Improvements to LHCbPR2

Maciej Szymański [1], A.Mazurov [2], B.Couturier [3]

15.05.2017

[1] University of Chinese Academy of Sciences
[2] University of Birmingham
[3] CERN

Necessity of code validation

- **Systematic measurements of software performance critical** for the profiling and validation

Necessity of code validation

- **Systematic measurements of software performance critical** for the profiling and validation
- Need for performance baseline in **controlled conditions**

Necessity of code validation

- **Systematic measurements of software performance critical** for the profiling and validation
- Need for performance baseline in **controlled conditions**
- **Inspect any changes** due to e.g. detector description, MC generators, physics of Geant4, new external libraries

Necessity of code validation

- **Systematic measurements of software performance critical** for the profiling and validation
- Need for performance baseline in **controlled conditions**
- **Inspect any changes** due to e.g. detector description, MC generators, physics of Geant4, new external libraries
- **Compare results** across different compilers and architectures

Necessity of code validation

- **Systematic measurements of software performance critical** for the profiling and validation
- Need for performance baseline in **controlled conditions**
- **Inspect any changes** due to e.g. detector description, MC generators, physics of Geant4, new external libraries
- **Compare results** across different compilers and architectures
- Testing **special configurations** and features

Necessity of code validation

- **Systematic measurements of software performance critical** for the profiling and validation
- Need for performance baseline in **controlled conditions**
- **Inspect any changes** due to e.g. detector description, MC generators, physics of Geant4, new external libraries
- **Compare results** across different compilers and architectures
- Testing **special configurations** and features
- Efficient **check for bugs**

Necessity of code validation

- **Systematic measurements of software performance critical** for the profiling and validation
- Need for performance baseline in **controlled conditions**
- **Inspect any changes** due to e.g. detector description, MC generators, physics of Geant4, new external libraries
- **Compare results** across different compilers and architectures
- Testing **special configurations** and features
- Efficient **check for bugs**
- Not only to monitor resource consumption, but also to **measure the physics performance**

Necessity of code validation

- **Systematic measurements of software performance critical** for the profiling and validation
- Need for performance baseline in **controlled conditions**
- **Inspect any changes** due to e.g. detector description, MC generators, physics of Geant4, new external libraries
- **Compare results** across different compilers and architectures
- Testing **special configurations** and features
- Efficient **check for bugs**
- Not only to monitor resource consumption, but also to **measure the physics performance**
- **Daily maintenance** and user support crucial

The issue

- **Nightlies** infrastructure **not suitable** for performance test
  - only boolean result and list of errors or warnings
  - running on all slots usually not needed
  - risk of interference with builds in case of long tests

The issue

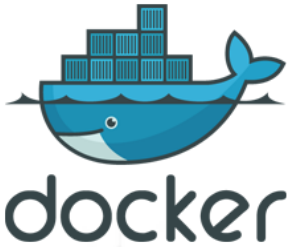
- **Nightlies** infrastructure **not suitable** for performance test
  - only boolean result and list of errors or warnings
  - running on all slots usually not needed
  - risk of interference with builds in case of long tests
- Tests of software performance often **ran manually by experts**
  - **specific knowledge** required to setup, run and understand the test
  - **resource consuming** to run the code on personal computer
  - **manual comparison** with other versions
  - results usually **not available publicly**

Requirements for the framework

- Definition of tests in **software repository**
  - e.g. PRConfig

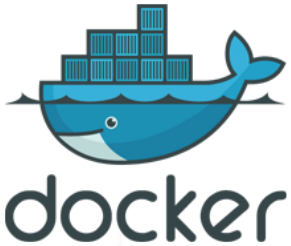
Requirements for the framework

- Definition of tests in **software repository**
  - e.g. PRConfig
- **Flexibility** in running the software
  - choice of test frequency
  - usage of specific platform



Requirements for the framework

- Definition of tests in **software repository**
  - e.g. PRConfig
- **Flexibility** in running the software
  - choice of test frequency
  - usage of specific platform
- **Friendly reporting** of the results (e.g. comparing histograms, trend analysis, memory and CPU consumption)



LHCb Performance and Regression

- Based on **microservice architecture**

wso2.com

LHCb Performance and Regression

- Based on **microservice architecture**

wso2.com

- **Loosely coupled modules** communicating through simple API's

LHCb Performance and Regression

- Based on **microservice architecture**

wso2.com

- **Loosely coupled modules** communicating through simple API's
- **Easy** to understand, maintain and scale

LHCb Performance and Regression

- Based on **microservice architecture**

wso2.com

- **Loosely coupled modules** communicating through simple API's
- **Easy** to understand, maintain and scale
- Usage of **software containers** and management tools facilitates deployment process

