GEMC: speed up, docker and singularity

Deployment with Containers
Shifts, Rotations of magnetic fields
Code speed up
Workflow for simulations on the grid (OSG), and offsite farms

New Tag: 4a.2.4
Docker, Singularity Images

M. Ungaro, Hall-B

GEMC Development: clas12Tag 4a.2.4

JLAB_VERSION goes from **2.1** to **2.2**. In practical terms, this mean geant4 upgraded

4.10.02.p03 to **4.10.03.p02**

(improved track navigation, e.m. physics, physics api small but non backward compatible changes)

Other changes:

CLHEP upgraded to 2.3.4.5
Qt upgraded to 5.9.1
ROOT upgraded to 6.12.06

GEMC Development: clas12Tag 4a.2.4

Magnetic Fields

Torus Map upgraded from 2008 version to "April 2018":

Torus Symmetric map

From gemc.jlab.org:

Translating/Rotating a Field

The options **DISPLACE_FIELDMAP** and **ROTATE_FIELDMAP** can be used to shift a field map origin and/or rotate a field map.

Shift example:

```
-DISPLACE_FIELDMAP="srr-solenoid, 3.5*mm, 0*mm, 0*mm"
```

this will shift the map origin from (0, 0, 0)mm to (3.5, 0, 0)mm

Rotation example:

```
-ROTATE_FIELDMAP="srr-solenoid, 15*deg, 0*deg, 0*deg"
```

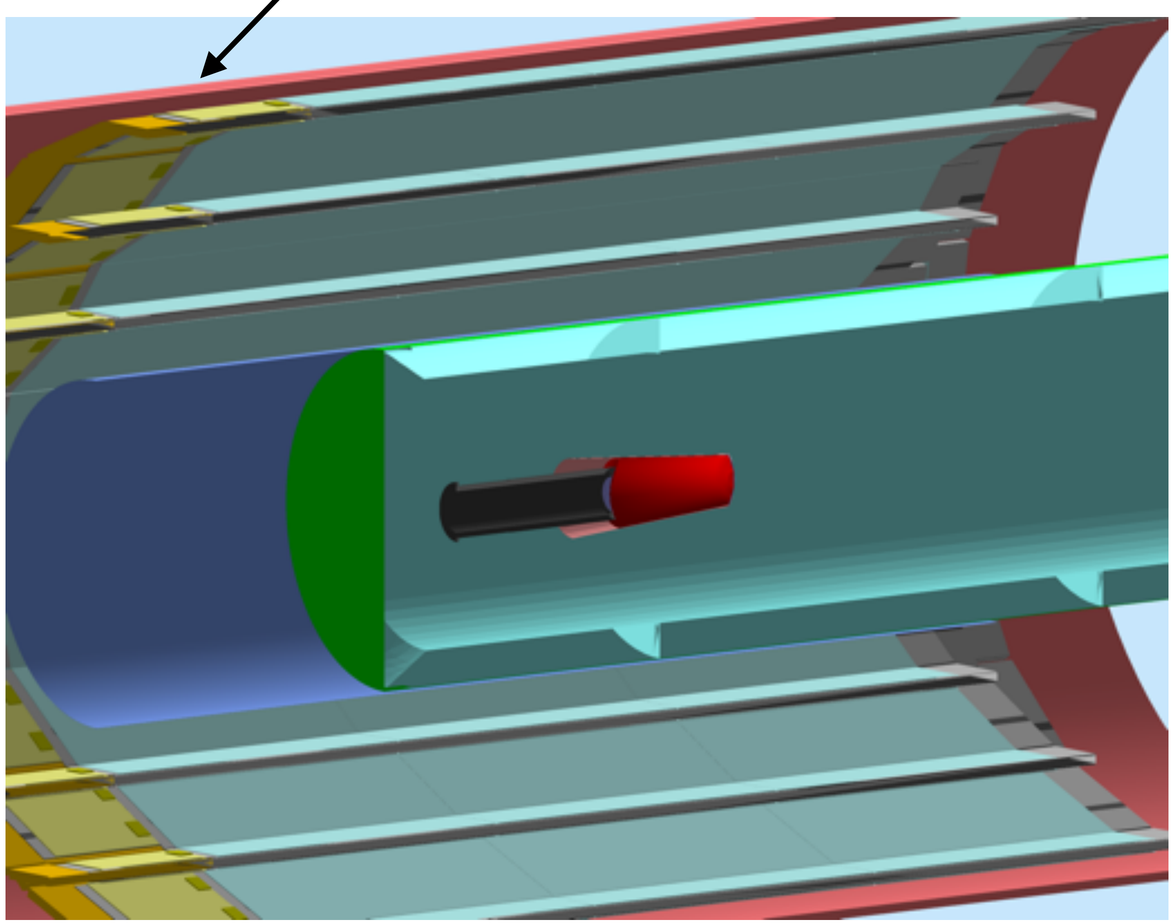
this will rotate the map 15 degrees along the x-axis.

GEMC Development: clas12Tag 4a.2.4

Geometry / Digitization Changes

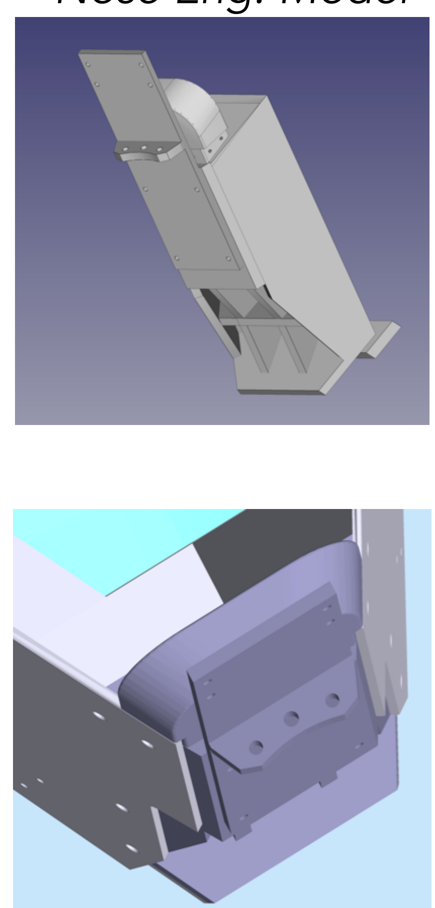
- FMT shift by 8mm
- FMT background hits
- use run number 11 as default in the gcard
- bst tungsten and heat shield
- LTCC Nose CAD model

