



The background image is a 3D schematic of the sPHENIX detector. It shows a central collision point with various particle tracks and vertices. Labels include 'Secondary Vertex' in red, 'Primary Vertex' in blue, 'b-jet' in blue, 'b-quark' in red, 'Distance of Closest Approach (DCA)' in blue, and 'B-hadron or photon' in red. There are also labels for 'h' (hadrons) and 'L<sub>sp</sub>' (longitudinal separation). The detector structure is shown in a semi-transparent grey, revealing internal components like the barrel and endcap calorimeters.

# JupyterLab for sPHENIX

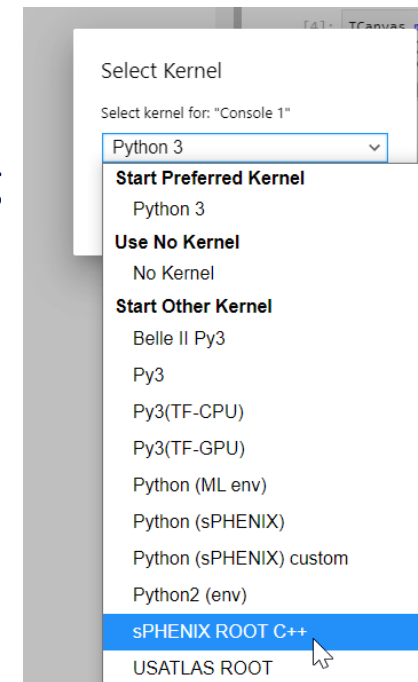
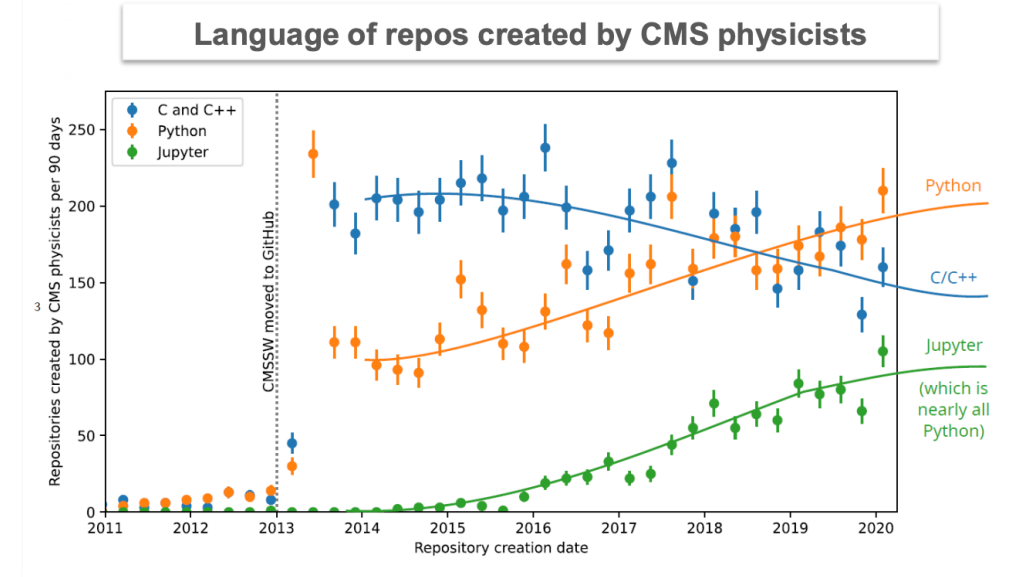
Jin Huang

Brookhaven National Lab

$B$ -hadron  
or photon

# JupyterLab for sPHENIX

- ▶ JupyterLab has been gain popularity for final stage analysis interface
- ▶ sPHENIX software are distributed with JupyROOT and Jupyter installed, available for local use of JupyterLab
- ▶ Thanks to your support, sPHENIX C++ and Python kernel also install in SDCC HTC JupyterHub, available for using SDCC resource via web interface
- ▶ Tutorials available [next page]
- ▶ However, given the heavy development stage of ROOT+JupyterLab, we decided to not to actively pushing collaborator to use JupyterLab (e.g. crash with macro bug is hard to trace)
- ▶ Meanwhile, a new use case in daily operation come up in Jenkins QA report [next section]
- ▶ And user participation may ramp up significantly as sPHENIX collect data and going to analysis stage



SDCC HTC JupyterHub  
Kernel menu


# Tutorial

- ▶ SDCC JupyterHub has been available for sPHENIX collaborator with Tutorials:
- ▶ <https://github.com/sPHENIX-Collaboration/tutorials/tree/master/JupyterLab>
- ▶ <https://indico.bnl.gov/event/7254/>

252 lines (252 sloc) 138 KB

Introduction

This is an example [sPHENIX ROOT Jupyter Notebook](#) to run the sPHENIX analysis on the [BNL SDCC Jupyter Lab](#). Please use launch the HTC Jupyter hub for this tutorial.



Jupyter Notebook provide an analysis interface that can integrate documentation, code and output plot in a single file. It supports all sPHENIX environment. It is suitable for final analysis step, while large scale analysis and reconstruction should go through the compiled c++ analysis module.


Here is also a pyROOT macro perform similar task if you would like to use python for analysis. In the same folder there is an example notebook to run the analysis in ROOT C++ macros

```
In [1]: # Setup the imports
import ROOT
from ROOT import TCanvas, TH3F, TH1F
```

Welcome to JupyROOT 6.16/00

Input

The example input file is 2019 TPC test beam ROOT file from the production with [sPHENIX Fun4All framework](#). More on the full test beam documentation is at [this wiki page](#) and analysis overview is in the [Offline analysis overview slides on TPC general meeting July 3, 2019](#).



Here we just load one output file that is put on a public web server. That is from run 300, when we put 120 GeV proton beam through the center of the TPC prototype that measures the beam position in 16 planes. The ROOT file contains the reconstructed tracks and clusters.



# Example use case: 1<sup>st</sup> CF4 TPC dE/dx test beam data

- ▶ Notebooks : [https://nbviewer.jupyter.org/github/sPHENIX-Collaboration/analysis\\_tpc\\_prototype/blob/master/fnal\\_2019/dEdx/main.ipynb](https://nbviewer.jupyter.org/github/sPHENIX-Collaboration/analysis_tpc_prototype/blob/master/fnal_2019/dEdx/main.ipynb)
- ▶ Presented in sPHENIX TPC meeting and EIC eRD6 meetings

## Introduction

dEdx analysis for sPHENIX TPC 2019 data from [FNAL beam test](#). The prototype contains 16 layer readout pads with 50/50 Ne/CF4 gas. The track segment in each layer is ~1.25cm long. The beam is 120 GeV/c proton beam.

Run with sPHENIX ROOT C++ [Jupyter Lab env at RCF](#)

## Gas comparison

Here is back-envelope comparison on the dE/dx resolution. Gas property table are from *Properties of some gas mixtures used in tracking detectors*, Archana Sharma (Darmstadt, GSI), SLAC-J-ICFA-16-3, SLAC-JOURNAL-ICFA-16-3. This table is generated with [this Google Sheet](#).

TPC	Pad rows	Gas	Radial Drift Vol. [cm]	dE/dx [keV/cm]	Primary Ionization [cm]	Total Ionization [cm]	Total Ionization/Initial Ionization	Integrated Primary Ionization	dE/dx resolution $\sigma$ [cm]	F
STAR w/ iTPC	72	P10 - 10% methane, 90% argon	150	2.344	23.2	89.9	3.9	3,480	6.5%	RHIC S&T rev Caines; iTPC
ALICE 2010	160	(Ne/CO2 90/10)/N2 5% (N2 not in calculation)	161.8	1.705	14.35	47.8	3.3	2,322	5% (cosmic)	doi:10.1016/j.nima.20
sPHENIX2019 w/ EIC R1	48	Ne/CF4 50/50	60	4.28	31.5	71.5	2.3	1,890	This study	sPHE arXiv:1402.1201

By using 50% CF4, sPHENIX 2019 gas produce very high primary ionization per cm of traverse, roughly two times that of ALICE and about 50% higher than STAR. With the compact active volume, this translate to ~1900 primary ionization that is 20% lower than ALICE and roughly half of STAR. Meanwhile, the total ionization to primary ionization ratio for sPHENIX 2019 gas is very low. Lower ratio leads to less high side tail that improves the efficiency of each measurement towards the dE/dx resolution.

## This study

This study extracts the dE/dx resolution from the sPHENIX 2019 beam test at FNAL by projecting the 16 layer prototype device to fully fledged 48-layer configuration envisioned for EIC.

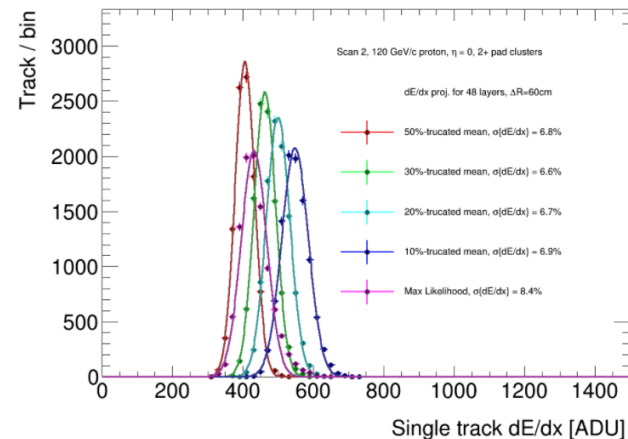
Note the energy deposition from 120 GeV/proton is higher than MIP due to radiative rise that leads to slightly better dE/dx resolution due to ionization statistics.

