Status of Software at ANL and Containerization

David Blyth
Simulation and Reconstruction

Legacy chain: SLIC + LCSim + slicPandora
- Full simulation and reconstruction with PFA for SiD-based detectors
- Has allowed us to study the applicability of a SiD-based detector for the EIC
- Limited to SiD subdetectors and symmetry

Evolution chain: lcgeo + LCSim
- Drops SLIC in favor of the DD4Hep-based lcgeo simulation
- Does not currently include PFA
Legacy Chain

- Adaptation of the SiD simulation and reconstruction software chain
- Full simulation + tracking + PFA
- Event visualization with Jas4pp (S. Chekanov)
- Thanks to a few efficiency improvements, digitization and tracking time in LCSim has been dramatically reduced
  - E.g. for $\sqrt{s} = 35$ GeV DIS events, time has been reduced by a factor of ~35
Evolution Chain (DD4Hep)

- Created in order to evolve away from SiD chain
- DD4Hep and LCSim made to work together for SiD-based detectors
- LCSim will soon be replaced with digitization and reconstruction that leverages DD4Hep detector description
- With LCSim replaced, the chain will be used to simulate and reconstruct detectors that are very different from SiD (e.g. Cherenkov components)
- See presentation by W. Armstrong tomorrow morning
NPDet

- DD4Hep-based parameterized detector library for nuclear physics experiments (W. Armstrong, S. Johnston)
- Compatible with the “Evolution Chain”
- Provides foundations for a number of detector concepts
  - JLEIC
  - SiEIC
  - ...
- Excellent place to collaborate right now! Any effort put into developing detector concepts here will not be wasted.
GenFind

- Generic track finding library in its early stages coupled to GenFit
  - Uses Hough transform and conformal mapping
- Working track finding for JLEIC case thanks to S. Johnston
  - However, still uses “SimTrackerHit” portion of LCIO model as input
- Near future:
  - Update to use digitized + reconstructed hits
  - Generalize using SiEIC as test case
Proio

- Language-neutral IO library for storing and transmitting intermediate and reconstructed data
  - Primary motivator: data model evolution and sharing data
- Based on Protobuf, and inspired by ProMC (S. Chekanov) and EicMC (A. Kiselev)
  - Conceptual merger of LCIO and ProMC/EicMC
- Implemented in
  - Go (tools mostly written in go: portable and performant)
  - Python
  - C++
  - Java (read-only for now)
- Will present on this in detail tomorrow
  - To get a head start: https://github.com/decibelcooper/proio
HepSim

- A simple but powerful tool for building a “Repository with MC simulations for particle physics”
  - Consists of a web interface and command-line tools
- Already contains ~2 Billion events
  - LO+PS, NLO, and NLO+PS
- Environment to study detector effects with fast and full simulations
- See next talk by S. Chekanov
Logical organization of HepSim

<table>
<thead>
<tr>
<th>Truth-level event samples</th>
<th>Simulated event samples</th>
<th>Detector descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low q2 DIS events</td>
<td>rfull056</td>
<td>SiEIC2: compact, GDML, etc</td>
</tr>
<tr>
<td>Gen: LEPTO/ARIADNE</td>
<td>fpadsim-1.3</td>
<td>SiEIC3: compact, GDML, etc</td>
</tr>
<tr>
<td>L: 57 pb⁻¹</td>
<td>rfull057</td>
<td>SiEIC4: compact, GDML, etc</td>
</tr>
<tr>
<td>CM energy: 35 GeV</td>
<td>rfull058</td>
<td>SiEIC5: compact, GDML, etc</td>
</tr>
<tr>
<td></td>
<td>fpadsim-1.3.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rfull058</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fpadsim-1.3.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rfull059</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fpadsim-1.4</td>
<td></td>
</tr>
</tbody>
</table>
HepSim and Containers

- Would like to standardize layout of reconstruction container images
  - Standard entry-point script within container that takes input and output directories as arguments?
- Reconstruction tags in HepSim will specify/correspond to Docker Hub tags
- Anyone with Singularity or Docker will be able to process arbitrary MC data with the reconstruction software on
  - Desktop
  - OSG
  - HPC
  - etc...

Simulated event samples:
- rfull056
- fpadsim-1.3
  - Det: SiEIC5
- rfull057
- fpadsim-1.3.2
  - Det: SiEIC5
- rfull058
- fpadsim-1.3.4
  - Det: SiEIC5
- rfull059
- fpadsim-1.4
  - Det: SiEIC5
Container Implementations

- **Docker**
  - Developed for IT industry
  - Integrated into cloud services such as AWS, Google Cloud, and Azure
  - Docker Hub (hub.docker.com)

- **Singularity**
  - Developed at LBL
  - Easier to use interactively on desktop
  - Better suited for grid and HPC
  - Can import from Docker Hub

- **Shifter**
  - Developed at NERSC
  - Specifically for deployment of images on HPC clusters
  - Imports from Docker Hub
Experiences with Containerization at ANL

