Table of Contents

LHCbPR - Performance and Regression Tests

Introduction

LHCbPR is responsible for systematically running the regression tests, collecting and comparing results of these tests so that any changes between different setups can be easily observed. The framework is based on a microservices architecture which breaks a project into loosely coupled modules communicating with each other through APIs. Most of the developed modules are generic which means that the proposed framework can be adopted for other experiments.

Architecture

The sequence diagram of the LHCbPR is shown in this figure:
https://twiki.cern.ch/twiki/pub/LHCb/LHCbPR/SequenceDiagram.pdf.

Infrastructure

Configuration of the test

The test is configured via https://gitlab.cern.ch/lhcb-core/LHCbNightlyConf/blob/master/test_schedule2.xml. The example of the configuration is the following:

```
<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>
```

The meaning of the keys:

- `schedule type` : you can specify `week` or `month`
- `schedule time` : usually ignored, see LHCbPR#Launching_the_tests below.
- `schedule` : the day of the week if `type="week"` (always respected).
- `slot`, `project`, `platform` : you can use globbing, e.g. `x86_64-slc6-gcc*-opt` to run the test for all gcc versions
- `test runner` : use `lhcbpr`
- `test group` : description of the option file, please contact us to add it to LHCbPR database https://lblhcbpr.cern.ch/api/options/
- `test env` : description of the command to run the test and the name of the handler. Please contact us to add the executable to LHCbPR database https://lblhcbpr.cern.ch/api/executables in case it is not already there. After `|` please specify the list of handlers separated by comma. The name should correspond to the file committed to https://gitlab.cern.ch/lhcb-core/LHCbPR2HD
- `os_label` : use `perf` for `slc6` tests or `perf-centos7` for `centos7` or `perf-centos7-timing` for the timing tests on `centos7` (the dedicated machine with the single executor in jenkins will be used)
- `count` : specify number of runs for the test (you can run multiple tests to check statistics - standard deviation will be computed for the extracted metrics)

Dashboard

Here you can find the dashboard for the periodic tests. One can verify here the status of the executed tests. The colour code is the following:

- running tests in blue
- successful tests in green
- failed tests in red
- tests which have been executed with success, but the handler failed (so that there is no output results for LHCbPR) in yellow

The urls to log files of the test (stored on EOS) and output of the jenkins job are provided.

Launching the tests

- **Automatic starting**. Tests defined in configuration file are started automatically by the Jenkins job. The job checks (messaging infrastructure described here is used for this purpose) every 5 minutes if there any new builds for the defined tests for the given day (the exact time from the configuration is ignored). If yes, another Jenkins job is triggered (the same which is used for tests of nightly builds) which actually runs the test. If the messaging infrastructure is down, as a backup, the tests may be started according to the time in the configuration when tests-poll Jenkins jobs will be enabled.
- **Manual start on demand**. After login to the dashboard, there is an orange button in top right, called `Start new periodic test`. After clicking it, you need to provide the same information which is in configuration file (except for scheduling). The message will be sent to the queue checked by Jenkins job every 5 minutes. Alternatively you can use command line:

  ```
  export RMQPWD=lhcbpr/lhcbpr
  lbq-requesttest <slot> <buildid> <project> <config> <group> <env>
  ```

  For example:

  ```
  lbq-requesttest 1467 lhcb-sim09  Gauss x86_64-slc6-gcc49-opt "GAUSS-RADLENGTHSCAN" "lb-run
  ```

  One can use also -l flag to specify the machine label in jenkins.

Machines

Here are the machines used for running the periodic tests:

- `volhcb05` with SLC6 (8 executors)
- `lblhcbpr6` with Centos7 (8 executors)
- `lblhcbpr1` with Centos7 (1 executor)
- `lbhltperf01` node devoted for throughput testing

Handlers

Handlers are python scripts used to extract relevant information from the the output produced by the test runs. The `BaseHandler` class enables to save Int, Float, String, JSON and File. The LHCbPR framework produces the zip file with the collected results which is sent to database through Dirac Storage element (`/lhcb/prdata/zips`). The description how to create handler and test it can be found here.

Front-end

The results of the tests are automatically picked up by web front-end available here. For the development of the specific analysis module please see here. The generic tool to compare the plots can be found by going to *LHCbPR Jobs* and *ROOT file viewer* tabs. To perform the trend analysis see the example below.

There is an ongoing work on the new version of the web front-end available here.

Results of the throughput tests are available here.

