Modernizing the CMS software stack

Mircho Rodozov
On behalf of the CMS collaboration

Outline

- Introduction
- CMS Software stack
  - How big/complex is our stack
- Modernizing our stack
  - Python2-3 support
  - Code formatting
- CI/CD improvements & “exotic” experiments
  - Heterogeneous computing
  - Managing CI for 120+ github repositories
  - CI resources utilization

Introduction

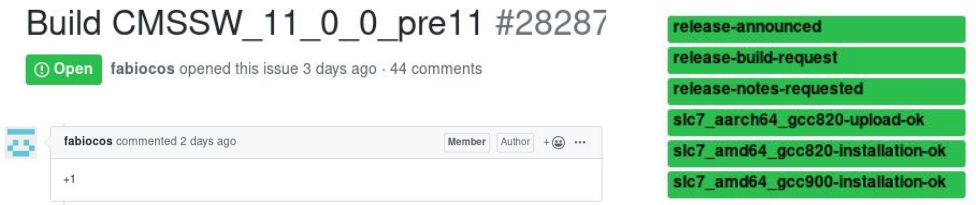
- We are:
  - Core software @ CMS
  - Build releases, Integration builds (IBs), externals, test & deploy them
  - Continuous integration / delivery (CI/CD) system
  - Maintaining CI/CD infrastructure

CMS Software stack size

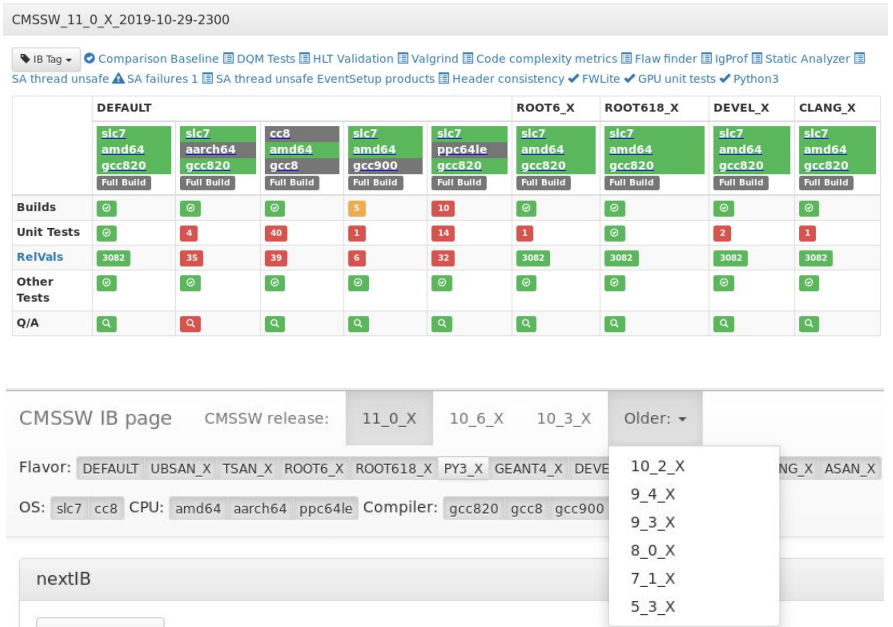
- Size of CMS Offline Software (CMSSW)
  - Over 4.5 M lines of code
    - 3.2 M C/C++ 1.4 M Python, 275k Fortran
  - Packages - 1300+
  - Multiple versions (10+ release flavors)
- Dependency on external packages
  - 500+, from source
  - custom build system (cmsBuild)
- Contributions
  - 200+ commits/week
    - 160+ github pull requests (PRs)
  - 800+ contributors
- 200+ GitHub repositories for CMSSW and externals

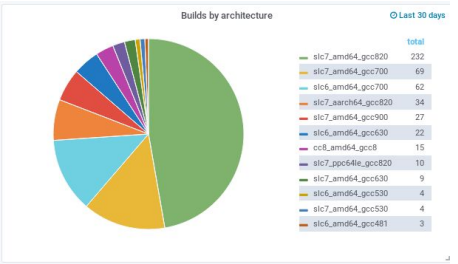
Software Distribution rate

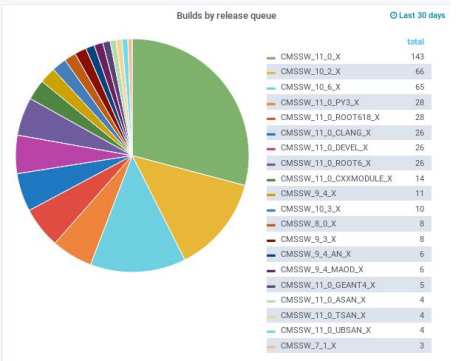
- Release builds
  - Every two weeks
  - via github issues



