



Continuous Integration Service at Fermilab

Vito Di Benedetto for CI Project

Vladimir Podstavkov, Michele Fattoruso, Bruno Coimbra
Fermilab

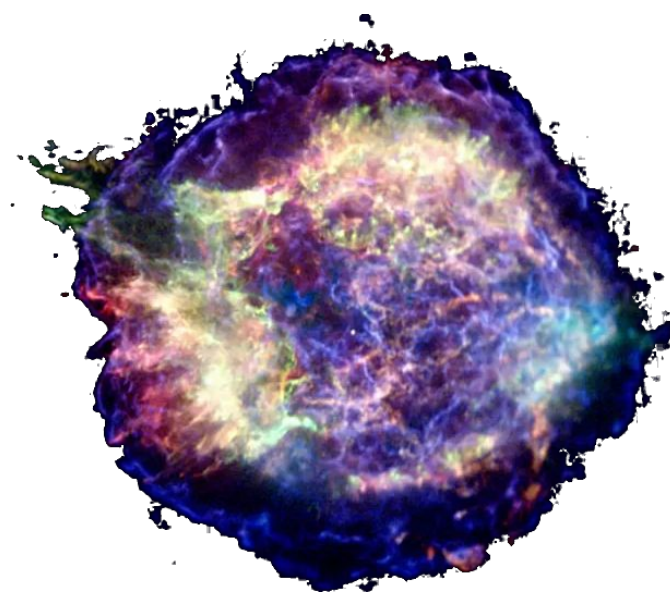
CHEP 2018 July 12, 2018

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



Why we care about the Continuous Integration

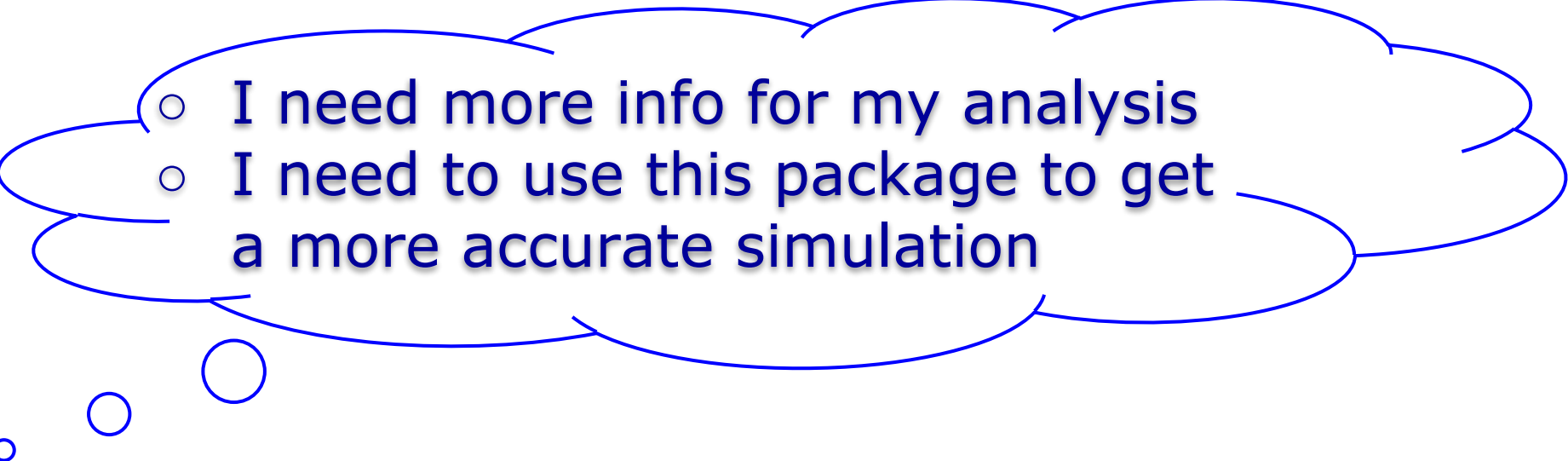
- All developers write code following good practice... and they test their code before committing it
 - but sometime the code:
 - blows up at build time

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

Select date range:

From: 12/01/2015

To: 05/01/2016

Update

Clear



the memory footprint
doesn't fit my system
anymore



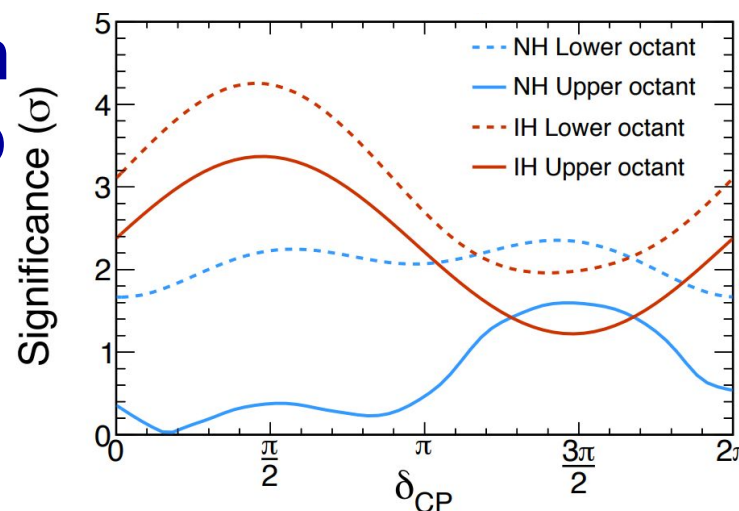
Why we care about the Continuous Integration

