

Monitoring LHCb Trigger developments using nightly integration tests and a new interactive web UI

Robert Currie, Edinburgh University,
on behalf of the LHCb Collaboratoion,



THE UNIVERSITY
of EDINBURGH



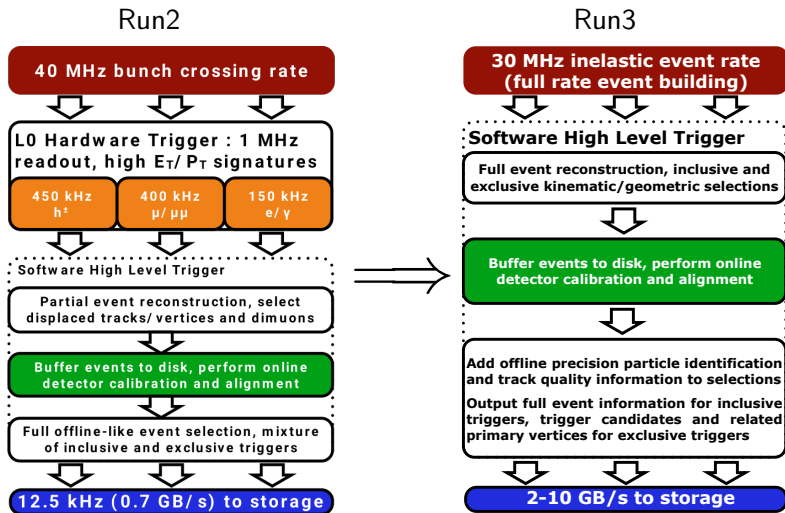
LHCb in Run3

During LHC Run3, LHCb will be using a software High Level Trigger to do all of the physics reconstruction. This will involve removing the existing L0 hardware trigger and processing events at 30Mhz in software.

Due to this it is important to understand the performance of the software from a computing and physics perspective.

Nightly testing makes it possible to understand the impact that software development has on both of these. However, keeping track of all of the different possible metrics collected when testing a piece of software can be difficult.

LHCb Trigger in Run2 vs Run3

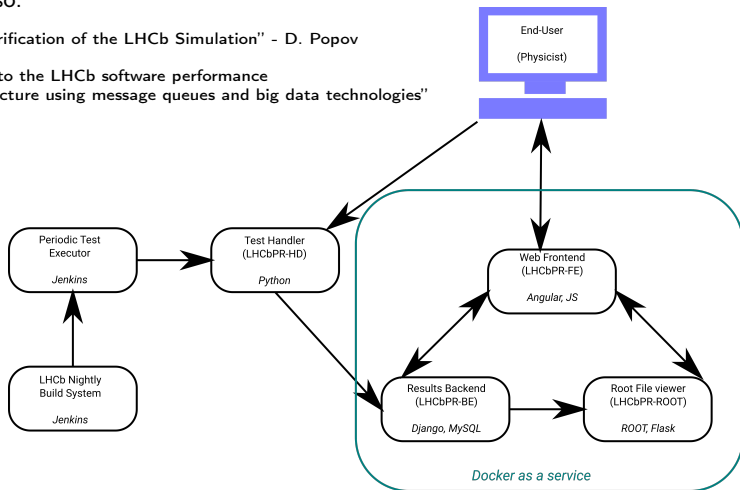


LHCb Performance Regression

See also:

“Testing and verification of the LHCb Simulation” - D. Popov

“Improvements to the LHCb software performance testing infrastructure using message queues and big data technologies” - M. Szymański



LHCbPR

The LHCbPR framework is built atop the LHCb nightly build system making use of the nightly Jenkins build infrastructure.

This framework allows for tracking of various parameters such as profiling metrics or high-level physics results. LHCbPR is already used in the verification of new releases of the LHCb simulation application.

Can track the performance of a piece of software against different metrics.

The framework is capable of storing multiple forms of information after running such as simple Python objects, ROOT files and JSON objects.

LHCbPR cont

In LHCbPR terminology the various factors that can be used to sort or filter through the results of different tests. These are referred to as:

Metric	Description
application	The application which was built
slot	The build configuration of the application
platform	Platform the application was built for
options	The configuration of the test
executable	The environment which the test was run in

These various different parameters allow for sorting and comparing the results of many different tests in a useful way.

