Monitoring software performance with LHCbPR

Maciej Szymański
University of Chinese Academy of Sciences
Kraków, 25 Jun 2018
LHCb Performance and Regression framework

- **LHCbPR** is a framework for systematic monitoring of the LHCb software
- Provides performance baseline in **controlled conditions**
- Enables to **inspect any changes** due to e.g. MC generators, physics of Geant4, new external libraries, new MRs, DDDB tags, ...
- **Compare results** across different compilers and architectures
- Not only to monitor resource consumption, but also to **measure the physics performance**
- Cf. nightly tests: larger statistics and more than boolean value
Infrastructure

- Periodic tests started by the **Jenkins** job
  - Configuration of Jenkins job
  - tests triggered when corresponding nightly builds ready (using RabbitMQ)

Maciej Szymański (UCAS)
12th hackathon on software for the upgrade, Kraków, 25 Jun 2018
Infrastructure

- Periodic tests started by the **Jenkins** job
  - Configuration of Jenkins job
  - tests triggered when corresponding nightly builds ready (using RabbitMQ)

- Configuration in **XML** files
  - **LHCbNightlyConf**
Infrastructure

- Periodic tests started by the **Jenkins** job
  - Configuration of Jenkins job
  - tests triggered when corresponding nightly builds ready (using RabbitMQ)
- Configuration in **XML** files
  - LHCbNightlyConf
- Machines that tests are currently running on
  - lblhcbpr1 with CC7 dedicated for timing tests (single executor in Jenkins), label: perf-centos7-timing
  - lblhcbpr4 with CC7 (8 executors), labels: perf-centos7, perf
  - volhcb05 with SLC6 (8 executors), labels: perf-slc6, perf
  - hltpref-quanta01-e52630v4 for HLT throughput test
  - lbhltpref01 for upgrade tests
Infrastructure

• Results of the tests parsed by the specific handlers
  ○ LHCbPR2HD
  ○ to save relevant metrics (int, float, string, ROOT, JSON)

• Zip file sent to the database through Dirac Storage Element
  /lhcb/prdata/zips

• Web front-end lblhcbpr.cern.ch
  ○ LHCbPR2BE
  ○ generic ROOT files viewer
  ○ trend analysis
  ○ custom modules

• Flexibility to push the results as HTML to EOS (at the level of LHCbPR2HD)
  ○ HLT rate and throughput tests
  ○ upgrade benchmarking
Infrastructure

- Results of the tests parsed by the specific handlers
  - LHCbPR2HD
  - to save relevant metrics (int, float, string, ROOT, JSON)

- Zip file sent to the database through Dirac Storage Element /lhcb/prdata/zips
  - LHCbPR2BE

Maciej Szymański (UCAS)
12th hackathon on software for the upgrade, Kraków, 25 Jun 2018
Infrastructure

- Results of the tests **parsed by the specific handlers**
  - LHCbPR2HD
  - to save relevant metrics (int, float, string, ROOT, JSON)

- Zip file sent to the database through **Dirac Storage Element**
  - lhcb/prdata/zips
  - LHCbPR2BE

- Flexibility to push the results as HTML to EOS (at the level of LHCbPR2HD)
  - HLT rate and throughput tests
  - upgrade benchmarking
Infrastructure

- Results of the tests **parsed by the specific handlers**
  - LHCbPR2HD
  - to save relevant metrics (int, float, string, ROOT, JSON)
- Zip file sent to the database through **Dirac Storage Element** `/lhcb/prdata/zips`
  - LHCbPR2BE
- **Web front-end** [lblhcbpr.cern.ch](http://lblhcbpr.cern.ch)
  - LHCbPR2FE
  - generic ROOT files viewer
  - trend analysis
  - custom modules

Maciej Szymański (UCAS)
12th hackathon on software for the upgrade, Kraków, 25 Jun 2018
Infrastructure

- Results of the tests **parsed by the specific handlers**
  - LHCbPR2HD
  - to save relevant metrics (int, float, string, ROOT, JSON)
- Zip file sent to the database through **Dirac Storage Element**
  - /lhcb/prdata/zips
  - LHCbPR2BE
- **Web front-end** (lblhcbpr.cern.ch)
  - LHCbPR2FE
  - generic ROOT files viewer
  - trend analysis
  - custom modules
- Flexibility to push the results as HTML to EOS (at the level of LHCbPR2HD)
  - HLT rate and throughput tests
  - upgrade benchmarking
Some statistics as of today

