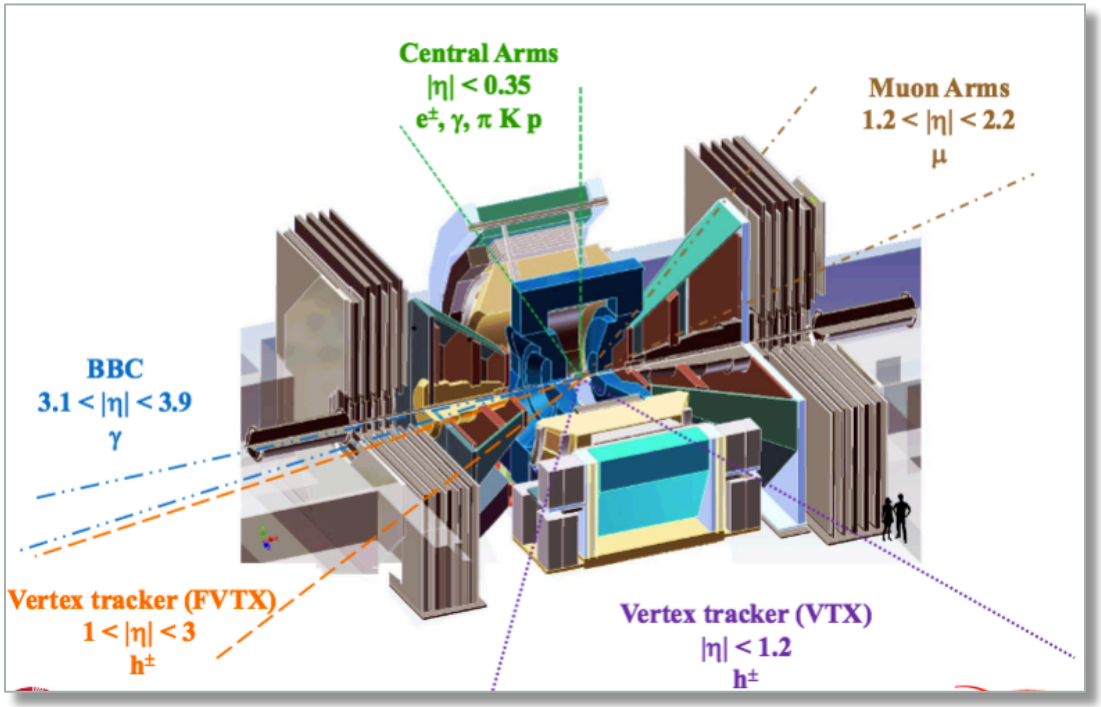
SIMULATING 5 TRILLION EVENTS ON THE OPEN SCIENCE GRID

Martin L. Purschke, Jin Huang, BNL

for the PHENIX collaboration

PHENIX and sPHENIX

- Currently, we are running the PHENIX detector at the Relativistic Heavy Ion Collider about a mile away from here
- 15 years of running
- Hundreds of publications
- One more Run to come
- Big upgrade to sPHENIX in the works
- Timeline early 20's
- sPHENIX is an almost completely new experiment



The PHENIX Detector

PHENIX made history

- In physics results, *and* in the DAQ/Computing arena

Tiny Drops of Early Universe 'Perfect' Fluid

First results from collisions of three-particle ions with gold nuclei reveal clear-cut evidence of primordial soup's signature particle flow

Need for Speed: Where we are

Lvl1-Triggers in Heavy Ions have a notoriously low rejection factor
that's because so many events have *something* that's interesting (different from LHC)
But hey, we could write out almost everything that RHIC gave us, so why bother... this approach has served us really well.
It also opened up access to processes that you can't exactly trigger on, it "just" takes some more work offline.

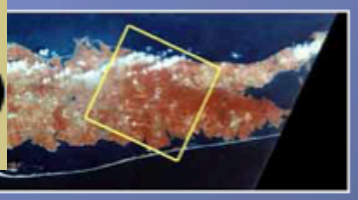
15

LHC-Era Data Rates in 2004 and 2005 Experiences of the PHENIX Experiment with a PetaByte of Data