## Input

```
1): const TString infile = "scan2/tpc_beam_ALL-0000.evt_TpcPrototypeGenFitTrkFitter.root";
// const TString description = "Position scan 2, #eta = 0, 3+ pad clusters";
// const Ticut cut = "TPCTrack.nCluster>14 && Sum$(ClusterY_Rotated<-3) == 0 && TPCTrack.cLusterSizePhi :
const TString description = "Scan 2, 120 GeV/c proton, #eta = 0, 2+ pad clusters";
const Ticut cut = "TPCTrack.nCluster>14 && Sum$(ClusterY_Rotated<-3) == 0 && Sum$(ClusterY_Rotated<-1.5)
```

```
leg->AddEntry(hClusterEnergyTMean1, Form("10%-truncated mean, #sigma(dE/dx) = %.1f%%", fitfunc->GetPar:
//TH1 * dEdxStatML(const TH1 * hPDF, TString name, const int N_sample, const int N_layer, const TF1 *
hClusterEnergyML->Draw("same");
fitfunc = (TF1 *)hClusterEnergyML->GetListOfFunctions()->At(0);
fitfunc->Draw("same");
leg->AddEntry(hClusterEnergyML, Form("Max Likelihood, #sigma(dE/dx) = %.1f%%", fitfunc->GetParameter(2,
leg->Draw();
c1->Draw();
SaveCanvas(c1,
TString(_file0->GetName()) + TString(c1->GetName()), kFALSE);
```

Info in <TCanvas::Print>: png file scan2/tpc\_beam\_ALL-0000.evt\_TpcPrototypeGenFitTrkFitter.rootTruckat

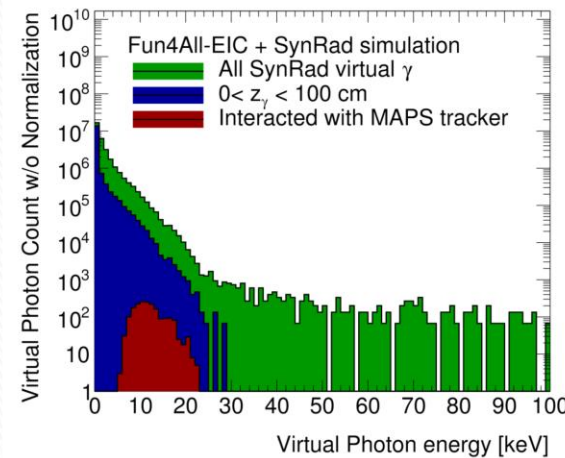


# Example use case: first full detector sim for EIC Synchrotron

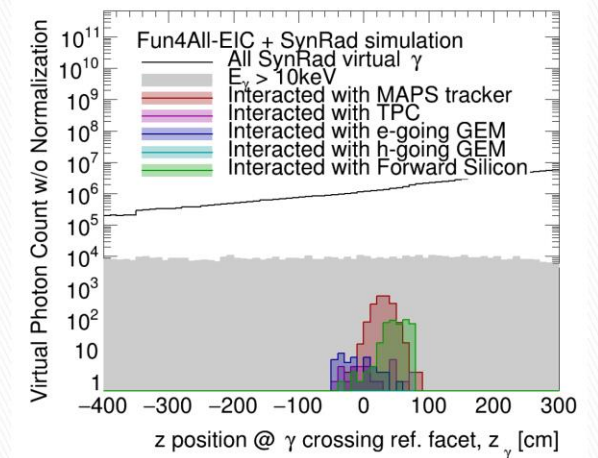
- ▶ Notebooks not disclosed outside EIC project
- ▶ Mix of Pandas, PyROOT and ROOT-clang code
- ▶ Presented in EIC YR workshops and detector workshops, e.g. <https://indico.jlab.org/event/400/contributions/6540/>

## Synchrotron background: detector response

- Iterating with accelerator design to avoid 10keV photon that exits -50 to +100cm from beam pipe



Energy dependence of MAPS vertex tracker to synchrotron



Beam-pipe exit-location

Note: all photons simulated for detector interaction, without cuts on z or energy. July-2020 lattice/chamber

Jin Huang <jhuang@bnl.gov>

SVTX @ EIC Workshop

13

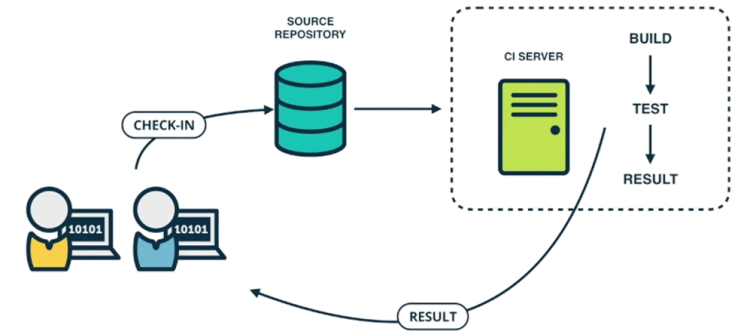
# New Case for Jupyter Notebook in daily operation



QA checks in sPHENIX Jenkins Continuous Integration (CI)

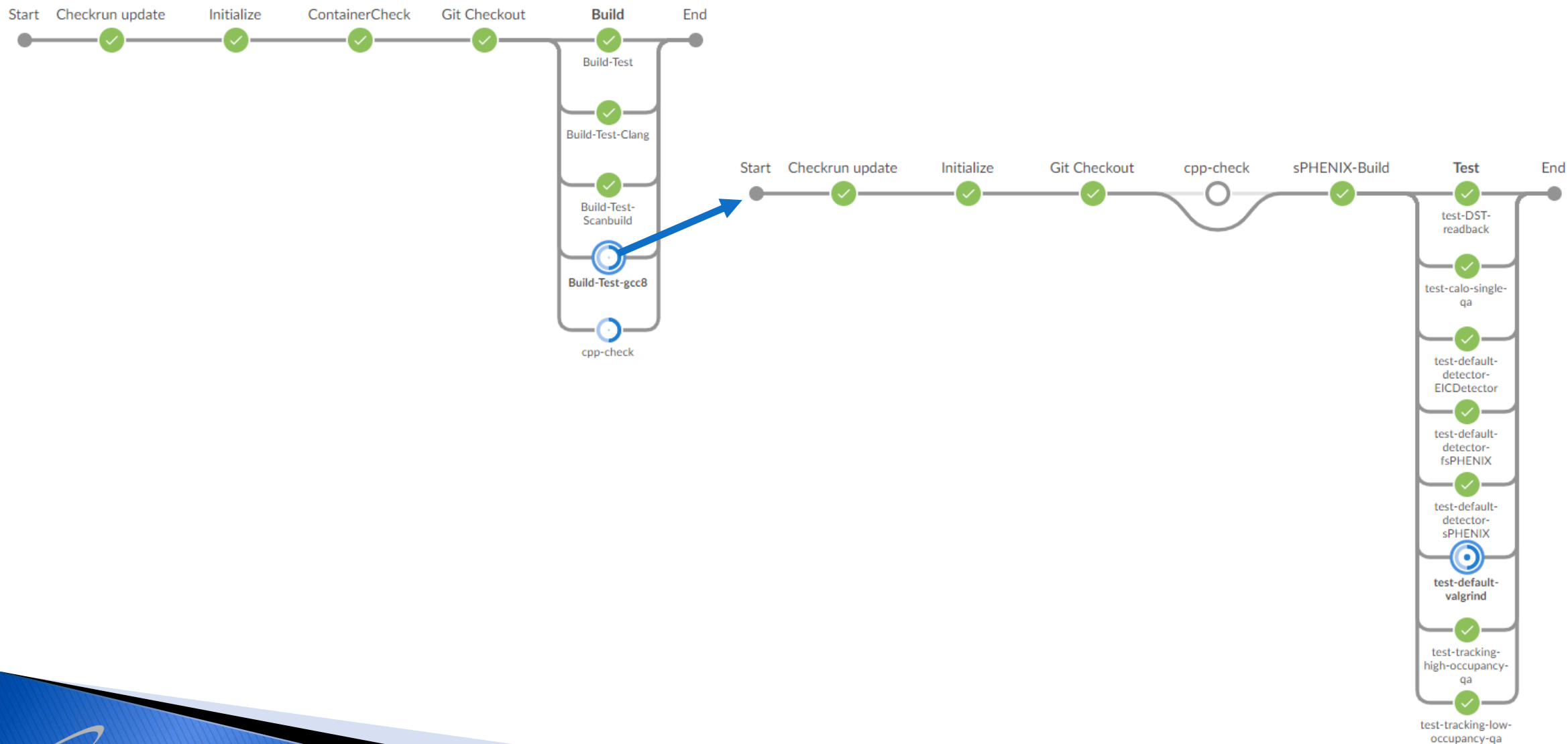
# What is Jenkins Continuous Integration (CI)

- ▶ Jenkins CI: watch over new pull request for code quality and software reproducibility
- ▶ Why Jenkins:
  - Supported by SDCC, and dedicated nodes with O(100) CPU O(200) GB memory : Thank you!
  - Widely used tool and avoid vendor lock to GitHub/GitLab
  - Highly configurable to run complex custom builds
- ▶ Two main development campaigns
  - 2018 x-mas holiday: initial introduction [[62th GM](#)]
  - 2020 x-mas holiday: upgrade infrastructure [this update]





# Jenkins Pipeline Workflow

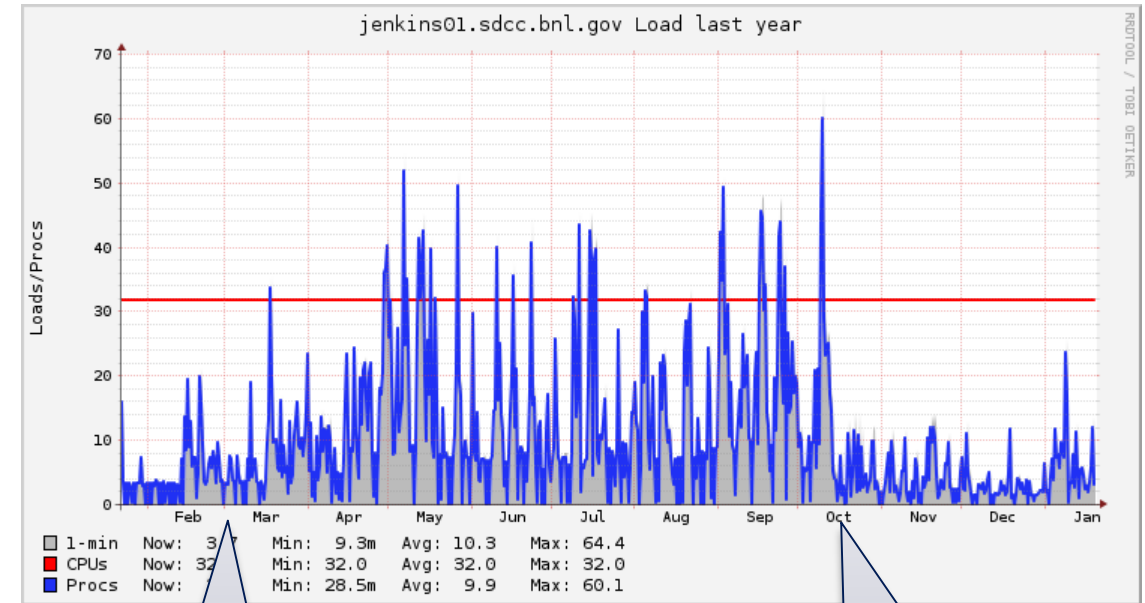




# Jenkins operation in the past two years

- ▶ Checked every Pull Request (PR) since 2019 and daily checks
  - O(100) CPU-hour test for each PR
  - Built sPHENIX code O(4000) times
  - O(2000) simulation QA tests
  - O(2000) cpp-checks
- ▶ Automatically builds the Doxygen code reference for sPHENIX (and EIC)
- ▶ Checks validity of sPHENIX containers for offsite computing

