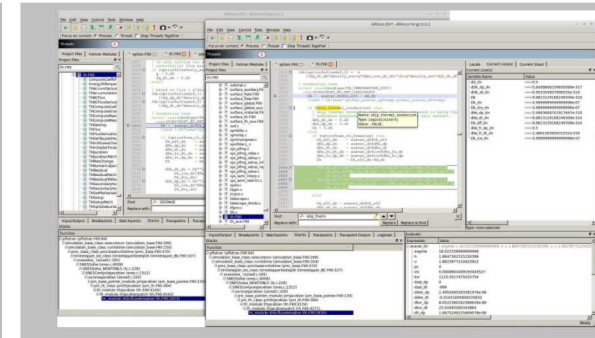


```

Test log file : pflotran-tests-2016-07-29_10-16-31.testlog
Running pflotran regression tests :
.....
-----
Regression test summary:
  Total run time: 185.067 [s]
  Total tests : 179
  Tests run : 179
  All tests passed.

```

last build	dev-std-gnu-linux	dev-std-gnu-mac	
	build successful	build successful	
current activity	waiting next in ~ 9 hrs 40 mins at 01:00	waiting next in ~ 9 hrs 40 mins at 01:00	
PDT	changes	dev-std-gnu-linux	dev-std-gnu-mac
01:12:50			
01:12:00			
01:04:54		test pflotran stdio	test pflotran stdio
01:04:36		build pflotran stdio	build pflotran stdio
		petsc stdio	petsc stdio
		set build script path	set build script path
		property 'pflotran_dir' set stdio property changes	property 'pflotran_dir' set stdio property changes



# PFLOTRAN Testing

# Cloud-based Continuous Integration



[bitbucket.org/pflotran/pflotran-qa](https://bitbucket.org/pflotran/pflotran-qa)  
[bitbucket.org/pflotran/pflotran-documentation](https://bitbucket.org/pflotran/pflotran-documentation)  
[bitbucket.org/pflotran/pflotran](https://bitbucket.org/pflotran/pflotran)

PFLOTRAN / node / pflotran  
 Commits  
 PFLOTRAN / node / pflotran  
 Wiki  
 PFLOTRAN / Home  
**PFLOTRAN**  
 PFLOTRAN is an open source, state-of-the-art modeling parallel subsurface flow and reactive transport code. The code is developed under a GNU GPL license allowing for third parties to distribute proprietary software with the code, however any modifications to the code shall need to be distributed and remain open source. PFLOTRAN is written in object oriented, low level Fortran 2003. The choice of Fortran over C/C++ was based primarily on the need to utilize an existing, proven, high performance and experimental domain context, without which PFLOTRAN's complex parallelization through domain decomposition using the MPI based PETSc framework with efficient tracking the git 4012 tag of PETSc, available through Bitbucket.



Travis CI

[travis-ci.org/pflotran/pflotran](https://travis-ci.org/pflotran/pflotran)  
 pflotran / pflotran

~300 Regression and 60 Unit Tests

Unlimited builds



pflotran.org

doc-dev.pflotran.org

**PFLOTRAN**  
 PFLOTRAN Documentation  
**PFLOTRAN**  
 PFLOTRAN is an open source, state-of-the-art modeling parallel subsurface flow and reactive transport code. The code is developed under a GNU GPL license allowing for third parties to distribute proprietary software with the code, however any modifications to the code shall need to be distributed and remain open source. PFLOTRAN is written in object oriented, low level Fortran 2003. The choice of Fortran over C/C++ was based primarily on the need to utilize an existing, proven, high performance and experimental domain context, without which PFLOTRAN's complex parallelization through domain decomposition using the MPI based PETSc framework with efficient tracking the git 4012 tag of PETSc, available through Bitbucket.