- 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

...and physicist
can spend more
time on analysis

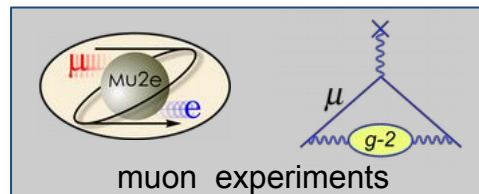


FIFE CI Goals

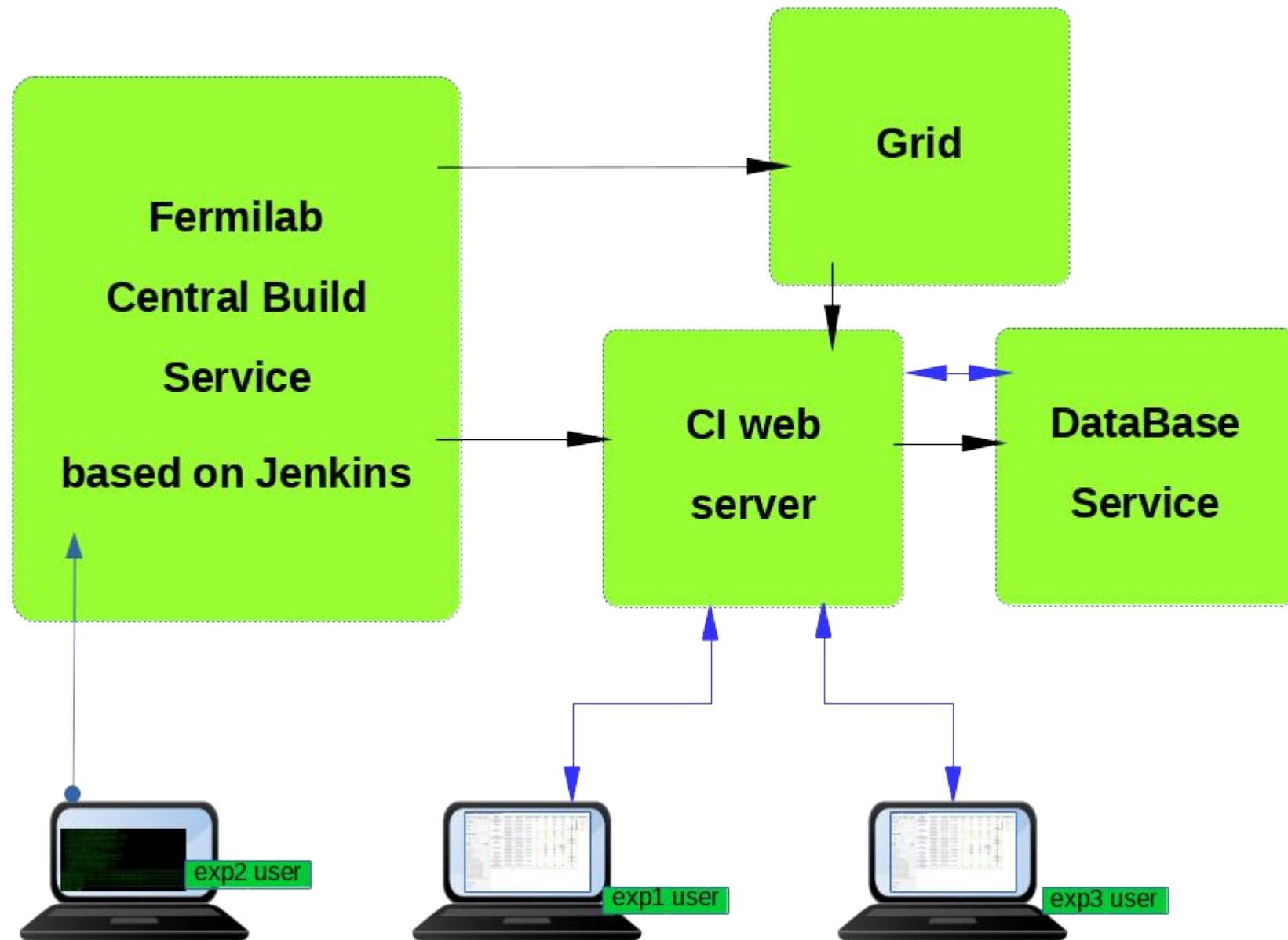
- Provide a common framework for all FIFE experiments and projects to build and test their software.
- Offer an access to comprehensive testing from unit tests to physics validation through regression test.
- Allow to trace code revision and resource usage stats for every build.
- CI builds can be triggered:
 - 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 ...

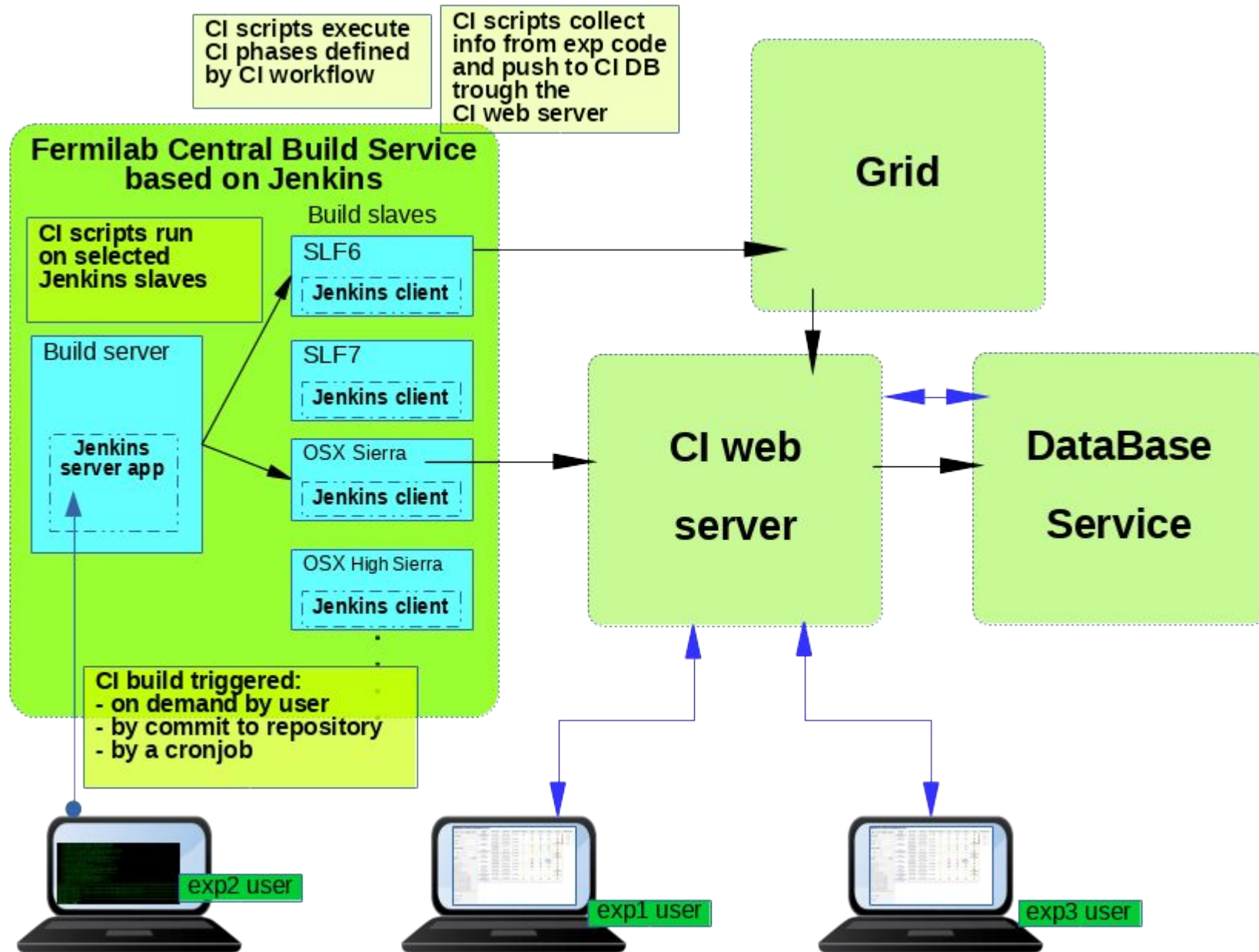
- The CI is used by a wide variety of experiments and projects
- Build systems:
 - CMake
 - Make
 - Custom one (MRB, SRT, scon)
- Version Control Systems:
 - CVS
 - SVN
 - GIT
 - Mercurial
- Repository is hosted:
 - Fermilab Redmine and/or elsewhere
- Code is tested on:
 - SL6/7
 - Mac OSX