Main Jenkins server loads over last 12 month

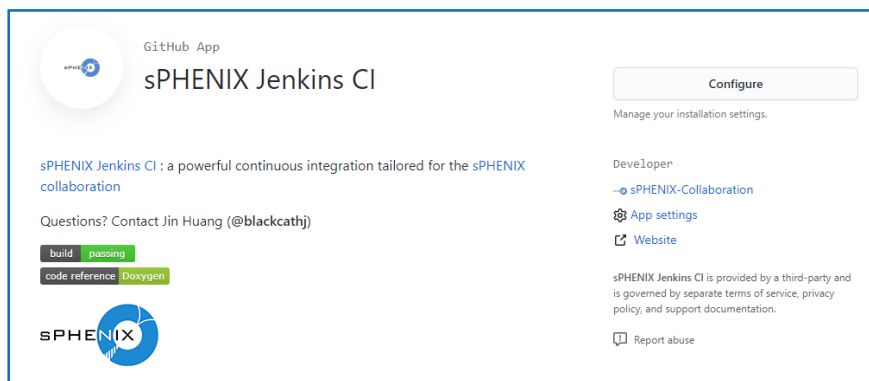


Introduction of tracking QA suites

Switch to ACTS and rebuild of macro setup broke QA chain

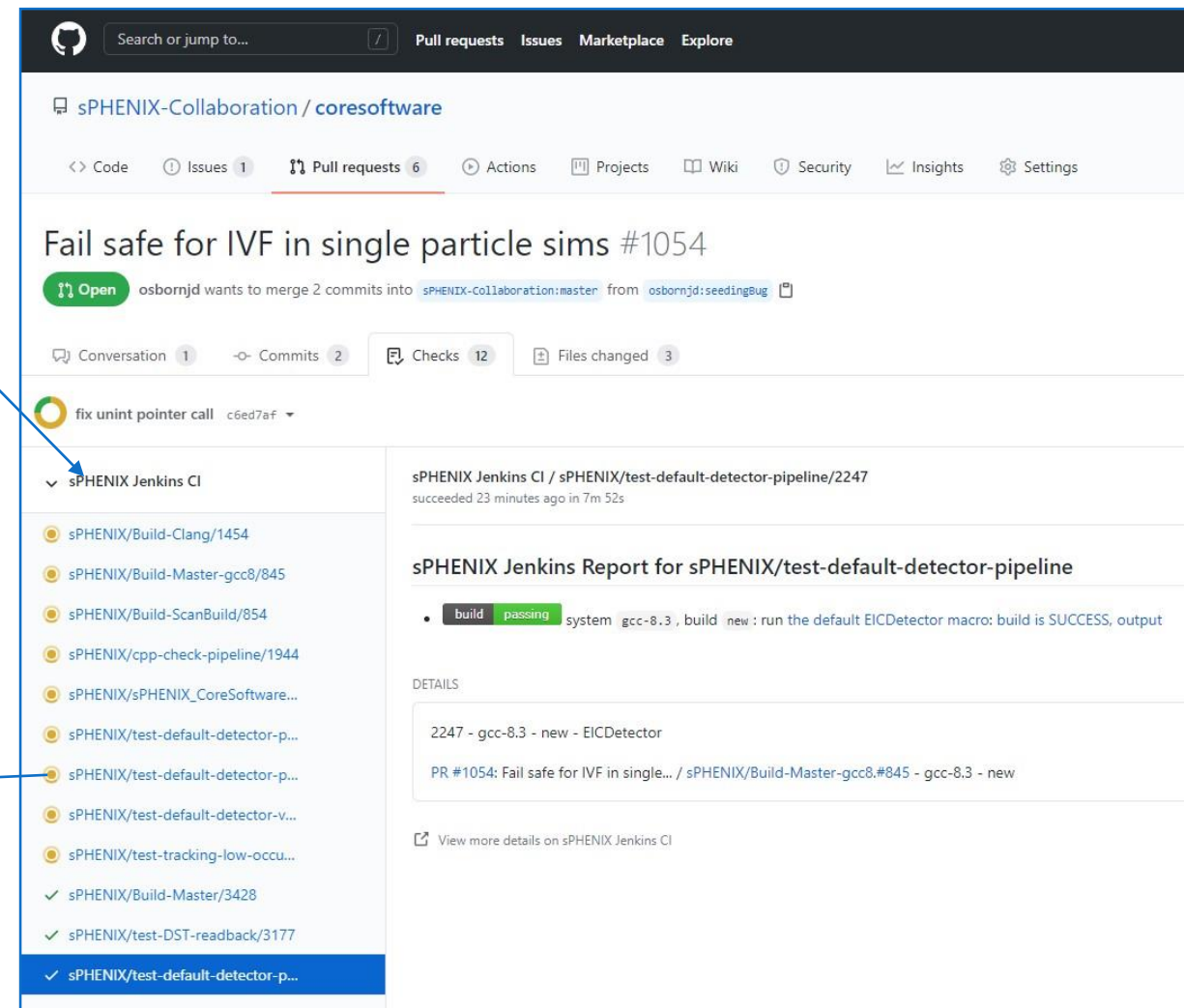
# Most recent updates

## Our own GitHub App



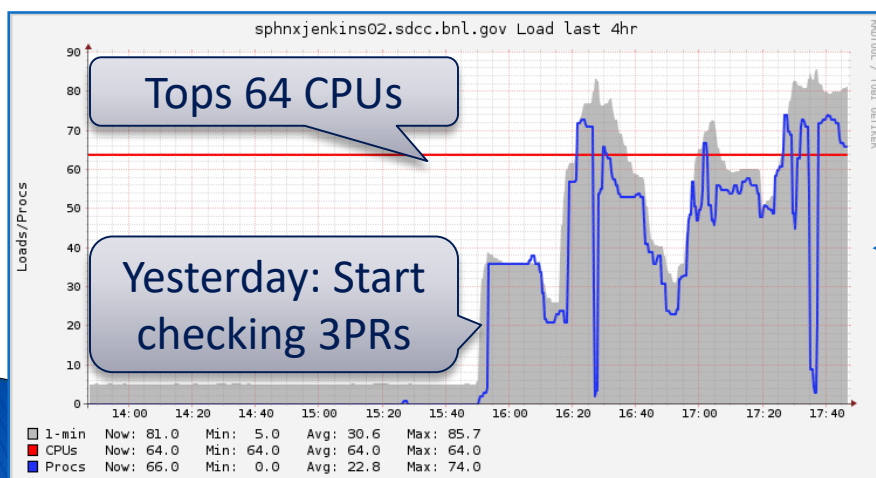
The screenshot shows the GitHub App configuration for 'sPHENIX Jenkins CI'. It includes a 'Configure' button, a description of the app as a powerful continuous integration tool for sPHENIX collaboration, and contact information for Jin Huang (@blackcathj). A status bar shows 'build passing' and 'code reference Doxygen'. The sPHENIX logo is at the bottom left.

## GitHub Checks API



The screenshot shows the GitHub Checks API page for the repository 'sPHENIX-Collaboration / coressoftware'. It displays a pull request titled 'Fail safe for IVF in single particle sims #1054' by osbornjd. The page shows 12 checks, with the 'sPHENIX Jenkins CI' check highlighted. The Jenkins CI check is successful, with a report for 'sPHENIX/test-default-detector-pipeline' showing a 'build passing' status. The report details the build process and the output of the EICDetector macro.

Brought online a 64-CPU dedicated computing node:



## Updated all checks supporting new macro and ACTS

sphenix-jenkins-ci bot commented 22 hours ago

### Build & test report

Report for commit 5b0992ee436470acc84ff7102d0a82ba42c4646d:



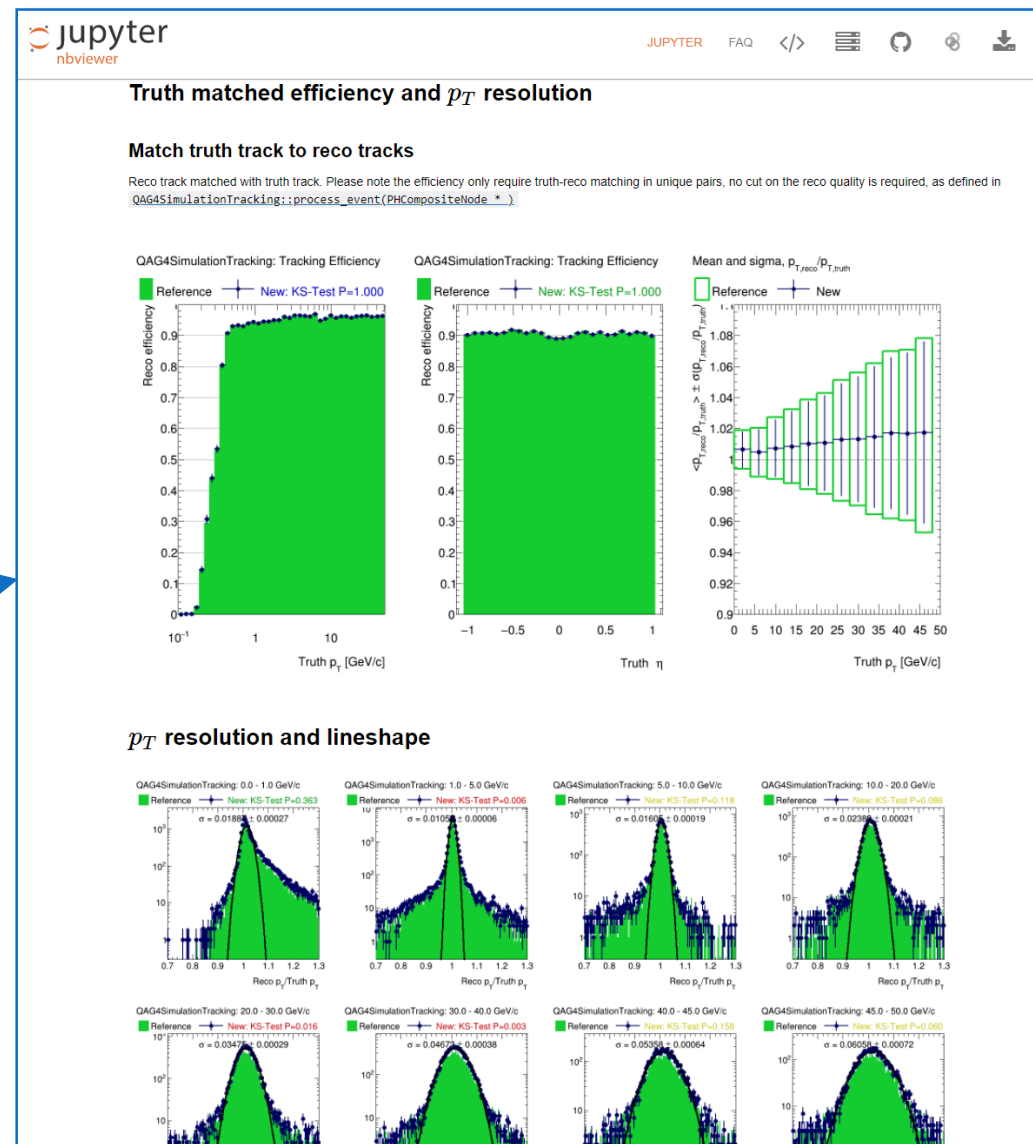
- build passing builds and tests overall are SUCCESS.
- build passing Build with configuration of gcc-8.3 / new is SUCCESS, Compiler report (full)/(new), build log
  - build passing Generating DST and readback: build is SUCCESS
  - build passing Calorimeter QA: build is SUCCESS
    - QA-calorimeter for e- at p<sub>T</sub>=4GeV : combined Chi2/nDoF = -0 / 72, and combined p-Value = 1
    - QA-calorimeter for pi+ at p<sub>T</sub>=30GeV : combined Chi2/nDoF = -0 / 72, and combined p-Value = 1
    - QA-calorimetric-jet for e- at p<sub>T</sub>=4GeV : combined Chi2/nDoF = -0 / 42, and combined p-Value = 1
    - QA-calorimetric-jet for pi+ at p<sub>T</sub>=30GeV : combined Chi2/nDoF = -0 / 42, and combined p-Value = 1
  - build passing Tracking QA at high occupancy: build is SUCCESS
    - QA-Intt : combined Chi2/nDoF = -0 / 72, and combined p-Value = 1
    - QA-Mvtx : combined Chi2/nDoF = -0 / 54, and combined p-Value = 1
    - QA-Tpc : combined Chi2/nDoF = -0 / 56, and combined p-Value = 1
    - QA-tracking : combined Chi2/nDoF = 0.0956583 / 26, and combined p-Value = 1
    - QA-vertexing : combined Chi2/nDoF = 330 / 98, and combined p-Value = 6.83194e-27
  - build passing Tracking QA at low occupancy: build is SUCCESS
    - QA-Intt : combined Chi2/nDoF = 0.406392 / 72, and combined p-Value = 1
    - QA-Mvtx : combined Chi2/nDoF = 0.0664111 / 54, and combined p-Value = 1
    - QA-Tpc : combined Chi2/nDoF = 0.40023 / 56, and combined p-Value = 1
    - QA-tracking : combined Chi2/nDoF = 0.297808 / 42, and combined p-Value = 1
    - QA-vertexing : combined Chi2/nDoF = 150.141 / 98, and combined p-Value = 0.000556263
- build passing system gcc-8.3, build new: run the default EICDetector macro: build is SUCCESS, output
- build passing system gcc-8.3, build new: run the default sPHENIX macro: build is SUCCESS, output
- build passing system gcc-8.3, build new: run the default sPHENIX macro: build is SUCCESS, output
- build unstable system gcc-8.3, build new: Valgrind test: build is UNSTABLE, valgrind report
- build passing Build with configuration of x8664\_s17 / clang is SUCCESS, clang report (full)/(new), build log
- build passing Build with configuration of x8664\_s17 / new is SUCCESS, Compiler report (full)/(new), build log
- build passing Build with configuration of x8664\_s17 / scan is SUCCESS, scan-build report (full)/(new), build log
- build passing cpp-check is SUCCESS, cppcheck report (full)/(new)