Codecov

codecov.io/gh/pflotran/pflotran/tree/master/src/pflotran

Files	Pass	Fail	Skipped	Coverage	
src/pflotran/PFS	47	0	0	47	0.00%
src/pflotran/PFS	40	46	0	47	100.00%
src/pflotran/FlowPFS	4	0	0	4	0.00%
src/pflotran/FlowPFS	34	55	0	1	98.21%
src/pflotran/Flow_pemsgPFS	23	22	0	1	95.65%
src/pflotran/FlowPFS	11	50	0	1	90.91%
src/pflotran/FlowPFS	197	147	0	30	84.77%
src/Block/BlockPFS	65	25	0	40	36.46%
src/Block/BlockPFS	145	0	0	145	0.00%
src/Block/BlockPFS	5	429	0	723	45.32%
src/Block/BlockPFS	150	42	0	108	28.00%
src/Block/BlockPFS	938	397	0	241	85.65%
src/Block/BlockPFS	5	386	0	423	47.98%

docker  
 SPHINX  
 Pflotran Documentation Generator  
**CODESHIP**  
 by CloudBees  
 app.codeship.com/projects/316699

Merge branch 'master' of bitbucket.org/pflotran/pflotran-documentation  
 Merge in jenkins build request P1 Updated the documentation with the SEED keyword  
 Added virtual machine instructions  
 Added SKIP\_RESTART  
 Updated environment variables  
 Updated PFLOTRAN\_DIR -> PFLOTRAN/DIR to match folder nomenclature  
 Indefinite error to null  
 Remove interactive frontend to install  
 Added python path to qa test arguments

Limit of 100 builds per month

**PFLOTRAN QA Test Suite**  
**PFLOTRAN**  
 Links to PFLOTRAN's QA Test Suite:  
 - Home To Run The Test Suite  
 - Install The Test Suite on a Single Python Host Environment  
 - QA Test Report Card  
 - Thermal QA Tests  
 - Steady Thermal Conduction  
 - Transient Thermal Conduction  
 - Pore QA Tests  
 - Steady Flow (Pressure)  
 - Transient Flow (Pressure)  
 - Gas QA Tests  
 - Steady Gas (Pressure)  
 - Transport QA Tests  
 - Steady Transport  
 - Transient Transport

Hammond and Bisht, 2019  
<https://doi.org/10.6084/m9.figshare.7761950.v1>

# Testing

- Unit testing
  - Equations of state
    - Viscosity
    - Density
    - Enthalpy / internal energy
    - Saturation pressure
  - Constitutive relations
    - Capillary pressure functions
    - Saturation functions
    - Relative permeability functions
  - pFUnit
    - Open source Fortran unit testing framework

# Testing (cont.)

- Regression testing (Did the solution change?)
  - Driven by custom python scripting
  - Regression module in PFLOTRAN used to sample solution variables at the end of a simulation
  - Locations (cell ids) are specified in a REGRESSION block

```
REGRESSION
  CELLS_PER_PROCESS 2
  CELLS
    29
  /
END
```

- Variables are specified in the OUTPUT block
- .regression file compared to .regression.gold file
- Solutions outside absolute- or relative-change convergence tolerances are flagged.

# Regression Entries in Input File

```
#===== regression =====
REGRESSION
  CELLS_PER_PROCESS 2
  CELLS
    29
  /
END

#===== output options =====
OUTPUT
  VARIABLES
    LIQUID_PRESSURE
    LIQUID_SATURATION
    PERMEABILITY_X
    PERMEABILITY_Y
    PERMEABILITY_Z
    POROSITY
    PH
    TOTAL
    TOTAL_SORBED
    KD
    MINERAL_SATURATION_INDEX
  /
END
```

# .regression Output File

```
-- PRESSURE: Liquid Pressure --
  Max:  3.6987012374958E+05
  Min: -2.9546226998033E+04
  Mean: 1.7789073395768E+05
  29:  1.9047483535024E+05
  1:   3.6938752319775E+05
  31:  1.9834550275718E+05
-- RATE: Metatorbernite Rate --
  Max:  0.0000000000000E+00
  Min: -1.9999999998411E-11
  Mean: -2.6666666663803E-12
  29:  0.0000000000000E+00
  1:   0.0000000000000E+00
  31:  0.0000000000000E+00
-- GENERIC: LIQUID VELOCITY [m/d] --
  29:  8.5124089175370E-02 -1.2877090842582E-01  5.2164253201197E-04
  1:   1.7617702348986E-02 -9.6502560583815E-04 -8.0086946328361E-04
  31:  2.5578704112122E-01  2.2363909790982E-02  1.8169543176494E-02
-- SOLUTION: Flow --
Time (seconds):  2.6570320129395E-02
Time Steps:      14
Newton Iterations:      28
Solver Iterations:      28
Time Step Cuts:        0
Solution 2-Norm:  1.8527721282346E+06
Residual 2-Norm:  3.7495263161587E-13
```

