Continuous Integration Service at Fermilab

Vito Di Benedetto for CI Project
Vladimir Podstavkov, Michele Fattoruso, Bruno Coimbra
Fermilab

CHEP 2018 July 12, 2018
Why we care about the Continuous Integration

- All developers write code following good practice... and they test their code before committing it
Why we care about the Continuous Integration

- All developers write code following good practice... and they test their code before committing it
  - but sometime the code:
    - blows up at build time
    - crashes at run time
    - when used in production trashes down computing resources
All developers write code following good practice and they test their code before committing it. But sometime the code:  
• blows up at build time.

Yeah, in the meantime module “A” has been updated to vX.Y and my code doesn’t fit anymore.
Why we care about the Continuous Integration

- All developers write code following good practice... and they test their code before committing it
  - I need more info for my analysis
  - I need to use this package to get a more accurate simulation
Why we care about the Continuous Integration

- All developers write code following good practice... and they test their code before committing it
  - I need more info for my analysis
  - I need to use this package to get a more accurate simulation

the memory footprint doesn’t fit my system anymore
Why we care about the Continuous Integration

- All developers write code following good practice... and they test their code before committing it

---

the memory footprint doesn’t fit my system anymore
Why we care about the Continuous Integration

- All developers write code following good practice... and they test their code before committing it
  - A good testing strategy that provides rapid feedback helps to identify defects introduced by code changes quickly
  - Issues detected early on in development are typically smaller, less complex and easier to resolve
Why we care about the Continuous Integration

- All developers write code following good practice... and they test their code before committing it
  - A good testing strategy that provides rapid feedback helps to identify defects introduced by code changes quickly
  - Issues detected early on in development are typically smaller, less complex and easier to resolve

...and physicist can spend more time on analysis
FIFE CI Goals

- Provide a common framework for all FIFE experiments and projects to build and test their software.
- Offer an access to comprehensive testing from unit tests to physics validation through regression test.
- Allow to trace code revision and resource usage stats for every build.
- CI builds can be triggered:
  - on demand by users
  - by commit in monitored repositories
  - periodically (cron)
- Provide UI that clearly indicates issues with current release
- Enable automatic notification in case of failures.
CI: It is Complicated ...

- The CI is used by a wide variety of experiments and projects

- Build systems:
  - CMake
  - Make
  - Custom one (MRB, SRT, scons)

- Version Control Systems:
  - CVS
  - SVN
  - GIT
  - Mercurial

- Repository is hosted:
  - Fermilab Redmine and/or elsewhere

- Code is tested on:
  - SL6/7
  - Mac OSX
CI Architecture

- Fermilab Central Build Service based on Jenkins
- Grid
- CI web server
- Database Service

Users: exp1 user, exp2 user, exp3 user
CI Architecture: Build service

Build server

Jenkins server app

CI scripts run on selected Jenkins slaves

CI build triggered:
- on demand by user
- by commit to repository
- by a cronjob

CI web server

CI web server app

Build slaves

SLF6
- Jenkins client

SLF7
- Jenkins client

OSX Sierra
- Jenkins client

OSX High Sierra
- Jenkins client

CI scripts execute CI phases defined by CI workflow

CI scripts collect info from exp code and push to CI DB trough the CI web server

Grid

DataBase Service

exp1 user

exp2 user

exp3 user
CI Architecture: Build service

Fermilab Central Build Service based on Jenkins

- CI scripts run on selected Jenkins slaves
- Build slaves
  - SLF6
    - Jenkins client
  - SLF7
    - Jenkins client
  - OSX Sierra
    - Jenkins client
  - OSX High Sierra
    - Jenkins client

CI web server

Grid

For CI Validation builds the CI script submits jobs for the validation workflow

DataBase Service

CI build triggered:
- on demand by user
- by commit to repository
- by a cronjob

exp2 user
exp1 user
exp3 user
CI Architecture: CI web server

Fermilab Central Build Service based on Jenkins

Grid

CI web server

server1
CI web app exp1
CI web app exp2

server2
CI web app exp1
CI web app exp2
CI web app exp3
redirector

DataBase Service

exp2 user
exp1 user
exp3 user
The workflow configuration allow the user to run an “almost” arbitrary procedure.