Martin L. Purschke, Brookhaven National Laboratory
PHENIX Collaboration

chep 06

Mumbai



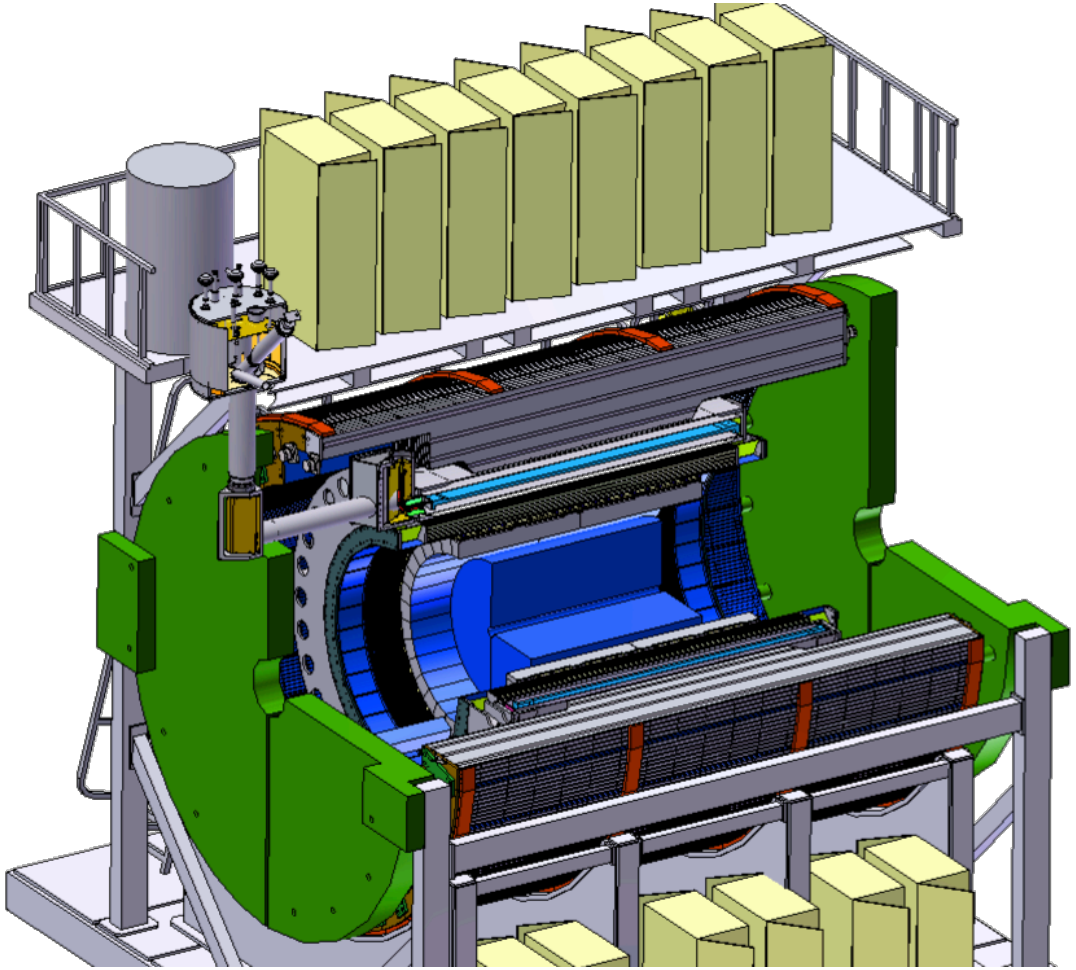
Long Island, NY

1

In with the new…

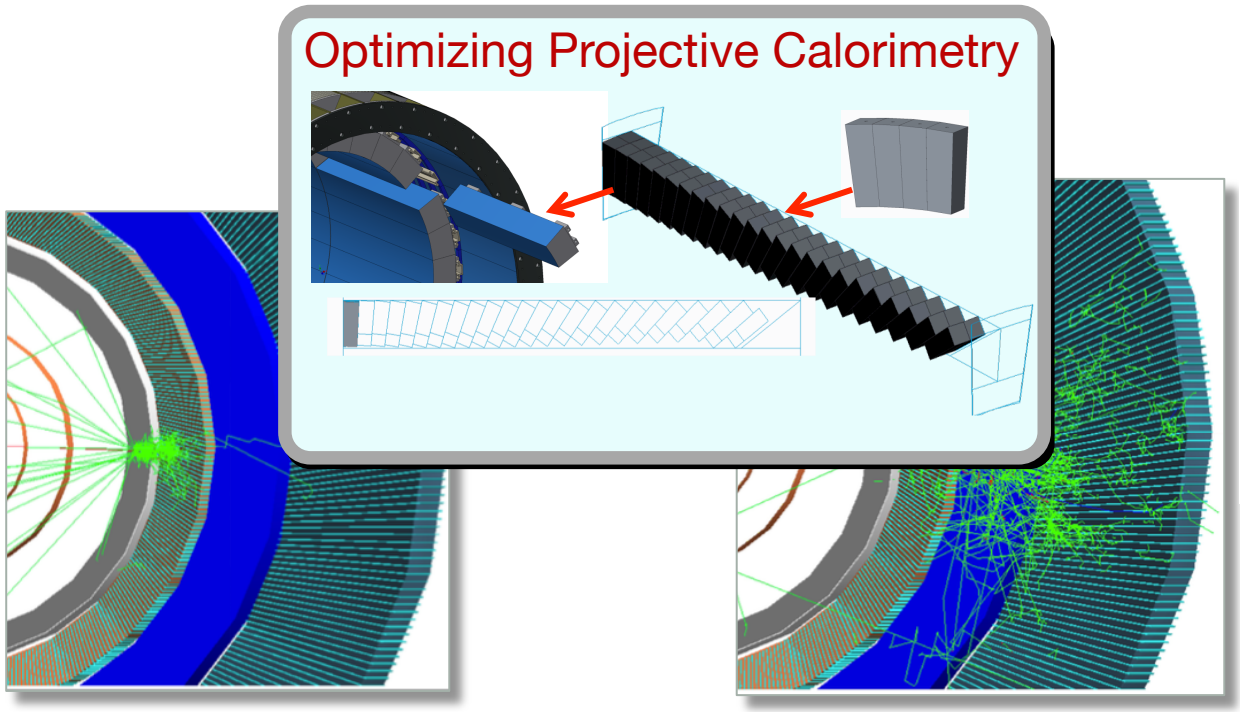
- sPHENIX is based on the former BaBar magnet
- Calorimetry and tracking
- Focus on jet physics