# Testing (cont.)

- Tests can be launched through the PFLOTRAN makefile
  - `make rtest` (regression tests only)
  - `make utest` (unit tests only)
  - `make test` (regression and unit tests)
- Regression tests can be launched separately from the command line within `$PFLOTRAN_DIR/regression_tests`  
`python regression_tests.py <args>`

# python regression\_tests.py --help

```
usage: regression_tests.py [-h] [--backtrace] [--advanced]
                          [-c CONFIG_FILES [CONFIG_FILES ...]] [--check-only]
                          [--check-performance] [--debug] [-d]
                          [-e EXECUTABLE] [--list-suites] [--list-tests]
                          [-m MPIEXEC] [-n]
                          [-r [RECURSIVE_SEARCH [RECURSIVE_SEARCH ...]]]
                          [-s SUITES [SUITES ...]] [-t TESTS [TESTS ...]]
                          [--timeout TIMEOUT] [-u]
```

Run a pflotran regression tests or suite of tests.

optional arguments:

```
-h, --help            show this help message and exit
--backtrace           show exception backtraces as extra debugging output
--advanced            enable advanced options for developers
-c CONFIG_FILES [CONFIG_FILES ...], --config-files CONFIG_FILES [CONFIG_FILES ...]
                    test configuration file to use
--check-only          diff the existing regression files without running
                    pflotran again.
--check-performance  include the performance metrics ('SOLUTION' blocks) in
                    regression checks.
--debug              extra debugging output
-d, --dry-run        perform a dry run, setup the test commands but don't
                    run them
-e EXECUTABLE, --executable EXECUTABLE
                    path to executable to use for testing
--list-suites        print the list of test suites from the config file and
                    exit
```

...

# 543.cfg – Regression Test Configuration File

```
[suites]
standard = 543_flow
           543_flow_dbase
           543_flow_eos_default
           543_flow_eos_constant
           543_flow_eos_exponential
           543_flow_and_tracer
           543_flow_and_tracer_dbase
           543_hanford_srfcplx_base
           543_hanford_srfcplx_base_restart
           543_hanford_srfcplx_base_restart_hdf5
           543_hanford_overwrite_restart
           543_hanford_srfcplx_param
standard_parallel = 543_flow-np8
                   543_flow_and_tracer-np8
                   543_hanford_srfcplx_param-np8

[default-test-criteria]
# default criteria for all tests, can be overwritten by specific tests
time = 500 percent
generic = 1.0e-12 absolute
concentration = 1.0e-9 relative
discrete = 0 absolute
rate = 1.0e-12 absolute
volume_fraction = 1.0e-12 absolute
pressure = 1.0e-12 relative
saturation = 1.0e-12 absolute

...

[543_flow-np8]
np=8

[543_hanford_srfcplx_param]
generic = 1.0e-12 relative
```

# 'make test' Screen Output

```
Running pflotran unit tests :
```

```
.....  
Time:          0.007 seconds
```

```
OK  
(60 tests)
```

```
-----  
Test log file : pflotran-tests-2018-09-25_17-52-05.testlog
```

```
Running pflotran regression tests :
```

```
.....  
.....  
.....  
.....
```

```
-----  
Regression test summary:
```

```
  Total run time: 266.636 [s]  
  Total tests : 297  
  Tests run : 297  
  All tests passed.
```