Automatically generated by sPHENIX Jenkins continuous integration




Jenkins

## New representation for QA reports based on Jupyter notebooks [new GitHub repo]





# Expect more QAs

- ▶ More QA checks and plots
- ▶ Weekly checks of longer-run jobs, e.g. test beam-base validation, entire jet sample
- ▶ Readback of production data, reproducibility of analysis

## DCA QA - Pull request 259

<https://github.com/sPHENIX-Collaboration/macros/pull/259>

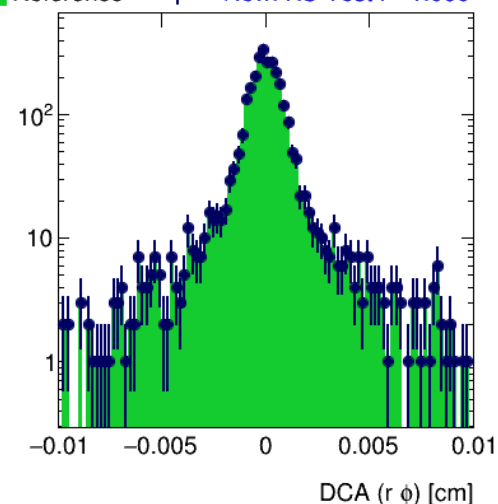
Thomas Marshal

(When developing:  
incoming grad student @ UCLA)



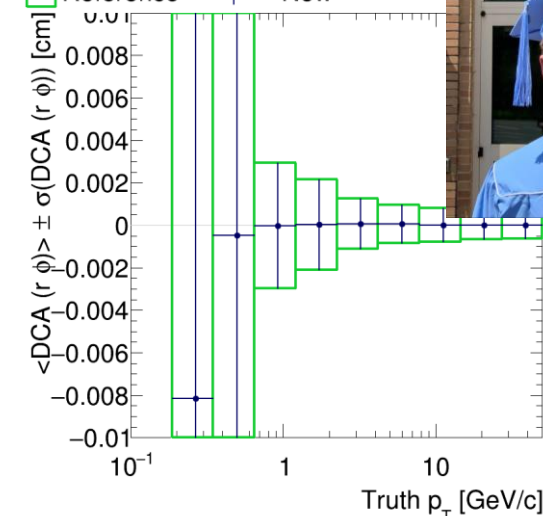
QAG4SimulationTracking: 8.0 - 16.0 GeV/c

Reference New: KS-Test P=1.000



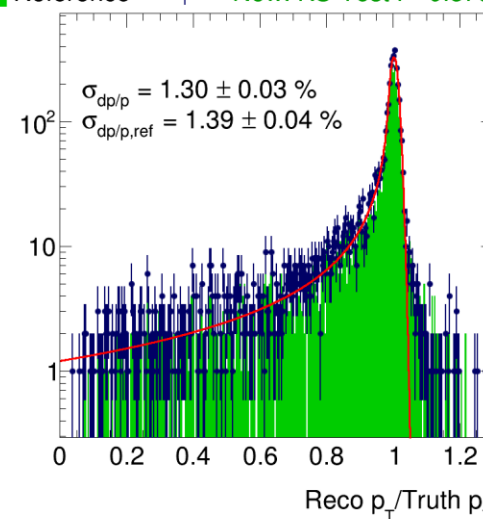
Mean and sigma, DCA (r phi) [cm]

Reference New



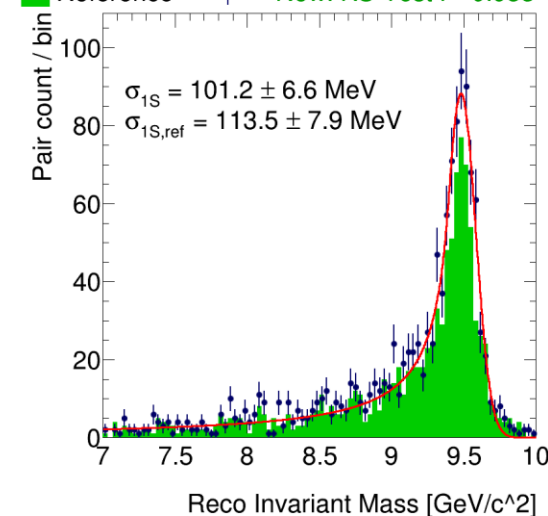
QAG4SimulationUpsilon: Electron lineshape

Reference New: KS-Test P=0.879



QAG4SimulationUpsilon: Y -> e+e- lineshape

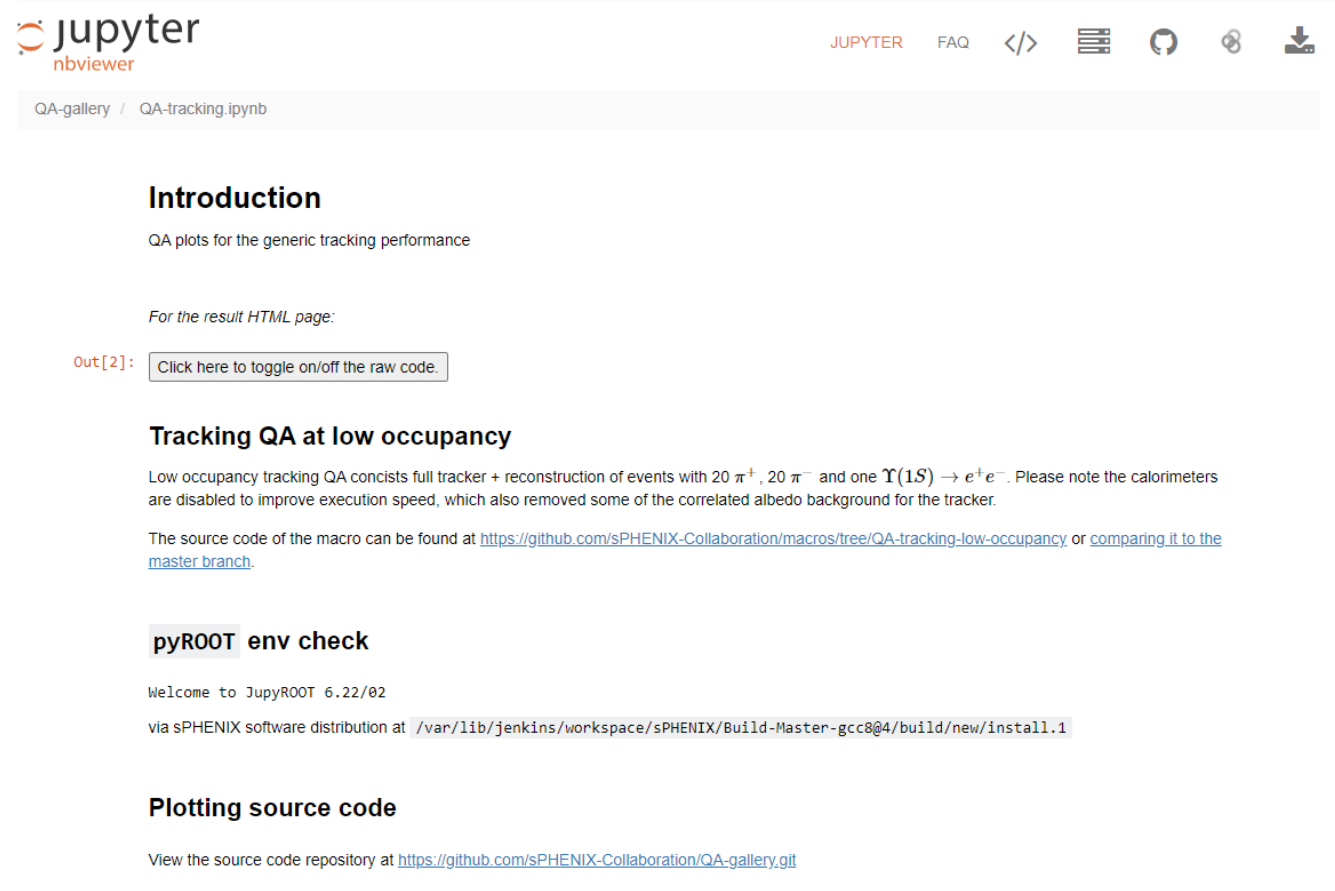
Reference New: KS-Test P=0.933



# Temp. QA plot hosting

- ▶ Currently use free cloud storage and render
  - Storage :  
<https://github.com/sPHENIX-Collaboration/QA-gallery>
    - Limited 1GB free
    - About one year operation fills it up
  - Render:
    - Free service at [nbviewer.jupyter.org](https://nbviewer.jupyter.org)
- ▶ No guarantee these reports would be available for history tracing few years later

**Example:** [https://nbviewer.jupyter.org/github/sPHENIX-Collaboration/QA-gallery/blob/jenkins-sPHENIX-test-tracking-low-occupancy-qa-689-test-tracking\\_Event300\\_Sum10/QA-tracking.ipynb](https://nbviewer.jupyter.org/github/sPHENIX-Collaboration/QA-gallery/blob/jenkins-sPHENIX-test-tracking-low-occupancy-qa-689-test-tracking_Event300_Sum10/QA-tracking.ipynb)



The screenshot displays the Jupyter Notebook interface for the 'QA-gallery' notebook. The header shows the 'jupyter nbviewer' logo and navigation links for 'JUPYTER', 'FAQ', and code editing icons. The breadcrumb path is 'QA-gallery / QA-tracking.ipynb'. The notebook content includes an 'Introduction' section with a description of QA plots, a code output 'Out[2]:' with a button to toggle raw code, a 'Tracking QA at low occupancy' section explaining the data and source code, a 'pyROOT env check' section showing the environment path, and a 'Plotting source code' section with a link to the repository.