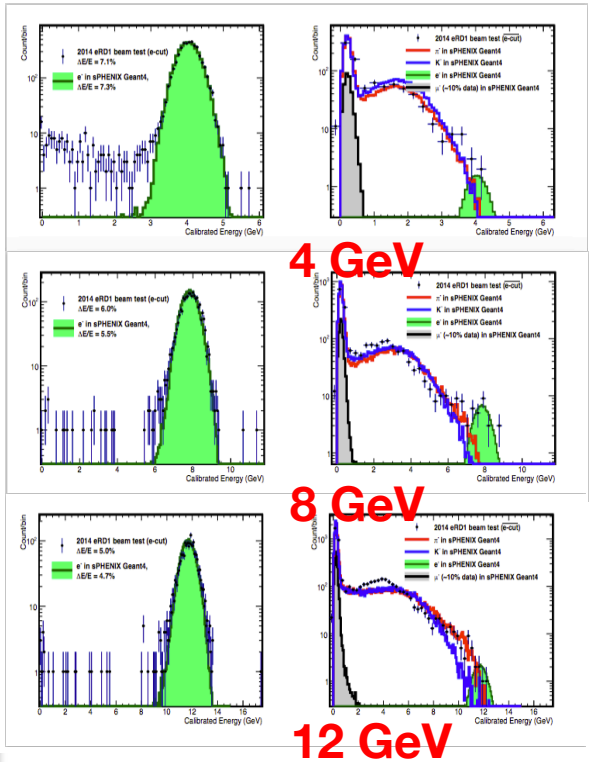
The magnet arrives in Bldg 912



Extensive G4 simulations in progress

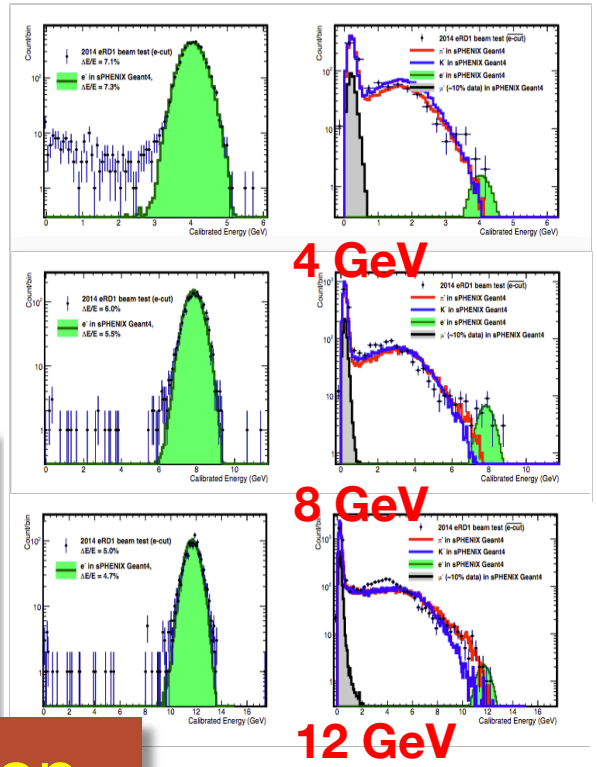
- Calorimeter module design (“2-d projective design”)
- Performance & resolution studies
- sPHENIX simulations currently using ~1/3 of RCF’s resources

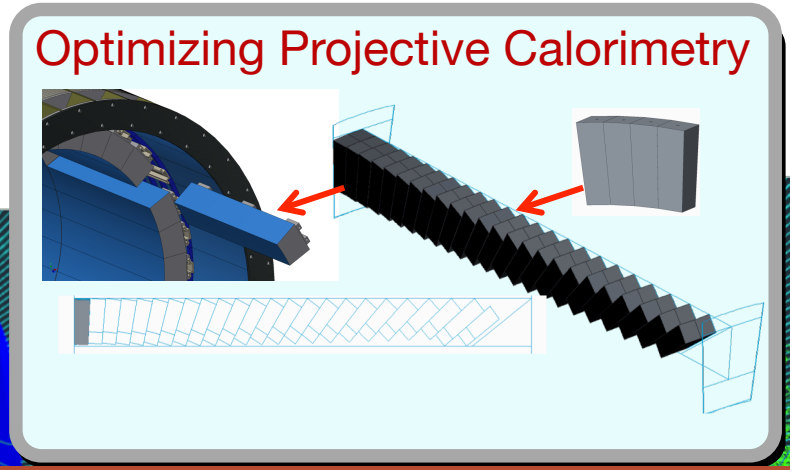




Extensive G4 simulations in progress

- Calorimeter module design ("2-d projective design")
- Performance & resolution studies
- sPHENIX simulations currently using ~1/3 of RCF's resources





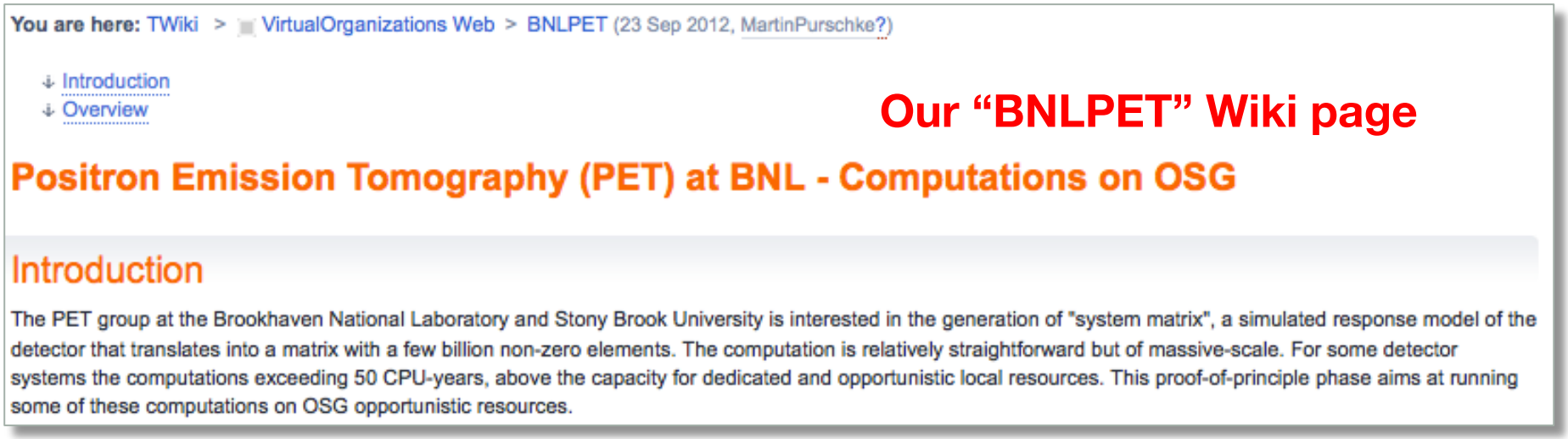
... and then, a *really* big simulation project came along…