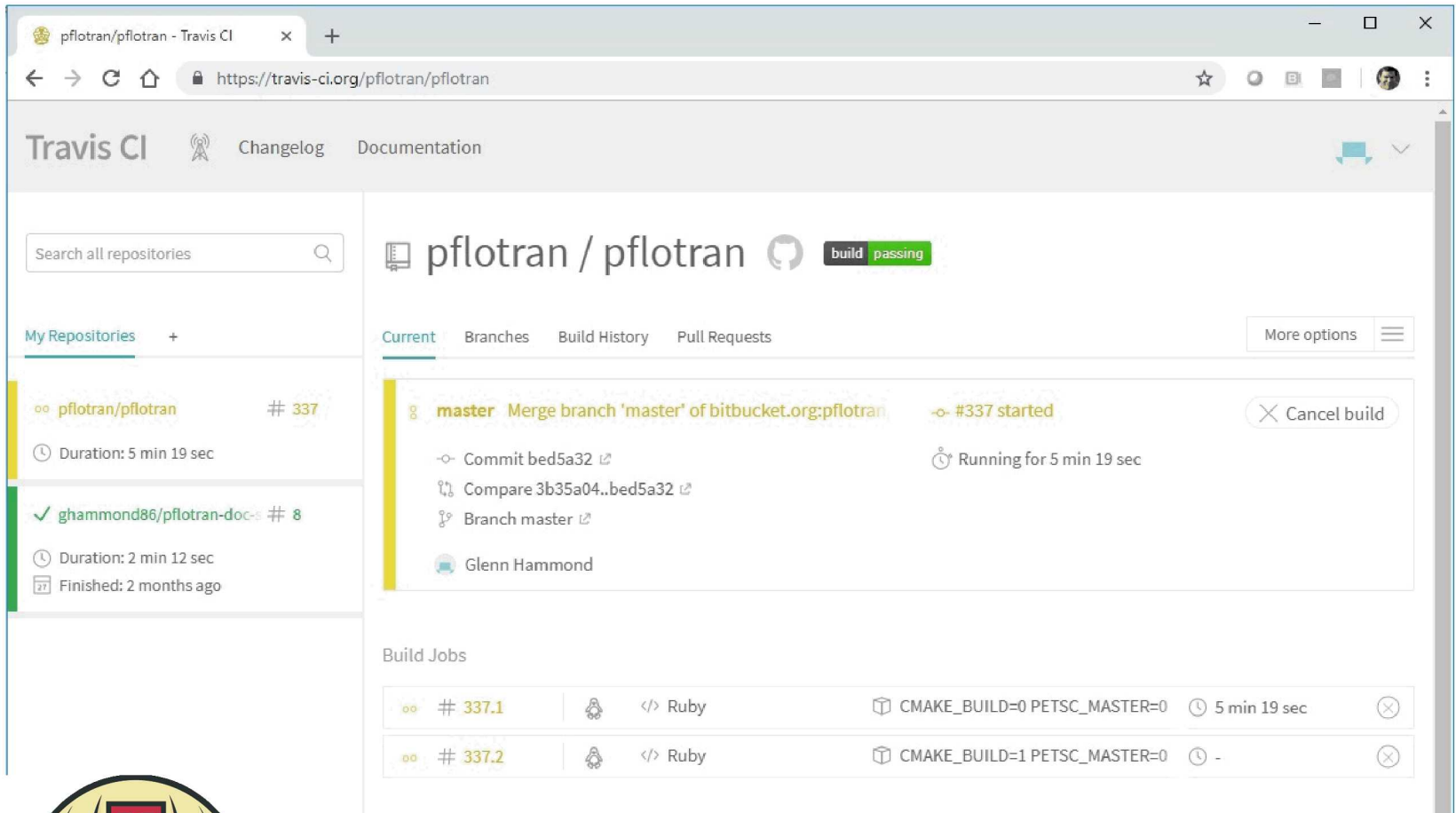
# Testing (cont.)