- Integration Builds
  - ~450/month







- Complexity

- Extensive validation of IBs
  - 4000+ tests

CI setup

submit PR

DQM: Reduce TH1 Usage #28342

Open **schneiml** wants to merge 24 commits into `cms-sw:master` from `schneiml:dqm-reduce-th1-usage`

cmsbuild added **code-checks-pending** **comparison-pending** **dqm-pending** **geometry-pending** **orp-pending** **pending-signatures** **reconstruction-pending** **simulation-pending** **tests-pending** labels

8 days ago

A new Pull Request was created by **@kpedro88** (Kevin Pedro) for master.

It involves the following packages:

FastSimulation/SimplifiedGeometryPropagator
RecoTracker/Configuration
RecoTracker/DeDx

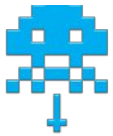
+code-checks

Logs: https://cmssdt.cern.ch/SDT/code-checks/cms-sw-PR-28254/12414

- This PR adds an extra 24KB to repository

please test

+1
Tested at: `3ffdf28`



+1

+1

+1

cmsbuild added **reconstruction-approved** and removed **reconstruction-pending** labels

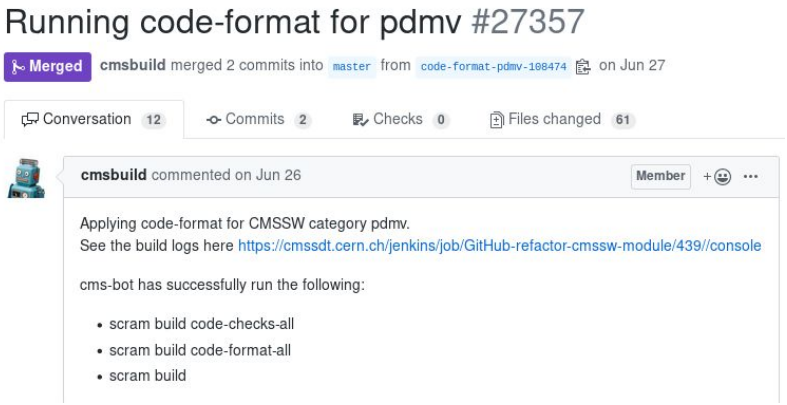
cmsbuild added **orp-approved** and removed **orp-pending** labels 2 days ago

cmsbuild added **fastsim-approved** and removed **fastsim-pending** labels 4 days ago

code-checks-approved
comparison-available
fastsim-approved
fully-signed
orp-approved
reconstruction-approved
tests-approved

Code format

- Clang format and clang tidy
  - Formatted 3.2 M lines
  - Collision with open PRs avoided
  - Style used: Google
  - Rules enforced via PR testing
- Automated transition



Clang format

```
33  31 +  class BeamSplash : public edm::EDFilter {
34  32    public:
35      -   explicit BeamSplash( const edm::ParameterSet & );
    33  +   explicit BeamSplash(const edm::ParameterSet &);
36  34    ~BeamSplash() override;
37      -
    35  +
38  36    private:
39      -   bool filter ( edm::Event &, const edm::EventSetup&) override;
40      -
    37  +   bool filter(edm::Event &, const edm::EventSetup &) override;
    38  +
```

Clang tidy

```
78      -   if ( EBRecHitCollection_.label() != "" && EBRecHitCollection_.instance() != "" )
    78  +   if ( !EBRecHitCollection_.label().empty() && !EBRecHitCollection_.instance().empty() )
79  79      {
80  80        iEvent.getByLabel( EBRecHitCollection_, pEBRecHits);
81  81        if ( pEBRecHits.isValid() )
      @@ -88,7 +88,7 @@ bool BeamSplash::filter( edm::Event& iEvent, const edm::EventSetup& iSetup)
88  88      }
89  89    }
90  90
91      -   if ( EERecHitCollection_.label() != "" && EERecHitCollection_.instance() != "" )
    91  +   if ( !EERecHitCollection_.label().empty() && !EERecHitCollection_.instance().empty() )
92  92      {
93  93        iEvent.getByLabel( EERecHitCollection_, pEERecHits);
94  94
      @@ -102,7 +102,7 @@ bool BeamSplash::filter( edm::Event& iEvent, const edm::EventSetup& iSetup)
```