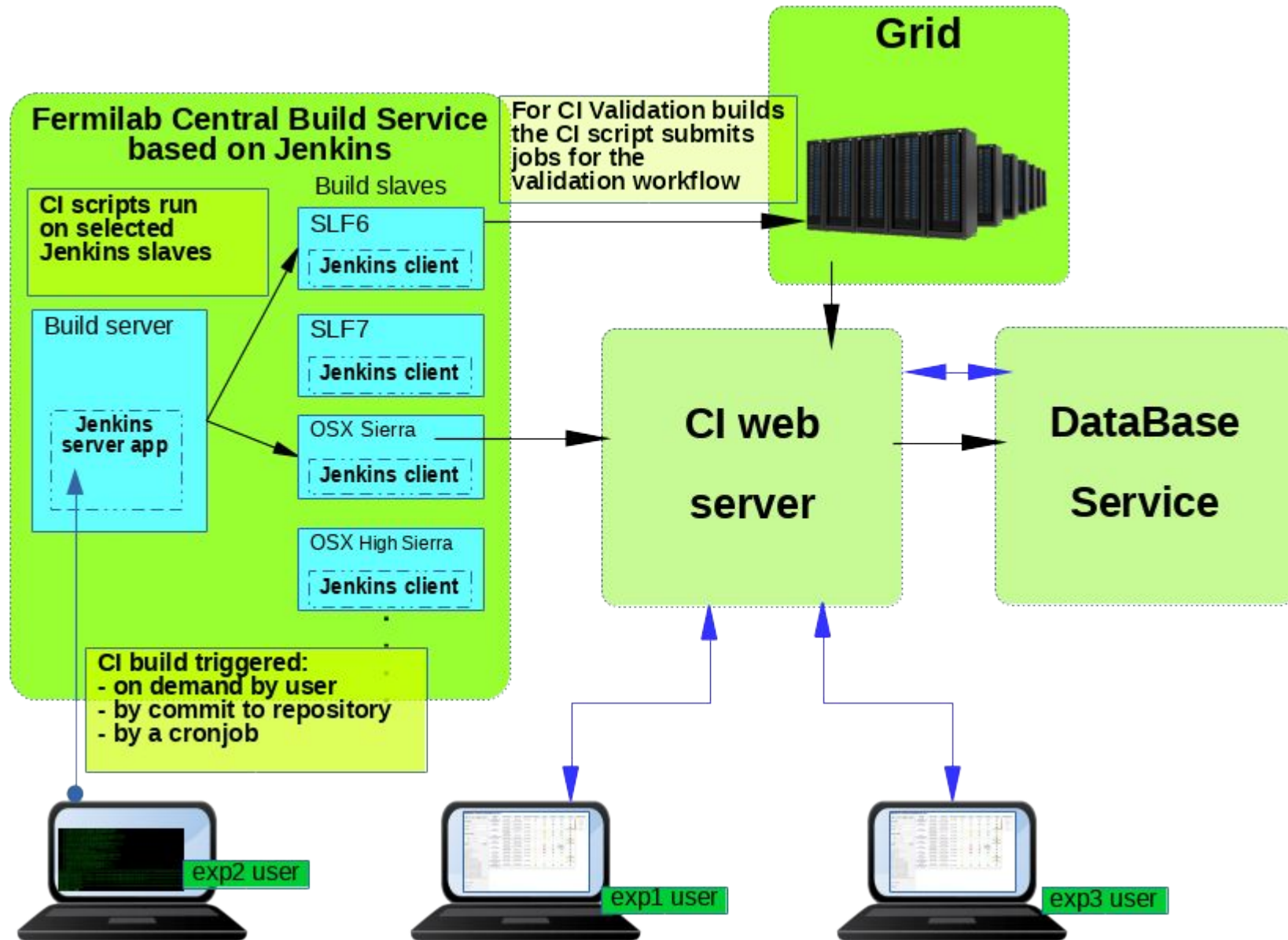
CI Architecture



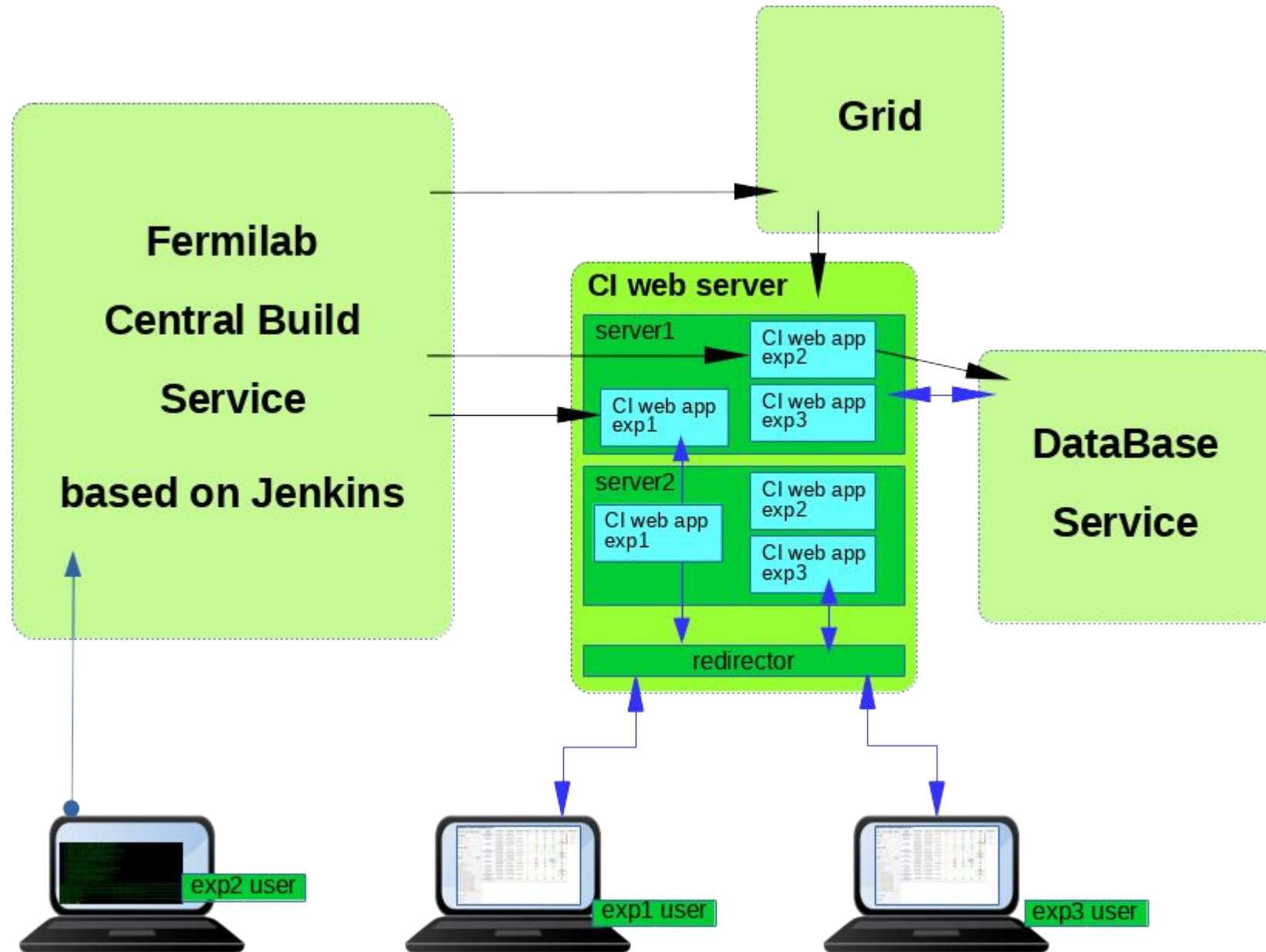
CI Architecture: Build service



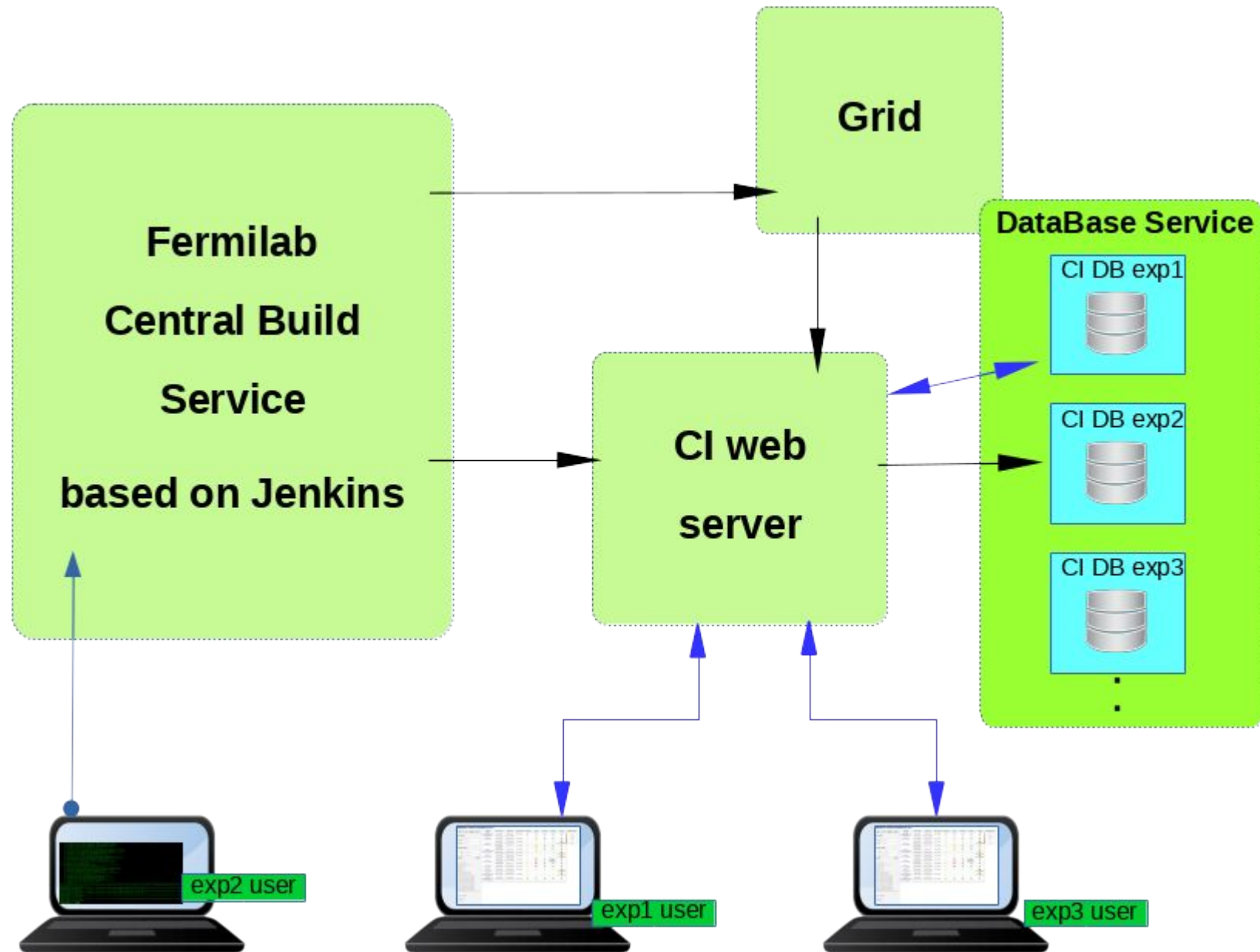
CI Architecture: Build service



CI Architecture: CI web server



CI Architecture: DataBase



CI workflow configuration

- 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

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

Multiplatform Continuous Integration for LarCI

⏮

🏠

?

⏭

Select builds:

From build:

of builds:

Select build date range:

From:

To:

Update

Clear

Platforms

Build types

Test suites

Validation suites

Update

Clear

Build ?	Start Time ?	Build Type ?	setup_environment ?	checkout ?	build ?	unit_test ?	install ?	gen_ref_files ?	ci_tests ?	update_re
dune_ci/820 (LArSoft DUNE)	2018-06-26 08:51:45	slf6 e15:prof	✓	✓	✓	✓	✓	✓	✓	✓
dune_ci/819 (LArSoft DUNE)	2018-06-26 08:50:09	slf6 e15:prof	✓	✓	✓	✓	✓	✓	✓	✓