- 5 applications
  - Brunel
  - Gauss
  - Geant4
  - Moore
  - MooreOnline

- 67 option files

- 128 tests (running on several slots and platforms)

- ~50 tests daily

- ~20 tests dedicated for timing (thus running only on lblhcbpr1 machine)

- Single tests run from several minutes up to 10 hours
How to take part

- Prepare the **options file** for the test and commit to e.g. PRConfig
How to take part

- Prepare the **options file** for the test and commit to e.g. PRConfig
- Specify the **command** to run
  - see whether it’s already defined: [lblcbpr.cern.ch/api/executables](lblcbpr.cern.ch/api/executables), if not, we’ll add it
    
    ```json
    "name": "lb-run-gaudirun",
    "content": "lb-run -c {platform} --user-area={build} {app_name}/{app_version} gaudirun.py {options}"
    
    "name": "lb-run-callgrind",
    "content": "( lb-run -c {platform} --user-area={build} {app_name}/{app_version} gaudirun.py 
    --printsequence {options} ; lb-run -c {platform} --user-area={build} 
    {app_name}/{app_version} valgrind --tool=callgrind --dump-instr=yes 
    --Instr-atstart=no --cache-sim=yes --branch-sim=yes python 
    $(lb-run -c {platform} --user-area={build} {app_name}/{app_version} 
    which gaudirun.py) {options} )"
    
    "name": "perf-lb-run-gaudirun",
    "content": "( perf record --call-graph=lbr -o perf.log lb-run -c {platform} 
    --user-area={build} {app_name}/{app_version} gaudirun.py {options} ; 
    perf report -i perf.log > perf.lbr.txt )"
    ```
How to take part

• Prepare the **options file** for the test and commit to e.g. PRConfig

• Specify the **command** to run
  ○ see whether it’s already defined: lbhcpr.cern.ch/api/executables, if not, we’ll add it

```json
"name": "lb-run-gaudirun",
"content": "lb-run -c {platform} --user-area={build} {app_name}/{app_version}
gaudirun.py {options}"

"name": "lb-run-callgrind",
"content": "( lb-run -c {platform} --user-area={build} {app_name}/{app_version} gaudirun.py
 --printsequence {options} ; lb-run -c {platform} --user-area={build}
{app_name}/{app_version} valgrind --tool=callgrind --dump-instr=yes
--instr-atstart=no --cache-sim=yes --branch-sim=yes python
$(lb-run -c {platform} --user-area={build} {app_name}/{app_version}
which gaudirun.py) {options} )"

"name": "perf-lb-run-gaudirun",
"content": "( perf record --call-graph=lbr -o perf.log lb-run -c {platform}
--user-area={build} {app_name}/{app_version} gaudirun.py {options} ;
perf report -i perf.log > perf.lbr.txt )"
```

• Create the **handler** to parse the output
  ○ many handlers already there, e.g. to parse TimingAuditor, output of perf, etc.
  ○ see README on: LHCbPR2HD
How to take part

• **Schedule** your test in [LHCbNightlyConf](lblhcbpr.cern.ch/api/options)
How to take part

- **Schedule** your test in [LHCbNightlyConf](lblhcbpr.cern.ch/api/options)

```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>
```
How to take part

- **Schedule** your test in LHCbNightlyConf

```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)
How to take part

• **Schedule** your test in [LHCbNightlyConf](lblhcbpr.cern.ch/api/options)

```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="1b-run-gaudirun|TimelineHandler"/>
  <os_label>perf</os_label>
  <count>5</count>
</periodictest>
```
How to take part

- **Schedule** your test in LHCbNightlyConf

```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: lblhcbpr.cern.ch/api/options
- Executable and handler
How to take part