- Example regression test failure
  - Perturb critical pressure for water equation of state by 10 billionths of a percent

```
subroutine EOSWaterDensityIFC67()
```

```
    tc1 = H2O_CRITICAL_TEMPERATURE      ! K  
-   pc1 = H2O_CRITICAL_PRESSURE        ! Pa  
+   pc1 = H2O_CRITICAL_PRESSURE + 1.d-10*H2O_CRITICAL_PRESSURE    ! Pa  
    vc1 = 0.00317d0    ! m^3/kg
```

# Bitbucket → Github → Travis CI



Travis CI

Search all repositories

My Repositories

- pflotran/pflotran # 337  
Duration: 5 min 19 sec
- ghammond86/pflotran-docs # 8  
Duration: 2 min 12 sec  
Finished: 2 months ago

pflotran / pflotran build passing

Current Branches Build History Pull Requests More options

g master Merge branch 'master' of bitbucket.org:pflotran -> #337 started  
Cancel build

- Commit bed5a32
- Compare 3b35a04..bed5a32
- Branch master
- Glenn Hammond

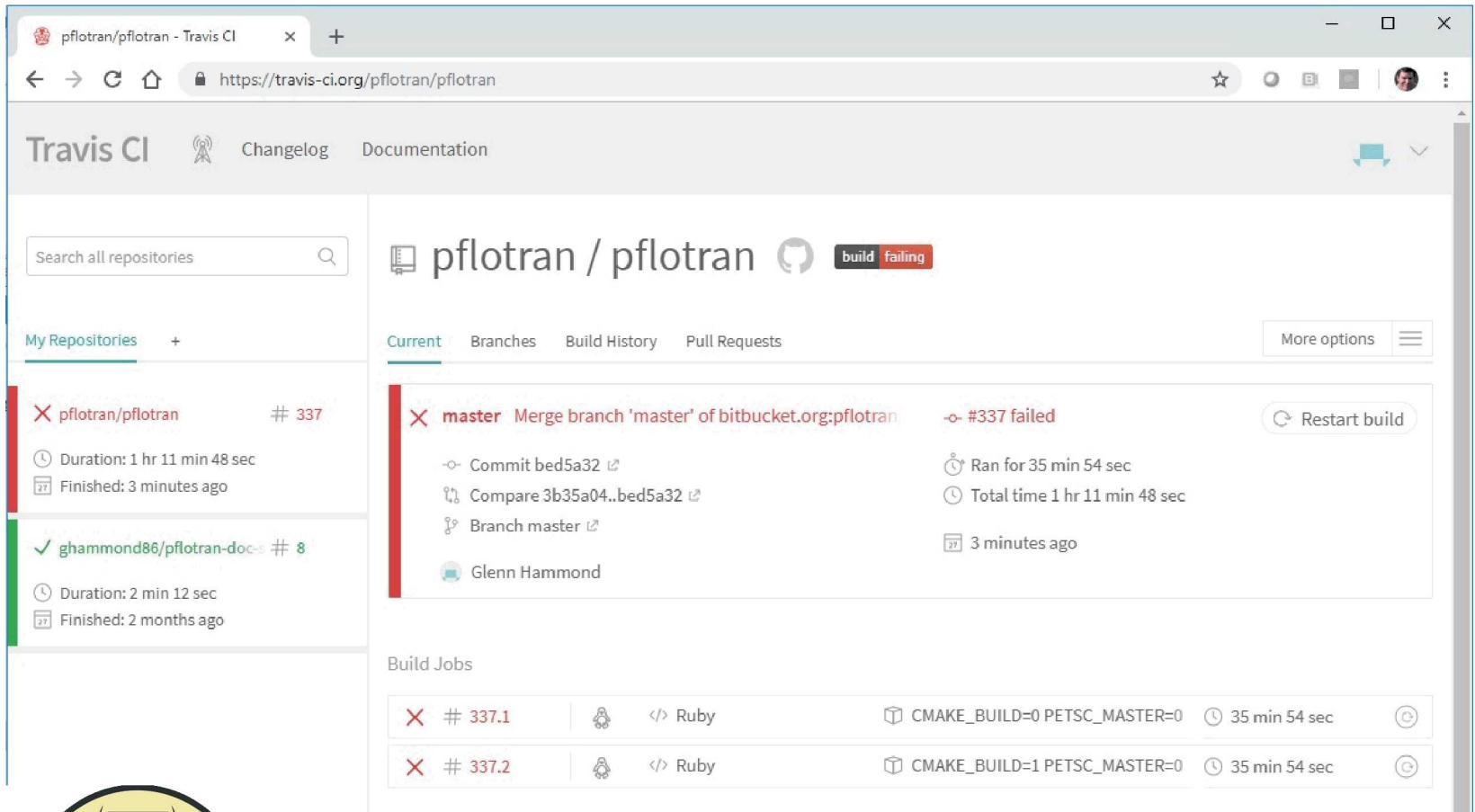
Running for 5 min 19 sec

Build Jobs

# 337.1	</> Ruby	CMAKE_BUILD=0 PETSC_MASTER=0	5 min 19 sec	⊗
# 337.2	</> Ruby	CMAKE_BUILD=1 PETSC_MASTER=0	-	⊗



# Bitbucket → Github → Travis CI



Travis CI

Search all repositories

My Repositories

- ✗ pflotran/pflotran # 337
  - Duration: 1 hr 11 min 48 sec
  - Finished: 3 minutes ago
- ✓ ghammond86/pflotran-doc # 8
  - Duration: 2 min 12 sec
  - Finished: 2 months ago

pflotran / pflotran **build failing**



Current Branches Build History Pull Requests More options

✗ **master** Merge branch 'master' of bitbucket.org:pflotran - #337 failed [Restart build](#)

- Commit bed5a32
- Compare 3b35a04..bed5a32
- Branch master
- Glenn Hammond

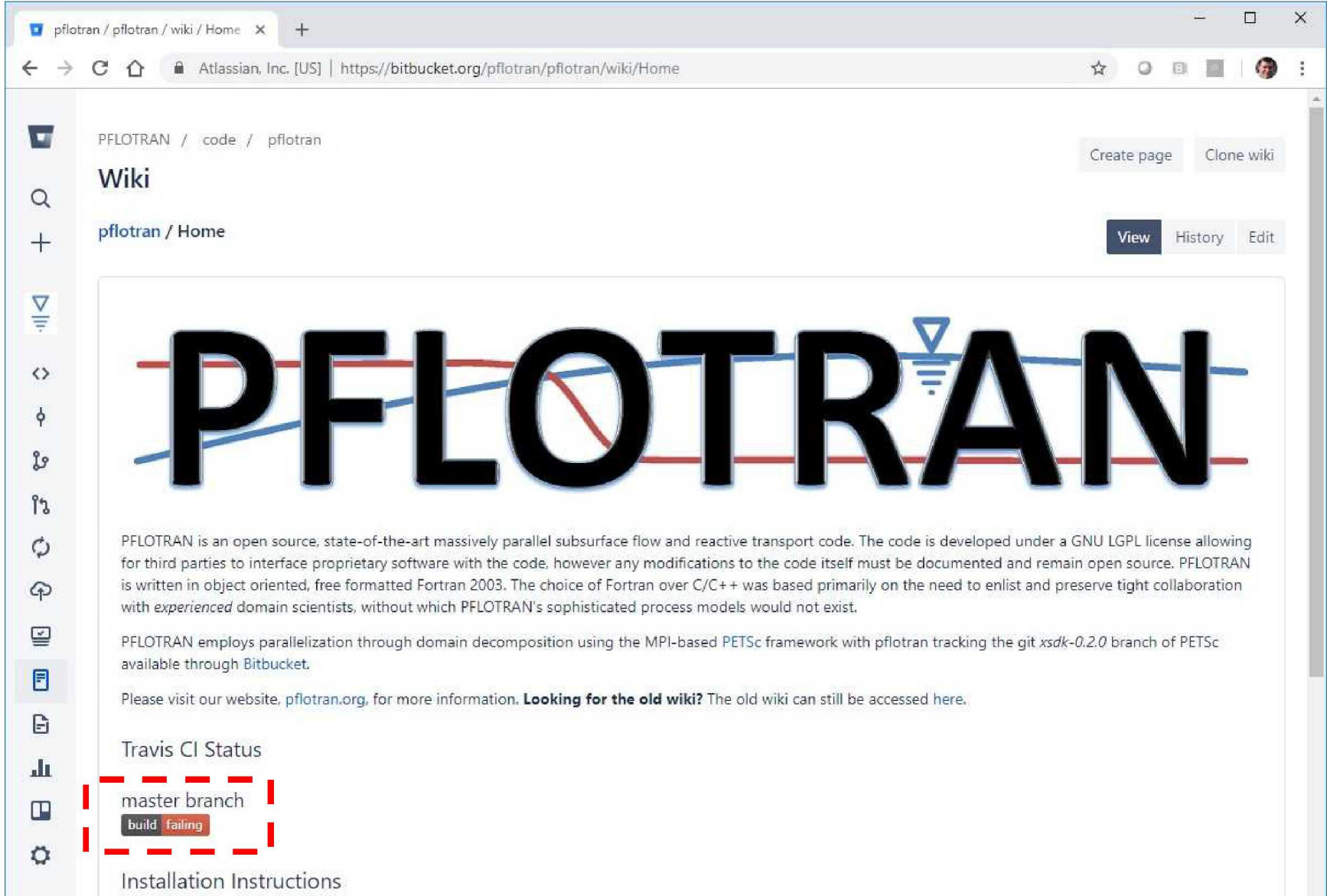
Ran for 35 min 54 sec  
Total time 1 hr 11 min 48 sec  
3 minutes ago