Workflow consists of multiple steps (phases) executed sequentially:
- env setup
- checkout
- build
- unit tests
- install
- regression tests
- validation tests

During some phases tasks could be executed in parallel.

For each phase users can define:
- initialization and finalizing procedures
- work to do
- logs to report
- the format of the report generated from the logs
CI dashboard overview

**CI dashboard used by LArSoft and LArSoft-based experiment code**

Multiplatform Continuous Integration for LarCI

---

**Platform**

- LArSoft
- ArgoNeuT
- DUNE
- LArIAT
- uBooNE
- SBND

**Builds**

- `dune_c/020 (LArSoft DUNE)`
- `dune_c/819 (LArSoft DUNE)`
- `dune_c/816 (LArSoft DUNE)`
- `dune_c/817 (LArSoft DUNE)`
- `dune_c/816 (LArSoft DUNE)`
- `lar_CI/2706 (DUNE)`
- `lar_CI/2705 (DUNE)`

**Build Types**

- LArSoft
- DUNE

**Build Details**

- **Start Time**: 2018-06-26 08:51:45
- **Build Type**: `dune_c/020 (LArSoft DUNE)`
- **Setup Environment**: `all6 e15prof`
- **Checkout**: ✔
- **Build**: ✔
- **Unit Test**: ✔
- **Install**: ✔
- **CI Tests**: ✔

---

**Fermilab**
This panel allows to filter CI builds using different criteria.

### Multiplatform Continuous Integration for LarCI

<table>
<thead>
<tr>
<th>Build</th>
<th>Start Time</th>
<th>Build Type</th>
<th>setup_environment</th>
<th>checkout</th>
<th>build</th>
<th>unit_test</th>
<th>install</th>
<th>gen_ref_files</th>
<th>ci_tests</th>
<th>ci_validation</th>
<th>update_re</th>
</tr>
</thead>
<tbody>
<tr>
<td>dune_c1/820 (LArSoft DUNE)</td>
<td>2018-06-26 08:51:45</td>
<td>al/6 e15prof</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>dune_c1/819 (LArSoft DUNE)</td>
<td>2018-06-26 08:50:09</td>
<td>al/6 e15prof</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>dune_c1/818 (LArSoft DUNE)</td>
<td>2018-06-26 08:19:28</td>
<td>dt/6 c2:prof</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>dune_c1/817 (LArSoft DUNE)</td>
<td>2018-06-26 08:15:10</td>
<td>al/6 e15prof</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>dune_c1/816 (LArSoft DUNE)</td>
<td>2018-06-26 08:13:29</td>
<td>al/6 e15prof</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>dune_c1/815 (LArSoft DUNE)</td>
<td>2018-06-25 15:40:47</td>
<td>dt/6 c2:prof</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>dune_c1/814 (LArSoft DUNE)</td>
<td>2018-06-25 15:37:32</td>
<td>al/6 e15prof</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>dune_c1/813 (LArSoft DUNE)</td>
<td>2018-06-25 15:36:35</td>
<td>al/6 e15prof</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
CI dashboard details