- **Schedule** your test in LHCbNightlyConf

```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 `lblhcbpr.cern.ch/api/options`
- Executable and handler
- Machine label
How to use LHCbPR

- Tests will automatically start on a day given by schedule (if the nightly build is ok)
- You will see PR link in the nightlies dashboard, close to the project name in the given slot

Watch the tests being executed in dashboard
- colour code: running tests, successful tests, failed tests, tests which have been executed with success, but the handler failed
- URL to log files of the test, handler and output of the Jenkins job

You can launch the test yourself from the dashboard
- e.g. to the test the handler
- click on Start new periodic test button (available after login)

To see the results of the test, by default you can use generic trend analysis and ROOT file viewer on lblhcbpr.cern.ch

(Optionally) create custom analysis module LHCbPR2FE
- or re-use existing one...

Maciej Szymański (UCAS)
12th hackathon on software for the upgrade, Kraków, 25 Jun 2018
How to use LHCbPR

• Tests will automatically start on a day given by schedule (if the nightly build is ok)
• You will see PR link in the nightlies dashboard, close to the project name in the given slot
• Watch the tests being executed in dashboard
  ◦ colour code: running tests, successful tests, failed tests, tests which have been executed with success, but the handler failed
  ◦ URL to log files of the test, handler and output of the Jenkins job

Maciej Szymański (UCAS)
12th hackathon on software for the upgrade, Kraków, 25 Jun 2018
How to use LHCbPR

• Tests will automatically start on a day given by schedule (if the nightly build is ok)
• You will see PR link in the nightlies dashboard, close to the project name in the given slot

• Watch the tests being executed in dashboard
  ◦ colour code: running tests, successful tests, failed tests, tests which have been executed with success, but the handler failed
  ◦ URL to log files of the test, handler and output of the Jenkins job

• You can launch the test yourself from the dashboard
  ◦ e.g. to the test the handler
  ◦ click on Start new periodic test button (available after login)
How to use LHCbPR

- Tests will automatically start on a day given by schedule (if the nightly build is ok)
- You will see PR link in the nightlies dashboard, close to the project name in the given slot
- Watch the tests being executed in dashboard
  - colour code: running tests, successful tests, failed tests, tests which have been executed with success, but the handler failed
  - URL to log files of the test, handler and output of the Jenkins job
- You can launch the test yourself from the dashboard
  - e.g. to the test the handler
  - click on Start new periodic test button (available after login)
- To see the results of the test, by default you can use generic trend analysis and ROOT file viewer on lblhcbpr.cern.ch
How to use LHCbPR

• Tests will automatically start on a day given by schedule (if the nightly build is ok)
• You will see PR link in the nightlies dashboard, close to the project name in the given slot
• Watch the tests being executed in dashboard
  ○ colour code: running tests, successful tests, failed tests, tests which have been executed with success, but the handler failed
  ○ URL to log files of the test, handler and output of the Jenkins job
• You can launch the test yourself from the dashboard
  ○ e.g. to the test the handler
  ○ click on Start new periodic test button (available after login)
• To see the results of the test, by default you can use generic trend analysis and ROOT file viewer on lblhcbpr.cern.ch
• (Optionally) create custom analysis module LHCbPR2FE
  ○ or re-use existing one ...
Trend module with predefined parameters

Plot last 10 measurements for a given algorithm

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>8176</td>
<td>min_RichPhotonRecoDown</td>