Build Jobs

✗ # 337.1	 </> Ruby	CMAKE_BUILD=0 PETSC_MASTER=0	🕒 35 min 54 sec	🔄
✗ # 337.2	 </> Ruby	CMAKE_BUILD=1 PETSC_MASTER=0	🕒 35 min 54 sec	🔄



# Travis CI Badge at pflotran.org



pflotran / pflotran / wiki / Home

Atlassian, Inc. [US] | https://bitbucket.org/pflotran/pflotran/wiki/Home

PFLOTRAN / code / pflotran

Wiki

pflotran / Home

Create page Clone wiki

View History Edit

# PFLOTRAN

PFLOTRAN is an open source, state-of-the-art massively parallel subsurface flow and reactive transport code. The code is developed under a GNU LGPL license allowing for third parties to interface proprietary software with the code, however any modifications to the code itself must be documented and remain open source. PFLOTRAN is written in object oriented, free formatted Fortran 2003. The choice of Fortran over C/C++ was based primarily on the need to enlist and preserve tight collaboration with *experienced* domain scientists, without which PFLOTRAN's sophisticated process models would not exist.

PFLOTRAN employs parallelization through domain decomposition using the MPI-based PETSc framework with pflotran tracking the git *xsdk-0.2.0* branch of PETSc available through Bitbucket.

Please visit our website, [pflotran.org](http://pflotran.org), for more information. **Looking for the old wiki?** The old wiki can still be accessed here.

### Travis CI Status

master branch  
build failing

Installation Instructions

```
Running pflotran unit tests :
```

```
.....F.....
```

```
Time:          0.006 seconds
```

```
Failure in: testEOSWater_DensitySTP
```

```
Location: [test_eos_water.pf:157]
```

```
expected: +998.3234 but found: +998.3234;      difference: |+0.4774847E-11| >
```

```
tolerance:+0.1000000E-15.
```

```
FAILURES!!!
```

```
Tests run: 60, Failures: 1, Errors: 0
```

```
-----  
Test log file : pflotran-tests-2018-09-25_17-01-20.testlog
```

```
Running pflotran regression tests :
```

```
.....F.F....FFFFF.....F.....F.....F..FF.FFF.FF.....FF.....
```

```
F.....F...FFFF...FFFFFF.....F.F.F...FF.....
```

```
.F.....FF.....F..F.....
```

```
.....FF..F.....FF.....  
-----
```

```
Regression test summary:
```

```
Total run time: 279.741 [s]
```

```
Total tests : 297
```

```
Tests run : 297
```

```
Failed : 46
```

# pflotran-tests-2018-09-25\_17-01-20.testlog



PFLOTRAN Regression Test Log

Date : 2018-09-25\_17-01-20

System Info :

platform : linux2

Test directory :

/home/travis/build/pflotran/pflotran/regression\_tests

PFLOTRAN repository status :