HLT Integration into LHCbPR

Integrating HLT monitoring into the LHCbPR framework involved porting existing tests which monitor the selection rates against a known data sample.

The requirements of this were to display a high-level comparison of the physics performance between day to day builds as well as providing the ability to examine the behaviour specific components on demand.

An interactive web front-end lends itself well to presenting data in this way.

HLT Integration cont

In order to improve the interaction between the end-user and the stored data most of the data sorting/filtering is done through the desktop browser.

This is made possible by web frameworks such as Angular which makes it possible to build interfaces allowing users to interact with the data within the browser.

To track the performance of the components of the HLT software the LHCbPR API was expanded to reduce the amount of client/server interaction from the web front-end.

LHCbPR Front End

The LHCbPR web front-end is based on the Angular JS framework making use of a RESTful interface to query for test results from the Django based backend.

Originally queries against this backend were designed to allow for interactive exploration of the data and test results.

This was found to not always scale well and has now been enhanced with new features to allow for more direct querying of certain pieces of key information.

LHCbPR Front End Cont

Web development can present a unique set of challenges.

Typically this is because these technologies are in constant rapid development which has both advantages and disadvantages.

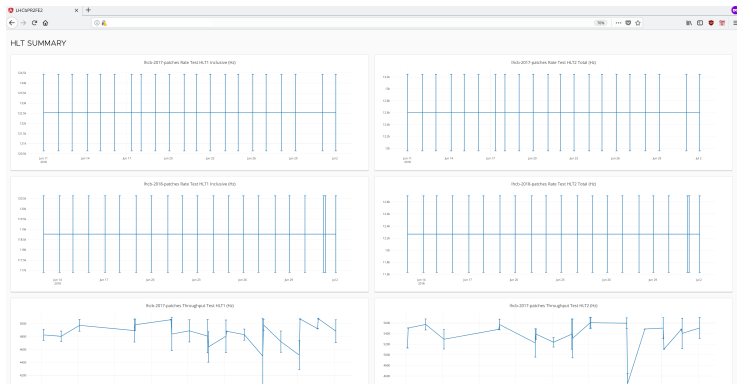
Currently working to deploy to a new front-end based on Angular6 and Project Clarity.

This has advantages including being built on a more modern JavaScript/TypeScript framework and delivering a more responsive UI.

Monitoring HLT in LHCbPR-FE

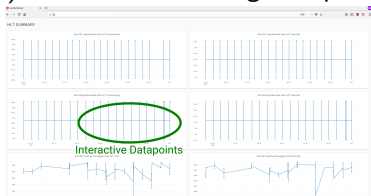


Monitoring HLT in LHCbPR-FE2

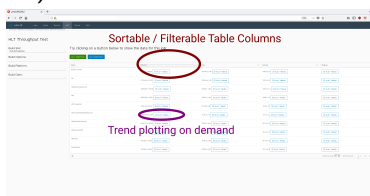


Interactive Web-UI

1) Click on interesting datapoint



2) Select a table and search



3) Plot evolution of physics parameter vs build



Conclusions

The LHCbPR framework has proven to be an expandable framework.

Able to track the results of physics tests across multiple branches of code, build architectures and execution environments.

Can use LHCbPR to track CPU and memory use of an application whilst executing different tests.

Currently tracking the computational and physics performance of several LHCb applications from their nightly builds.

Conclusions

LHCbPR-FE has proven to be a good framework for giving an interactive presentation of the data to users.

Developing a web front-end has unique challenges and advantages when displaying and examining data.

Working to integrate the full LHCb data reconstruction software stack into nightly tests.

Thank you for listening!

BACKUPS

Uniform Resource Identifier

Generic URI structure:

URI = `http(s)://service.cern.ch[/path/] [?query]`

Originally the LHCbPR service constructed all of the queries according to a `[/path/]` which allowed for an interactive exploration of the results stored in the database.

New `[?query]` functionality allows for a more direct access to the information of interest. Additionally this has reduced the amount of data that needs to be transferred improving the responsiveness of the web interface.

e.g.

`https://service.cern.ch/api/unique-build-ids/?application=MOORE`