</tr>
<tr>
<td>8177</td>
<td>max_RichPhotonRecoDown</td>
</tr>
<tr>
<td>8178</td>
<td>mean_RichPhotonRecoDown</td>
</tr>
<tr>
<td>8179</td>
<td>sigma_RichPhotonRecoDown</td>
</tr>
<tr>
<td>8180</td>
<td>min_RichTrackGloPointsDown</td>
</tr>
<tr>
<td>8181</td>
<td>max_RichTrackGloPointsDown</td>
</tr>
<tr>
<td>8182</td>
<td>mean_RichTrackGloPointsDown</td>
</tr>
<tr>
<td>8183</td>
<td>sigma_RichTrackGloPointsDown</td>
</tr>
<tr>
<td>8184</td>
<td>min_PrStoreFTHt</td>
</tr>
<tr>
<td>8185</td>
<td>max_PrStoreFTHt</td>
</tr>
</tbody>
</table>

Filter Attributes

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>BRUNEL (9521)</td>
</tr>
<tr>
<td>Options</td>
<td>MiniBrunel (1559)</td>
</tr>
<tr>
<td>Executables</td>
<td>lib-run-gaudi/run (426)</td>
</tr>
<tr>
<td></td>
<td>lib-run-callgrid (674)</td>
</tr>
<tr>
<td></td>
<td>lib-run-gaudi-run-perconfig (4424)</td>
</tr>
<tr>
<td></td>
<td>lib-run-gaudi-run-perconfig: no-auto-override (92)</td>
</tr>
<tr>
<td></td>
<td>gaudiun (1)</td>
</tr>
<tr>
<td>Platforms</td>
<td>x86_64-centos7-gcc62-opt (4849)</td>
</tr>
<tr>
<td>Hosts</td>
<td>libhcbpr1.cern.ch (3515)</td>
</tr>
<tr>
<td></td>
<td>libhcbpr3.cern.ch (3599)</td>
</tr>
<tr>
<td></td>
<td>libhcbpr4.cern.ch (603)</td>
</tr>
<tr>
<td></td>
<td>public10 (1)</td>
</tr>
<tr>
<td></td>
<td>vohcbs05.cern.ch (3803)</td>
</tr>
<tr>
<td>Versions</td>
<td>Show Nightly versions</td>
</tr>
<tr>
<td></td>
<td>Number of nightly versions to show: 10</td>
</tr>
</tbody>
</table>

Maciej Szymański (UCAS)

12th hackathon on software for the upgrade, Kraków, 25 Jun 2018
Using trend analysis

Plot the time spent by EVENT LOOP in Brunel test as a function of the software version

go to Trends/Trends tab
Using trend analysis

Plot the time spent by EVENT LOOP in Brunel test as a function of the software version

1. Go to the Trends/Trends tab.
2. Select Brunel from the list of applications.
3. Choose versions.
4. Type the name of the algorithm.
Using trend analysis

Plot the time spent by EVENT LOOP in Brunel test as a function of the software version

- Go to the Trends/Trends tab.
- Select Brunel from the list of applications.
- Select the option, platform and host you are interested in.
- Choose versions.
- Type the name of the algorithm.

Select the option, platform and host you are interested in.
Using trend analysis

Plot the time spent by EVENT LOOP in Brunel test as a function of the software version

- Go to Trends/Trends tab
- Select Brunel from the list of applications
- Select the option, platform and host you are interested in

Choose versions

Maciej Szymański (UCAS)
12th hackathon on software for the upgrade, Kraków, 25 Jun 2018
Using trend analysis

Plot the time spent by EVENT LOOP in Brunel test as a function of the software version

type the name of the algorithm
Using trend analysis

Plot the time spent by EVENT LOOP in Brunel test as a function of the software version

![Graph of EVENT_LOOP time spent as a function of software version]
Event loop timing in Gauss vs. SW version

- Gauss164 (RICH fast options): (61.7±2.3)s
- Gauss169 (new tag): (53.5±1.2)s
- Gauss169 (new tag): (40.3±0.8)s
Using ROOT file viewer

1. Go to LHCbPR Jobs/ROOT file viewer tab.
2. Select applications, filter options, etc.
3. Select jobs to analyze and click on "Analyze".
4. Click on "Select plots".
5. Select your histogram.
6. Click on "Plot selected".
7. Choose the way of display.
Using ROOT file viewer

- Go to LHCbPR Jobs/ROOT file viewer tab
- Select applications, filter options etc
- Select jobs to analyze and click Analyze
- Click on Select plots
- Select your histogram
- Click on Plot selected
- Choose the way of display
Using ROOT file viewer

select jobs to analyze and click Analyze
Using ROOT file viewer

- Go to LHCbPR Jobs/ROOT file viewer tab
- Select applications, filter options etc.
- Select jobs to analyze and click Analyze
- Click on Select plots
- Select your histogram