"Can we simulate 5 Trillion Events?"

- This is the odd-one-out simulations project
- Determining the various contributions to the muon production (Drell-Yan, Heavy Flavor, combinatorial) in the forward region
- Looking for (simulated) events which have one or more muons in the extreme forward region ($1 < \eta < 5$) with the Pythia event generator
- These are somewhat rare processes ($\sim 1.4 * 10^{-3}$)
- No way to influence the event generator to only produce desired event topologies – leads to severe biases
- brute-force crank through events and discard the unwanted topologies
- This gives this project an unusually low IO/CPU ratio

Can we? Yes

- Just 2 years or so ago, the answer would have been “no”
- I had run simulations at the half-million CPU hours level on the OSG in a completely different context (Medical Imaging)
- cashed in on that expertise here


You are here: TWiki > VirtualOrganizations Web > BNLPET (23 Sep 2012, MartinPurschke?)

Introduction
Overview

Our “BNLPET” Wiki page

Positron Emission Tomography (PET) at BNL - Computations on OSG

Introduction

The PET group at the Brookhaven National Laboratory and Stony Brook University is interested in the generation of "system matrix", a simulated response model of the detector that translates into a matrix with a few billion non-zero elements. The computation is relatively straightforward but of massive-scale. For some detector systems the computations exceeding 50 CPU-years, above the capacity for dedicated and opportunistic local resources. This proof-of-principle phase aims at running some of these computations on OSG opportunistic resources.

- In principle, the sPHENIX sims are “just a much larger project”…
- The issues are in the details, though

Work Breakdown

Goal is to run jobs for ~10-12 hours

Optimum was found to be 10 million events/job (try and error)

Half a million jobs!

No way to do this manually!

Challenges

- The earlier PET-related simulations only had a simple monolithic executable, a few scripts and data files, and 20,000 jobs or so
- Here we are dragging an entire framework along
- How to get the framework to the execution node?
- How to get the data back “home” (the RCF, that is)
- How to automate everything that it remains a manageable endeavor that leaves time for my day job?
- Target was a commitment of 10 days setup + 3 hours/ week for 10 weeks of routine running
- Also, “just me” – with no offense to anyone, automation beats more warm bodies

How to bring the project to and from the remote node

- “to” is, in principle, the easier part
- Looked at cvmfs, but… not set up for us, not agile enough by far
- No shortage of anonymous – not authenticated – ways to pull data from somewhere. wget, ftp, git,…
- Getting data back to RCF is harder – no non-authenticated way to transfer, no safe method to furnish a job with the proper credentials (usually ssh keys)
- Settled on condor file transfers for both directions
- Data flowing back to the submitter host’s /local-scratch
- I/O levels well below “the radar”