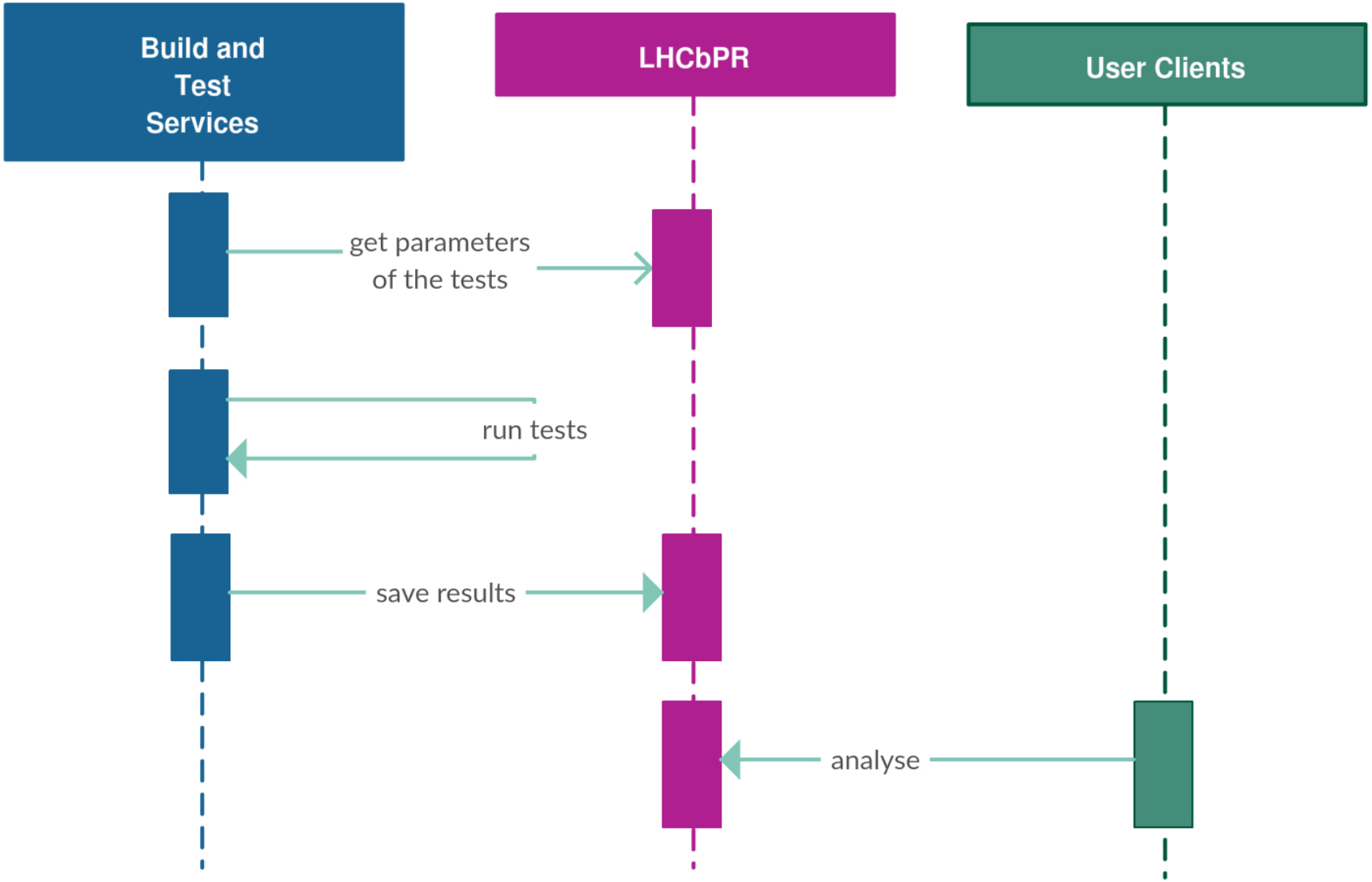
LHCb Performance and Regression

- Based on **microservice architecture**

wso2.com

- **Loosely coupled modules** communicating through simple API's
- **Easy** to understand, maintain and scale
- Usage of **software containers** and management tools facilitates deployment process
- **Generic form** of modules enables to apply the proposed architecture not only in LHCb, but can be adopted elsewhere

Sequence diagram



A.Mazurov, CHEP 2016

Infrastructure (1)

- Periodic tests started by the **Jenkins** job
  - `https://jenkins-lhcb-nightlies.web.cern.ch/job/periodic-tests/job/tests-poll/`
  - nightly builds as input flavour

Infrastructure (1)

- Periodic tests started by the **Jenkins** job
  - https://jenkins-lhcb-nightlies.web.cern.ch/job/periodic-tests/job/tests-poll/
  - nightly builds as input flavour

- Configuration in **XML** files
  - https://gitlab.cern.ch/lhcb-core/LHCbNightlyConf/blob/master/test_schedule2.xml

```xml
<periodictest>
  <schedule type="week" time="10:00">Mon,Tue,Wed,Thu,Fri</schedule>
  <slot>lhcb-future</slot>
  <project>Brunel</project>
  <platform>x86_64-slc6-gcc62-opt</platform>
  <test runner="lhcbpr" group="MiniBrunel" env="lb-run-gaudirun|TimeLineHandler"/>
  <os_label>perf</os_label>
  <count>5</count>
</periodictest>
```

Infrastructure (1)

- Periodic tests started by the **Jenkins** job
  - https://jenkins-lhcb-nightlies.web.cern.ch/job/periodic-tests/job/tests-poll/
  - nightly builds as input flavour

- Configuration in **XML** files
  - https://gitlab.cern.ch/lhcb-core/LHCbNightlyConf/blob/master/test_schedule2.xml

schedule weekly or monthly

```
<periodictest>
  <schedule type="week" time="10:00">Mon,Tue,Wed,Thu,Fri</schedule>
  <slot>lhcb-future</slot>
  <project>Brunel</project>
  <platform>x86_64-slc6-gcc62-opt</platform>
  <test runner="lhcbpr" group="MiniBrunel" env="lb-run-gaudirun|TimeLineHandler"/>
  <os_label>perf</os_label>
  <count>5</count>
</periodictest>
```

Infrastructure (1)

- Periodic tests started by the **Jenkins** job
  - https://jenkins-lhcb-nightlies.web.cern.ch/job/periodic-tests/job/tests-poll/
  - nightly builds as input flavour

- Configuration in **XML** files
  - https://gitlab.cern.ch/lhcb-core/LHCbNightlyConf/blob/master/test_schedule2.xml

schedule weekly or monthly

```
<periodictest>
  <schedule type="week" time="10:00">Mon,Tue,Wed,Thu,Fri</schedule>
  <slot>lhcb-future</slot>
  <project>Brunel</project>
  <platform>x86_64-slc6-gcc62-opt</platform>
  <test runner="lhcbpr" group="MiniBrunel" env="lb-run-gaudirun|TimeLineHandler"/>
  <os_label>perf</os_label>
  <count>5</count>
</periodictest>
```

description of options (stored in PRConfig)

Infrastructure (1)

- Periodic tests started by the **Jenkins** job
  - https://jenkins-lhcb-nightlies.web.cern.ch/job/periodic-tests/job/tests-poll/
  - nightly builds as input flavour

- Configuration in **XML** files
  - https://gitlab.cern.ch/lhcb-core/LHCbNightlyConf/blob/master/test_schedule2.xml

```
<periodictest>
  <schedule type="week" time="10:00">Mon,Tue,Wed,Thu,Fri</schedule>
  <slot>lhcb-future</slot>
  <project>Brunel</project>
  <platform>x86_64-slc6-gcc62-opt</platform>
  <test runner="lhcbpr" group="MiniBrunel" env="lb-run-gaudirun|TimeLineHandler"/>
  <os_label>perf</os_label>
  <count>5</count>
</periodictest>
```

schedule weekly or monthly

description of options (stored in PRConfig)

executable and handler (LHCbPR API)

Infrastructure (1)

- Periodic tests started by the **Jenkins** job
  - https://jenkins-lhcb-nightlies.web.cern.ch/job/periodic-tests/job/tests-poll/
  - nightly builds as input flavour

- Configuration in **XML** files
  - https://gitlab.cern.ch/lhcb-core/LHCbNightlyConf/blob/master/test_schedule2.xml

```
<periodictest>
  <schedule type="week" time="10:00">Mon,Tue,Wed,Thu,Fri</schedule>
  <slot>lhcb-future</slot>
  <project>Brunel</project>
  <platform>x86_64-slc6-gcc62-opt</platform>
  <test runner="lhcbpr" group="MiniBrunel" env="lb-run-gaudirun|TimeLineHandler"/>
  <os_label>perf</os_label>
  <count>5</count>
</periodictest>
```

schedule weekly or monthly

description of options (stored in PRConfig)

executable and handler (LHCbPR API)

machine label

Infrastructure (2)