Build ?	Start Time ?	Build Type ?	setup_environment ?	checkout ?	build ?	unit_test ?	install ?	ci_tests ?		
dune_ci/818 (LArSoft DUNE)	2018-06-26 08:19:28	d16 c2:prof	✓	✓	✓	✓	✓	✗		
dune_ci/817 (LArSoft DUNE)	2018-06-26 08:15:10	slf6 e15:prof	✓	✓	✓	✓	✓	⚠		
dune_ci/816 (LArSoft DUNE)	2018-06-26 08:13:20	slf7 e15:prof	✓	✓	✓	✓	✓	⚠		
Build ?	Start Time ?	Build Type ?	checkout ?	build ?	unit_test ?	install ?	ci_tests ?	ci_validation ?		
lar_ci/2706 (DUNE)	2018-06-26 02:58:37	slf6 e15:prof	✓	✓	✓	✓	⚠	✓		
lar_ci/2705 (DUNE)	2018-06-26 02:48:51	slf6 e15:prof	✓	✓	✓	✓	⚠	✓		
Build ?	Start Time ?	Build Type ?	setup_environment ?	checkout ?	build ?	unit_test ?	install ?	ci_tests ?		
dune_ci/814 (LArSoft DUNE)	2018-06-25 15:40:47	d16 c2:prof	✓	✓	✗	?	?	?		
dune_ci/815 (LArSoft DUNE)	2018-06-25 15:37:32	slf6 e15:prof	✓	✓	✓	✓	✓	⚠		
dune_ci/813 (LArSoft DUNE)	2018-06-25 15:36:35	slf7 e15:prof	✓	✓	✓	✓	✓	⚠		