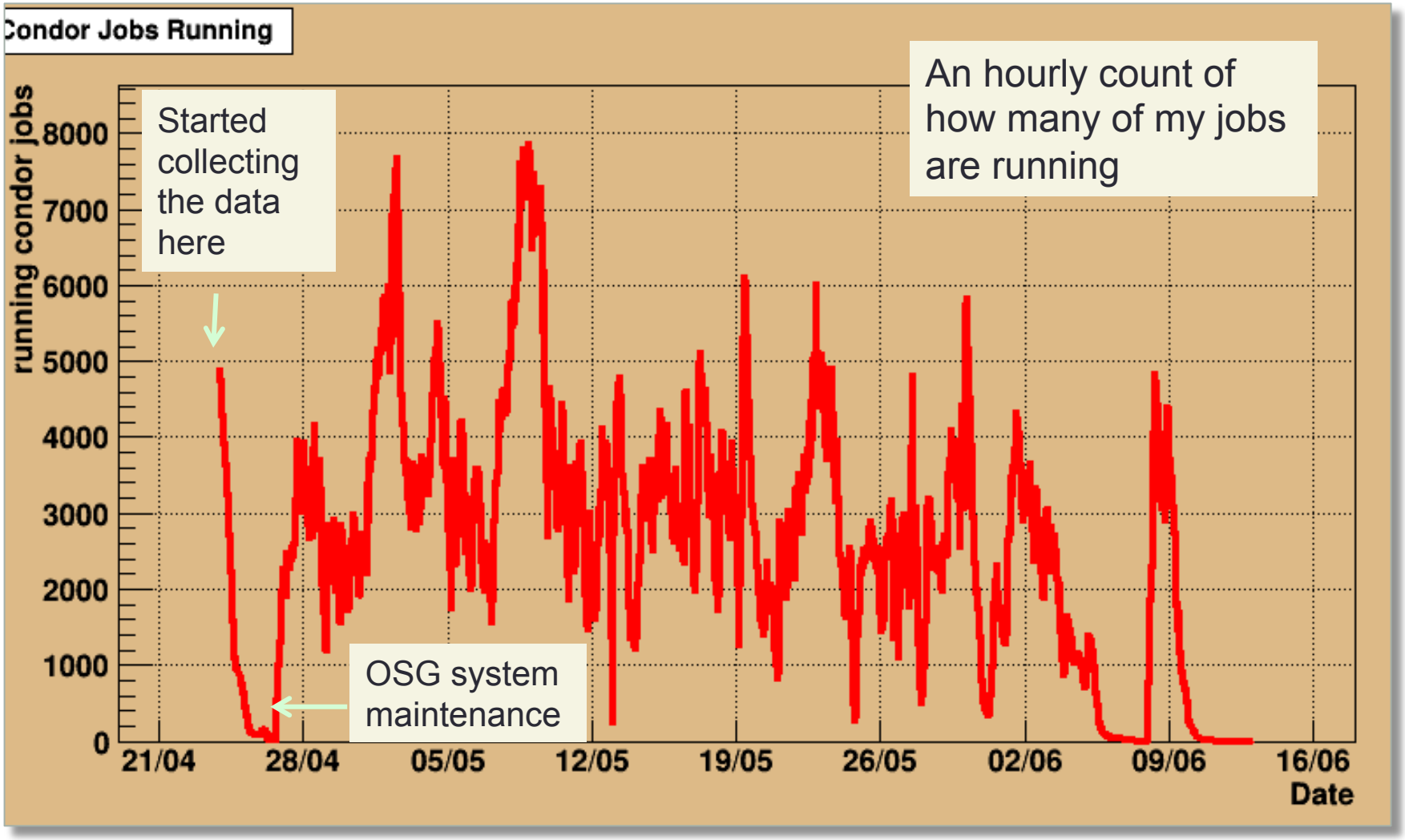
In numbers…

- A job needs about 1550MB in executables and shared libraries to run
- All are 64bit binaries (more later)
- Put a “support bundle” together with all required files, bzip2-compressed to 17% -> 253MB
- Output is just a few 10’s of MB
- Output flowing back to submitter host is getting rsync’ed back to RCF continuously, every 2hrs or so (ssh-agent is your friend)
- Got into a routine to delete files from SH Wednesdays and Sundays – “safe delete procedure”
- Got close to quota only once when I was alone on the OSG for a while

Worker nodes varieties

- Initially encountered strange failures on nodes that lacked standard, system-level libraries
- Not sure why a console application would need a nvidia graphics lib on *some* nodes, but that's what happened
- Identified a superset of 60 additional libraries which were missing on some nodes, 2nd on-demand support bundle
- Tested the executable with ldd, fetched and unpacked 2nd bundle if failure, bend LD_LIBRARY_PATH accordingly
- After that, less than 0.082% failure rate (422 failed jobs)

Job statistics



How much CPU?

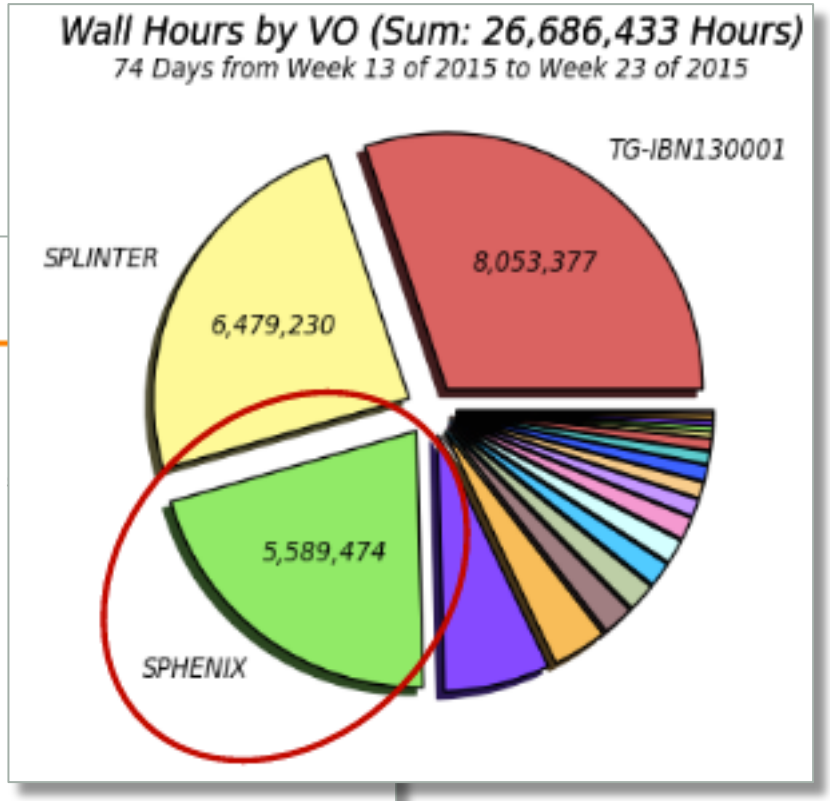
- This is early in the project.
- Didn't keep all log files so I cannot run the final tally

```
$ find log/ -name ``*.log'' -exec grep 'Total Remote Usage' {} \; | \
   sed -e 's/,//g' | awk '{print $3}' | \
   awk -F: '{X += ($1 *3600 + $2*60 + $3)/3600} END {print X}'

1.34523e06

$ bc -l
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
1345230 / 24
56051.25000000000000000000
1345230 / 24/365
153.56506849315068493150
```