- Tests are running on dedicated machines
  - volhcb05 with SLC6
  - lblhcbpr3 with CERN CentOS7

Infrastructure (2)

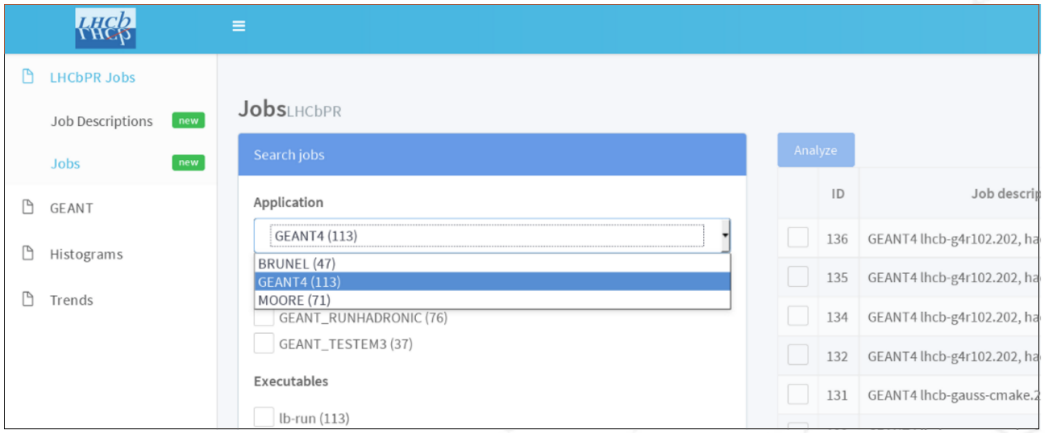
- Tests are running on dedicated machines
  - volhcb05 with SLC6
  - lblhcbpr3 with CERN CentOS7
- Results of the tests **parsed by the specific handlers**
  - LHCbPR2HD `https://gitlab.cern.ch/lhcb-core/LHCbPR2HD`
  - to save int, float, string, file and json

Infrastructure (2)

- Tests are running on dedicated machines
  - volhcb05 with SLC6
  - lblhcbpr3 with CERN CentOS7
- Results of the tests **parsed by the specific handlers**
  - LHCbPR2HD `https://gitlab.cern.ch/lhcb-core/LHCbPR2HD`
  - to save int, float, string, file and json
- Zip file sent to the database through **Dirac Storage Element** `/lhcb/prdata/zips`
  - LHCbPR2BE `https://gitlab.cern.ch/lhcb-core/LHCbPR2BE`

Infrastructure (2)

- Tests are running on dedicated machines
  - volhcb05 with SLC6
  - lblhcbpr3 with CERN CentOS7
- Results of the tests **parsed by the specific handlers**
  - LHCbPR2HD `https://gitlab.cern.ch/lhcb-core/LHCbPR2HD`
  - to save int, float, string, file and json
- Zip file sent to the database through **Dirac Storage Element** `/lhcb/prdata/zips`
  - LHCbPR2BE `https://gitlab.cern.ch/lhcb-core/LHCbPR2BE`
- **Web front–end** http://lblhcbpr2.cern.ch/
  - LHCbPR2FE `https://gitlab.cern.ch/lhcb-core/LHCbPR2FE`



Latest improvements

- **Migration** of the job polling the periodic tests to the **new Jenkins instance**
- Storing logs of test jobs on EOS (`/eos/lhcb/storage/lhcbpr/logs/`)

Latest improvements

- **Migration** of the job polling the periodic tests to the **new Jenkins instance**
- Storing logs of test jobs on EOS (`/eos/lhcb/storage/lhcbpr/logs/`)
- **New dashboard** to monitor how the periodic test jobs finalise (in collaboration with Stefan Chitic)