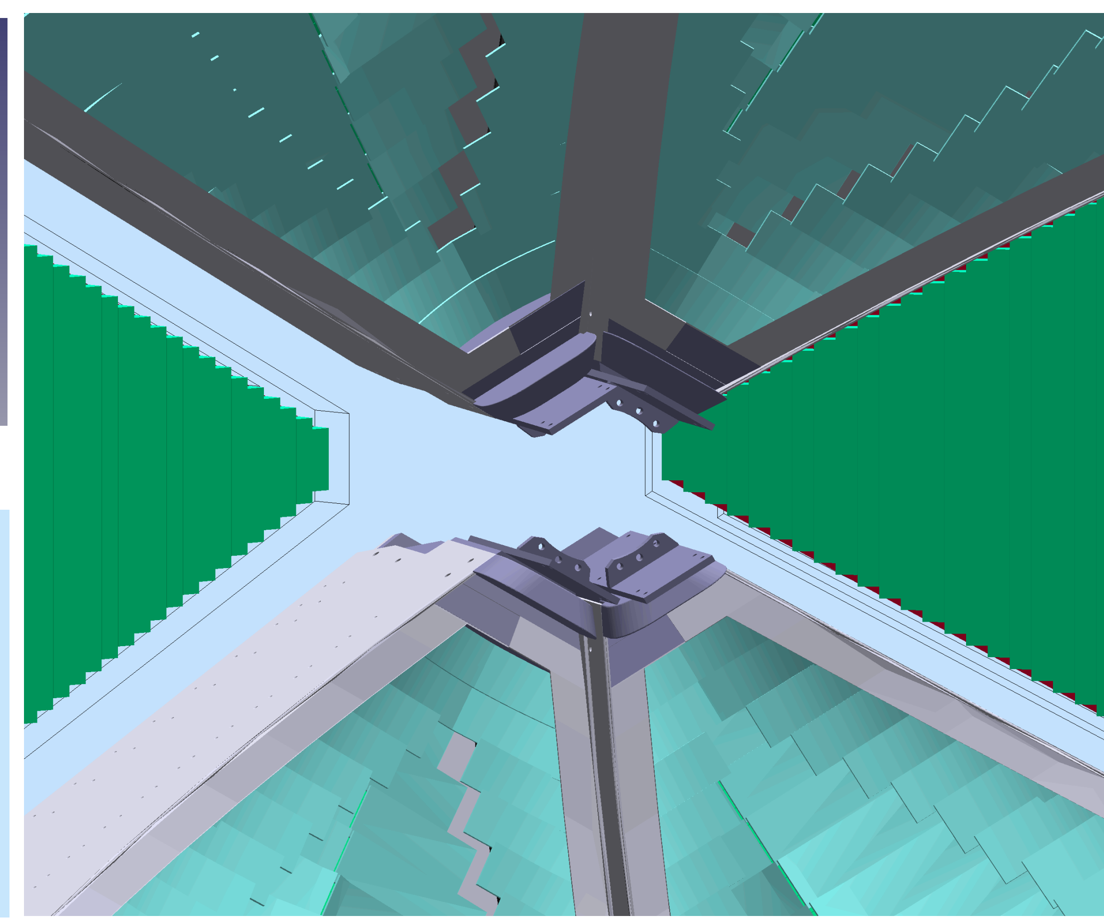
Neoprene 2.7mm



BST Tungsten and Heat Shield

Nose Eng. Model





Nose GEMC implementation

M. Ungaro, Hall-B

GEMC Development: clas12Tag 4a.2.4

env variable "**GEMC_DATA_DIR**" as a base path in the gcard (gcard is now portable to other systems)
to be ported to a command line / gcard option as well

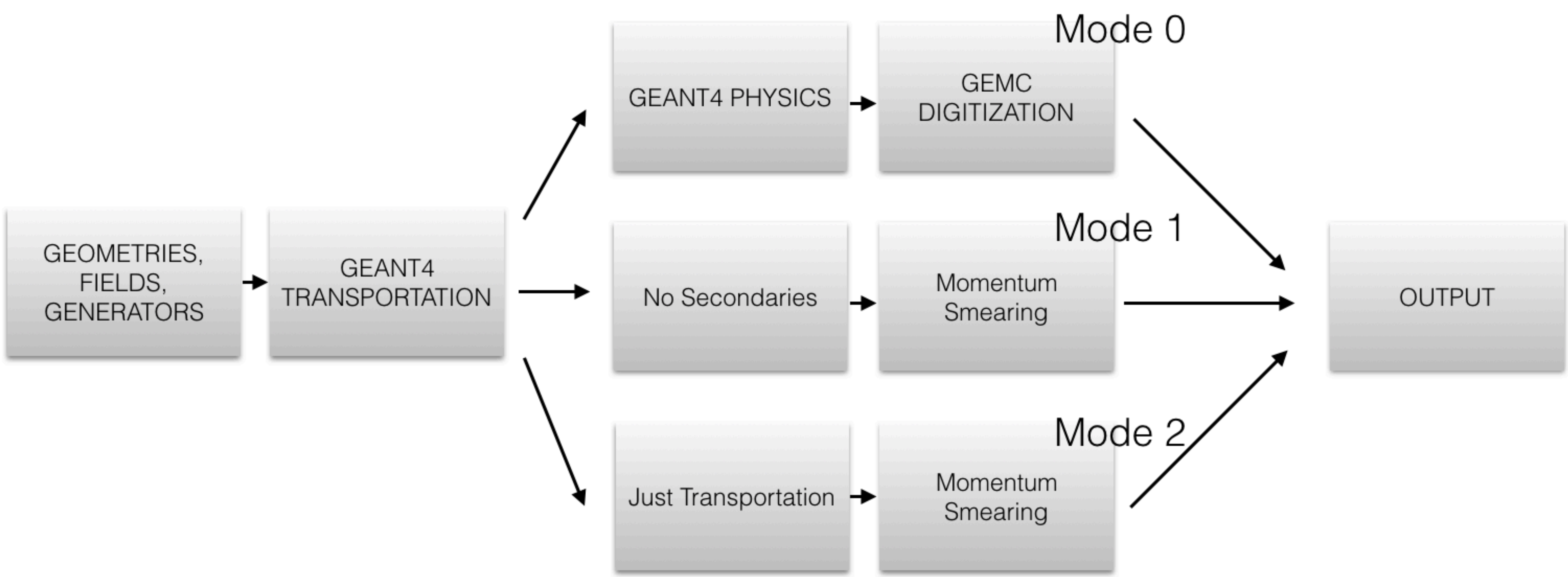
```
<!-- target. Notice variation give the target type. Can be: lH2, lD2, ND3 -->
<detector name="/group/clas12/gemc/4a.2.3/experiments/clas12/targets/cad/"   factory="CAD"/>
<detector name="/group/clas12/gemc/4a.2.3/experiments/clas12/targets/target" factory="TEXT" variation="cad"/>
```

↓

```
<!-- target. Notice variation give the target type. Can be: lH2, lD2, ND3 -->
<detector name="experiments/clas12/targets/cad/"   factory="CAD"/>
<detector name="experiments/clas12/targets/target" factory="TEXT" variation="cad"/>
```

GEMC Development: clas12Tag 4a.2.4

FAST MC Mode fixes for: -FASTMCMODE=[10,20]



Mode 0: is the full geant4 simulation.
Mode 1: turns off secondaries.
Mode 2: turns off all physics except transportation.
Mode 10: Same as mode 1 but with process routines enabled.
Mode 20: Same as mode 2 but with process routines enabled.