Pending

Unknown

Succeeded

Warning

Failed

Skipped

CI dashboard details

This panel allows to filter CI builds using different criteria

LArSoft | ArgoNeuT | **DUNE** | LArIAT | uBooNE | SBND

Multiplatform Continuous Integration for LarCI

◀ ⌂ ? ▶

Select builds:

From build: dune_ci/820

of builds: number

Select build date range:

From: mm/dd/yyyy

To: mm/dd/yyyy

Update Clear

Platforms +

Build types +

Test suites +

Validation suites +

Update Clear

Build ?	Start Time ?	Build Type ?	setup_environment ?	checkout ?	build ?	unit_test ?	install ?	gen_ref_files ?	ci_tests ?	update_re
dune_ci/820 (LArSoft DUNE)	2018-06-26 08:51:45	slf6 e15:prof	✓	✓	✓	✓	✓	✓	✓	✓
dune_ci/819 (LArSoft DUNE)	2018-06-26 08:50:09	slf6 e15:prof	✓	✓	✓	✓	✓	✓	✓	✓
Build ?	Start Time ?	Build Type ?	setup_environment ?	checkout ?	build ?	unit_test ?	install ?	ci_tests ?		
dune_ci/818 (LArSoft DUNE)	2018-06-26 08:19:28	d16 c2:prof	✓	✓	✓	✓	✓	✗		
dune_ci/817 (LArSoft DUNE)	2018-06-26 08:15:10	slf6 e15:prof	✓	✓	✓	✓	✓	⚠		
dune_ci/816 (LArSoft DUNE)	2018-06-26 08:13:20	slf7 e15:prof	✓	✓	✓	✓	✓	⚠		
Build ?	Start Time ?	Build Type ?	checkout ?	build ?	unit_test ?	install ?	ci_tests ?	ci_validation ?		
lar_ci/2706 (DUNE)	2018-06-26 02:58:37	slf6 e15:prof	✓	✓	✓	✓	⚠	✓		
lar_ci/2705 (DUNE)	2018-06-26 02:48:51	slf6 e15:prof	✓	✓	✓	✓	⚠	✓		
Build ?	Start Time ?	Build Type ?	setup_environment ?	checkout ?	build ?	unit_test ?	install ?	ci_tests ?		
dune_ci/814 (LArSoft DUNE)	2018-06-25 15:40:47	d16 c2:prof	✓	✓	✗	?	?	?		
dune_ci/815 (LArSoft DUNE)	2018-06-25 15:37:32	slf6 e15:prof	✓	✓	✓	✓	✓	⚠		
dune_ci/813 (LArSoft DUNE)	2018-06-25 15:36:35	slf7 e15:prof	✓	✓	✓	✓	✓	⚠		

Progress Legend

Pending

Unknown

Succeeded

Warning

Failed

Skipped

CI dashboard details

(?) markers give access do wiki documentation

LArSoft | ArgoNeuT | **DUNE** | LArIAT | uBooNE | SBND

Multiplatform Continuous Integration for LarCI

⏮

🏠

?

⏭

Select builds:

From build:

of builds:

Select build date range:

From:

To:

Update

Clear

Platforms

+

Build types

+

Test suites

+

Validation suites

+

Update

Clear

Build ?	Start Time ?	Build Type ?	setup_environment ?	checkout ?	build ?	unit_test ?	install ?	gen_ref_files ?	ci_tests ?	update_re
dune_ci/820 (LArSoft DUNE)	2018-06-26 08:51:45	slf6 e15:prof	✓	✓	✓	✓	✓	✓	✓	✓
dune_ci/819 (LArSoft DUNE)	2018-06-26 08:50:09	slf6 e15:prof	✓	✓	✓	✓	✓	✓	✓	✓
Build ?	Start Time ?	Build Type ?	setup_environment ?	checkout ?	build ?	unit_test ?	install ?	ci_tests ?		
dune_ci/818 (LArSoft DUNE)	2018-06-26 08:19:28	d16 c2:prof	✓	✓	✓	✓	✓	✗		
dune_ci/817 (LArSoft DUNE)	2018-06-26 08:15:10	slf6 e15:prof	✓	✓	✓	✓	✓	⚠		
dune_ci/816 (LArSoft DUNE)	2018-06-26 08:13:20	slf7 e15:prof	✓	✓	✓	✓	✓	⚠		
Build ?	Start Time ?	Build Type ?	checkout ?	build ?	unit_test ?	install ?	ci_tests ?	ci_validation ?		
lar_ci/2706 (DUNE)	2018-06-26 02:58:37	slf6 e15:prof	✓	✓	✓	✓	⚠	✓		
lar_ci/2705 (DUNE)	2018-06-26 02:48:51	slf6 e15:prof	✓	✓	✓	✓	⚠	✓		
Build ?	Start Time ?	Build Type ?	setup_environment ?	checkout ?	build ?	unit_test ?	install ?	ci_tests ?		
dune_ci/814 (LArSoft DUNE)	2018-06-25 15:40:47	d16 c2:prof	✓	✓	✗	?	?	?		
dune_ci/815 (LArSoft DUNE)	2018-06-25 15:37:32	slf6 e15:prof	✓	✓	✓	✓	✓	⚠		
dune_ci/813 (LArSoft DUNE)	2018-06-25 15:36:35	slf7 e15:prof	✓	✓	✓	✓	✓	⚠		