`https://lbnightlies.cern.ch/periodic/`

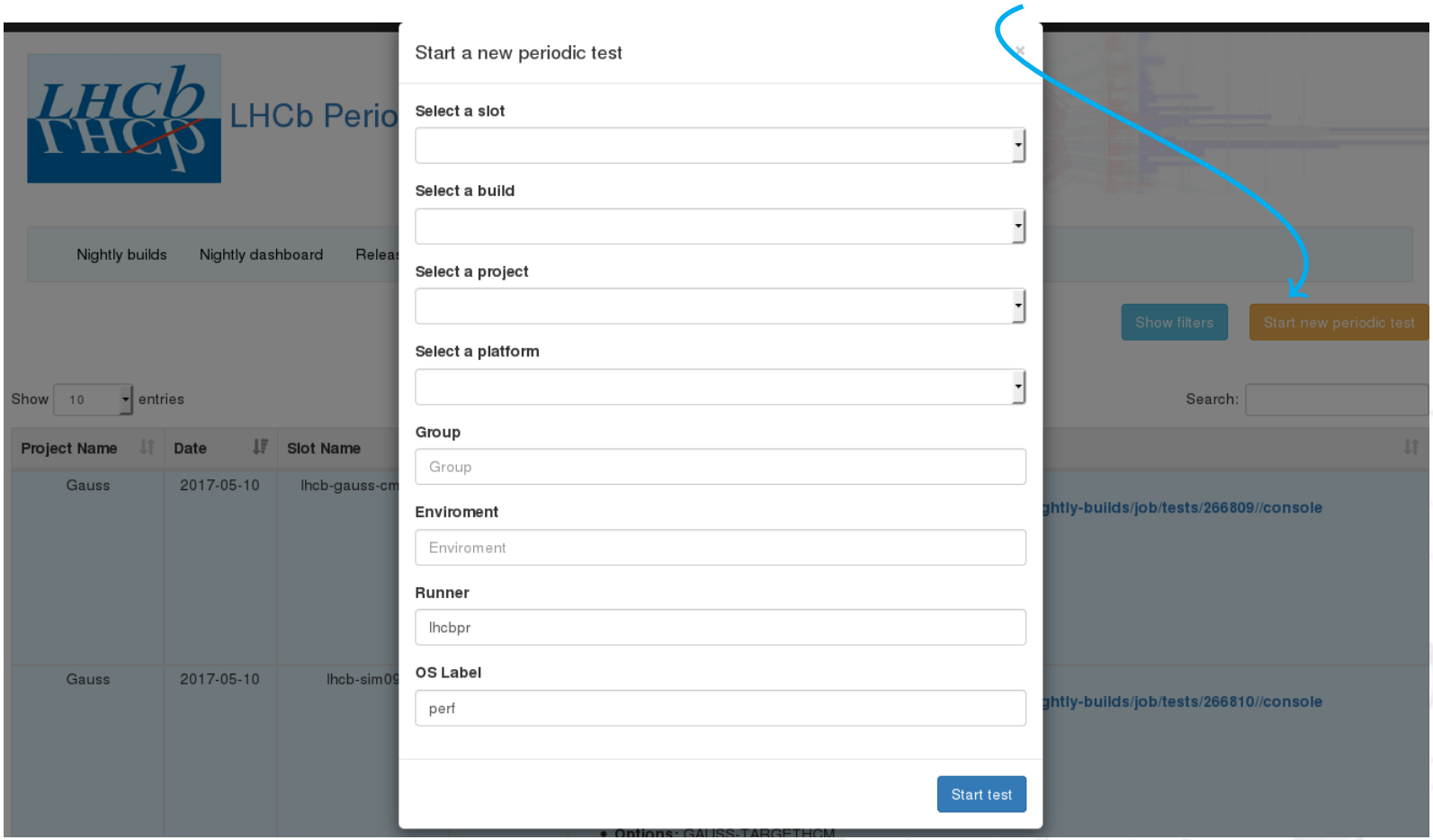
Latest improvements

- Upgrades with **messaging system** (RabbitMQ)

Latest improvements

- Upgrades with **messaging system** (RabbitMQ)
- Users may run tests on demand
- The message is sent to the queue checked by Jenkins job every 10 minutes

'Start new periodic test' button available after login



Latest improvements

- Running tests without explicit time scheduling

RabbitMQ

Latest improvements

- Running tests without explicit time scheduling
  - nightlies scripts send the **message once the builds are done**

RabbitMQ

Latest improvements

- Running tests without explicit time scheduling
  - nightlies scripts send the **message once the builds are done**
  - Jenkins job checks periodically the queue whether there are new builds for the defined tests for the given day
    - exact time is ignored
    - no need for waiting
    - build may not be ready on scheduled time

RabbitMQ

Latest improvements

- Running tests without explicit time scheduling
  - nightlies scripts send the **message once the builds are done**
  - Jenkins job checks periodically the queue whether there are new builds for the defined tests for the given day
    - exact time is ignored
    - no need for waiting
    - build may not be ready on scheduled time
  - if yes, the test is scheduled to run

RabbitMQ

Latest improvements

- Running tests without explicit time scheduling
  - nightlies scripts send the **message once the builds are done**
  - Jenkins job checks periodically the queue whether there are new builds for the defined tests for the given day
    - exact time is ignored
    - no need for waiting
    - build may not be ready on scheduled time
  - if yes, the test is scheduled to run
  - implemented, testing, not yet in production