(? markers give access do wiki documentation

Multiplatform Continuous Integration for LarCI

<table>
<thead>
<tr>
<th>Build</th>
<th>Start Time</th>
<th>Build Type</th>
<th>setup_environment</th>
<th>checkout</th>
<th>build</th>
<th>unit_test</th>
<th>install</th>
<th>gen_ref_files</th>
<th>ci_tests</th>
<th>update_re</th>
</tr>
</thead>
<tbody>
<tr>
<td>dune_ci/020 (LarSoft DUNE)</td>
<td>2018-06-26 08:51:45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dune_ci/019 (LarSoft DUNE)</td>
<td>2018-06-26 08:50:09</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dune_ci/016 (LarSoft DUNE)</td>
<td>2018-06-26 08:19:28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dune_ci/017 (LarSoft DUNE)</td>
<td>2018-06-26 08:15:10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dune_ci/016 (LarSoft DUNE)</td>
<td>2018-06-26 08:13:29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lar_ci/2706 (DUNE)</td>
<td>2018-06-26 02:58:37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lar_ci/2705 (DUNE)</td>
<td>2018-06-26 02:48:51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dune_ci/014 (LarSoft DUNE)</td>
<td>2018-06-25 15:40:47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CI dashboard details

CI Validation reports some failed jobs but validation results are OK

Multiplatform Continuous Integration for LarCI

<table>
<thead>
<tr>
<th>Build</th>
<th>Start Time</th>
<th>Build Type</th>
<th>setup_environment</th>
<th>checkout</th>
<th>build</th>
<th>unit_test</th>
<th>install</th>
<th>get-ref_files</th>
<th>ci_tests</th>
<th>ci_validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>dune_ci/820 (LArSoft DUNE)</td>
<td>2018-06-26 08:51:45</td>
<td>e15prom</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dune_ci/819 (LArSoft DUNE)</td>
<td>2018-06-26 08:50:09</td>
<td>e15prof</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dune_ci/818 (LArSoft DUNE)</td>
<td>2018-06-26 08:19:28</td>
<td>c2prof</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dune_ci/817 (LArSoft DUNE)</td>
<td>2018-06-26 08:15:10</td>
<td>e15prof</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dune_ci/816 (LArSoft DUNE)</td>
<td>2018-06-26 08:13:20</td>
<td>e15prof</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>lar_ci/2706 (DUNE)</td>
<td>2018-06-26 02:58:37</td>
<td>e15prof</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>lar_ci/2705 (DUNE)</td>
<td>2018-06-26 02:48:51</td>
<td>e15prof</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dune_ci/813 (LArSoft DUNE)</td>
<td>2018-06-25 15:36:35</td>
<td>e15prof</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Warning in regression tests notifies that some stage output is changed.
CI builds to update reference files

Multiplatform Continuous Integration for LarCI

<table>
<thead>
<tr>
<th>Build</th>
<th>Start Time</th>
<th>Build Type</th>
<th>setup_environment</th>
<th>checkout</th>
<th>build</th>
<th>unit_test</th>
<th>install</th>
<th>gen_ref_files</th>
<th>ci_tests</th>
<th>ci_validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>dune_ci/020 (LArSoft DUNE)</td>
<td>2018-06-26 08:51:45</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>dune_ci/819 (LArSoft DUNE)</td>
<td>2018-06-26 08:50:09</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>dune_ci/816 (LArSoft DUNE)</td>
<td>2018-06-26 08:19:28</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>dune_ci/381 (LArSoft DUNE)</td>
<td>2018-06-26 08:15:10</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>dune_ci/816 (LArSoft DUNE)</td>
<td>2018-06-26 08:13:29</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>lar_ci/2706 (DUNE)</td>
<td>2018-06-26 02:58:37</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>lar_ci/2705 (DUNE)</td>
<td>2018-06-26 02:48:51</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>dune_ci/813 (LArSoft DUNE)</td>
<td>2018-06-25 15:36:35</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td>✔</td>
<td></td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>
## CI (regression) tests status