1345230 hours
56051 days
153 years

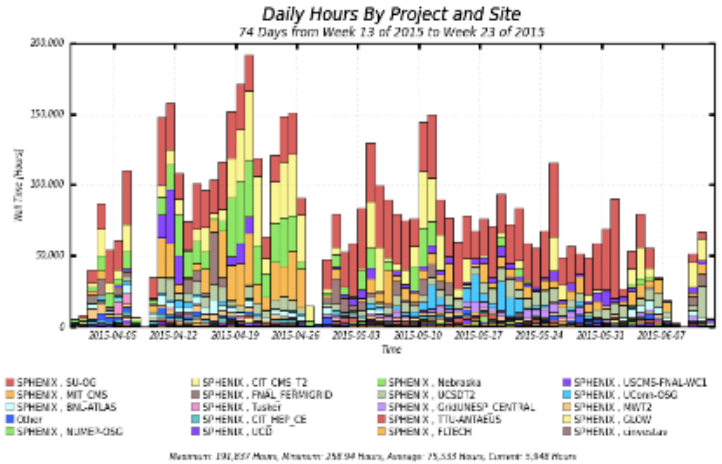
We did not stay under the radar…

- Of course our OSG marathon got noticed…



Open Science Grid

sPHENIX



- Studies for upgraded PHENIX detector at BNL (~5 trillion collisions)
- ~10 week computing campaign (M. Purschke)
  - 5.6M hours
  - Able to ramp up rapidly - #3 project on OSG in that time

https://indico.fnal.gov/getFile.py/access?contribId=9&resId=0&materialId=slides&confId=10055

And we got some press…

Open Science Grid

Home About News Contact

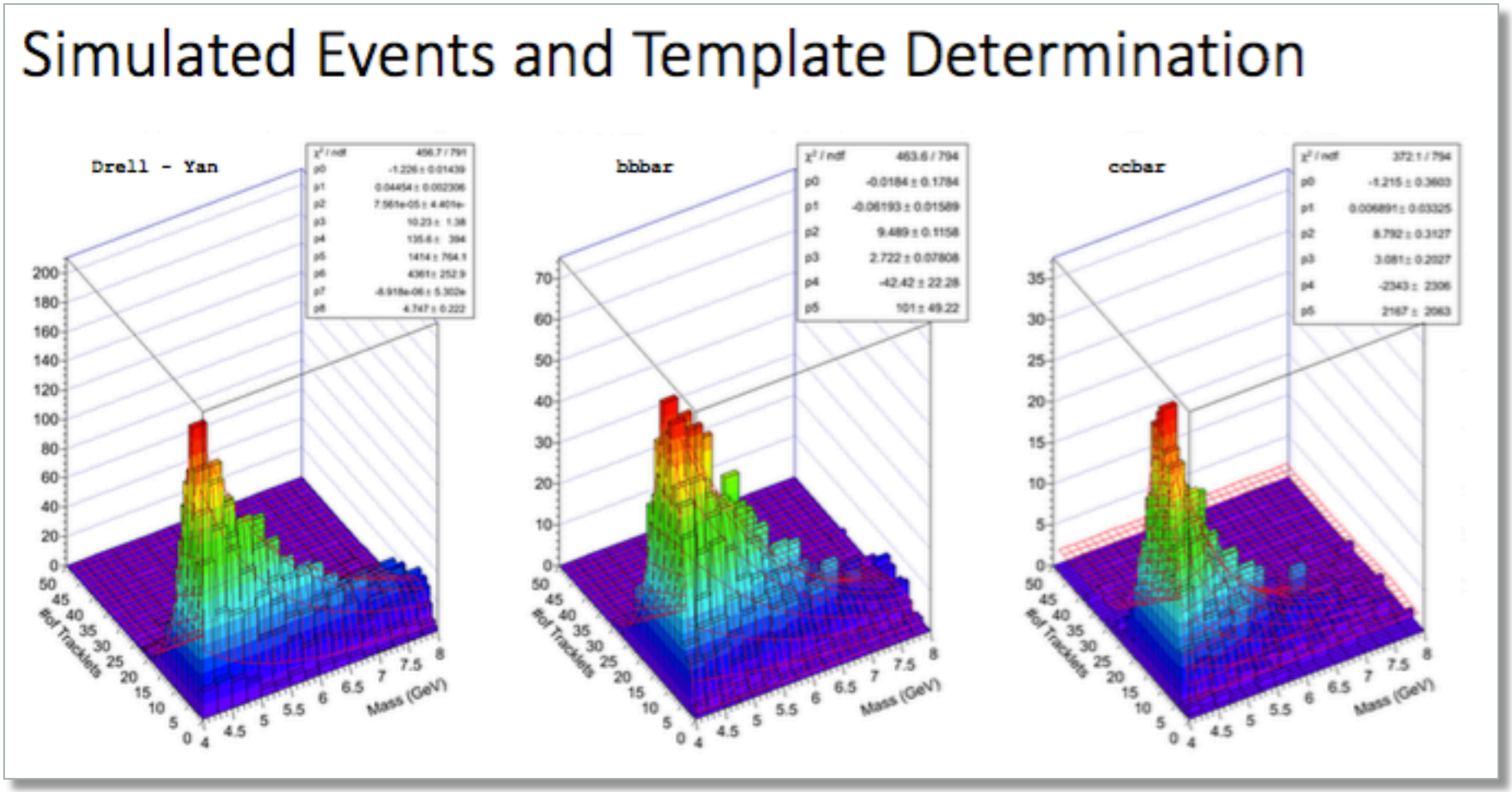
Using Open Science Grid to prepare for 'the next big thing' at Brookhaven

In November 2014, the PHENIX (**P**ioneering **H**igh **E**nergy **N**uclear **I**nteraction e**X**periment) collaboration, that conducts research in heavy ion collisions, updated its 2010 proposal to upgrade the PHENIX detector at the Relativistic Heavy Ion Collider (RHIC) at Brookhaven National Laboratory (BNL). The detector recorded its first relativistic heavy ion collision in 2000, and now records many different particles emerging from collisions at RHIC. As in every year, PHENIX will stop running around the end of June due to power consumption and maintenance needs. After that, the collaboration will gear up for one final run in 2016, starting preparations after Thanksgiving and running through about June 2016.