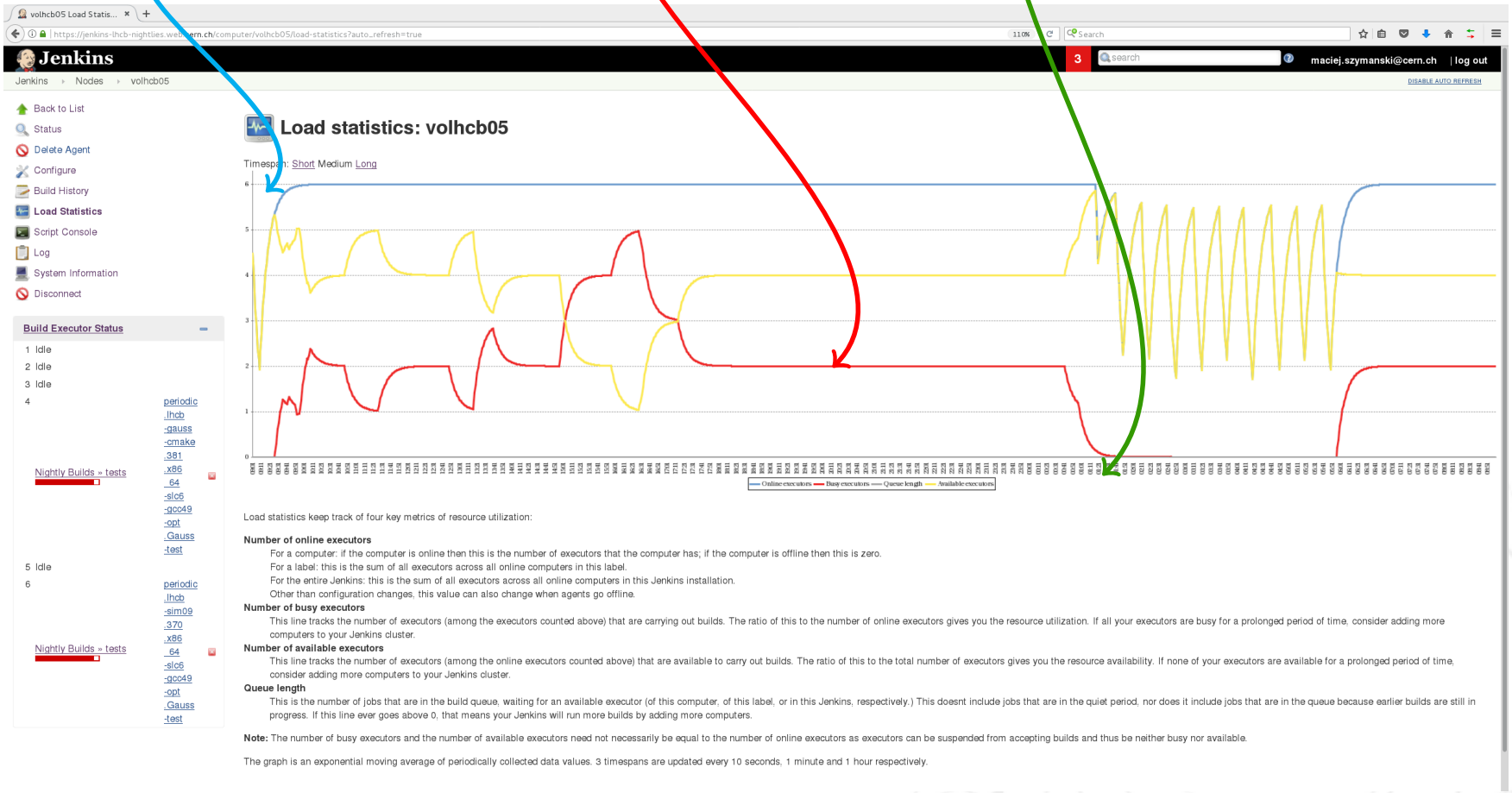
RabbitMQ

Load of volhcb05

6 executors

busy executors

time span of 24h

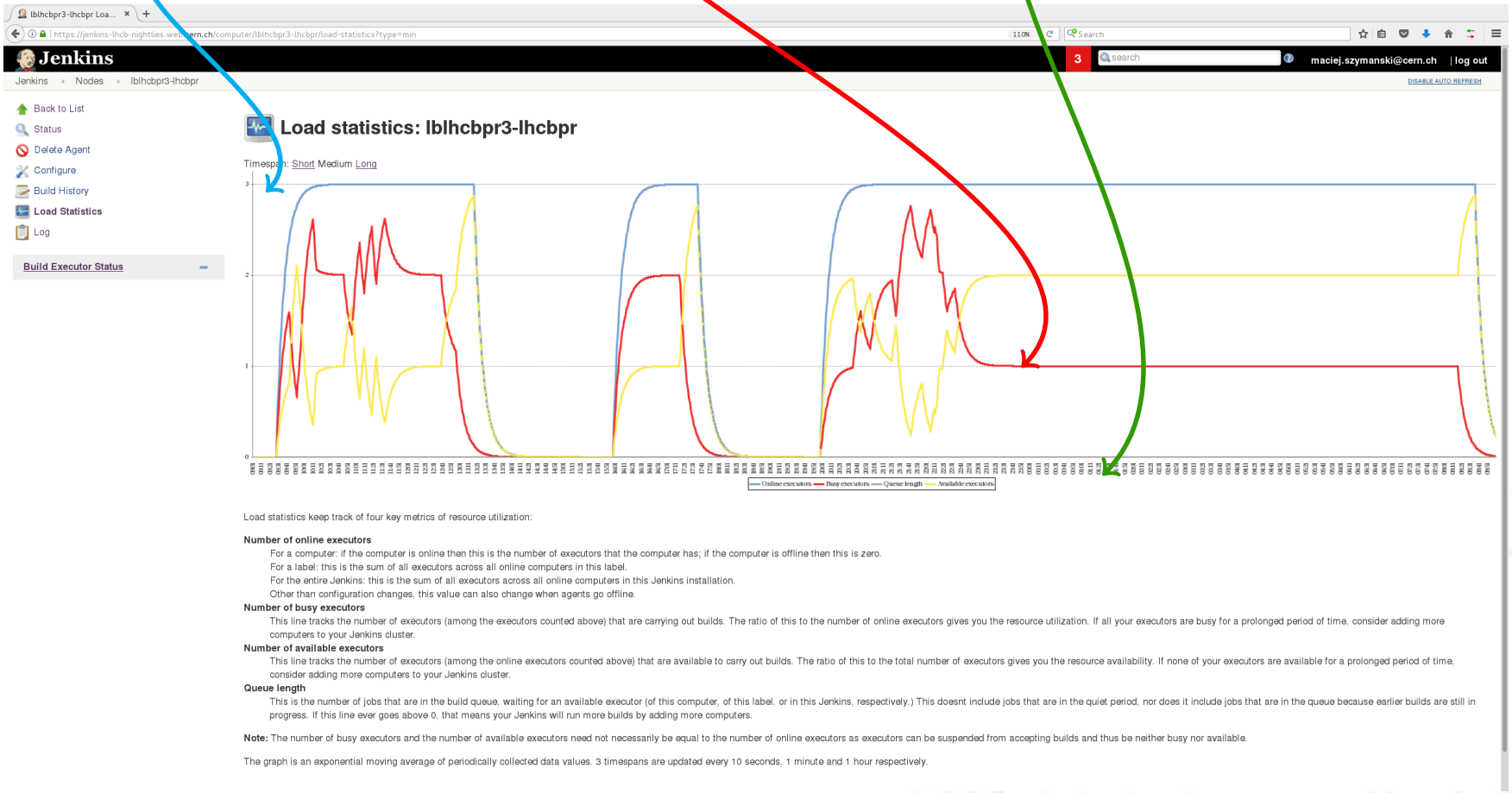


Load of lblhcbpr3

3 executors

busy executors

time span of 24h



LHCbPR - current tests

- Brunel, timing information
- MiniBrunel
  - work of Stefan Roiser and Chris Hasse (see Upgrade session on Thursday morning)
  - callgrind, timing, physics performance

LHCbPR - current tests

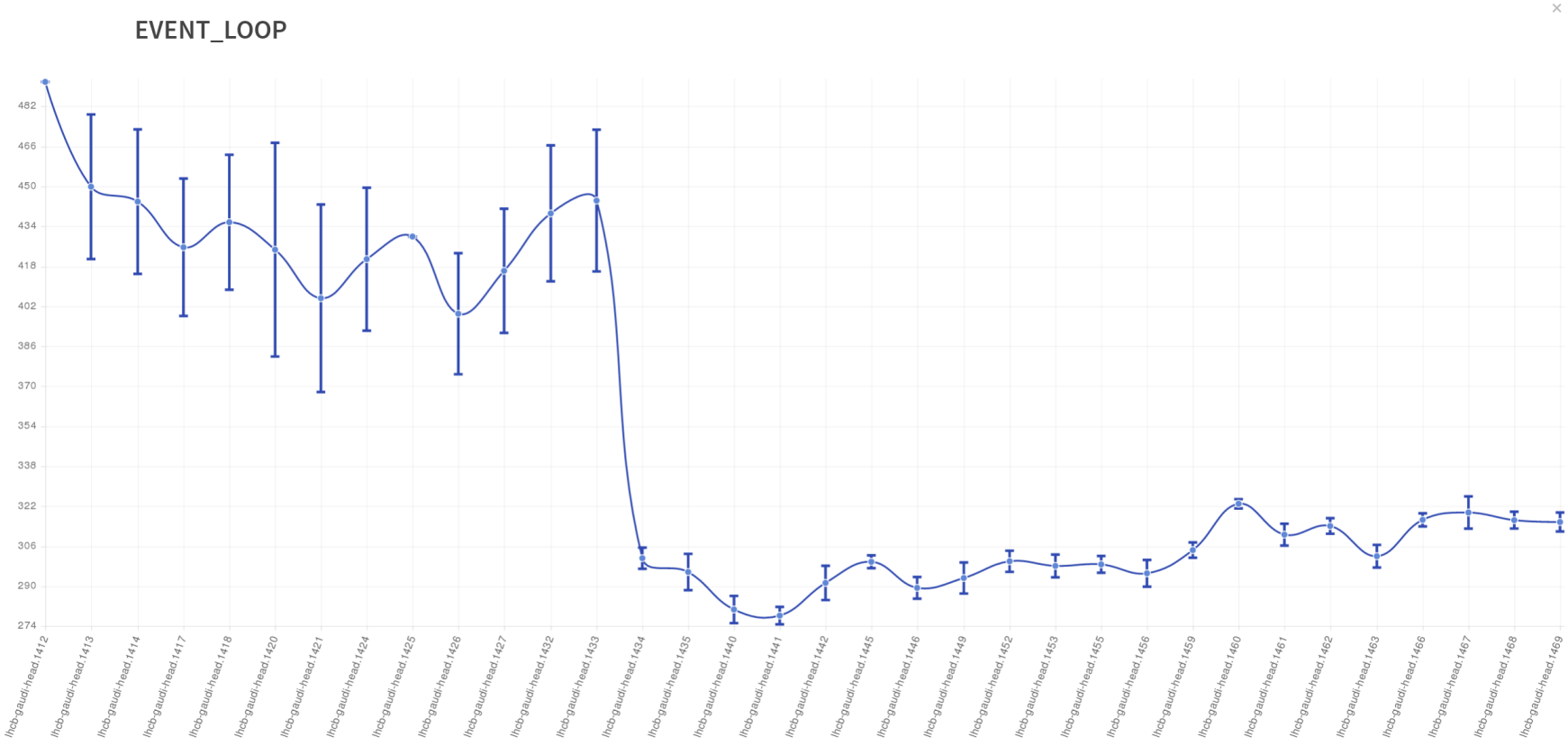
- Brunel, timing information
- MiniBrunel
  - work of Stefan Roiser and Chris Hasse (see Upgrade session on Thursday morning)
  - callgrind, timing, physics performance
- Moore
  - testing rates of High Level Trigger