**Phase:** `ci_tests`

<table>
<thead>
<tr>
<th>#</th>
<th>Test Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>ci_reco_regression_test_dunefd</code></td>
</tr>
<tr>
<td>2</td>
<td><code>ci_reco_regression_test_protoDUNE</code></td>
</tr>
<tr>
<td>3</td>
<td><code>ci_select_regression_test_dunefd</code></td>
</tr>
<tr>
<td>4</td>
<td><code>ci_g4_regression_test_dunefd</code></td>
</tr>
<tr>
<td>5</td>
<td><code>ci_g4_regression_test_protoDUNE</code></td>
</tr>
<tr>
<td>6</td>
<td><code>ci_g4_regression_test_protoDUNE_2</code></td>
</tr>
<tr>
<td>7</td>
<td><code>ci_reco_regression_test_protoDUNE_2</code></td>
</tr>
<tr>
<td>8</td>
<td><code>ci_detsim_regression_test_dunefd</code></td>
</tr>
<tr>
<td>9</td>
<td><code>ci_detsim_regression_test_protoDUNE</code></td>
</tr>
<tr>
<td>10</td>
<td><code>ci_detsim_regression_test_protoDUNE_2</code></td>
</tr>
<tr>
<td>11</td>
<td><code>ci_gen_regression_test_dunefd</code></td>
</tr>
<tr>
<td>12</td>
<td><code>ci_gen_regression_test_protoDUNE</code></td>
</tr>
<tr>
<td>13</td>
<td><code>ci_gen_regression_test_protoDUNE_2</code></td>
</tr>
<tr>
<td>14</td>
<td><code>ci_mergeana_regression_test_dunefd</code></td>
</tr>
<tr>
<td>15</td>
<td><code>ci_mergeana_regression_test_protoDUNE</code></td>
</tr>
</tbody>
</table>

- CI tests are sorted according to the status severity:
  - failure (red)
  - warning (orange)
  - successful (green)
CI (regression) tests status

Phase: ci_tests

- CI tests are sorted according to the status severity:
  - failure (red)
  - problem to run the code
  - in case of crash there is the backtrace extracted from the coredump

<table>
<thead>
<tr>
<th>#</th>
<th>Test Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ci_reco_regression_test_dunefd</td>
</tr>
<tr>
<td>2</td>
<td>ci_reco_regression_test_protoDUNE</td>
</tr>
<tr>
<td>3</td>
<td>ci_select_regression_test_dunefd</td>
</tr>
<tr>
<td>4</td>
<td>ci_g4_regression_test_dunefd</td>
</tr>
<tr>
<td>5</td>
<td>ci_g4_regression_test_protoDUNE</td>
</tr>
<tr>
<td>6</td>
<td>ci_g4_regression_test_protoDUNEdp</td>
</tr>
<tr>
<td>7</td>
<td>ci_reco_regression_test_protoDUNEdp</td>
</tr>
<tr>
<td>8</td>
<td>ci_detsim_regression_test_dunefd</td>
</tr>
<tr>
<td>9</td>
<td>ci_detsim_regression_test_protoDUNE</td>
</tr>
<tr>
<td>10</td>
<td>ci_detsim_regression_test_protoDUNEdp</td>
</tr>
<tr>
<td>11</td>
<td>ci_gen_regression_test_dunefd</td>
</tr>
<tr>
<td>12</td>
<td>ci_gen_regression_test_protoDUNE</td>
</tr>
<tr>
<td>13</td>
<td>ci_gen_regression_test_protoDUNEdp</td>
</tr>
<tr>
<td>14</td>
<td>ci_mergeana_regression_test_dunefd</td>
</tr>
<tr>
<td>15</td>
<td>ci_mergeana_regression_test_protoDUNE</td>
</tr>
</tbody>
</table>
## CI (regression) tests status