**Introduction**

QA plots for the generic tracking performance

For the result HTML page:

Out[2]: [Click here to toggle on/off the raw code.](#)

**Tracking QA at low occupancy**

Low occupancy tracking QA consists full tracker + reconstruction of events with  $20 \pi^+$ ,  $20 \pi^-$  and one  $\Upsilon(1S) \rightarrow e^+e^-$ . Please note the calorimeters are disabled to improve execution speed, which also removed some of the correlated albedo background for the tracker.

The source code of the macro can be found at <https://github.com/sPHENIX-Collaboration/macros/tree/QA-tracking-low-occupancy> or [comparing it to the master branch](#).

**pyROOT env check**

Welcome to JupyROOT 6.22/02

via sPHENIX software distribution at `/var/lib/jenkins/workspace/sPHENIX/Build-Master-gcc8@4/build/new/install.1`

**Plotting source code**

View the source code repository at <https://github.com/sPHENIX-Collaboration/QA-gallery.git>

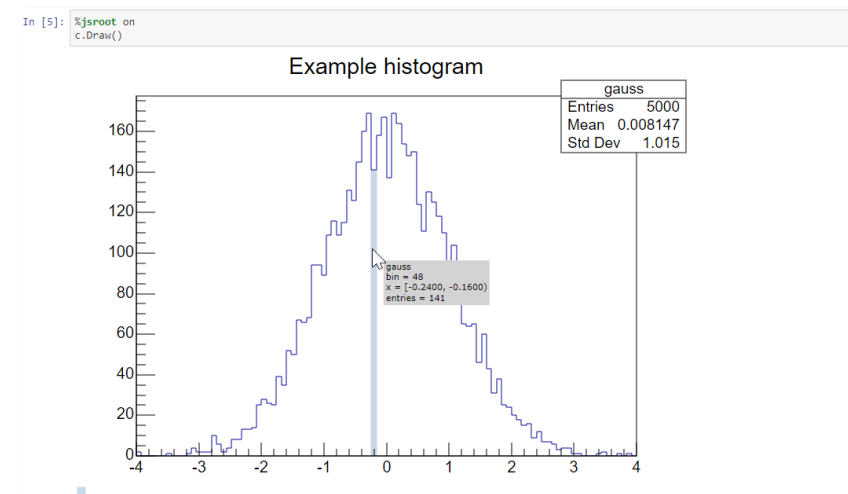
# Request and discussions





# Request and discussions

- ▶ For Jenkins QA report archival:
  - Nbviewer hosting with SDCC GitTea, or other SDCC publication site
    - i.e. moving one of these link to completely SDCC hosted:  
[https://nbviewer.jupyter.org/github/sPHENIX-Collaboration/QA-gallery/blob/jenkins-sPHENIX-test-tracking-low-occupancy-qa-689-test-tracking\\_Event300\\_Sum10/QA-tracking.ipynb](https://nbviewer.jupyter.org/github/sPHENIX-Collaboration/QA-gallery/blob/jenkins-sPHENIX-test-tracking-low-occupancy-qa-689-test-tracking_Event300_Sum10/QA-tracking.ipynb)
- ▶ SDCC notebook binder?
  - Open one above notebook directly on SDCC JupyterHub
  - Rerun by user interactively when needed
- ▶ Plug-ins
  - Example contrib-nbextensions: TOC2
  - Example HEP specific extensions: jsROOT
- ▶ Dynamic kernelling
  - sPHENIX kernel now fixed to use newest build
  - However, we generate a software snapshot every week as well as production builds. Nice the kernel can be configurable dynamically to select those builds
  - Note this is *not* a problem for JupyterLab via sPHENIX Singulartiy container, as user select build first before launching JupyterLab
- ▶ Major version upgrade
  - JupyterLab @ SDCC is two major version behind latest

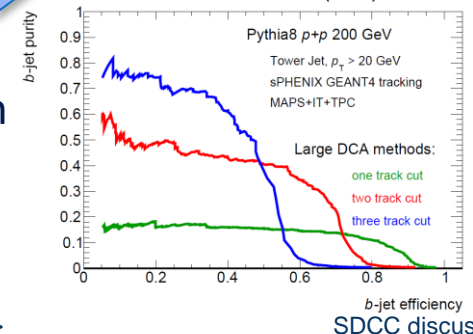
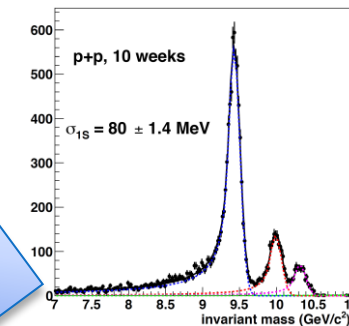
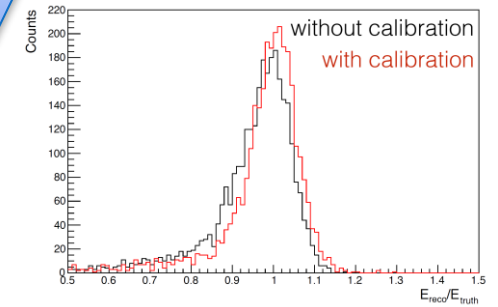
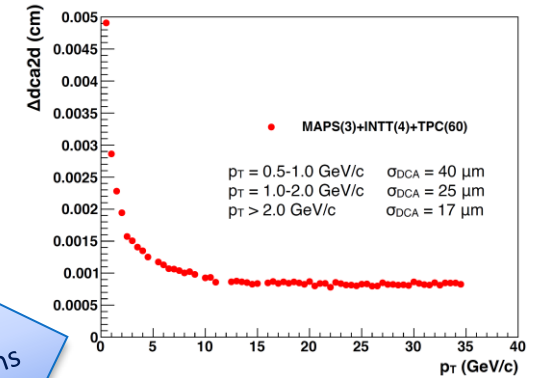
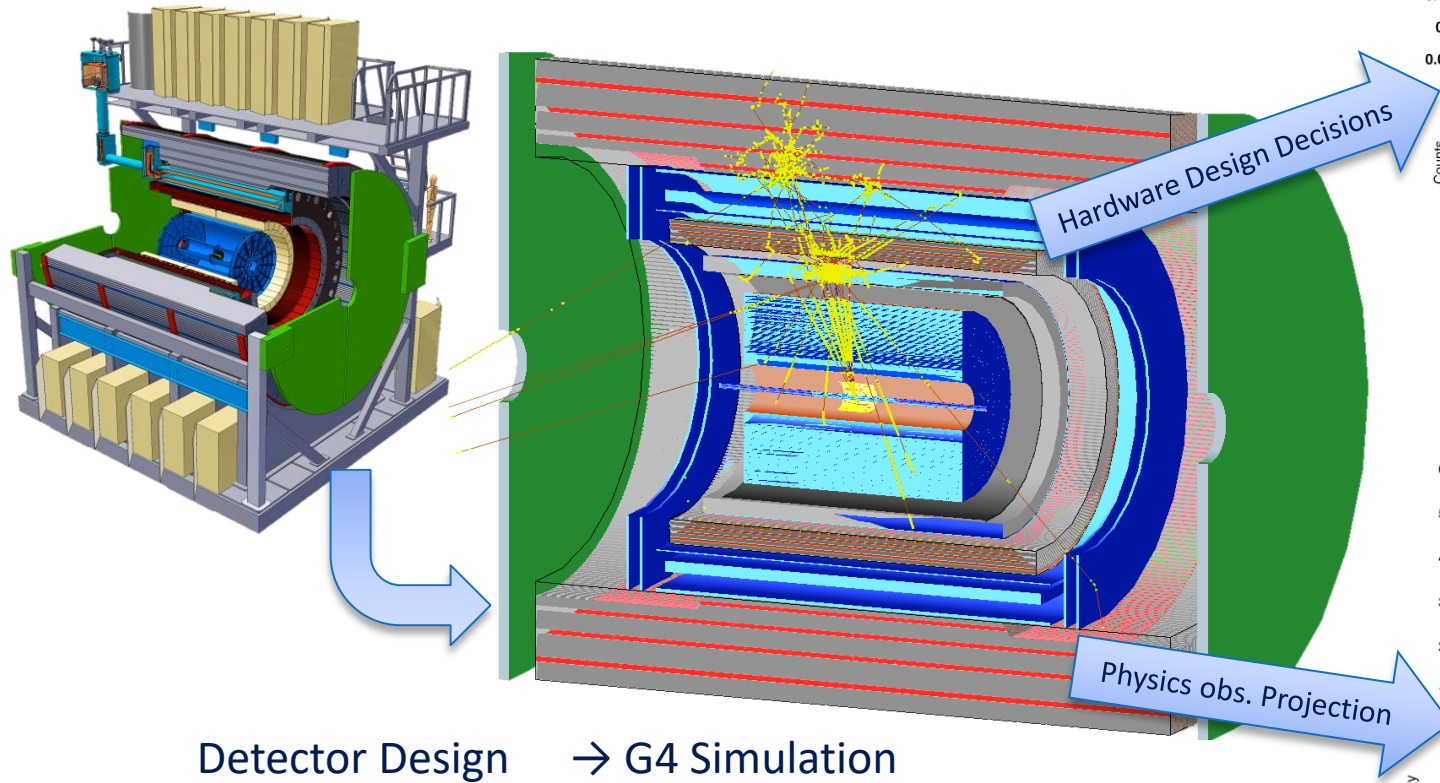


```
SDCC:/u0b/software/jupyter/kernels/sphenix-  
root/kernel.json  
{  
  "language": "c++",  
  "display_name": "sPHENIX ROOT C++",  
  "argv": [  
    "/u0b/software/jupyter/bin/sphenix_setup.sh",  
    "-m",  
    "JupyROOT.kernel.rootkernel",  
    "-f",  
    "{connection_file}"  
  ]  
}
```