LHCbPR - current tests

- Brunel, timing information
- MiniBrunel
  - work of Stefan Roiser and Chris Hasse (see Upgrade session on Thursday morning)
  - callgrind, timing, physics performance
- Moore
  - testing rates of High Level Trigger
- Gauss and Geant4
  - see talk by Tim Williams on 67th A&S week
  - timing the simulation software per detector and particle
  - validation of different build systems
  - early physics validation
  - standalone tests isolating other changes (calorimetry, hadronic cross sections, multiple scattering)

Brunel timing

https://lblhcbpr2.cern.ch/#/app/trend?apps=1&options=6&executables=4&platforms=18&withNightly&nightlyVersionNumber=340&attr=5



- time spent by reconstruction algorithm per event as a function of the software version

HLT rate tests

https://lhcbpr-docs.web.cern.ch/lhcbpr-docs/

LHCB PR

Results of HLT Tests

Last 20 tests

Date	Test	Slot.build
2016-12-11 20:16:53	2016NB_25ns_L0Filt0x160B Physics_pp_Draft2016_DefHlt1	lhcb-head.1319
2016-12-11 18:59:37	2016NB_25ns_L0Filt0x160B Physics_pp_LooseDraft2016_TEST2	lhcb-2016-patches.234
2016-12-11 15:16:07	2016NB_25ns_L0Filt0x160B Physics_pp_LooseDraft2016_DefHlt1	lhcb-2016-patches.234
2016-12-11 12:31:21	2015NB_25ns_L0Filt0x1606 Physics_pp_Draft2016_DefHlt1	lhcb-2016-patches.234
2016-12-10 20:21:01	2016NB_25ns_L0Filt0x160B Physics_pp_Draft2016_DefHlt1	lhcb-head.1318
2016-12-10 19:07:57	2016NB_25ns_L0Filt0x160B Physics_pp_LooseDraft2016_TEST2	lhcb-2016-patches.233
2016-12-10 15:23:41	2016NB_25ns_L0Filt0x160B Physics_pp_LooseDraft2016_DefHlt1	lhcb-2016-patches.233
2016-12-10 12:34:06	2015NB_25ns_L0Filt0x1606 Physics_pp_Draft2016_DefHlt1	lhcb-2016-patches.233
2016-12-09 19:03:41	2016NB_25ns_L0Filt0x160B Physics_pp_LooseDraft2016_TEST2	lhcb-2016-patches.232

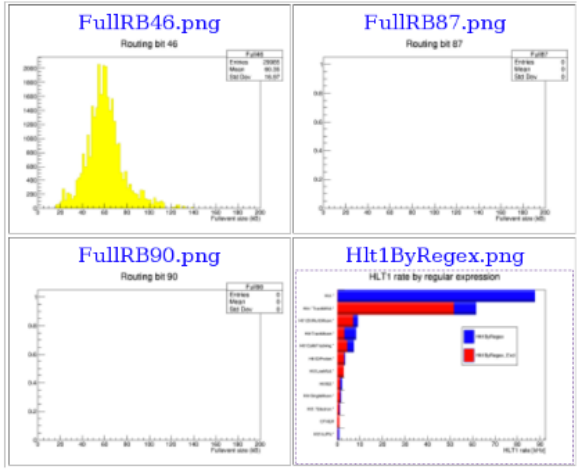
Hlt1ByRegex

#	Regex	Inclusive [kHz]	[MB/s]	[kB/event]	Exclusive [kHz]	[MB/s]	[kB/event]
1	Hlt1.*	87.9 ± 1.6	5063.3	59.0	0.0 ± 0.0	0.0	0.0
2	Hlt1.*TrackMVA.*	61.8 ± 1.4	3627.2	60.1	51.9 ± 1.3	3037.3	60.0
3	Hlt1(Di|Multi)Muon.*	9.2 ± 0.6	611.0	67.8	7.0 ± 0.5	460.5	67.1
4	Hlt1TrackMuon.*	8.4 ± 0.5	472.8	57.6	3.2 ± 0.3	177.7	56.3
5	Hlt1CalibTracking.*	7.4 ± 0.5	438.9	60.7	4.4 ± 0.4	263.4	60.8
6	Hlt1DiProton.*	3.5 ± 0.3	166.0	48.6	3.1 ± 0.3	145.2	48.5
7	Hlt1LowMult.*	2.8 ± 0.3	70.3	25.4	2.8 ± 0.3	70.3	25.4
8	Hlt1B2.*	2.1 ± 0.3	140.8	67.6	1.3 ± 0.2	83.2	65.5
9	Hlt1SingleMuon.*	1.7 ± 0.2	90.8	54.7	1.2 ± 0.2	59.3	50.6
10	Hlt1.*Electron.*	1.3 ± 0.2	67.8	53.4	0.8 ± 0.2	42.3	52.0
11	OTHER	1.0 ± 0.2	52.0	55.0	1.0 ± 0.2	52.0	55.0
12	Hlt1IncPhi.*	0.9 ± 0.2	59.1	67.3	0.3 ± 0.1	18.4	70.7

Hlt2Rates_Totals

#	Line	Incl. (kHz)	MB/s
-	Total	9.17 ± 0.55	618.39
-	Total_Full	6.17 ± 0.45	412.99
-	Total_Turbo	4.10 ± 0.37	179.72
-	Total_TurboCalib	0.37 ± 0.11	25.68
-	Total_TurboAndFull	1.23 ± 0.20	-
-	Total_FullAndTurboCalib	0.17 ± 0.07	-
-	Total_TurboAndTurboCalib	0.13 ± 0.07	-

plots



HLT rates

root file: download it
log file: download it
Table of Settings
Table of InputStats
Table of EventSizes
Table of Hlt1ByRegex
Table of Hlt2Rates_Totals
Table of Hlt1Rates
Table of Hlt2Rates_Full
Table of Hlt2Rates_Turbo
Table of Hlt2Rates_TurboCalib
Table of RatesByStream_Modules
Table of RatesByStream_AllStreams
Table of RateMatrix
Table of persistRecoLines
Table of Hlt2Overlap_TurboAndFull
Table of TimingHlt2_combinatorics
Table of TimingHlt2_total
Table of TimingHlt1_total
Table of plots

