LbMCSSubmit: A new flexible and scalable request submission system for LHCb simulation

Chris Burr on behalf of the LHCb Computing and Simulation projects

9th May 2023
➤ Current simulation requests have to span from 2011 to future upgrades

➤ Wide variety of needs - varied physics program and detector studies
  ➤ Flavour physics analyses typically need many specific samples
  ➤ Typically not much overlap between analyses

➤ Massive compute requirements
  ➤ An increasing number of “fast” simulation options available

➤ Thousands of “basic” configurations (without considering decays)
  ➤ Multiply by many thousands of “event types”
  ➤ Effectively infinite possible customisations

How can we handle this complexity without making things complicated?
Simulated LHCb data is typically processed by the full processing chain

- Typically ~8 separate applications to run
- Lots of commonality but not trivial to isolate

Representative of Run 1+2 proton-proton running, exact dataflow varies by year and analyst needs
Web application for submitting productions
- Based around “models” which are prepared by experts
- A model is made up of a series of “steps” (correspond to an executable + arguments)
- Liaisons* clone the model and make small edits for specific productions

Worked well for many years but shows problems
- Large maintenance burden on already busy experts
- Lack of frontend developer expertise for improvements

*Human contact between the simulation project and a specific area of the LHCb physics programme
What do we actually want?

- 95%+ of simulation requests can be summarised in a few words:

I want 100,000 events per magnet polarity of

\[
\begin{align*}
D^+ & \rightarrow K^- \ pi^+ \ pi^+ \\
D^{*+} & \rightarrow D^0 \rightarrow (\pi^+ \ pi^-) \ pi^- 
\end{align*}
\]

in 2012 and 2016 proton-proton conditions

- Should result in 8 requests being created in LHCbDIRAC (the LHCb grid middleware)
  - 2 magnet polarities × 2 data taking years × 2 event types

- What does LHCbDIRAC actually need to know?
What does LHCbDIRAC need to know?
➤ That’s a lot...

➤ Undesirable to move abstractions into LHCbDIRAC itself
  ➤ Experts regularly need to manually provide that information for the O(%) special cases
  ➤ LHCbDIRAC developers are disjoint from the Simulation experts

➤ LHCbDIRAC and Simulation are very small teams (≤10 people), enables:
  ➤ Simulation experts to focus on simulation
  ➤ LHCbDIRAC experts to focus on grid/production activities
  ➤ Analysts to focus on analysis

➤ Instead, start from a minimal YAML file which can be written by anyone
Previous YAML file is “stage 0”

Apply a series of transformations to add more information at each stage

Eventually get “stage 6” which can be directly executed/submitted

Transformation between stages is performed using a Python package \( \text{LbMCS\text{Submit}} \)

Benefits:
- **Safe**: Non-expert users can only modify things which are deemed suitable
- **Maintainable**: Python expertise common in LHCb
- **Flexible**: Experts can edit any stage directly to handle special requests
Very LHCb specific, targeted around how LHCb evolved in the last 15 years

At a high level:
1. Include defaults for optional parameters
2. Convert to a list of requests
3. Flatten each request into the final list of distinct requests
4. Apply conditions specific information (beam info, geometry, trigger config)
5. Apply special fixes and sanity checks
6. Convert the high level configuration into a format LHCbDIRAC can execute

Actual example: [https://gitlab.cern.ch/lhcb-simulation/lbmcsubmit/-/blob/main/docs/transformations.md](https://gitlab.cern.ch/lhcb-simulation/lbmcsubmit/-/blob/main/docs/transformations.md)
▷ At this point the request is the same as the legacy system
  ▷ Can we do better? - Take inspiration from LHCb’s Analysis Productions*

▷ Use GitLab to submit the request to LHCbDIRAC
  ▷ CI can run test jobs for each production
  ▷ Catch bugs before submission
  ▷ Opportunity for review prior to submission
  ▷ Expose estimated resource usage to requesters

▷ Complements the existing testing procedure
  1. CI checks if the configuration works prior to submission
  2. LHCbDIRAC testing phase generates more events to catch rarer issues
  3. Full production launches

*Analysis Productions: A declarative approach to ntupling (9 May 2023, 17:45)
Overall summary

Jobs (12 total)

Test job statistics

<table>
<thead>
<tr>
<th>Status</th>
<th>Job Name</th>
<th>Generated</th>
<th>Stored</th>
<th>Size</th>
<th>Requested</th>
<th>Generated</th>
<th>Disk</th>
<th>CPU years</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUCCESS</td>
<td>request-submit</td>
<td>10</td>
<td>2.3</td>
<td>MB</td>
<td>500,000</td>
<td>500,000</td>
<td>15.63</td>
<td>12</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>request-submit</td>
<td>10</td>
<td>2.85</td>
<td>MB</td>
<td>500,000</td>
<td>500,000</td>
<td>122.61</td>
<td>22</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>request-submit</td>
<td>10</td>
<td>3.69</td>
<td>MB</td>
<td>500,000</td>
<td>500,000</td>
<td>126.16</td>
<td>23</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>request-submit</td>
<td>10</td>
<td>2.59</td>
<td>MB</td>
<td>500,000</td>
<td>500,000</td>
<td>125.66</td>
<td>22</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>request-submit</td>
<td>10</td>
<td>2.39</td>
<td>MB</td>
<td>500,000</td>
<td>500,000</td>
<td>118.16</td>
<td>27</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>request-submit</td>
<td>10</td>
<td>2.84</td>
<td>MB</td>
<td>500,000</td>
<td>500,000</td>
<td>151.06</td>
<td>26</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>request-submit</td>
<td>10</td>
<td>2.39</td>
<td>MB</td>
<td>500,000</td>
<td>500,000</td>
<td>156.54</td>
<td>22</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>request-submit</td>
<td>10</td>
<td>2.45</td>
<td>MB</td>
<td>500,000</td>
<td>500,000</td>
<td>123.27</td>
<td>22</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>request-submit</td>
<td>10</td>
<td>2.77</td>
<td>MB</td>
<td>500,000</td>
<td>500,000</td>
<td>100.47</td>
<td>18</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>request-submit</td>
<td>10</td>
<td>3.61</td>
<td>MB</td>
<td>500,000</td>
<td>500,000</td>
<td>150.01</td>
<td>18</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>request-submit</td>
<td>10</td>
<td>2.57</td>
<td>MB</td>
<td>500,000</td>
<td>500,000</td>
<td>103.28</td>
<td>6</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>request-submit</td>
<td>10</td>
<td>1.73</td>
<td>MB</td>
<td>500,000</td>
<td>500,000</td>
<td>68.52</td>
<td>17</td>
</tr>
</tbody>
</table>

Can view detailed information about each test
Recently replaced the LHCb Simulation requests system

Essential to manage the complexity of supporting over a decade of data taking
  ▶ Frees up expert time so they can focus on more impactful work

Especially valuable non-mainstream areas of the physics programme
  ▶ No need to prioritise which request “models” get updated frequency

Simpler, safer, more flexible - pick three!
Questions?