- Mode 1: ~10x faster.
- Mode 2: ~200x faster.

GEMC Benchmarks

- 7 GeV e- between 15 and 25 degrees in theta, all phis
- 2 GeV gamma between 15 and 25 degrees in theta, phi=0
- 2 GeV proton at 90 degrees in theta

Each cell adds its component plus all the components above.

Adds Solenoid Field →

Adds Torus Field →

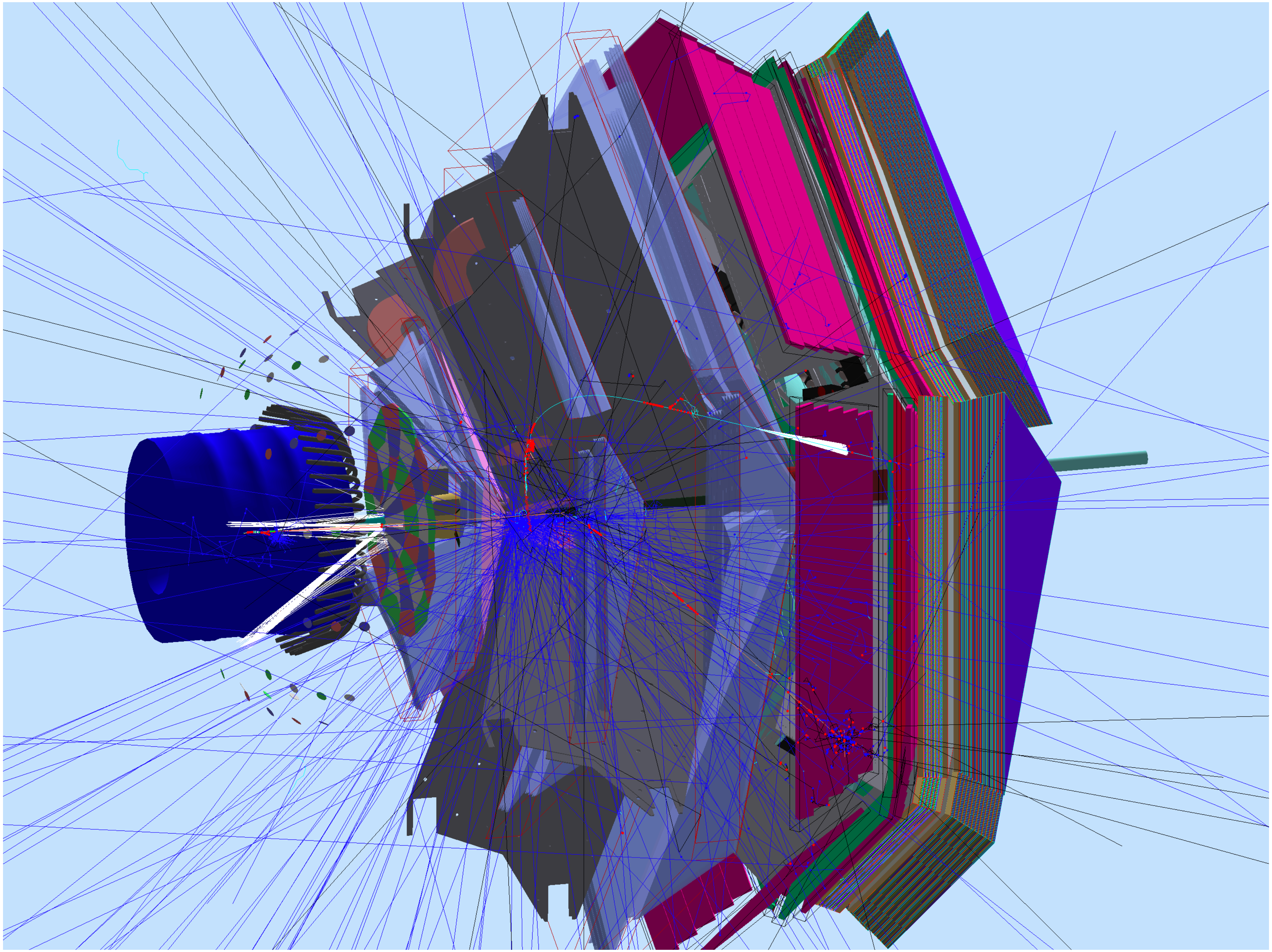
	Production Cut 10 mm		
	time fo 2000 events	Event / seconds	% increase
Target	3.6	555.8	
SVT	5.5	365.9	51.9%
MM	40.9	48.9	648.6%
CTOF	41.1	48.7	0.4%
Solenoid	126.7	15.8	208.4%
HTCC	174.6	11.5	37.9%
FT	179.4	11.2	2.7%
DC	245.8	8.1	37.0%
LTCC	307.1	6.5	24.9%
FTOF	326.3	6.1	6.3%
PCAL	718.8	2.8	120.3%
EC	987.8	2.0	37.4%
Torus	1057.2	1.9	7.0%

3 particles in the CLAS12 fiducial volume at ~500 ms / event

GEMC Benchmarks

What happens if the particles are generated **everywhere**?
(typical generator working conditions)