Modernizing python

- 1.2 M lines converted to be py3 compatible
- Migration completed
  - Made it compatible while running py2
  - Works with both
- Mostly a manual job
  - Limited number of cases
  - Some conversion tools exist (futurize, autopep)


```
 9  + from builtins import int
10
11    ROOT.gSystem.Load("libFWCoreFWLite")
12    ROOT.FWLiteEnabler.enable()
    @@ -512,7 +512,7 @@ def to (self, entryIndex):
512             """Jumps to event entryIndex"""
513             if self._veryFirstTime:
514                 self._createFWLiteEvent()
    -           return self._event.to ( long(entryIndex) )
515 +           return self._event.to ( int(entryIndex) )
```

```
 7  + if sys.version_info[0]>2:
 8  +     import _pickle as cPickle
 9  + else:
10  +     import cPickle
```

Python packaging upgrade

- Integrated 200+ python externals
  - Build from sources using pip
- Simplified format
  - one line update
  - py2 and py3 recipes unified
- Customization for the new packages
  - a package can have patch
  - build only for py2 or py3 if required
  - conditionally build
    - python version
    - architecture

Before

```
Source: https://github.com/matplotlib/matplotlib/archive/v%{realversion}.tar.gz
Requires: py2-pytz py2-numpy py2-python-dateutil zlib libpng freetype py2-pyparsing py2-six
BuildRequires: py2-pip

%prep
%setup -n matplotlib-%{realversion}
cat >> setup.cfg <<- EOF
[directories]
basedirlist = ${FREETYPE_ROOT}:${LIBPNG_ROOT}:${ZLIB_ROOT}:${PY2_NUMPY_ROOT}:${PY2_PYTZ_ROOT}:${PY2_SIX}
[gui_support]
gtk = False
gtkagg = False
tkagg = True
wxagg = False
macosx = False
EOF
mkdir no-pkg-config
(echo '#!/bin/sh'; echo 'exit 1') > no-pkg-config/pkg-config
chmod +x no-pkg-config/pkg-config

%install
export CPLUS_INCLUDE_PATH=${FREETYPE_ROOT}/include/freetype2:${LIBPNG_ROOT}/include/libpng16
export MPLCONFIGDIR=$PWD/no-pkg-config
PATH=$PWD/no-pkg-config:$PATH \
export CPLUS_INCLUDE_PATH=${FREETYPE_ROOT}/include/freetype2:${LIBPNG_ROOT}/include/libpng16
export PYTHONUSERBASE=%i
pip install . --user
pip3 install . --user
```

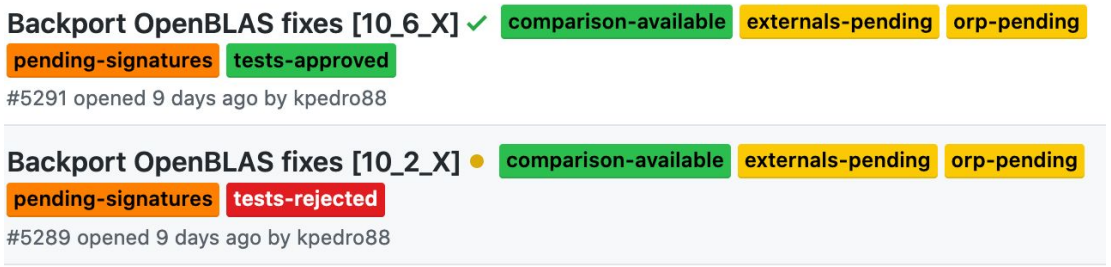
After

```
numpy==1.15.1 ; python_version<'3.0' ; platform_machine=='aarch64'
numpy==1.15.1 ; python_version>'3.0' ; platform_machine=='aarch64'
oamap==0.12.4
onnx==1.6.0
ordereddict==1.1
packaging==19.2
matplotlib==2.2.4 ; python_version<'3.0'
matplotlib==3.1.1 ; python_version>'3.0'
```

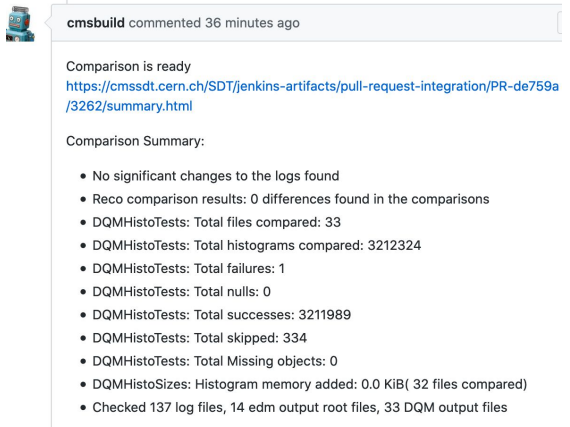
```
Requires: py2-cycler py2-kiwisolver py2-pyparsing py2-python-dateutil
Requires: zlib libpng freetype
```