Pending

Unknown

Succeeded

Warning

Failed

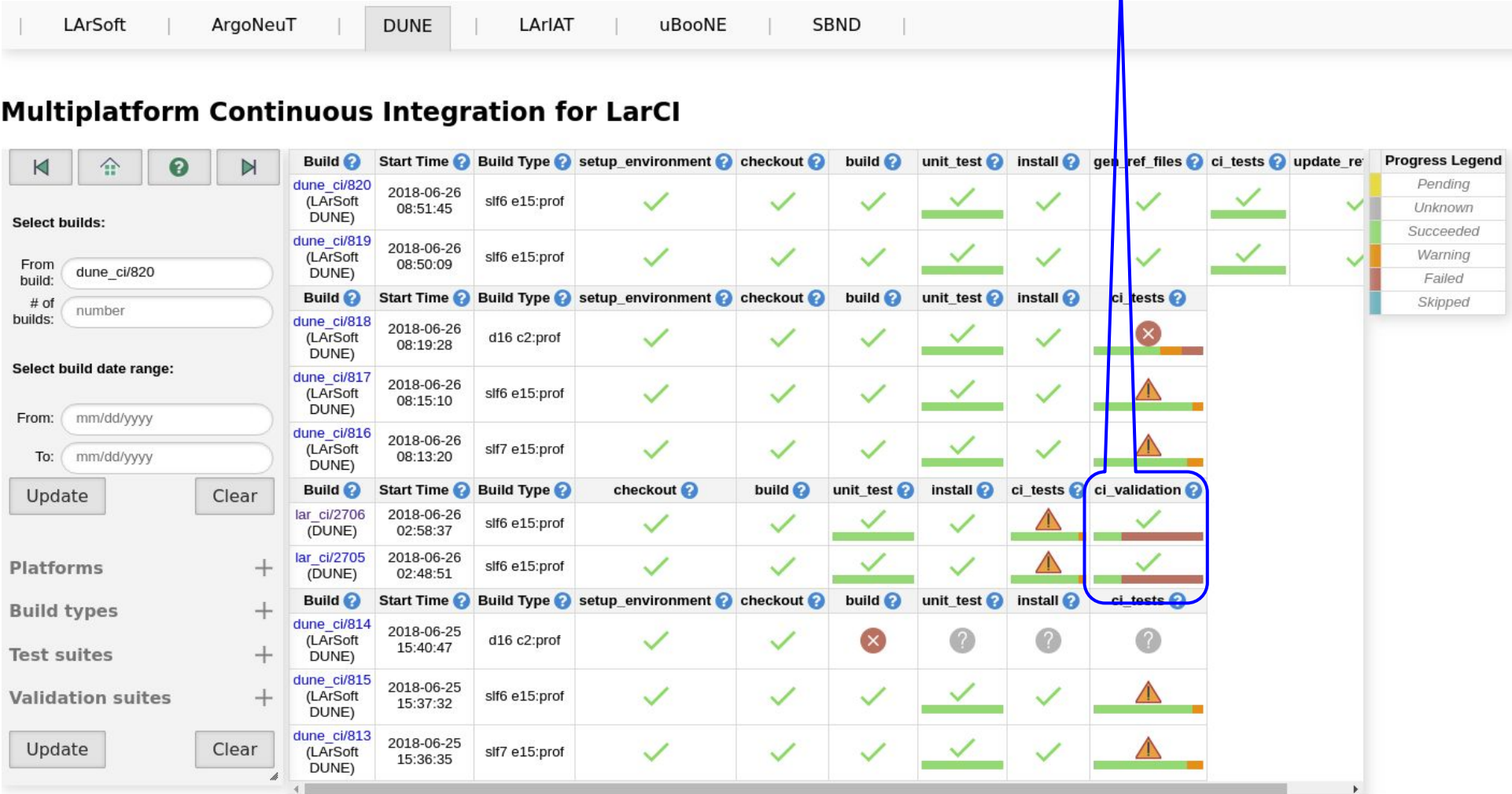
Skipped

207/12/18Vito Di Benedetto | Continuous Integration service at Fermilab

Fermilab

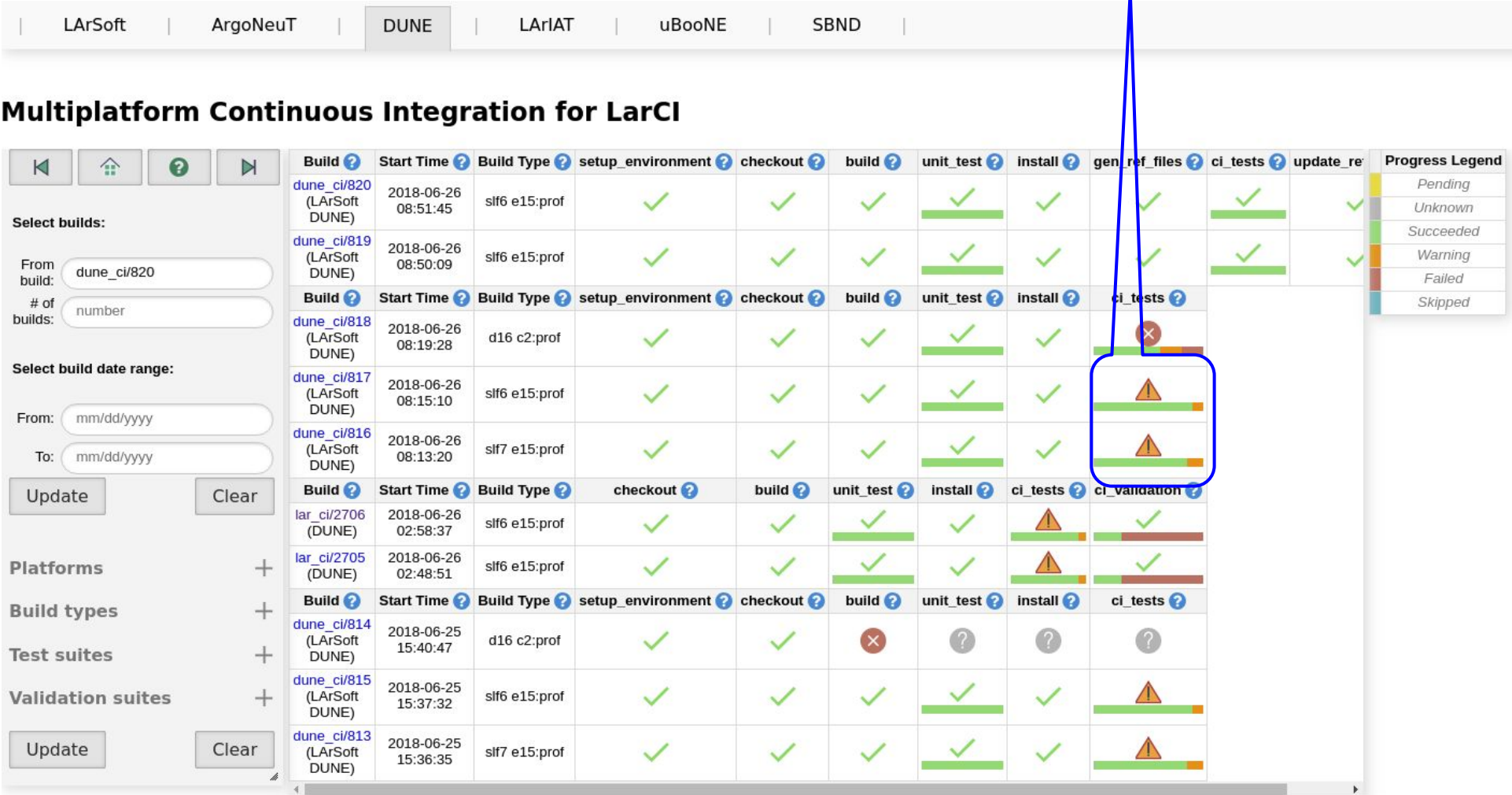
CI dashboard details

CI Validation reports some failed jobs but validation results are OK



CI dashboard details

Warning in regression tests notifies that some stage output is changed



CI dashboard details

CI builds to update reference files

LArSoft

ArgoNeuT

DUNE

LArIAT

uBooNE

SBND

Multiplatform Continuous Integration for LarCI

Select builds:

From build: dune_ci/820

of builds: number

Select build date range:

