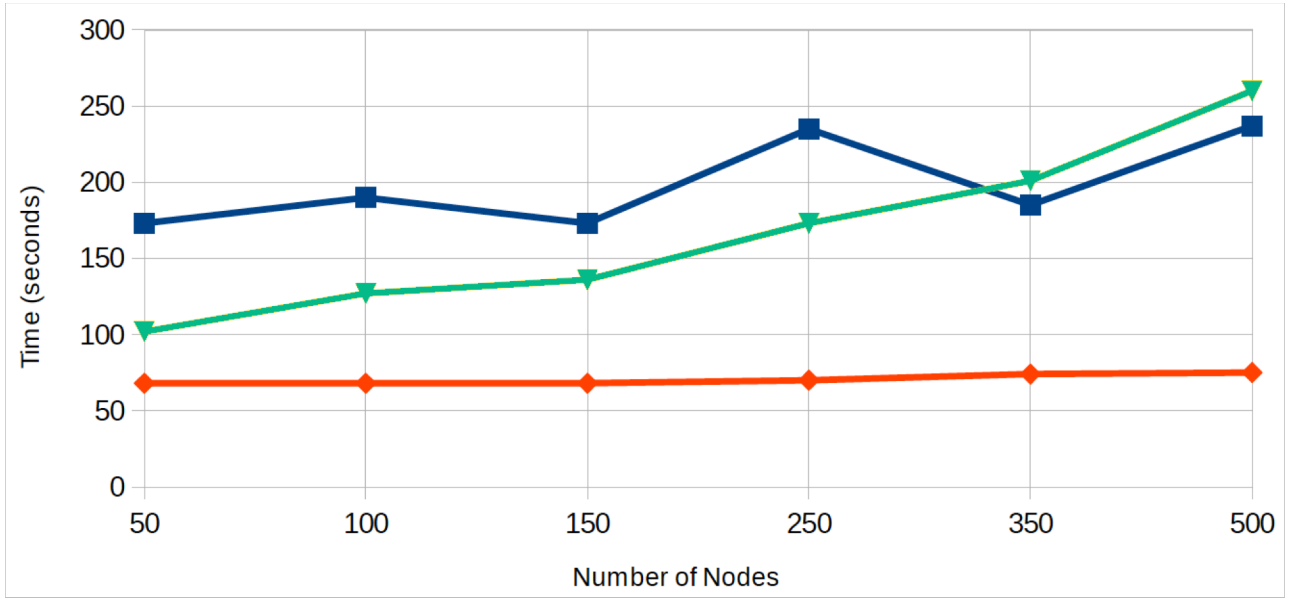




EXASCALE COMPUTING PROJECT

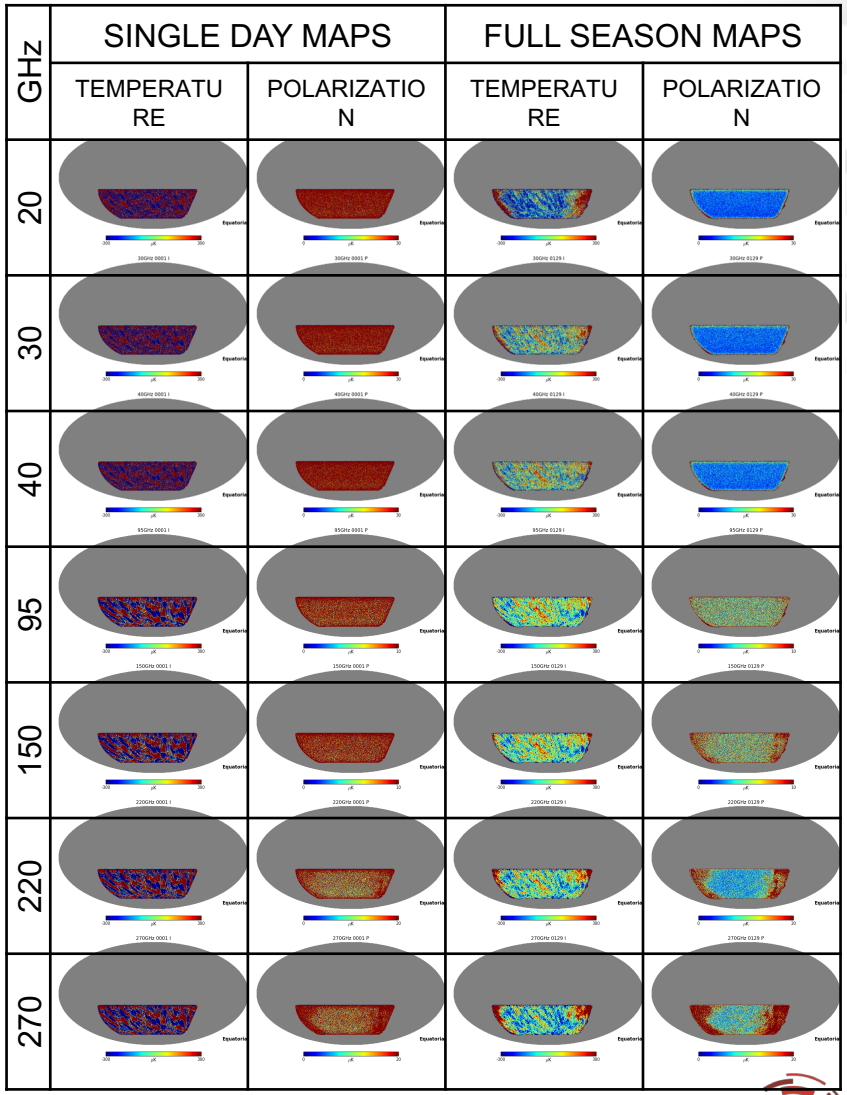
Creating a Monster

- The software stack is very big: 3.5 TB and 20M inodes
- Compress and deduplicate with squashfs: 350 GB
- Start up time shows excellent scaling out to 500 nodes (16,000 cores)



Measuring the Composition of the Universe

- CMB – S4
  - Ambitious collection of telescopes to measure the remnants of the Big Bang with unprecedented precision
- Simulated 50,000 instances of telescope using 600,000 cores on Cori KNL nodes.
- Why Shifter?
  - Python wrapped code needs to start at scale



COMPUTATIONAL COSMOLOGY CENTER

EXASCALE COMPUTING PROJECT

Shifter Enables Science

- Shifter is making scientific analysis easier at NERSC
  - Successful use across many scientific disciplines
  - Shifter framework can be extended to other systems and shifter images can be run at any “Docker-friendly” computing center

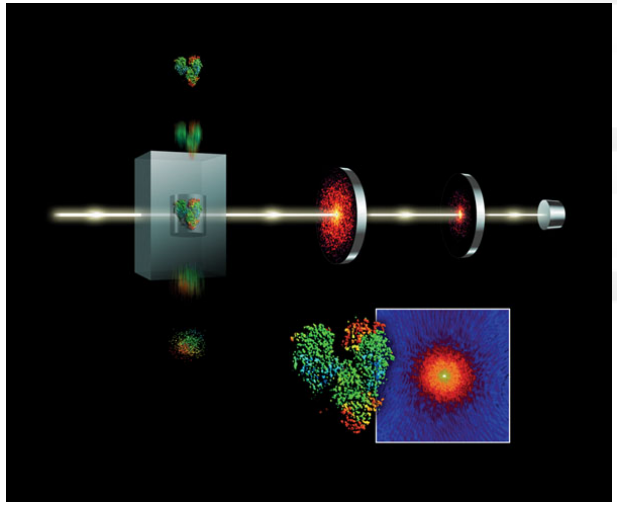
National Energy Research Scientific Computing Center

Exploring the Universe

- Dark Energy Survey: Visualizing the universe
  - Measuring the expansion history of the universe to understand the nature of Dark Energy.
- Data analysis code: identify objects (stars, galaxies, quasars, asteroids etc.) in images, calibrate, measure their properties.
- Why Containers?
  - Complicated software stack – runs on laptops to supercomputers
  - Python-based code; lots of imports

Imaging the Heart of Things

- LCLS: Linac Coherent Light Source at SLAC
  - Using X-rays to image nanoscale particles and understand chemistry on the natural timescale of reactions
- Realtime image analysis based on python stack (tomo.py)
- Why Shifter?
  - Many library imports, complicated software stack



Loop Mounted FS for Super Fast I/O

- Shifter can mount an xfs file system on each node
  - Created when job starts and destroyed when job ends
  - Cray “local disk”
  - Excellent I/O rates:
    - Backed by the Lustre file system, metadata operations are all confined to a single node
    - Also good for “bad IO”