" … carefully choosing which jobs to run, Purschke says he is using a lot of OSG processing power—about 5 million hours in April and May alone, and 5.4 million hours since the start of the year. "

http://www.opensciencegrid.org/using-open-science-grid-to-prepare-for-the-next-big-thing-at-brookhaven/

Flashing some work-in-progress plots



These plots are directly generated from the OSG simulations

Meanwhile…

- There will be a large push for more simulations
- Current problem is a large memory footprint for Geant4 simulations – few suitable nodes
- We got set up with a cvmfs area
- Now have a PHENIX database slave server in the BNL Science DMZ
- More people are getting certificates
- Trying to get this out of expert-only mode
- We expect the continuous use of the OSG to grow as suitable simulation projects get pushed out to the grid

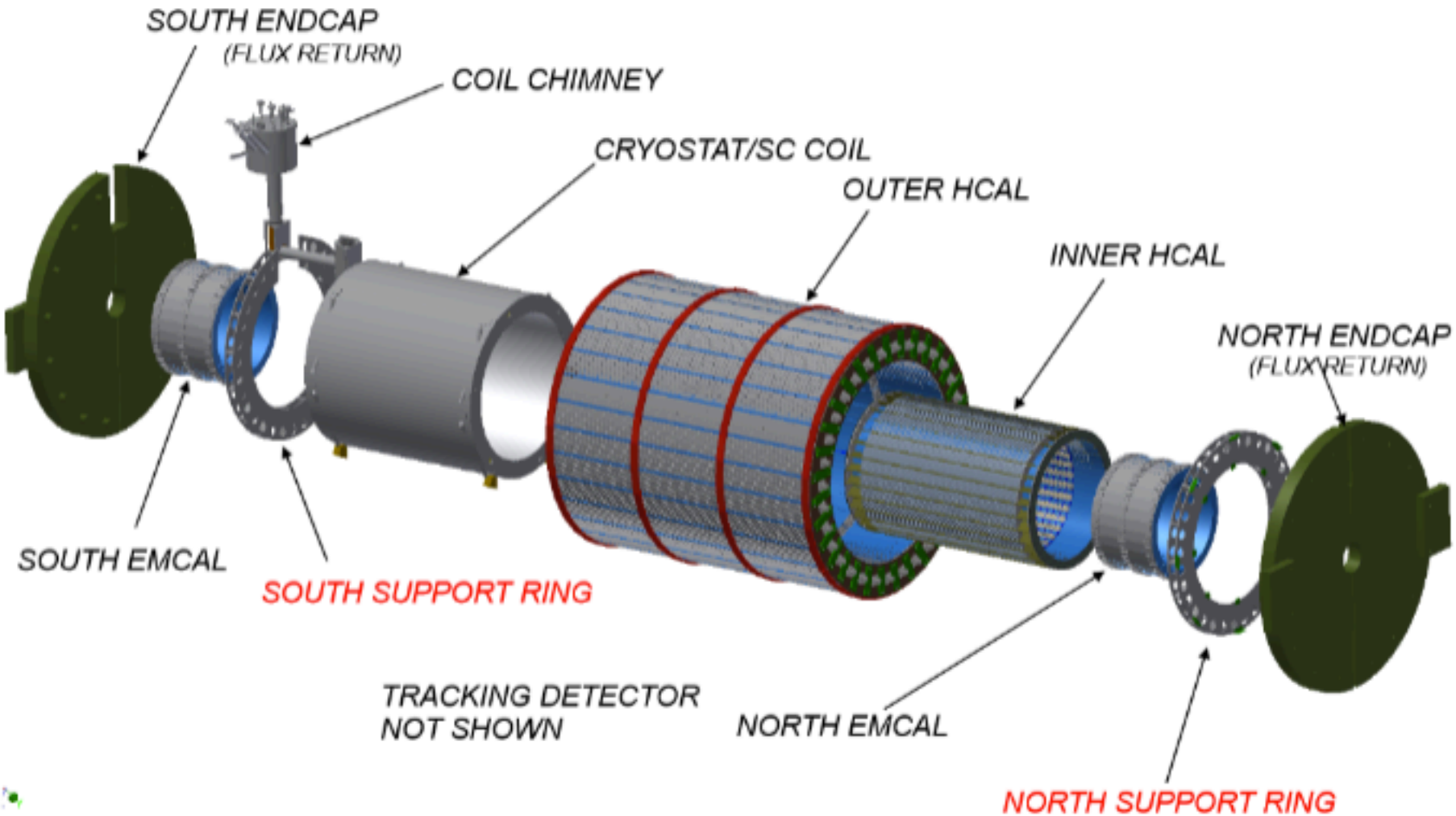
Summary

- Extremely successful simulation project run on the OSG
- Mega-project run almost fully on autopilot
- 5.5 million hours, ~22TB data output
- Very short (~10 days) setup time
- “good citizen” approach – tune job running time, stay well below quotas, clean up…
- Quirky node behavior workarounds necessary
- No guaranteed 32bit compatibility… is there a reason?
- Ran a project that most people deemed impossible
- OSG is a fantastic resource for us