# Extra information



# sPHENIX Software Repo.



Open source @ **GitHub**

<https://github.com/sPHENIX-Collaboration/>

Core software: 200k line of code

Analysis: 59k line of code

Jin Huang <jhuang@bnl.gov>

SDCC discussion



# Main repository activities

<https://github.com/sPHENIX-Collaboration/>

sPHENIX-Collaboration / coresoftware

- abinashpun / coresoftware
- adfrawley / coresoftware
- liuk / coresoftware
- belmonrj / coresoftware
- blackcatj / coresoftware
- bogui56 / coresoftware
- carlosperezlara / coresoftware
- Chongk / coresoftware
- damcglincy / coresoftware
- meeg / coresoftware
- dkapu001 / coresoftware
- dmgilman / coresoftware
- dvperpelitsa / coresoftware
- FrancescoVassalli / coresoftware
- gmitsuka / coresoftware
- GuannanXie / coresoftware
- HaiwangYu / coresoftware
- hvannieuwenh / coresoftware
- ibromberg / sphenix\_coresoftware
- jdoso / coresoftware
- jinlong1810 / coresoftware
- JinsTest / coresoftware
- jlabounty / coresoftware
- johnlajoie / coresoftware
- kurthill / coresoftware
- mccumbermike / coresoftware
- mchiu-bnl / coresoftware
- mckinziebrandon / coresoftware
- mpurschke / coresoftware
- msar15 / coresoftware
- nfeege / sPHENIX-coresoftware
- phyxch / coresoftware
- pingwong / coresoftware
- pinkenburg / coresoftware
- pragarg / coresoftware
- raisbeck / coresoftware
- rccorliss / coresoftware
- mitay958 / coresoftware
- duynguyen / coresoftware

December 11, 2018 – January 11, 2019

Period: 1 month

## Overview

10 Active Pull Requests

0 Active Issues

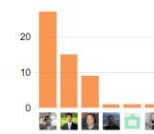
10 Merged Pull Requests

0 Proposed Pull Requests

0 Closed Issues

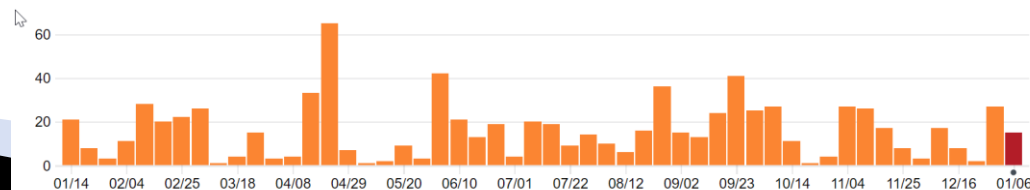
0 New Issues

Excluding merges, 6 authors have pushed 54 commits to master and 54 commits to all branches. On master, 365 files have changed and there have been 27,380 additions and 11,253 deletions.



10 Pull requests merged by 6 people

- Merged #538 Changes for ROOT6 a day ago
- Merged #537 Testing sPHENIX Jenkins continuous integration a day ago
- Merged #535 Tracking Modularization and Reorganization 3 days ago
- Merged #536 Change genfit for root6 8 days ago
- Merged #534 adding const access function for cluster/tower containers. 14 days ago
- Merged #533 Add E864 (cut) module forward EMCAL 20 days ago
- Merged #532 Tpc sims reorganization - clang-format 22 days ago
- Merged #529 Tpc sims reorganization 22 days ago
- Merged #531 prepare for Debug build 25 days ago
- Merged #530 Better treatment of reconstructed jet mass. on Dec 11, 2018



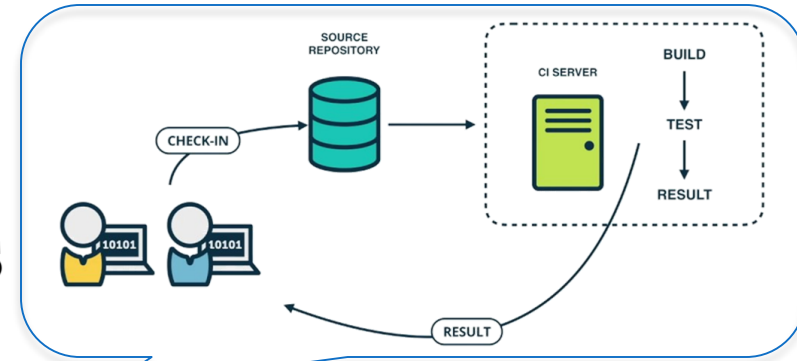
<https://github.com/sPHENIX-Collaboration/coresoftware/pulse/monthly>

# Questions and solutions

You might had these questions:

- ▶ When did **jet energy scale** changed?
- ▶ Did the code **run slower/faster** recently?
- ▶ Would my pull request **break the sPHENIX build**?
- ▶ I build this new module, may I have **some suggestion**?

Our new solution:  **Jenkins**



- ▶ Implementing **continuous integration** for sPHENIX software
- ▶ Automatically run compile, QA, and code analysis for new GitHub pull request. Report accessible with RCF SSO.
- ▶ Regularly test/archive default setups

# Pull request checks



<https://github.com/sPHENIX-Collaboration/coresoftware/pulls>

5 Open 509 Closed Author Labels Projects Milestones Reviews Assignee Sort

Testing sPHENIX Jenkins continuous integration • CI-build-new-PENDING CI-build-root6-PASS  
CI-calo-QA-PENDING CI-cpp-check-PENDING CI-release CI-valgrind-PENDING  
#537 opened 3 days ago by blackcathj

TPC with a rigid central membrane ✓ CI-build-new-PASS CI-build-root6-PASS CI-calo-QA-AVAILABLE  
CI-cpp-check-AVAILABLE CI-release CI-valgrind-AVAILABLE test and suggestion welcomed  
#509 opened on Oct 30, 2018 by blackcathj

Preparing a perimeter tower geometry for clustering preview  
#411 opened on Oct 13, 2018 by blackcathj

Tower Energy Cut and Tower Energy Scale for Jet Finder ✓ test and suggestion welcomed  
#368 opened on Oct 18, 2017 by SongkyoLee

Event display preview  
#194 opened on Oct 18, 2016 by sookhyun

On-going checks

Passed checks

Report available



# Would my pull request break the sPHENIX build?

<https://github.com/sPHENIX-Collaboration/coresoftware/pull/509#issuecomment-452144033>

sPHENIX-bot commented 2 days ago

Member + 😊 ...

### Build & test report

Report for commit 1b4a8ea3f7796174d30b73c37b5f88ccb4ce019e:

- build **passing** builds and tests overall are SUCCESS.
- build **passing** Build with configuration of new is SUCCESS
- build **passing** Build with configuration of root6 is SUCCESS
- build **passing** Calorimeter QA: build is SUCCESS, 📊 QA report
- build **passing** cpp-check on coressoftware: build return code 0, 📊 cpp-check report
- build **passing** run the default sPHENIX macro: build is SUCCESS, 📁 output files
- build **passing** Valgrind test: build is SUCCESS, 📊 valgrind report

Automatically generated by sPHENIX Jenkins continuous integration

sPHENIX  Jenkins

Reports automatically post to pull request

Also sent to email list:

<https://lists.bnl.gov/mailman/listinfo/sphenix-github-l>

← Passed tests



Failed tests →

sPHENIX-bot commented 2 minutes ago


Member + 😊 ...

### Build & test report

Report for commit 2ce16cbb6d11949b44d64f6e0a4727a077165701:

- build **failing** builds and tests overall are FAILURE.
- build **passing** Build with configuration of root6 is SUCCESS
- build **passing** cpp-check on coressoftware: build return code 0, 📊 cpp-check report

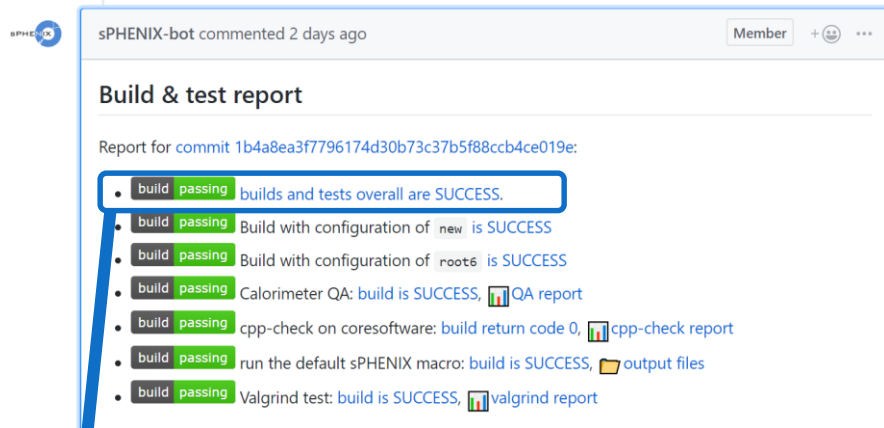
Automatically generated by sPHENIX Jenkins continuous integration

sPHENIX  Jenkins

Jin Huang <jhuang@bnl.gov>

# Behind the scene

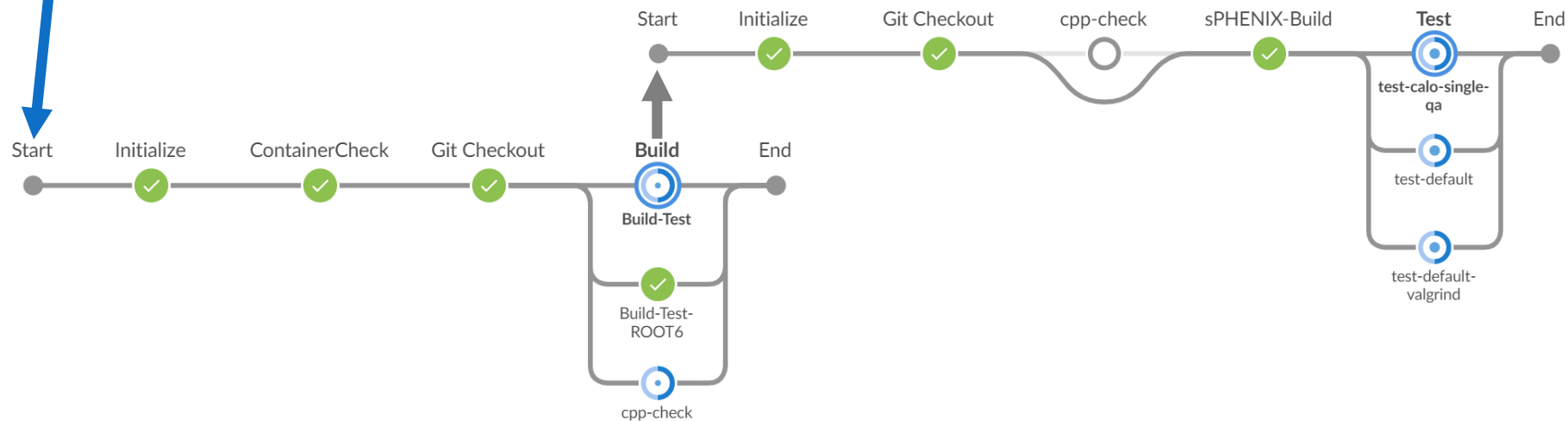
<https://github.com/SPHENIX-Collaboration/coresoftware/pull/509#issuecomment-452144033>



- ▶ Run pipelines of tests
- ▶ On a dedicated 32-core server
- ▶ Reproduce RCF env with sPHENIX Singularity container