Example of plotting trend

Let s assume we want to plot the time spent by `EVENT_LOOP` as a function of the software version. The option file can be found here, and the command used to run the test is:

```
lb-run --use=PRConfig -c x86_64-slc6-gcc62-opt --user-area=$(pwd)/../build Brunel/HEAD gaudirun.p
```

To plot the trend:

- go to *Trends/Trends* tab and select `Brunel` from the list of applications
- select the option you are interested in, in this case `PRTEST-COLLISION15-1000`
- tick *Show Nightly versions* and specify the number of versions to show, e.g. 50.
- start typing the name of the algorithm in the field *Filter attributes* and click *Show*

You should see the plot: https://twiki.cern.ch/twiki/pub/LHCb/LHCbPR/BrunelTiming.pdf.

Requirements for participation

To add your project to LHCbPR the following information is needed:

- Command to run the test
- The option file stored e.g. in PRConfig
- Handler to extract the relevant information from the test
- Analysis module in case you are interested in specific presentation of results (other than trend analysis and generic comparison of plots using ROOT file browser)

Analysing data from LHCbPR in SWAN

Since May 2018, results of LHCbPR tests are copied to Hadoop Distributed File System (HDFS), see the user guide, twiki and knowledge base articles for reference. Details of the procedure can be found in https://gitlab.cern.ch/maszyman/lhcbpr-hadoop.

Owing to that, one can use SWAN notebooks to create custom reports on test results. Examples are available in `/eos/user/m/maszyman/SWAN_projects/read_hdfs`. The requirement to read from HDFS is to belong to `ai-hadoop-users` e-group (which can be granted by opening a SNOW ticket to the Hadoop and Spark Service to request access).

To be able to read from HDFS, go to SWAN, open a new notebook (you may need to create a new project first), click on a star (second to last in top row - Spark clusters connection). By default you should be directed to `analytix` cluster, when you click 'Connect'.

Alternatively, you can use docker container to run pyspark:

```
# get docker image
docker login gitlab-registry.cern.ch
docker pull gitlab-registry.cern.ch/db/cerndb-infra-hadoop-conf:qa

# run it
docker run -d -it -p 5000-5300:5000-5300 --hostname $HOSTNAME --name "lhcbpr-hadoop"  -v /cvmfs/s

# go into docker image
docker exec -it lhcbpr-hadoop bash

# get kerberos token
kinit ${USER}@CERN.CH

# run pyspark interactively (alternatively write a script and run it using spark-submit command)
pyspark
```

The notebook producing trend plots of the throughput is available here.

Setup of the throughput tests

The throughput tests are running making use of lhcb-benchmark-scripts repository. Please see readme for details.

You can find an example of the setup in https://lblhcbpr.cern.ch/api/options/90/. Please note that lhcb-benchmark-scripts repository is setup in LbNightlyTools.

Development

For the development of LHCbPR, please see here.

Contact and Infos

Support:

- Ben Couturier
- Alexander Mazurov
- Maciej Szymański
- Mattermost channel for discussion and announcements about LHCbPR
- Mattermost channel for notifications about results of the throughput tests

Resources

Dashboard:

- https://lbnightlies.cern.ch/periodic/summary/

Web application:

- https://lblhcbpr.cern.ch
- https://gitlab.cern.ch/lhcb-core/LHCbPR2FE
- https://lblhcbpr.cern.ch/fe2
- https://gitlab.cern.ch/lhcb-core/LHCbPR2FE2
- https://lhcbpr-docs.web.cern.ch/lhcbpr-docs/ (Results of HLT Rate Tests)

- https://lhcbpr-hlt.web.cern.ch/lhcbpr-hlt/PerfTests/UpgradeThroughput/ (Results of Throughput Tests)

API service:

- https://lblhcbpr.cern.ch/api/
- 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://jenkins-lhcb-nightlies.web.cern.ch/job/periodic-tests/
- https://gitlab.cern.ch/lhcb-core/LbNightlyTools

Configuration of the periodic tests

- https://gitlab.cern.ch/lhcb-core/LHCbNightlyConf/blob/master/test_schedule2.xml

Collection of various talks given on the subject of LHCbPR

- Performance and Regression tests for Simulation, Alexander Mazurov. 7th Computing Workshop
- Microservices for systematic profiling and monitoring of the refactoring process at the LHCb experiment, Alexander Mazurov, CHEP 2016
- A dedicated infrastructure for performance and regression tests, LHCbPR2, abstract and summary sent for NSS 2017, Alexander Mazurov
- LHCbPR2 and simulation checks, Tim Williams, 67th A&S Week
- Improvements to LHCbPR2, Maciej Szymanski, 9th Computing Workshop
- Performance metrics in PR2, Stefan Roiser, 9th Computing Workshop
- Improvements to the LHCb software performance testing infrastructure using message queues and big data technologies, Maciej Szymanski, CHEP 2018

- For historical interest, the original design of LHCbPR was presented in this talk by Emmanouil Kiagias at the Core Software meeting on 24th April 2013

This topic: LHCb > LHCbPR
Topic revision: r19 - 2019-08-05 - MaciejSzymanski