Validating externals

- Integration tests for externals
  - Does it integrates (build + link against the IB)
  - Does it run

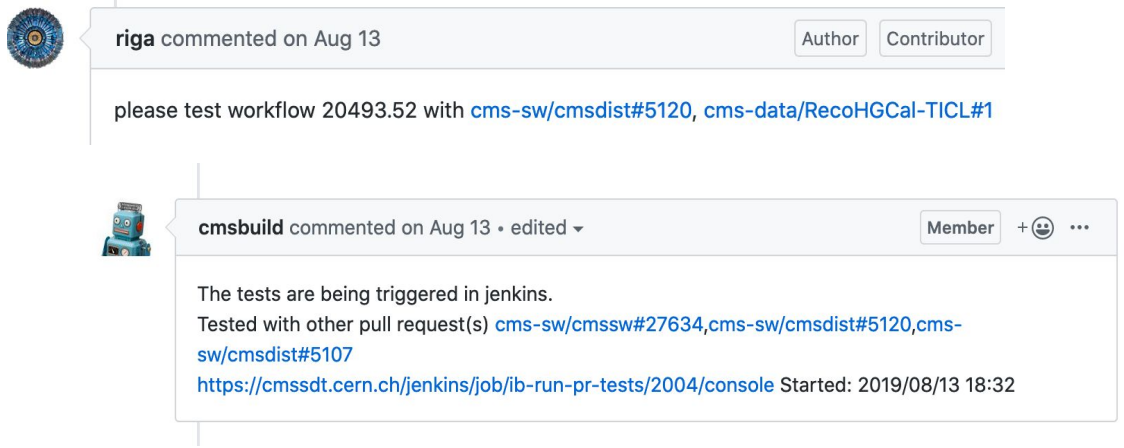


- Physics validation
  - Comparison with reference



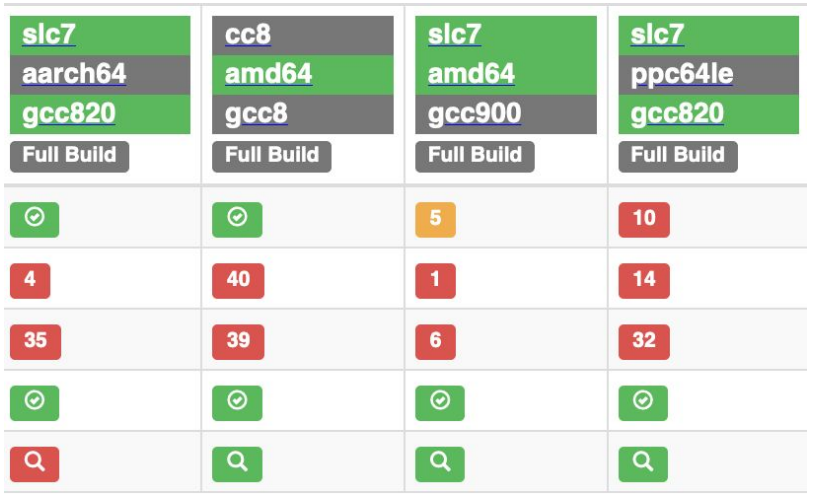
Testing multiple PRs

- Testing 200+ different repos (CMSSW, externals, data)
- Testing multiple PRs from different repos at once
- Convenient for testing pull requests that need external update



CMSSW on heterogeneous resources

- ARM
  - Cavium ThunderX 96 cores
  - Cavium ThunderX2 128 cores
- PowePC
  - Power 8
  - 1 Wistron 256 core
- GPUs
  - Shared Nvidia Tesla K20X
  - HTCondor GPUs resources
    - as Jenkins nodes
    - released on job completion
    - Thanks to IT-CM for adding a short queue to access these nodes for us !



CI resources utilization

- We run our own scheduler for CI validations
- Heuristics based, 20 to 25% faster
- https://tinyurl.com/yywh2py7

Conclusions

- CMS improves it's code base continuously
- Keep improving our CI/CD system
- Embraced modern code policies
- Migrated python
- Successfully running CI and tests on non Intel architectures
- Put some effort for better utilization of the CI resources