<table>
<thead>
<tr>
<th>Phase: ci_tests</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ci_testsDUNE.log</code></td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>Test Name</td>
</tr>
<tr>
<td>---</td>
<td>-----------</td>
</tr>
<tr>
<td>1</td>
<td><code>ci_reco_regression_test_dunefd</code></td>
</tr>
<tr>
<td>2</td>
<td><code>ci_reco_regression_test_protoDUNE</code></td>
</tr>
<tr>
<td>3</td>
<td><code>ci_select_regression_test_dunefd</code></td>
</tr>
<tr>
<td>4</td>
<td><code>ci_g4_regression_test_dunefd</code></td>
</tr>
<tr>
<td>5</td>
<td><code>ci_g4_regression_test_protoDUNE</code></td>
</tr>
<tr>
<td>6</td>
<td><code>ci_g4_regression_test_protoDUNEdp</code></td>
</tr>
<tr>
<td>7</td>
<td><code>ci_reco_regression_test_protoDUNEdp</code></td>
</tr>
<tr>
<td>8</td>
<td><code>ci_detsim_regression_test_dunefd</code></td>
</tr>
<tr>
<td>9</td>
<td><code>ci_detsim_regression_test_protoDUNE</code></td>
</tr>
<tr>
<td>10</td>
<td><code>ci_detsim_regression_test_protoDUNEdp</code></td>
</tr>
<tr>
<td>11</td>
<td><code>ci_gen_regression_test_dunefd</code></td>
</tr>
<tr>
<td>12</td>
<td><code>ci_gen_regression_test_protoDUNE</code></td>
</tr>
<tr>
<td>13</td>
<td><code>ci_gen_regression_test_protoDUNEdp</code></td>
</tr>
<tr>
<td>14</td>
<td><code>ci_mergeana_regression_test_dunefd</code></td>
</tr>
<tr>
<td>15</td>
<td><code>ci_mergeana_regression_test_protoDUNE</code></td>
</tr>
</tbody>
</table>

- **CI tests are sorted according to the status severity:**
  - failure (red)
  - warning (orange)
  - to code runs, but the output is different from the reference
CI (regression) tests: memory usage

By switching from CRY to CORSICA as cosmic shower generator, the memory usage jumped from ~2Gb to ~3.5Gb.

After memory optimization the memory usage went down to 1.2Gb.
CI Validation dashboard

**uBooNE Data/MC Cosmic validation v06_76_00**

<table>
<thead>
<tr>
<th>Build</th>
<th>Start Time</th>
<th>Build Type</th>
<th>sim_MC</th>
<th>reco_MC</th>
<th>mergeana_MC</th>
<th>validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>lar_ci2387</td>
<td>2018-05-15 03:57:02</td>
<td>sft6 e15:prof</td>
<td>⚠️</td>
<td>⚠️</td>
<td>⚠️</td>
<td>⚠️</td>
</tr>
</tbody>
</table>

**Number of events for each stage**

- **validation**: 0
- **sim_MC**: 11,900
- **reco_MC**: 10,550
- **mergeana_MC**: 0

**Phase: ci_validation**

- ![ci_validation](ci_validation_uBooNE.log)
- ![compare_MC](compare_MC_prevMC/)
- ![compare_MC](compare_MC_refMC/)

**Remarks**

- The markers give access to the stage job stats.
- Folders with validation plots.
- Links to time series for validation stats.
CI Validation dashboard

Folders with validation plots expanded

The stage marker gives access to job resource stats
The CI Validation reported a significant drop in the track reconstruction efficiency for one of the reconstruction algorithms.
Summary

● The FIFE CI service provides a comprehensive framework for FIFE experiments and projects to test and validate their offline production and analysis code on the supported platforms.
  ○ It allows to validate experiment code through grid jobs.
  ○ Provides a dashboard for easy access to logs and statistical graphs.

● Currently the CI service is in use by the ArgoNeuT, DUNE, g-2, LArIAT, MINERvA, mu2e, NOvA, SBND and uBooNE experiments and by the following projects: ART and LArSoft software suites, GENIE, GlideinWMS and Rivet.

● The service proved to be very useful to intercept issues in experiments’ code early on and get it fixed before running production jobs.

● Future development plans include:
  ○ enhanced physics validation features
  ○ support for code profiling
Thank you for your attention