From: mm/dd/yyyy

To: mm/dd/yyyy

Update

Clear

Platforms

Build types

Test suites

Validation suites

Update

Clear

Build	Start Time	Build Type	setup_environment	checkout	build	unit_test	install	gen_ref_files	ci_tests	update_re
dune_ci/820 (LArSoft DUNE)	2018-06-26 08:51:45	slf6 e15:prof	✓	✓	✓	✓	✓	✓	✓	✓
dune_ci/819 (LArSoft DUNE)	2018-06-26 08:50:09	slf6 e15:prof	✓	✓	✓	✓	✓	✓	✓	✓
dune_ci/818 (LArSoft DUNE)	2018-06-26 08:19:28	d16 c2:prof	✓	✓	✓	✓	✓	✗		
dune_ci/817 (LArSoft DUNE)	2018-06-26 08:15:10	slf6 e15:prof	✓	✓	✓	✓	✓	⚠		
dune_ci/816 (LArSoft DUNE)	2018-06-26 08:13:20	slf7 e15:prof	✓	✓	✓	✓	✓	⚠		
lar_ci/2706 (DUNE)	2018-06-26 02:58:37	slf6 e15:prof	✓	✓	✓	✓	⚠	✓		
lar_ci/2705 (DUNE)	2018-06-26 02:48:51	slf6 e15:prof	✓	✓	✓	✓	⚠	✓		
dune_ci/814 (LArSoft DUNE)	2018-06-25 15:40:47	d16 c2:prof	✓	✓	✗	?	?	?		
dune_ci/815 (LArSoft DUNE)	2018-06-25 15:37:32	slf6 e15:prof	✓	✓	✓	✓	✓	⚠		
dune_ci/813 (LArSoft DUNE)	2018-06-25 15:36:35	slf7 e15:prof	✓	✓	✓	✓	✓	⚠		

Build

Start Time

Build Type

setup_environment

checkout

build

unit_test

install

ci_tests

ci_validation

Progress Legend

Pending

Unknown

Succeeded

Warning

Failed
















Skipped

CI (regression) tests status

Phase: **ci_tests**



 ci_testsDUNE.log
















#	Test Name
1	 ci_reco_regression_test_dunefd
2	 ci_reco_regression_test_protoDUNE
3	 ci_select_regression_test_dunefd
4	 ci_g4_regression_test_dunefd
5	 ci_g4_regression_test_protoDUNE
6	 ci_g4_regression_test_protoDUNEdp
7	 ci_reco_regression_test_protoDUNEdp
8	 ci_detsim_regression_test_dunefd
9	 ci_detsim_regression_test_protoDUNE
10	 ci_detsim_regression_test_protoDUNEdp
11	 ci_gen_regression_test_dunefd
12	 ci_gen_regression_test_protoDUNE
13	 ci_gen_regression_test_protoDUNEdp
14	 ci_mergeana_regression_test_dunefd
15	 ci_mergeana_regression_test_protoDUNE

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

CI (regression) tests status

Phase: `ci_tests`

 `ci_testsDUNE.log`
















#	Test Name
1	 <code>ci_reco_regression_test_dunefd</code>
2	 <code>ci_reco_regression_test_protoDUNE</code>
3	 <code>ci_select_regression_test_dunefd</code>
4	 <code>ci_g4_regression_test_dunefd</code>
5	 <code>ci_g4_regression_test_protoDUNE</code>
6	 <code>ci_g4_regression_test_protoDUNEdp</code>
7	 <code>ci_reco_regression_test_protoDUNEdp</code>
8	 <code>ci_detsim_regression_test_dunefd</code>
9	 <code>ci_detsim_regression_test_protoDUNE</code>
10	 <code>ci_detsim_regression_test_protoDUNEdp</code>
11	 <code>ci_gen_regression_test_dunefd</code>
12	 <code>ci_gen_regression_test_protoDUNE</code>
13	 <code>ci_gen_regression_test_protoDUNEdp</code>
14	 <code>ci_mergeana_regression_test_dunefd</code>
15	 <code>ci_mergeana_regression_test_protoDUNE</code>

- 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

CI (regression) tests status

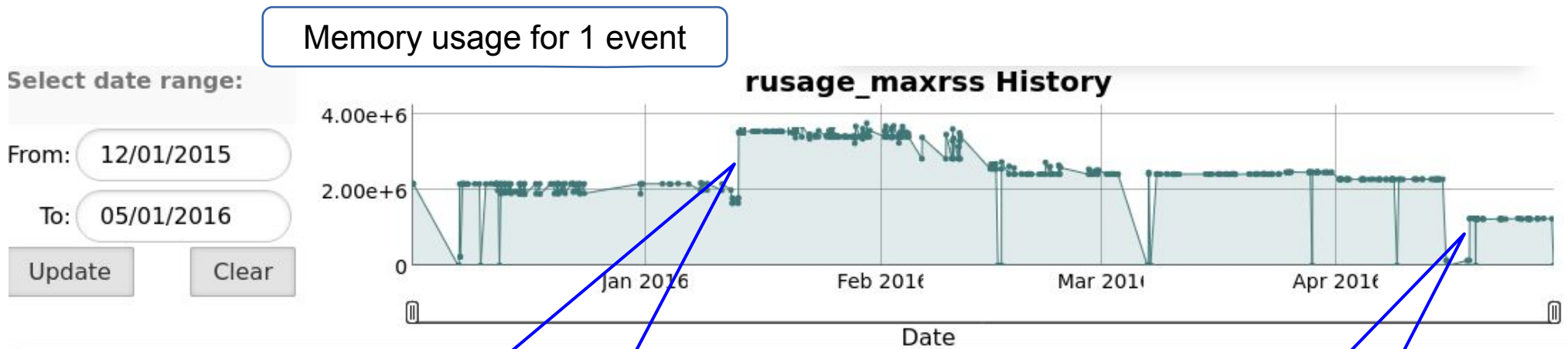
Phase: ci_tests

 ci_testsDUNE.log

#	Test Name
1	 ci_reco_regression_test_dunefd
2	 ci_reco_regression_test_protoDUNE
3	 ci_select_regression_test_dunefd
4	 ci_g4_regression_test_dunefd
5	 ci_g4_regression_test_protoDUNE
6	 ci_g4_regression_test_protoDUNEdp
7	 ci_reco_regression_test_protoDUNEdp
8	 ci_detsim_regression_test_dunefd
9	 ci_detsim_regression_test_protoDUNE
10	 ci_detsim_regression_test_protoDUNEdp
11	 ci_gen_regression_test_dunefd
12	 ci_gen_regression_test_protoDUNE
13	 ci_gen_regression_test_protoDUNEdp
14	 ci_mergeana_regression_test_dunefd
15	 ci_mergeana_regression_test_protoDUNE