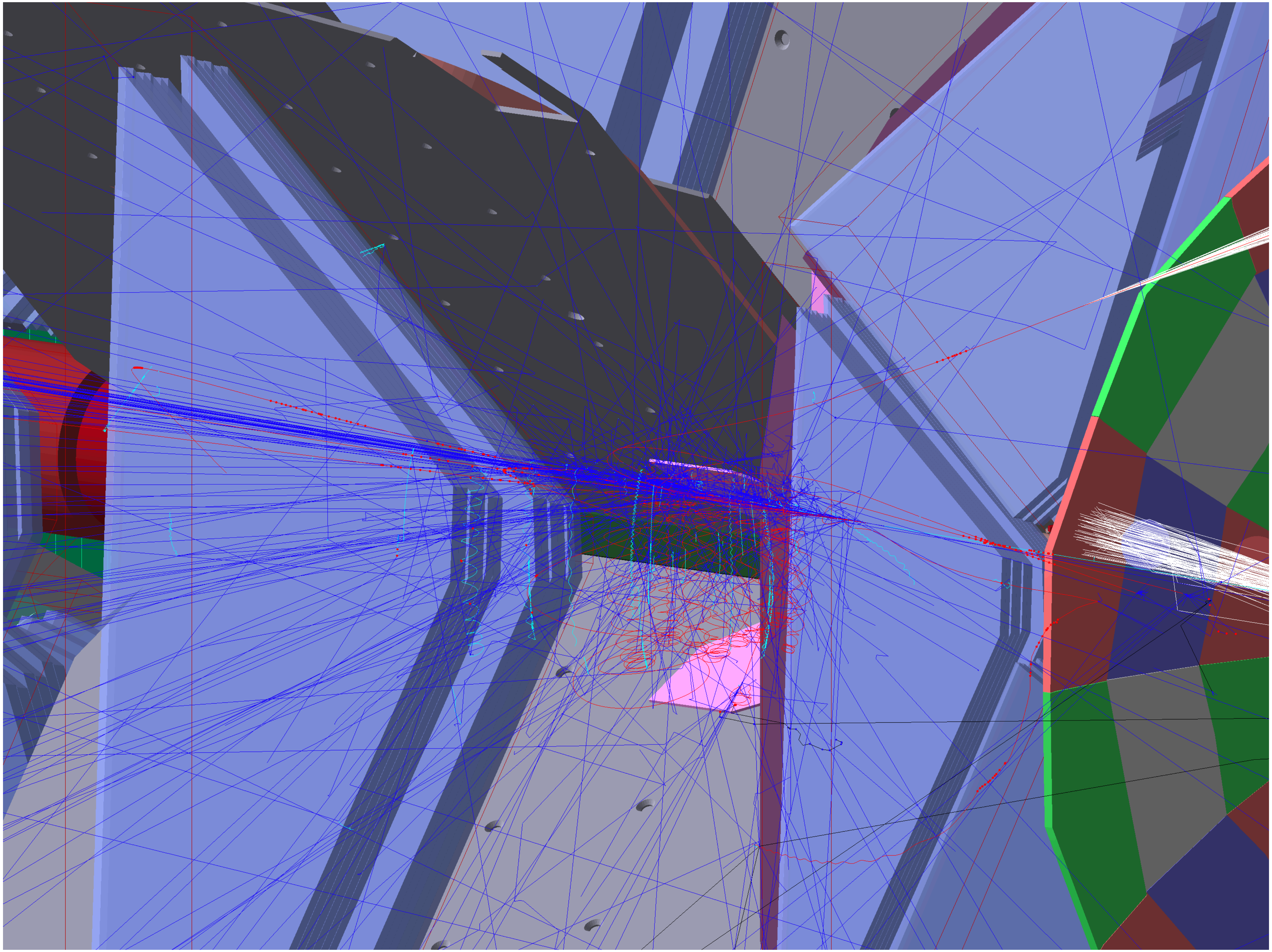
GEMC Benchmarks



electron in torus hardware. Notice: produce secondary electron giving signal in LTCC!

M. Ungaro, Hall-B

GEMC Benchmarks



electron in torus frame. Lots of e+ e- pairs looping in the field.

GEMC Benchmarks

Those (not rare) events came from a rho events generator.
Simulation slow down:

from **500 ms / event** to **1.6 second / event**

- Some of these secondaries produce real hits.
- **Most secondaries** DO NOT, and slow down the simulation unnecessary.

We could eliminate **those secondaries** with the PRODUCTION CUT:

Definition (from geant4, documented @ gemc.jlab.org)

in *geant4* **the production cut is a distance**, not an energy:

- if secondaries can travel more than that distance, they are produced.
- otherwise there is discrete energy loss.

GEMC Benchmarks

New option in gemc can set a production cut for a list of volumes.
(under the hood, this produce a G4Region in geant4):

PRODUCTIONCUTFORVOLUMES

Tentatively: 10cm for 4a.2.4. Bigger than probably needed. Studies undergoing.
The clas12 lines below do not change the DC occupancy coming from luminosity beam background.

```
<!-- production threshold for passive volumes -->

  <!-- beamline shielding: 10cm-->
  <option name="PRODUCTIONCUTFORVOLUMES" value="innerShieldAndFlange, outerFlange, outerMount, nut1, nut2, nut3, nut4,
  nut5, nut6, nut7, nut8, nut9, taggerInnerShield,  main-cone, main-cone, adjuster1, adjuster2, adjuster3,
  DSShieldFrontLead, DSShieldBackLead, DSShieldInnerAss, DSShieldBackAss, DSShieldFrontAss, DSShieldBackCover,
  DSShieldFrontCover, DSShieldFlangeAttachment, DSShieldFlange,  100" />

  <!-- vacuum line: 10cm-->
  <option name="PRODUCTIONCUTFORVOLUMES" value="connectUpstreamToTorusPipe, connectTorusToDownstreamPipe,
  downstreamPipeFlange,  100" />

  <!-- torus magnet: 10cm-->
  <option name="PRODUCTIONCUTFORVOLUMES" value="BoreShield, CenterTube, DownstreamShieldingPlate, DownstreamVacuumJacket,
  WarmBoreTube, apex, Shield1, Shield2, Shield3, Shield4, Shield5, Shield6, Shield7, shell1a, shell1b, shell2a, shell2b,
  shell3a, shell3b, shell4a, shell4b, shell5a, shell5b, shell6a, shell6b, 100" />
```

This reduces the rho events from 1.6 to 0.7 s / event. More studies needed.
More volumes can be added, cuts optimized.

GEMC Distribution

Software installation instructions (JLAB_VERSION 2.2)

These instructions are verified on:

Linux:

- CentOS 7

Mac:

- Sierra
- High Sierra

These are instructions to install JLAB_VERSION 2.2. You can find supported JLAB_VERSION releases instructions below:

- Development Version
- Version 2.2
- Version 2.1
- Version 2.0

General Requirements:

- csh shell
- cmake >= 3.3 (for geant4. check with *cmake --version*)
- network connection to download the various packages during installation
- wget
- scons version 2 to 2.5.1
- mysql
- git
- recommended: qt

⌘ MAC OS specific requirements

Make sure you have the latest xcode tools with:

```
xcode-select --install
```

You can install the dependencies with homebrew:

List of packages

Ubuntu (Debian-line) specific requirements

You can install the dependencies with apt-get:

List of packages

Fedora (red-hat line) specific requirements

You can install the dependencies with dnf (or yum):

List of packages

Installation:

1. Choose a place (JLAB_ROOT) to install the software

   A good choice for **JLAB_ROOT** is **/opt/jlab_software**:

   ```
   set JLAB_ROOT=/opt/jlab_software
   ```

   Create the working directory:

   ```
   mkdir -p $JLAB_ROOT/2.2
   ```

2. Get the installation scripts:

   ```
   cd $JLAB_ROOT
   wget http://www.jlab.org/12gev_phys/packages/sources/ceInstall/ceInstall_2.2.tar.gz
   tar -zxpf ceInstall_2.2.tar.gz --strip-components 1 -C 2.2
   ```

3. Set the environment variables. You should put these lines in your login script:

   ```
   setenv JLAB_ROOT /opt/jlab_software
   source $JLAB_ROOT/2.2/ce/jlab.csh
   ```

4. Start a new shell. Now you should see a log similar to this

   ```
   > Common Environment Version: <2.2>  (Wed March 18 2018)
   > Running as mauri on 1.800.gemc.cool.org
   > OS Release:    Darwin_macosx10.13-x86_64-clang9.0.0
   > JLAB_ROOT set to:     /opt/jlab_software
   > '/opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0' is not a directory. Creating it.

   !! Attention:  BANKS  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/banks/1.4/bin  not found
   !! Attention:  CCDB  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/ccdb/ccdb-1.06.02/lib  not found
   !! Attention:  CLHEP  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/clhep/2.3.4.5/lib  not found
   !! Attention:  EVIO  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/evio/5.1/bin  not found
   !! Attention:  GEANT4  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/geant4/4.10.03.p02/bin  not found
   !! Attention:  GEMC  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/gemc/2.7/gemc  not found
   !! Attention:  JANA  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/jana/0.7.7p1/bin/jana  not found
   !! Attention:  MLIBRARY  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/mlibrary/1.2/lib  not found
   !! Attention:  MYSQL  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/mysql/lib  not found
   !! Attention:  QT    installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/qt/5.9.1/5.9.1/clang_64/lib  not found
   !! Attention:  ROOT  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/root/6.12.06/bin/root-config  not found
   !! Attention:  SCONS  installation check /opt/jlab_software/2.2/scons_bm/1.6/site_tools  not found
   !! Attention:  XERCESC  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/xercesc/3.2.0/lib  not found
   ```

   If you do, great! The enviroment seems correct and you're ready to run the installation scripts. Every script that you run below will replace the warning log with a successful one. Notice: if you already have some existing environment variables that you want to keep, use the option "keepmine":

   ```
   source $JLAB_ROOT/2.2/ce/jlab.csh keepmine
   ```

5. Choose the packages you wish to install by executing the corresponding script:

   ```
   cd $JLAB_ROOT/2.2/install

   ./go_clhep
   ./go_xercesc
   ```

   If you have a system installation of qt (see instructions on requirements on how to install it):

   ```
   ./go_qt system
   ```

   Otherwise the following command will downlowad the qt installer for you. Follow instructions for the installation:

   ```
   ./go_qt
   ```

   Continue with the rest of the package installation:

   ```
   ./go_geant4
   ./go_sconsscript
   ./go_evio
   ./go_mysql
   ./go_ccdb
   ./go_mlibrary
   ./go_gemc
   ./go_fields
   ./go_root
   ./go_banks
   ./go_jana
   ```

GEMC Distribution

Software installation instructions (JLAB_VERSION 2.2)

These instructions are verified on:

Linux:

- CentOS 7

Mac:

- Sierra
- High Sierra

These are instructions to install JLAB_VERSION 2.2. You can find supported JLAB_VERSION releases instructions below:

- Development Version
- Version 2.2
- Version 2.1
- Version 2.0

General Requirements:

- csh shell
- cmake >= 3.3 (for geant4. check with *cmake --version*)
- network connection to download the various packages during installation
- wget
- scons version 2 to 2.5.1
- mysql
- git
- recommended: qt

⌘ MAC OS specific requirements

Make sure you have the latest xcode tools with:

```
xcode-select --install
```

You can install the dependencies with homebrew:

List of packages

Ubuntu (Debian-line) specific requirements

You can install the dependencies with apt-get:

List of packages

Fedora (red-hat line) specific requirements

You can install the dependencies with dnf (or yum):

List of packages

After many years, even with many improvements on the installation scripts, the compilation of gemc from scratch remains a **difficult task**.

geant4, qt, cadmesh, various compilers, scons and cmake, etc present challenges with their upgrades and virtually every release of Mac and Linux OSes. And that's ok: we want to use the latest technologies and bug fixes. But it makes the **installation grueling on the users and on system admins for offsite farms.**

Installation:

1. Choose a place (JLAB_ROOT) to install the software

   A good choice for **JLAB_ROOT** is **/opt/jlab_software**:

```
set JLAB_ROOT=/opt/jlab_software
```

   Create the working directory:

```
mkdir -p $JLAB_ROOT/2.2
```

2. Get the installation scripts:

```
cd $JLAB_ROOT
wget http://www.jlab.org/12gev_phys/packages/sources/ceInstall/ceInstall_2.2.tar.gz
tar -zxpf ceInstall_2.2.tar.gz --strip-components 1 -C 2.2
```

3. Set the environment variables. You should put these lines in your login script:

```
setenv JLAB_ROOT /opt/jlab_software
source $JLAB_ROOT/2.2/ce/jlab.csh
```

4. Start a new shell. Now you should see a log similar to this

```
> Common Environment Version: <2.2>  (Wed March 18 2018)
> Running as mauri on 1.800.gemc.cool.org
> OS Release:    Darwin_macosx10.13-x86_64-clang9.0.0
> JLAB_ROOT set to:     /opt/jlab_software
> '/opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0' is not a directory. Creating it.

!! Attention:  BANKS  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/banks/1.4/bin  not found
!! Attention:  CCDB  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/ccdb/ccdb-1.06.02/lib  not found
!! Attention:  CLHEP  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/clhep/2.3.4.5/lib  not found
!! Attention:  EVIO  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/evio/5.1/bin  not found
!! Attention:  GEANT4  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/geant4/4.10.03.p02/bin  not found
!! Attention:  GEMC  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/gemc/2.7/gemc  not found
!! Attention:  JANA  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/jana/0.7.7p1/bin/jana  not found
!! Attention:  MLIBRARY  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/mlibrary/1.2/lib  not found
!! Attention:  MYSQL  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/mysql/lib  not found
!! Attention:  QT    installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/qt/5.9.1/5.9.1/clang_64/lib  not found
!! Attention:  ROOT  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/root/6.12.06/bin/root-config  not found
!! Attention:  SCONS  installation check /opt/jlab_software/2.2/scons_bm/1.6/site_tools  not found
!! Attention:  XERCESC  installation check /opt/jlab_software/2.2/Darwin_macosx10.13-x86_64-clang9.0.0/xercesc/3.2.0/lib  not found
```