Settings

Property	Setting
settings	Physics_pp_Draft2016
inputdata	2015NB_25ns_L0Filt0x1606
Events	30000

InputStats

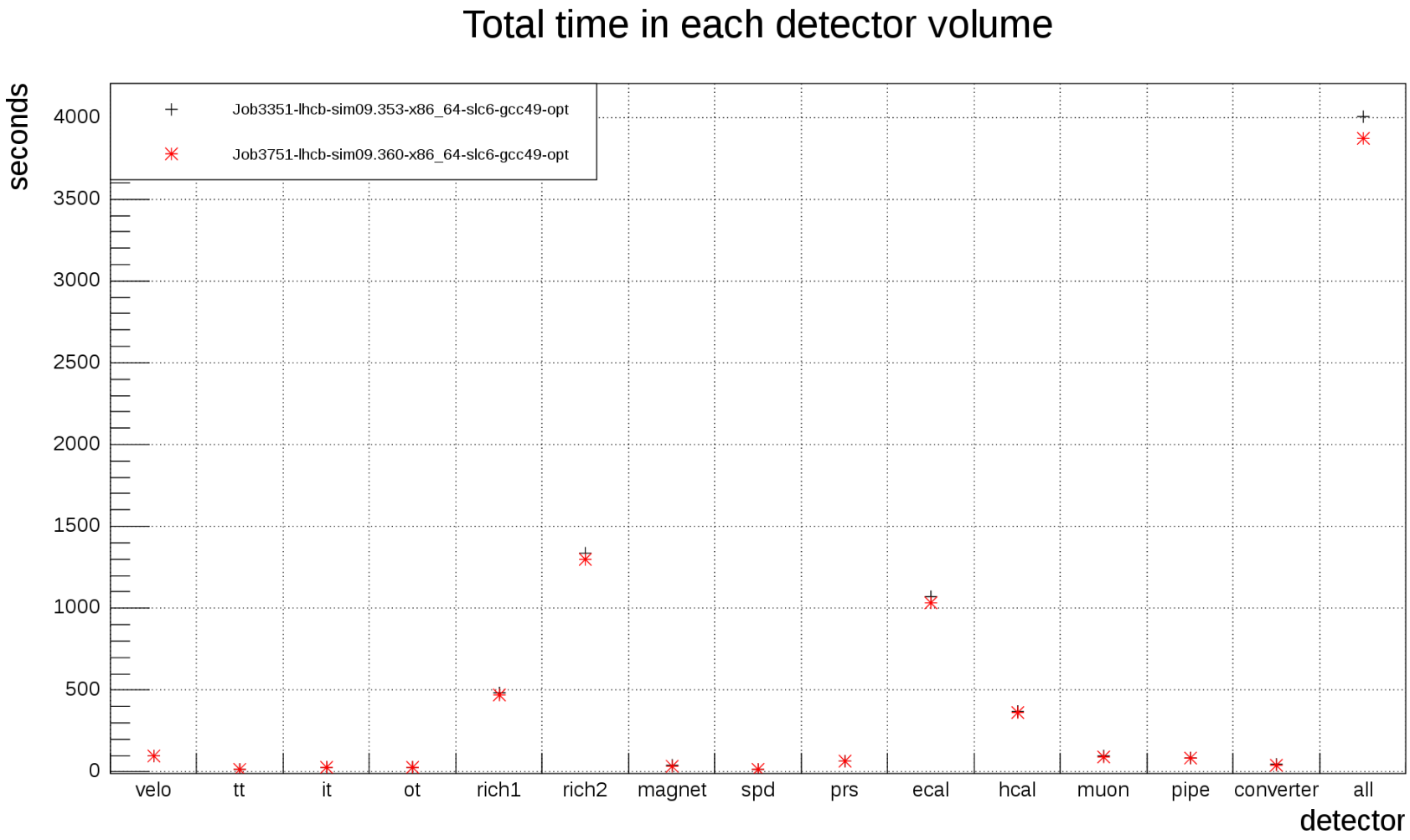
EventSizes

RB	descr.	Full evt (kB)	Turbo evt (kB)	PR part (kB)
RB88	Turbo	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
RB46	HLT1	60.4 ± 0.1	3.2 ± 0.0	47.9 ± 0.6
RB90	Turbocalib	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
RB87	Full	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

- R.Currie, R.Matev, M.Vesterinen, S.Stahl, C.Fitzpatrick

Gauss timing

https://lblhcbpr2.cern.ch/#/app/detime?apps=4&options=21&executables=3&executables=2&executables=4&platforms=14&withNightly=false&nightlyVersionNumber=1&jobs=3351&jobs=3751



D.Popov

- time spent by simulation software per detector

Gauss timing

https://lblhcbpr2.cern.ch/#/app/detime?apps=4&options=21&executables=3&executables=2&executables=4&platforms=14&withNightly=false&nightlyVersionNumber=1&jobs=3351&jobs=3751

Timing of particle groups per detector for Job3351-lhcb-sim09.353-x86_64-slc6-gcc49-opt

	prs	rich2	muon	velo	rich1	spd	it	converter	magnet	pipe	ot	tt	hcal	ecal
other	3.5700	19.9300	20.5700	23.9100	19.9200	3.1900	2.4300	3.1300	5.9100	6.1000	4	1.5600	109.4600	99.2300
opticalphoton		1261.4100	0.0200	1.8600	441.4000				0.1000	0.0100	0.0600	0.0600	0.0100	0.0200
gamma	43.4300	9.99	[illegible]	22.4100	5.7600	8.7500	13.0300	14.8600	9.9100	30.4900	11.2700	3	124.7200	511.9700
mu	0.0200	1.0600	0.4300	0.1000	0.1700	0.0900	0.0900	0.0200	0.0600	0.0100	0.2600	0.1100	0.4100	0.6300
K	0.1100	0.7500	0.5100	3.8800	0.3000	0.1700	0.4700	0.3600	0.1000	0.4100	0.3900	0.3300	1.5200	4.4400
pi	1.2600	12.1500	6.1800	35.8500	3.9600	1.6700	4.9200	2.6400	3.0900	7.7500	4.9000	3.2200	23.1300	48.1100
e	18.5900	32.2100	35.7000	11.9500	13.7600	4.0400	5.9700	22.8700	18.9500	40.3300	5.9500	5.1200	112.7300	406.8700