<https://github.com/SPHENIX-Collaboration/singularity>



# Did the code run slower recently?



<https://github.com/sPHENIX-Collaboration/coresoftware/pull/509#issuecomment-452144033>

sPHENIX-bot commented 2 days ago

Member

### Build & test report

Report for commit 1b4a8ea3f7796174d30b73c37b5f88ccb4ce019e:

- build **passing** builds and tests overall are SUCCESS.
- build **passing** Build with configuration of new is SUCCESS
- build **passing** Build with configuration of roots is SUCCESS
- build **passing** Calorimeter QA: build is SUCCESS, QA report
- build **passing** cpp-check on coresoftware: build return code 0, cpp-check
- build **passing** run the default sPHENIX macro build is SUCCESS

## Stage View

<https://web.racf.bnl.gov/jenkins-sphenix/job/sPHENIX/job/test-calo-single-qa/>

			Declarative: Checkout SCM	Prebuild-Cleanup	Initialize	Git Checkout	Copy reference	Test-e-	Test-pi+	html-report	Declarative: Post Actions
Average stage times: (Average full run time: ~47min 59s)			356ms	2s	3s	4s	522ms	20min 28s	22min 9s	15s	322ms
#18	Jan 09 10:36	No Changes	365ms	2s	2s	3s	524ms				
			42min 7s								
#17	Jan 09 03:50	No Changes	322ms	1s	2s	3s	603ms	22min 57s	24min 20s	13s	323ms
#16	Jan 08 14:22	50 commits	418ms	2s	3s	4s	520ms	22min 34s	24min 43s	17s	333ms
#15	Jan 08 03:50	No Changes	349ms	1s	2s	3s	523ms	22min 42s	25min 12s	18s	337ms
#14	Jan 07 19:46	No Changes	342ms	2s	3s	6s	564ms	23min 9s	24min 51s	18s	334ms
#13											

# When did jet energy scale changed?



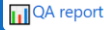


<https://github.com/sPHENIX-Collaboration/coresoftware/pull/509#issuecomment-452144033>

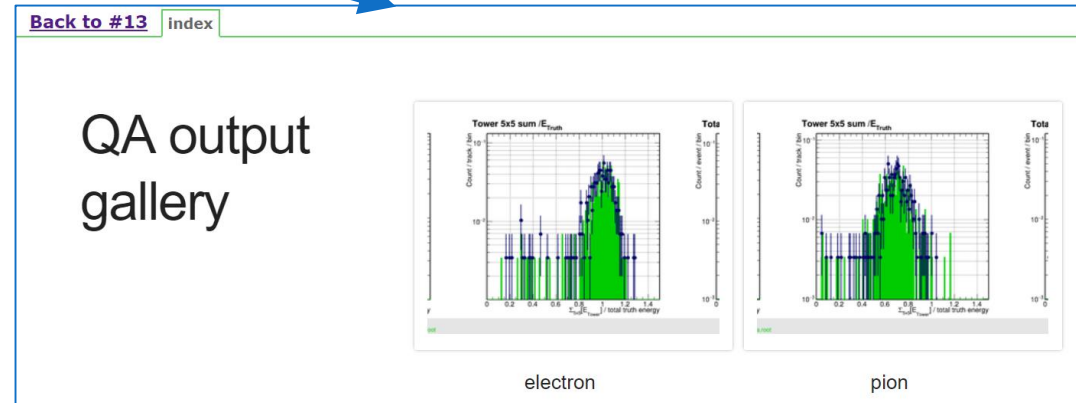
sPHENIX-bot commented 2 days ago

Member

### Build & test report

Report for commit 1b4a8ea3f7796174d30b73c37b5f88ccb4ce019e:

- build **passing** builds and tests overall are SUCCESS.
- build **passing** Build with configuration of new is SUCCESS
- build **passing** Build with configuration of roots is SUCCESS
- build **passing** Calorimeter QA: build is SUCCESS  QA report
- build **passing** cpp-check on coresoftware: build return code 0,  cpp-check report
- build **passing** run the default sPHENIX macro build is SUCCESS  output file



- <https://github.com/sPHENIX-Collaboration/coresoftware/pull/104>
- <https://github.com/sPHENIX-Collaboration/coresoftware/pull/105>
- <https://github.com/sPHENIX-Collaboration/coresoftware/pull/107>

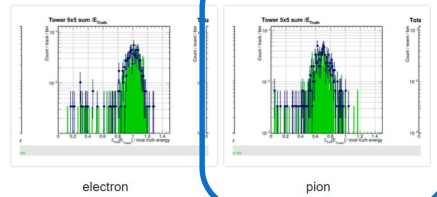


# When did jet energy scale changed?



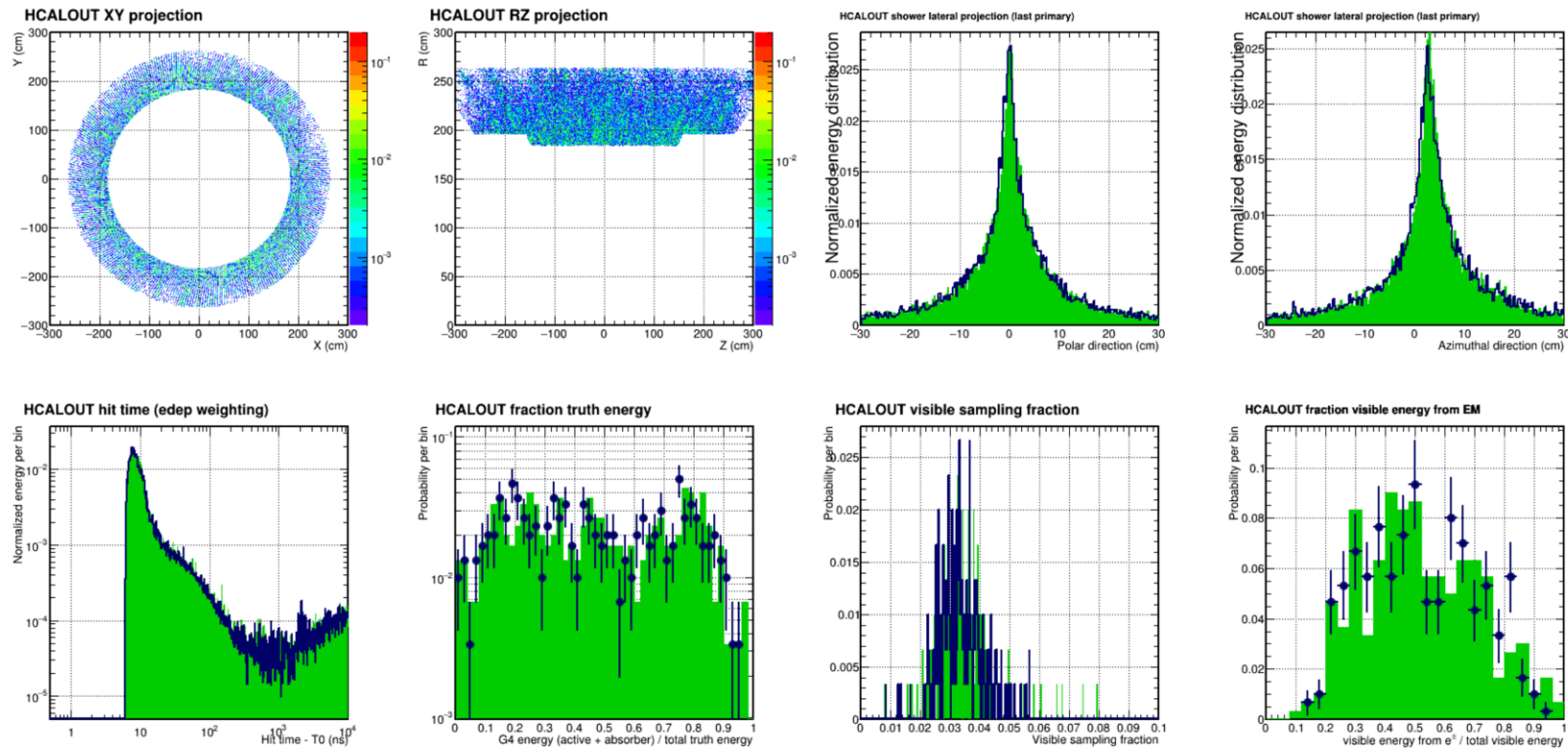
[Back to #13](#) [Index](#)

QA output  
gallery



[https://web.racf.bnl.gov/jenkins-sphenix/job/SPHENIX/job/test-calo-single-qa/13//QA\\_20Report/](https://web.racf.bnl.gov/jenkins-sphenix/job/SPHENIX/job/test-calo-single-qa/13//QA_20Report/)

[Back to #13](#) [Index](#)

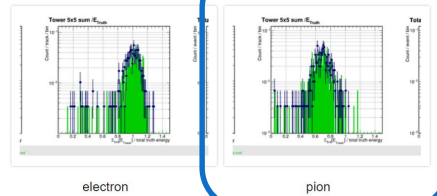


# When did jet energy scale changed?



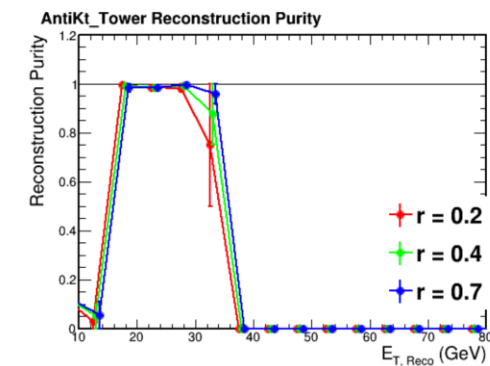
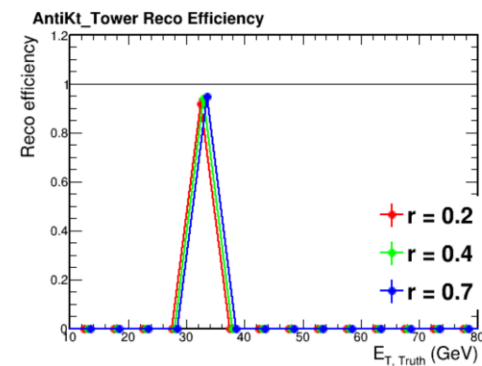
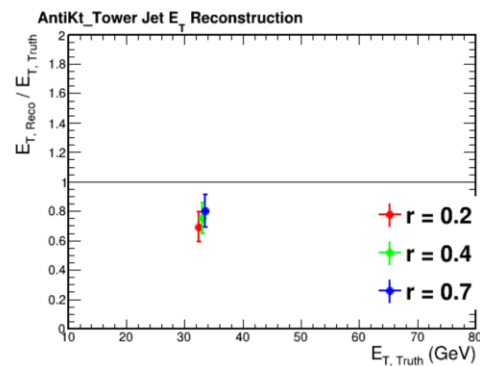
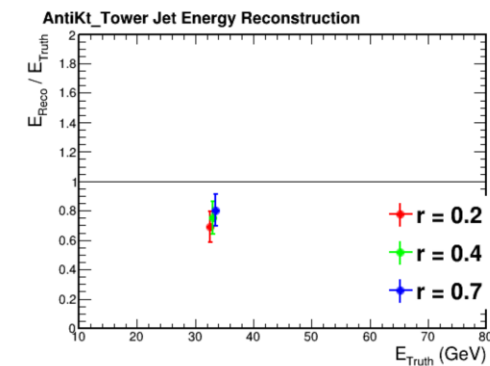
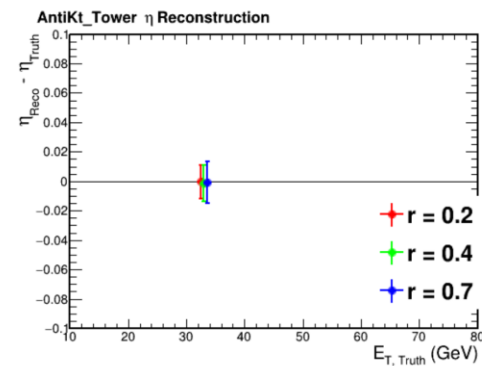
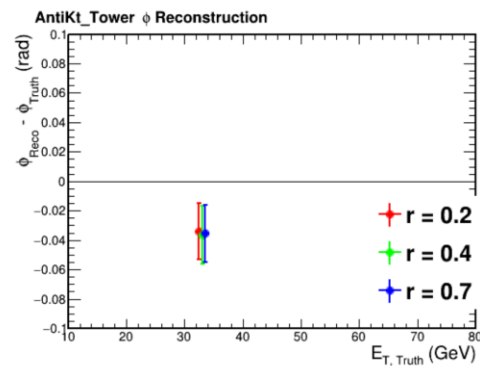
[Back to #13](#) [Index](#)