   If you do, great! The enviroment seems correct and you're ready to run the installation scripts. Every script that you run below will replace the warning log with a successful one. Notice: if you already have some existing environment variables that you want to keep, use the option "keepmine":

```
source $JLAB_ROOT/2.2/ce/jlab.csh keepmine
```

5. Choose the packages you wish to install by executing the corresponding script:

```
cd $JLAB_ROOT/2.2/install

./go_clhep
./go_xercesc
```

   If you have a system installation of qt (see instructions on requirements on how to install it):

```
./go_qt system
```

   Otherwise the following command will downlowad the qt installer for you. Follow instructions for the installation:

```
./go_qt
```

   Continue with the rest of the package installation:

```
./go_geant4
./go_sconsscript
./go_evio
./go_mysql
./go_ccdb
./go_mlibrary
./go_gemc
./go_fields
./go_root
./go_banks
./go_jana
```

GEMC Distribution: Docker images

No requirements on the OS version, or the OS (Mac, Linux, Windows)
No libraries needed other than have the "docker" app installed.
(https://www.docker.com/community-edition)

Installing gemc is now a one liner.

GEMC Distribution: Docker images

```
docker run -it --rm jeffersonlab/clas12tags:4a.2.4 bash
```

GEMC Distribution: Docker images

```
docker run -it --rm jeffersonlab/clas12tags:4a.2.4 bash
```

jefferson lab cloud docker repository

The first time you run this, it will download the image (just a few minutes).
Every subsequent time execution is instantaneous.

GEMC Distribution: Docker images

```
docker run -it --rm jeffersonlab/clas12tags:4a.2.4 bash
```

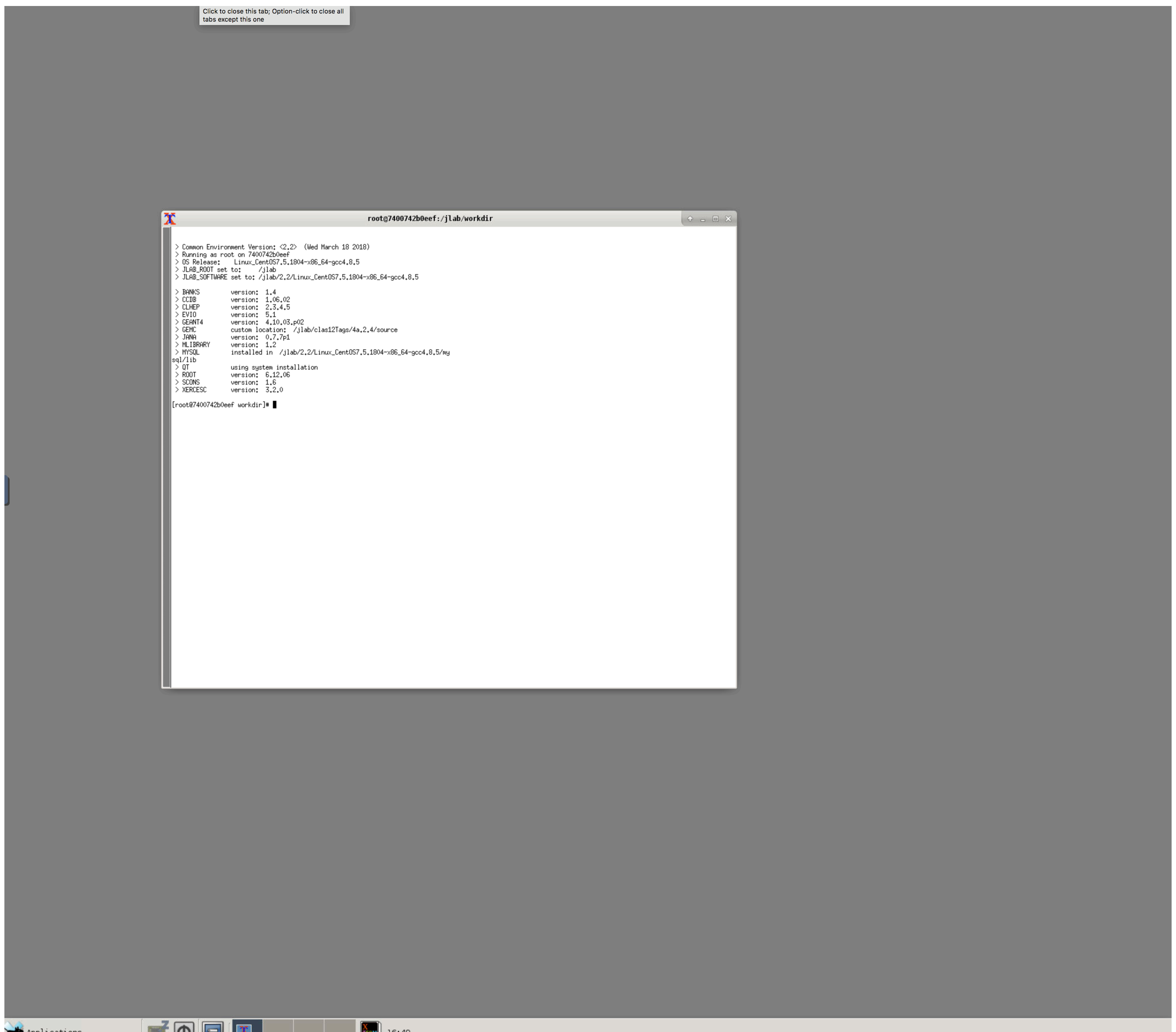
```
%>docker run -it --rm jeffersonlab/clas12tags:4a.2.4 bash


 > Common Environment Version: <2.2>  (Wed March 18 2018)
 > Running as root on d83191ce8cd4
 > OS Release:    Linux_CentOS7.5.1804-x86_64-gcc4.8.5
 > JLAB_ROOT set to:     /jlab
 > JLAB_SOFTWARE set to: /jlab/2.2/Linux_CentOS7.5.1804-x86_64-gcc4.8.5

 > BANKS    version:  1.4
 > CCDB     version:  1.06.02
 > CLHEP    version:  2.3.4.5
 > EVIO     version:  5.1
 > GEANT4   version:  4.10.03.p02
 > GEMC     custom location:  /jlab/clas12Tags/4a.2.4/source
 > JANA     version:  0.7.7p1
 > MLIBRARY    version:  1.2
 > MYSQL    installed in  /jlab/2.2/Linux_CentOS7.5.1804-x86_64-gcc4.8.5/mysql/lib
 > QT       using system installation
 > ROOT     version:  6.12.06
 > SCONS    version:  1.6
 > XERCESC     version:  3.2.0

[root@d83191ce8cd4 workdir]#
```

You can now run gemc (batch mode).