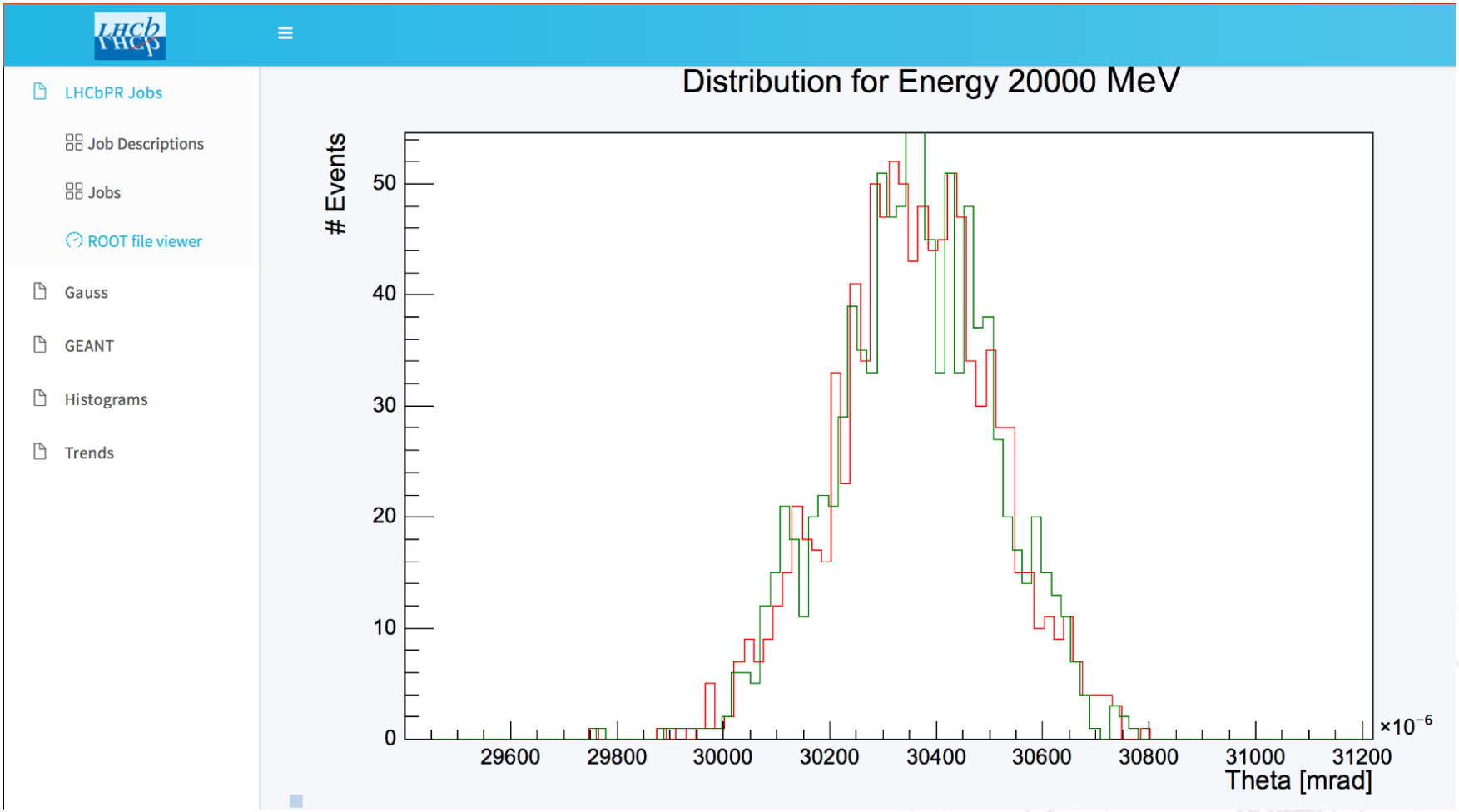
partimingh2d
x = rich2
y = opticalphoton
bin = 1, 5
entries = 1261

D.Popov

- time spent by simulation software per particle and per detector

Geant4 - CMT vs. Cmake

https://lblhcbpr2.cern.ch/#/app/rootfileviewer?apps=2&options=4&withNightly=false&nightlyVersionNumber=1&jobs=3750&jobs=3356



T.Williams

- comparison of physics results for different build systems

Ongoing developments

- Integration tests (Rob Currie)
  - Moore $\rightarrow$ Tesla $\rightarrow$ DaVinci

Ongoing developments

- Integration tests (Rob Currie)
  - Moore $\rightarrow$ Tesla $\rightarrow$ DaVinci
- New analysis modules
  - E.g. replacement of the web-based HLT rate tests monitoring

Ongoing developments

- Integration tests (Rob Currie)
  - Moore $\rightarrow$ Tesla $\rightarrow$ DaVinci
- New analysis modules
  - E.g. replacement of the web-based HLT rate tests monitoring
- Deploying new tests
  - Gauss
  - MiniBrunel

Ongoing developments

- Integration tests (Rob Currie)
  - Moore $\rightarrow$ Tesla $\rightarrow$ DaVinci
- New analysis modules
  - E.g. replacement of the web-based HLT rate tests monitoring
- Deploying new tests
  - Gauss
  - MiniBrunel
- Continuous feedback from users

LHCbPR - Resources

- Web application:
  `https://lblhcbpr2.cern.ch`
  `https://lblhcbpr2.cern.ch/api/`
  `https://gitlab.cern.ch/lhcb-core/LHCbPR2FE`
- API service:
  `https://gitlab.cern.ch/lhcb-core/LHCbPR2BE`
- ROOT HTTP service:
  `https://gitlab.cern.ch/lhcb-core/LHCbPR2ROOT`
- Tests' output handlers:
  `https://gitlab.cern.ch/lhcb-core/LHCbPR2HD`
- Project builder:
  `https://gitlab.cern.ch/amazurov/LHCbPR2`
- Jenkins configuration
  `https://gitlab.cern.ch/lhcb-core/LbNightlyTools`
- Configuration of the periodic tests
  `https://gitlab.cern.ch/lhcb-core/LHCbNightlyConf/`

Summary

- Monitoring of the software is **essential tool** in large scientific project

Summary

- Monitoring of the software is **essential tool** in large scientific project
- **LHCbPR** has already shown to be the **useful framework**

Summary

- Monitoring of the software is **essential tool** in large scientific project
- **LHCbPR** has already shown to be the **useful framework**
- **Extensible project** thanks to modular architecture (e.g. new analysis modules)

Summary

- Monitoring of the software is **essential tool** in large scientific project
- **LHCbPR** has already shown to be the **useful framework**
- **Extensible project** thanks to modular architecture (e.g. new analysis modules)
- Feedback is welcome, we are available to discuss new features, deploy new tests etc.