QA output  
gallery



[https://web.racf.bnl.gov/jenkins-sphenix/job/SPHENIX/job/test-calo-single-qa/13//QA\\_20Report/](https://web.racf.bnl.gov/jenkins-sphenix/job/SPHENIX/job/test-calo-single-qa/13//QA_20Report/)

[Back to #13](#) [Index](#)



# May I have some suggestion on my code?






<https://github.com/sPHENIX-Collaboration/coresoftware/pull/538#issuecomment-452992916>

sPHENIX-bot commented 2 days ago



Member

### Build & test report

Report for commit 1b4a8ea3f7796174d30b73c37b5f88ccb4ce019e:

- build passing builds and tests overall are SUCCESS.
- build passing Build with configuration of new is SUCCESS
- build passing Build with configuration of roots is SUCCESS
- build passing Calorimeter QA: build is SUCCESS, QA report
- build passing cpp-check on coressoftware: build return code 0,  [cpp-check report](#)
- build passing run the default sPHENIX macro: build is SUCCESS,  output files
- build passing Valgrind test: build is SUCCESS,  valgrind report

Automatically generated by sPHENIX Jenkins continuous integration

  Jenkins

## Cppcheck Results

### Summary

Severity	Count	Delta
Error	31	
Warning	271	+10
Style	0	
Performance	132	
Portability	0	
Information	0	
No category	0	
Total	434	+10

### Cppcheck Engine

[1.83]

### Details

Show issues highlighted on a single page

- [all](#)
- [new and solved](#)
- [new](#)
- [solved](#)
- [unchanged](#)

State	File	Line	Severity
new	<a href="#">coresoftware/offline/packages/trackreco/PHRaveVertexing.C</a>	<a href="#">89</a>	warning
new	<a href="#">coresoftware/offline/packages/trackreco/PHGenFitTrkProp.C</a>	<a href="#">975</a>	warning
new	<a href="#">coresoftware/offline/packages/trackreco/PHGenFitTrkProp.C</a>	<a href="#">121</a>	warning
new	<a href="#">coresoftware/offline/packages/trackreco/PHHoughSeeding.C</a>	<a href="#">114</a>	warning
new	<a href="#">coresoftware/offline/packages/trackreco/PHHoughAllInOne.C</a>	<a href="#">3156</a>	warning
new	<a href="#">coresoftware/offline/packages/trackreco/PHGenFitTrkProp.C</a>	<a href="#">1252</a>	warning

```
153 }
154
155 void PHPy6ParticleTrigger::AddParticles(std::string particles) {
156     std::vector<int> addedParts = convertToInts(particle
157     _theParticles.insert(_theParticles.end(),addedParts.
158 }
159
160 void PHPy6ParticleTrigger::AddParticles(int particle)
161     _theParticles.push_back(particle);
162 }
```

**passedByValue:** Function parameter 'particles' should be passed by reference.

Parameter 'particles' is passed by value. It could be passed as a (const) reference which is usually faster and recommended in C++.

# More checks ...



<https://github.com/sPHENIX-Collaboration/coresoftware/pull/538#issuecomment-452992916>



sPHENIX-bot commented 2 days ago

### Build & test report

Report for commit 1b4a8ea3f7796174d30b73c37b5f88ccb4ce019e:

- build **passing** builds and tests overall are SUCCESS.
- build **passing** Build with configuration of `new` is SUCCESS
- build **passing** Build with configuration of `root6` is SUCCESS
- build **passing** Calorimeter QA: build is SUCCESS, QA report
- build **passing** cpp-check on coressoftware: build return code 0, cpp-check report
- build **passing** run the default sPHENIX macro: build is SUCCESS, output files
- build **passing** Valgrind test: build is SUCCESS, valgrind report

Automatically generated by sPHENIX Jenkins continuous integration

  **Jenkins**

## ✓ Build #14 (Jan 7, 2019 6:16:03 PM)

PR #509: TPC with a rigid central me... / sPHENIX/Build-Master.#29



### Build Artifacts

 G4sPHENIX.root_g4cemc_eval.root	228.86 KB	 <a href="#">view</a>
 G4sPHENIX.root_g4hcalin_eval.root	72.10 KB	 <a href="#">view</a>
 G4sPHENIX.root_g4hcalout_eval.root	73.37 KB	 <a href="#">view</a>
 G4sPHENIX.root_g4jet_eval.root	54.55 KB	 <a href="#">view</a>
 G4sPHENIX.root_g4svtx_eval.root	1.08 MB	 <a href="#">view</a>
 test-default.md	453 B	 <a href="#">view</a>

## Valgrind Process Details

[back to processes overview](#)

### root.exe (process 160260)

Valgrind arguments	-v --num-callers=30 --leak-check=full --error-limit=no --log-file=Fun4All_G4_sPHENIX.valgrind --xml=yes --xml-file=Fun4All_G4_sPHENIX.valgrind.xml --leak-resolution=high
Target arguments	-b -q Fun4All_G4_sPHENIX.C(2)
Parent process	160258
Child processes	

Uninitialized condition errors	767
Memory leak errors (definitely lost)	1972 (2284603 bytes)
Memory leak errors (possibly lost)	151 (86555728 bytes)
total	2890

**Uninitialized Condition:** An initialised-condition use error is reported when a condition depends on uninitialised values. ([Valgrind Manual](#))

ID	Error
0x0	<a href="#">Conditional jump or move depends on uninitialised value(s)</a>
0x1	<a href="#">Conditional jump or move depends on uninitialised value(s)</a>
0x2	<a href="#">Conditional jump or move depends on uninitialised value(s)</a>
0x3	<a href="#">Conditional jump or move depends on uninitialised value(s)</a>



# Jenkins dashboard



<https://web.racf.bnl.gov/jenkins-sphenix/view/sPHENIX%20Pipelines/>

Build history										
Job	Stages								Commits	Build Finished Duration
sPHENIX » test-default-valgrind-pipeline	Declarative: Checkout SCM	Prebuild-Cleanup	Initialize	Git Checkout	Test				18	2019-01-09 10:36:15 0ms
sPHENIX » test-default-pipeline	Declarative: Checkout SCM	Prebuild-Cleanup	Initialize	Git Checkout	Test				19	2019-01-09 10:36:15 0ms
sPHENIX » test-calo-single-qa	Declarative: Checkout SCM	Prebuild-Cleanup	Initialize	Git Checkout	Copy reference	Test-e-	Test-pi+		18	2019-01-09 10:36:15 0ms
sPHENIX » Build-Master 40 - root6 - origin/pr/537/merge	Declarative: Checkout SCM	Initialize	Git Checkout	cpp-check	sPHENIX-Build	Test	test-default	test-default-valgrind	noreply Update README.md	40 2019-01-09 10:37:40 18m 9s
sPHENIX » Build-Master 39 - new - origin/pr/537/merge	Declarative: Checkout SCM	Initialize	Git Checkout	cpp-check	sPHENIX-Build	Test	test-calo-single-qa	test-default-valgrind	noreply Update README.md	39 2019-01-09 10:19:30 0ms
sPHENIX » sPHENIX_CoreSoftware_PullRequest 11 - origin/pr/537/merge	Declarative: Checkout SCM	Initialize	ContainerCheck	Git Checkout	Build	Build-Test	cpp-check	Build-Test-ROOT6		11 2019-01-09 10:19:10 0ms
sPHENIX » test-default-valgrind-pipeline	Declarative: Checkout SCM	Prebuild-Cleanup	Initialize	Git Checkout	Test	report	Declarative: Post Actions		17	2019-01-09 05:57:42 2h 6m 52s
sPHENIX » test-default-pipeline	Declarative: Checkout SCM	Prebuild-Cleanup	Initialize	Git Checkout	Test	report	Declarative: Post Actions		18	2019-01-09 04:23:11 32m 21s
sPHENIX » test-calo-single-qa	Declarative: Checkout SCM	Prebuild-Cleanup	Initialize	Git Checkout	Copy reference	Test-e-	Test-pi+	html-report	17	2019-01-09 04:38:33 47m 43s
sPHENIX » Build-Master 38 - root6 - origin/master	Declarative: Checkout SCM	Initialize	Git Checkout	cpp-check	sPHENIX-Build	Test	test-default	test-default-valgrind	38	2019-01-09 03:56:13 20m 54s
sPHENIX » Build-Master 37 - new - origin/master	Declarative: Checkout SCM	Initialize	Git Checkout	cpp-check	sPHENIX-Build	Test	test-default	test-default-valgrind	37	2019-01-09 05:57:46 2h 22m 26s
sPHENIX » sPHENIX_CoreSoftware_PullRequest 10 - origin/master	Declarative: Checkout SCM	Initialize	ContainerCheck	Git Checkout	Build	cpp-check	Build-Test	Build-Test-ROOT6	10	2019-01-09 05:57:48 2h 22m 47s
sPHENIX » sPHENIX_CoreSoftware_PullRequest 9 - origin/pr/535/merge	Declarative: Checkout SCM	Initialize	ContainerCheck	Git Checkout	Build	cpp-check	Build-Test	Build-Test-ROOT6	9	2019-01-08 14:39:26 11s

# Need your help

- ▶ Tracking QA modules → plots
- ▶ Jet samples to run/archive jet QA
- ▶ Summarize QA results, e.g. KS-tests
- ▶ Add more checks
  - Pull request check (~100 CPU-hr)
  - Weekly checks (~1000 CPU-hr)
- ▶ Suggestions welcomed
  - At Tue sim./software meeting: <https://indico.bnl.gov/category/88/>
  - At email list: [sphenix-software-l@lists.bnl.gov](mailto:sphenix-software-l@lists.bnl.gov)