GEMC Distribution: Docker images

```
docker run -it --rm -p 6080:6080 jeffersonlab/clas12tags:4a.2.4
```

http://localhost:6080



You can now run gemc (interactive mode).

GEMC Distribution: Docker images

Advantages of Docker over virtual machines

- Virtual machines needs to be booted every time.
- They are slow.
- They are resources hogs.
- Running intensive openGL apps (like GEMC) can be problematic.
- If I change a small thing in gemc, the whole image needs to be reinstalled. Unless users do it by hand (then we're back to square 1).

GEMC Distribution: Docker images

Advantages of Docker over virtual machines

- Virtual machines needs to be booted every time.
- They are slow.
- They are resources hogs.
- Running intensive openGL apps (like GEMC) can be problematic.
- If I change a small thing in gemc, the whole image needs to be reinstalled. Unless users do it by hand (then we're back to square 1).

- No booting: execution is immediate.
- Docker is fast.
- It's an app running on a shell. Can be seen with a browser or a vnc client.
- Containers are layered up. A small fix in gemc means re-downloading a small container only.
- Local disk can be made available to docker, so one can get gemc from repo and recompile. No libraries needed.
- Image storage in the cloud, tagged.
- Base for singularity.

GEMC Distribution: Docker images

How fast is it?

Test on 4a.2.4, full clas12 geometry, all fields on.

```
gemc clas12.gcard -USE_GUI=0 -N=100 -BEAM_P="e-, 4*GeV, 20*deg, 10*deg"
```

```
Total gemc time: 128.21 seconds.
Events only time: 32.5758 seconds.
```

```
Total gemc time: 92.46 seconds.
Events only time: 22.1616 seconds.
```

```
Total gemc time: 117.02 seconds.
Events only time: 22.6159 seconds.
```

GEMC Distribution: Docker images

How fast is it?

Test on 4a.2.4, full clas12 geometry, all fields on.

```
gemc clas12.gcard -USE_GUI=0 -N=100 -BEAM_P="e-, 4*GeV, 20*deg, 10*deg"
```

```
Total gemc time: 128.21 seconds.
Events only time: 32.5758 seconds.
```

ifarm1401

```
Total gemc time: 92.46 seconds.
Events only time: 22.1616 seconds.
```

```
Total gemc time: 117.02 seconds.
Events only time: 22.6159 seconds.
```

GEMC Distribution: Docker images

How fast is it?

Test on 4a.2.4, full clas12 geometry, all fields on.

```
gemc clas12.gcard -USE_GUI=0 -N=100 -BEAM_P="e-, 4*GeV, 20*deg, 10*deg"
```

```
Total gemc time: 128.21 seconds.
Events only time: 32.5758 seconds.
```

ifarm1401

```
Total gemc time: 92.46 seconds.
Events only time: 22.1616 seconds.
```

```
Total gemc time: 117.02 seconds.
Events only time: 22.6159 seconds.
```

my imac

GEMC Distribution: Docker images

How fast is it?

Test on 4a.2.4, full clas12 geometry, all fields on.

```
gemc clas12.gcard -USE_GUI=0 -N=100 -BEAM_P="e-, 4*GeV, 20*deg, 10*deg"
```

```
Total gemc time: 128.21 seconds.
Events only time: 32.5758 seconds.
```

ifarm1401

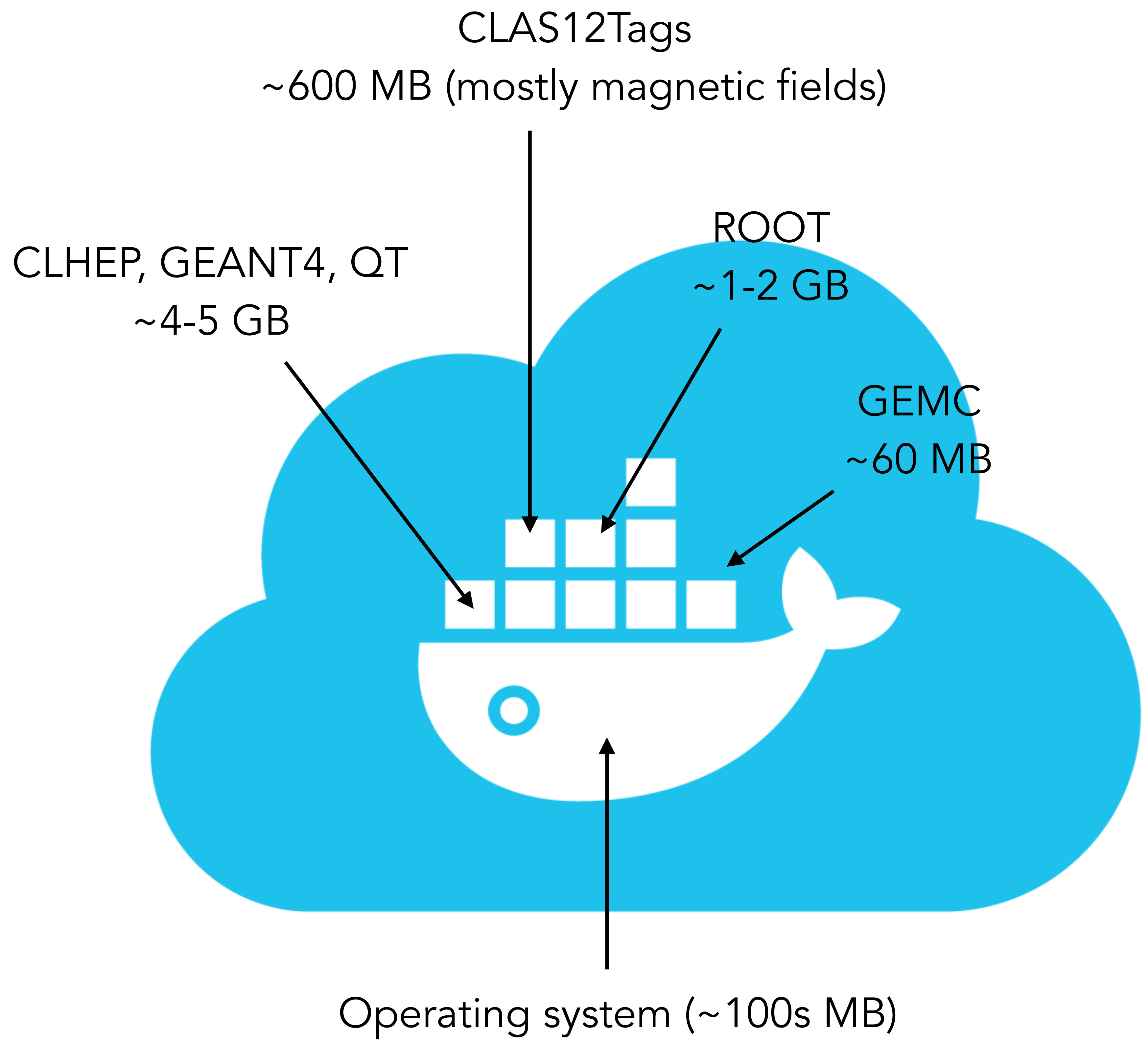
```
Total gemc time: 92.46 seconds.
Events only time: 22.1616 seconds.
```

docker

```
Total gemc time: 117.02 seconds.
Events only time: 22.6159 seconds.
```

my imac

GEMC Distribution: Docker images



Docker and Singularity

Singularity is "the" secure mode to deploy jobs based on docker images on farms.
Singularity is installed on the JLab farm.