- Click on Plot selected
- Choose the way of display
Using ROOT file viewer

1. Go to LHCbPR Jobs/ROOT file viewer tab.
2. Select applications, filter options etc. and select jobs to analyze. Click Analyze.
3. Click on Select plots.
4. Select your histogram.
5. Click on Plot selected.
6. Choose the way of display.

Select your histogram.
Using ROOT file viewer

1. Go to LHCbPR Jobs/ROOT file viewer tab
2. Select applications, filter options etc.
3. Select jobs to analyze and click Analyze
4. Click on Select plots
5. Select your histogram
6. Click on Plot selected

Maciej Szymański (UCAS)
12th hackathon on software for the upgrade, Kraków, 25 Jun 2018
Using ROOT file viewer

- Go to LHCbPR Jobs/ROOT file viewer tab
- Select applications, filter options etc
- Select jobs to analyze and click Analyze
- Click on Select plots
- Select your histogram
- Click on Plot selected

Choose the way of display
Upgrade benchmarking

- Special case for the upgrade studies

- Tests results pushed to eos website: https://cern.ch/lhcbpr-hlt/PerfTests/UpgradeVelo

- Throughput test (Velo only and best performance)

- PrChecker (Velo only)

- Running on lhcb-tdr-test (HEAD of everything)

- Notifications when results available sent to mattermost private channel (please let us know if you’re interested)

- More tests to come
Perf test for Gauss

• Linux tool integrated in the kernel to (see tutorial by Hadrien):
  ◦ measure CPU statistics
  ◦ investigate where the program spends time
  ◦ produce call graphs

• Test runs with command:
  `perf record --call-graph=lbr -o perf.log lb-run -c {platform}
  --user-area={build} {app_name}/{app_version} gaudirun.py {options}
  perf report -i perf.log > perf.lbr.txt`

• Using lhcb-sim09-upgrade and
  `$PRCONFIGOPTS/Gauss/PRTEST-2016-SIM-P8-10000000-100evts.py`

• Scheduled to run weekly in LHCbPR

• Example of results:
Flame Graph

Function: G4SteppingManager::InvokePostStepDoItProcs (1,783,175 samples, 13.06%)
Igprof test for Gauss

- Performance and memory profiling tool, here just focus on:
  - memory that hasn’t be freed
  - total memory allocated by any function

- Test runs with command:
  
  igprof_env="lb-run -c {platform} --ext igprof --user-area={build} {app_name}/{app_version}"
  
  $igprof_env igprof -d -mp -z -o igout.mp.gz gaudirun.py {options}
  
  $igprof_env igprof-analyse -v -g -r MEM_LIVE igout.mp.gz > igout.mp.live.txt
  
  $igprof_env igprof-analyse -v -g -r MEM_TOTAL igout.mp.gz > igout.mp.total.txt

- Using lhcb-sim09-upgrade and
  
  $PRCONFIGOPTS/Gauss/PRTEST-2016-SIM-P8-10000000-100evts.py

- Scheduled to run weekly in LHCbPR

- Example of the results:
  
Big data tools for LHCbPR

- Data from tests pushed to Hadoop Distributed File System
- Possibility to use Spark connector in SWAN notebooks to read those data
- Interactive exploration, collaboration on tests reports
- Needs to belong to the ai-hadoop-users e-group

- MySQL DBoD
- LHCbPR
- EOS
- LHCbPR2FE
- batch ingestion in Parquet format
- Apache Spark
- HDFS
- SWAN
- Apache Zeppelin
- Jupyter
- currently in production

Maciej Szymański (UCAS)
12th hackathon on software for the upgrade, Kraków, 25 Jun 2018
import matplotlib.pyplot as plt
x_val = [x[0] for x in reversed(results.collect())]
y_val = [float(x[1]) for x in reversed(results_unc.collect())]
yerr = [float(x[1]) for x in reversed(results_unc.collect())]
print(x_val)
print(y_val)
print(yerr)
plt.xticks(list(range(len(x_val))), x_val)
plt.xticks(rotation=90)
plt.xlim([-1, len(x_val)])
plt.errorbar(range(len(y_val)), y_val, yerr=yerr, fmt='o')
plt.show()
Thank you!
Please give feedback on our mattermost channel.
LHCbPR - Resources

- Web application:
  https://lblhcbpr.cern.ch
  https://lblhcbpr.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/lhcb-core/LHCbPR2

- Jenkins configuration
  https://gitlab.cern.ch/lhcb-core/LbNightlyTools

- Configuration of the periodic tests
  https://gitlab.cern.ch/lhcb-core/LHCbNightlyConf/