- Much of our simulation and reconstruction has been moved to containers on both the Open Science Grid (OSG) and HPC clusters.
- Primary Docker images have been developed and hosted on Docker Hub https://hub.docker.com/u/argonneeic/
- Singularity and Shifter containers have been run in Grid/HPC environments
Dockerfiles

- Essentially source code for Docker images
- Can be readily revision controlled
  - E.g. [https://eicweb.phy.anl.gov/dblyth/FPaDSimContainer](https://eicweb.phy.anl.gov/dblyth/FPaDSimContainer)
- Serves as
  - Instructions to building a Docker image
  - Documentation for image
- Good idea to
  - Import from image tags that are not subject to change
  - Reference specific software releases or commit hashes

```bash
FROM dbcooper/arch:2017-02-18

# Set up basic environment
## Required tools from Arch repository
RUN pacman -S --noconfirm \
    sed \n    sudo

RUN useradd -m -g wheel fpadsimuser; \
    sed -i.bak 's/\%(wheel ALL=(ALL) NOPASSWD: ALL)/\1/\1' /etc/sudoers;

USER fpadsimuser
WORKDIR /home/fpadsimuser

CMD /bin/bash -l

# ROOT
## Required tools from Arch repository
RUN sudo pacman -S --noconfirm \
    awk \n    base \n    base-devel \n    binutils \n    cmake \n    fakeroot \n    gcc \n    git \n    grep \n    gzip \n    mako \n    python & & \
```
### Dockerfiles

- Essentially source code for Docker images
- Can be readily revision controlled
  - E.g. [https://eicweb.phy.anl.gov/dblyth/FPaDSimContainer](https://eicweb.phy.anl.gov/dblyth/FPaDSimContainer)
- Serves as
  - Instructions to building a Docker image
  - Documentation for image
- Good idea to
  - Import from image tags that are not subject to change
  - Reference specific software releases or commit hashes

```bash
# CLHEP
RUN sudo pacman -S --noconfirm \
    wget \n    xerces-c
ENV CLHEP_VERSION 2.3.4.4

RUN wget http://proj-clhep.web.cern.ch/proj-clhep/DISTRIBUTION/tarFiles/clhep-
    tar -xzf clhep.tgz \n    mv $CLHEP_VERSION/CLHEP ./ \n    rm -rf $CLHEP_VERSION \n    mkdir build \n    cd build \n    CXXFLAGS=-std=c++14 cmake ../CLHEP \n    make -j30 \n    sudo make install \n    cd .. 
    rm -rf build CLHEP clhep.tgz

# GEANT4
ENV GEANT4_VERSION 10.3.1

RUN git clone https://github.com/Geant4/geant4.git \n    cd geant4 \n    git checkout tags/vsGEANT4_VERSION \n    cd .. 
    mkdir build \n    cd build \n    cmake ../geant4 
    -DGEANT4_BUILD_CXXSTD=14 
    -DGEANT4_INSTALL_DATA=ON \n    -DGEANT4_USE_GDML=ON \n    -DGEANT4_USE_SYSTEM_CLHEP=ON \n    make -j30 
    sudo make install
```
Container Image Development Practices...

- Can differ significantly from practices of IT industry
  - For IT, there is a strong incentive to have small, single-purpose images
    - IT industry uses containers in cloud
  - On OSG and HPC, it is a different story
    - Images can be large, and it does not affect the amount of IO
    - On OSG, images are fed unpacked over CVMFS, on-demand
    - On HPC, a high-bandwidth connection serves parts of image on-demand
- For me, all software components meant to work together are *packaged together* in an image
  - Images are large: ~5 GiB
  - Only storage quotas apply pressure to keep images from being much larger
- In this usage, container images are less about providing *appliances*, and more about providing a cohesive simulation/reconstruction *environment*
Singularity on OSG

- OSG scripts generate unpacked singularity images served over CVMFS
  - CVMFS offers aggressive caching
  - Docker import can lose some environment information
    - In this case, it is possible to copy proper image files to nodes, but in this case image size matters!
  - Using Singularity limits jobs to a subset of grid resources
- Difficulties with OSG image distribution ultimately has discouraged use
  - My work has instead grown to favor local HPC resources (namely Bebop)
Singularity on Bebop

- New Cray CS400 cluster at ANL
- Shockingly easy!
  1. Load Singularity module
  2. Pull Docker Hub or shub image into local image file
  3. Load image file from nodes over high-speed link
- “Legacy chain” and “Evochain” run out of the box on Broadwell nodes
  - Not so much on KNL nodes: Java apps raise exceptions over insufficient resources
  - J. Taylor Childers discovered that it is max thread limits that prevent Java GC threads from spawning
Summary

- The power of containers can be summed up in the following fact:
  - Our entire simulation and reconstruction was converted over from running on OSG to a brand new HPC cluster in about 2 hours.
- This kind of portability can be a very powerful collaboration tool
  - E.g., people with little to no knowledge of particular simulation/reconstruction software could evaluate the performance of a detector design and/or reconstruction procedure for their physics case
- Other ways to collaborate...
  - Share Dockerfiles
  - Share base images
  - ...