Docker and Singularity

Singularity is "the" secure mode to deploy jobs based on docker images on farms.
Singularity is installed on the JLab farm.

Singularity is used at the Open Science Grid (OSG).
Singularity is used at the National Energy Research Scientific Computing Center (NERSC)

How does it work on the grid?
A singularity container is **built from a docker container** and
automagically distributed over ALL the OSG nodes (all the offsite farms).

Docker and Singularity

Singularity is "the" secure mode to deploy jobs based on docker images on farms.
Singularity is installed on the JLab farm.

Singularity is used at the Open Science Grid (OSG).
Singularity is used at the National Energy Research Scientific Computing Center (NERSC)

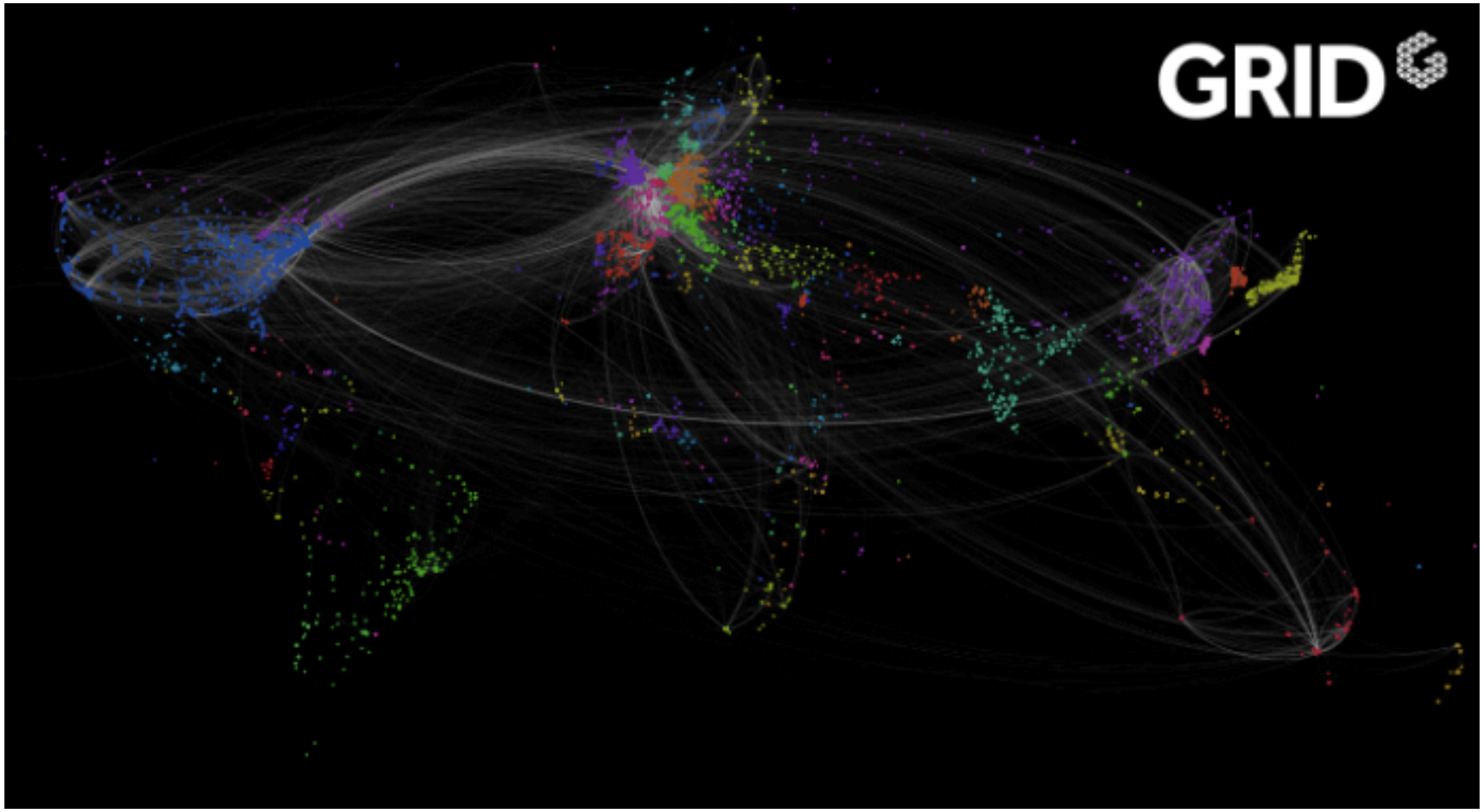
Singularity is in PRODUCTION by GLUEX for their simulation/reconstruction on the farm.
Last month: 100% simulation on OSG, 2B events.

JLab CC is working on Swift 2:
Seamless submission of jobs to OSG NERSC

CLAS12 Simulations, Docker and Singularity

- CLAS12 can now request to be a formal member of OSG (new applications were closed)
- MIT may hire post-doc to help CC in the Swift 2 / Grid effort.
- Universities, Offsite farms can use CLAS12 accounts to add their farm to the grid

JLAB USER
JSUB
Command



Output File

Currently: 10 GB / s connection.
FY2020: should be 100 GB connection.

Summary, Outlook

- 4a.2.4: Same as 4a.2.3 but uses JLAB_VERSION 2.2. In addition:
    - Use new torus field map
    - FMT shift by 8mm
    - use run number 11 as default in the gcard
    - FMT background hits
    - production cut set for individual volumes in the options
    - new geant4 version
    - env variable "GEMC_DATA_DIR" as a base path in the gcard (gcard is now portable to other systems)
    - bst tungsten and heat shield
    - LTCC Nose CAD model
    - magnetic field map displacements and rotations with command line options
    - FAST MC mode 10, 20 output fixed

- Geometry additions
- Shifts, Rotations of magnetic fields
- Code speed up: 2x improvements, can it be 3x? More studies needed.
- Deployment with Containers: Docker is operational: CLAS12 Tags 4a.2.4. Please try.
- Exciting prospects for CLAS12 simulations on OSG