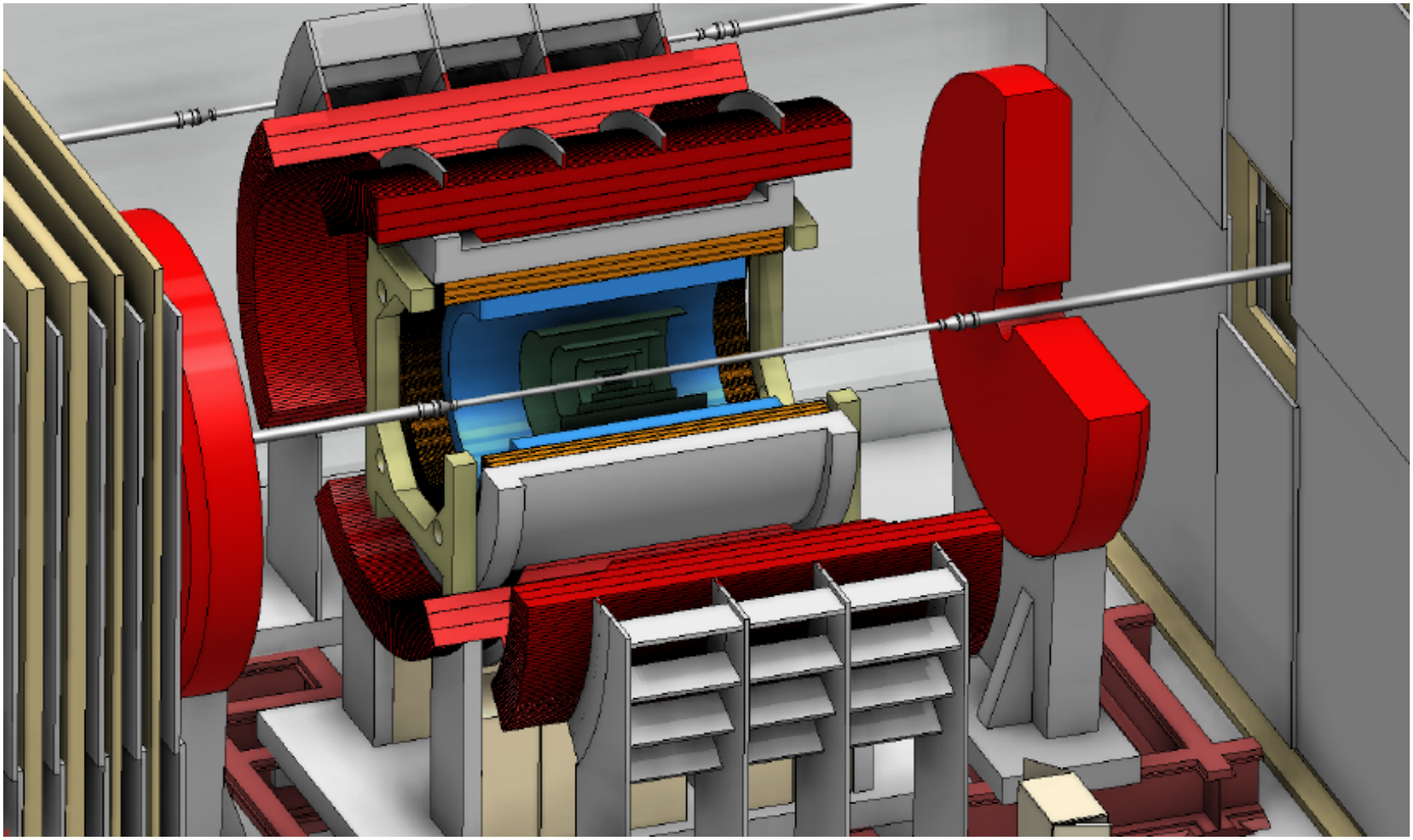
Thank you, OSGers!

This page is intentionally left blank

Putting it all together



sPHENIX Artist's View



Condor job

```
InitialDir = /local-scratch/purschke/pythia/output

Executable     =  run.sh
Universe = vanilla

when_to_transfer_output = ON_EXIT
notification     = Never

should_transfer_files = YES

transfer_input_files = ../files.tar.bz2

output = x.out
error = x.err
Log = x.log

requirements = (( OpSysAndVer == "rhel6" ) ||   ( OpSysAndVer == "SL6" ) ||
( OpSysAndVer == "CentOS6" )  )

+ProjectName = "sPHENIX"
queue
```

run.sh

```
#! /bin/sh
export JOBNR=$1
[ -z "$JOBNR" ] && JOBNR=0
NAMEEXT=$(printf "%08d\n" $JOBNR)

mkdir run_area
cd run_area
time tar xfj ../files.tar.bz2
rm -f ../files.tar.bz2

export LHAPATH=$_CONDOR_SCRATCH_DIR/run_area/PDFsets
export LD_LIBRARY_PATH=$_CONDOR_SCRATCH_DIR/run_area/install/lib:
$_CONDOR_SCRATCH_DIR/run_area/syslibs:$_CONDOR_SCRATCH_DIR/run_area/root/lib
export ROOTSYS=$_CONDOR_SCRATCH_DIR/run_area/root
PATH=$ROOTSYS/bin:$PATH

ldd $_CONDOR_SCRATCH_DIR/run_area/syslibs/libfun4all.so > ldd.txt 2>&1
if  grep -q "not found" ldd.txt; then
    export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$_CONDOR_SCRATCH_DIR/run_area/oslibs
fi

root -b -q phpythia.C\(10000000,\"pythia_MB.cfg\",1\)
if [ -e phpythia.root ] ; then
    mv phpythia.root ../phpythia_single_${NAMEEXT}.root
    mv phpy_xsec.root ../phpy_xsec_single_${NAMEEXT}.root
fi
cd ..
rm -rf run_area
```