- 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



Phase: ci_validation

ci_validationuBooNE.log

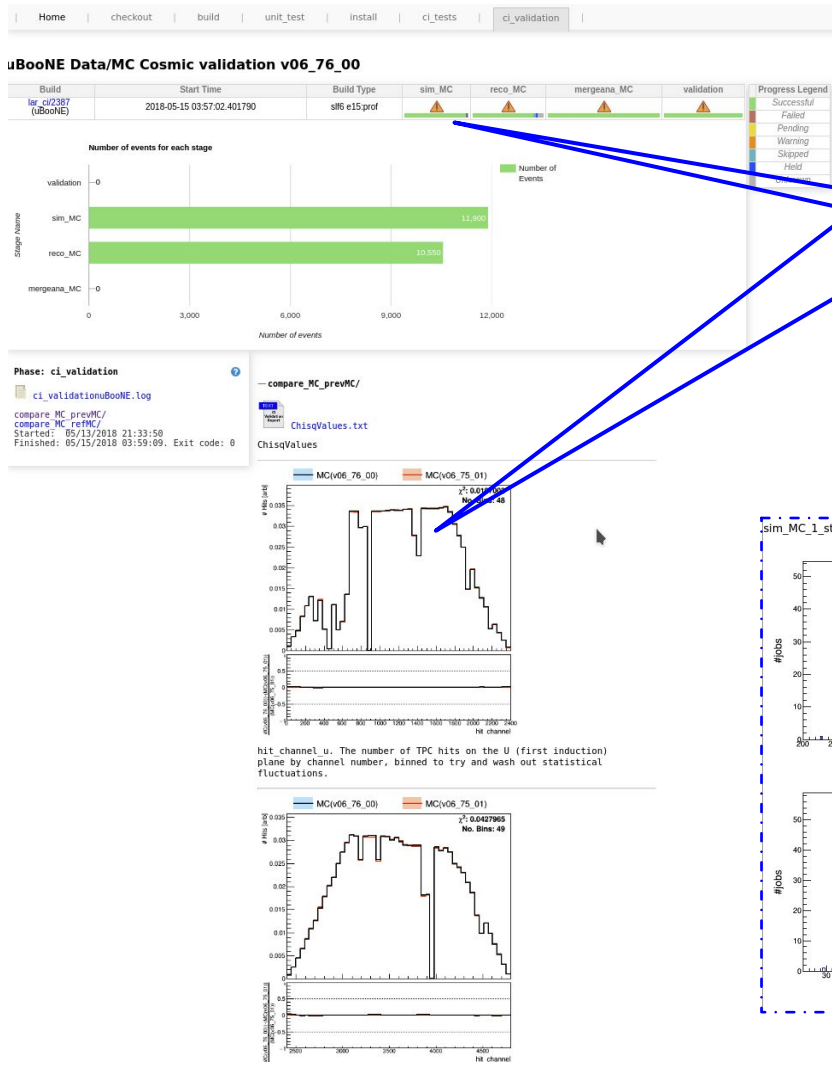
compare_MC_prevMC/
compare_MC_refMC/

Started: 05/13/2018 21:33:50
Finished: 05/15/2018 03:59:09. Exit code: 0

Folders with validation plots

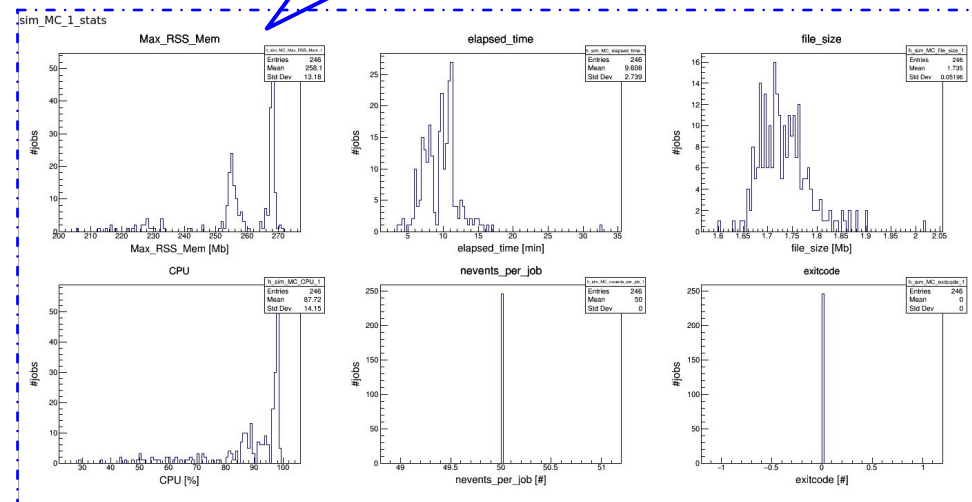
Links to time series for validation stats

CI Validation dashboard

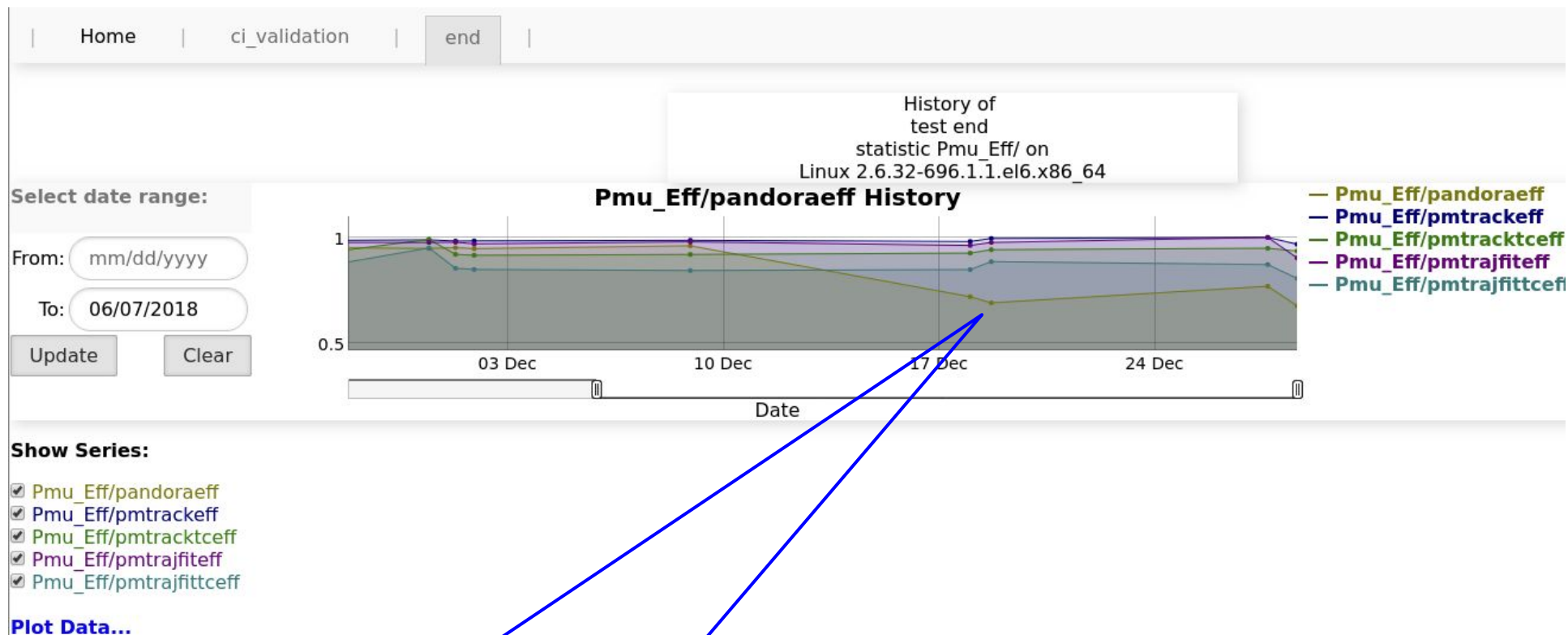


Folders with validation plots expanded

The stage marker gives access to job resource stats



CI Validation: efficiency time series



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