-----

```
$ git log -1 HEAD
```

```
commit bed5a32cb0ba391a41a64fcf9b7120f7a078a94d
```

```
Merge: 3b35a04 559915d
```

```
Author: Glenn Hammond <gehammo@sandia.gov>
```

```
Date: Tue Sep 25 11:36:49 2018 -0500
```

Merge branch 'master' of bitbucket.org:pflotran/pflotran

```
$ git status -u no
```

```
HEAD detached at bed5a32
```

```
nothing to commit, working tree clean
```

PETSc information :

-----

```
* WARNING * This information may be incorrect if you have more than one  
version of petsc installed.
```

```
PETSC_DIR : /home/travis/build/pflotran/pflotran/petsc
```

```
PETSC_ARCH : petsc-arch
```

```
petsc repository status :
```

```
$ git log -1 HEAD
```

```
commit 03c0fad46595e0cd29635b9188b232e623499437
```

```
Author: Barry Smith <bsmith@mcs.anl.gov>
```

```
Date: Thu Mar 30 14:27:53 2017 -0500
```

Add hack to get PETSc with gfortran to work on Apple with xcode 8.3

# pflotran-tests-2016-07-29\_10-27-50.testlog

```
-----
543_flow-np8...
  cd /home/travis/build/pflotran/pflotran/regression_tests/default/543
  /home/travis/build/pflotran/pflotran/petsc/petsc-arch/bin/mpiexec -np 8
/home/travis/build/pflotran/pflotran/src/pflotran/pflotran -malloc 0 -successful_exit_code 86 -input_prefix
543_flow-np8
  # 543_flow-np8 : run time : 2.91 seconds
  diff 543_flow-np8.regression.gold 543_flow-np8.regression
  Skipping SOLUTION : Flow
543_flow-np8... passed.
-----
543_hanford_srfcplx_param...
  cd /home/travis/build/pflotran/pflotran/regression_tests/default/543
  /home/travis/build/pflotran/pflotran/src/pflotran/pflotran -malloc 0 -successful_exit_code 86 -input_prefix
543_hanford_srfcplx_param
  # 543_hanford_srfcplx_param : run time : 3.11 seconds
  diff 543_hanford_srfcplx_param.regression.gold 543_hanford_srfcplx_param.regression
  FAIL: LIQUID VELOCITY [m/d]:1 : 1.084136795e-11 > 1e-12 [relative]
  FAIL: LIQUID VELOCITY [m/d]:1 : 7.63191790473e-11 > 1e-12 [relative]
  FAIL: LIQUID VELOCITY [m/d]:1 : 2.060265741e-11 > 1e-12 [relative]
  FAIL: LIQUID VELOCITY [m/d]:31 : 7.3779567027e-12 > 1e-12 [relative]
  FAIL: LIQUID VELOCITY [m/d]:31 : 1.76111798338e-12 > 1e-12 [relative]
  FAIL: LIQUID VELOCITY [m/d]:29 : 2.25552127701e-12 > 1e-12 [relative]
  FAIL: LIQUID VELOCITY [m/d]:29 : 1.32019648478e-12 > 1e-12 [relative]
  FAIL: LIQUID VELOCITY [m/d]:29 : 1.61796082447e-11 > 1e-12 [relative]
  FAIL: UO2(PO3)2 SI:Min : 4.39851271218e-12 > 1e-12 [relative]
  FAIL: UO2S04 SI:Min : 4.38535476851e-12 > 1e-12 [relative]
  FAIL: Torbernite SI:Min : 8.80326710311e-12 > 1e-12 [relative]
  FAIL: (UO2)3(PO4)2.4H2O SI:Min : 1.31617137493e-11 > 1e-12 [relative]
  FAIL: UO2CO3 SI:Min : 4.38491638053e-12 > 1e-12 [relative]
  FAIL: UO3.0.9H2O(alpha) SI:Min : 4.37498338731e-12 > 1e-12 [relative]
  FAIL: Metatorbernite SI:Min : 8.79872832628e-12 > 1e-12 [relative]
  ...
  FAIL: Schoepite SI:Min : 4.38503889665e-12 > 1e-12 [relative]
543_hanford_srfcplx_param... failed.
-----
```

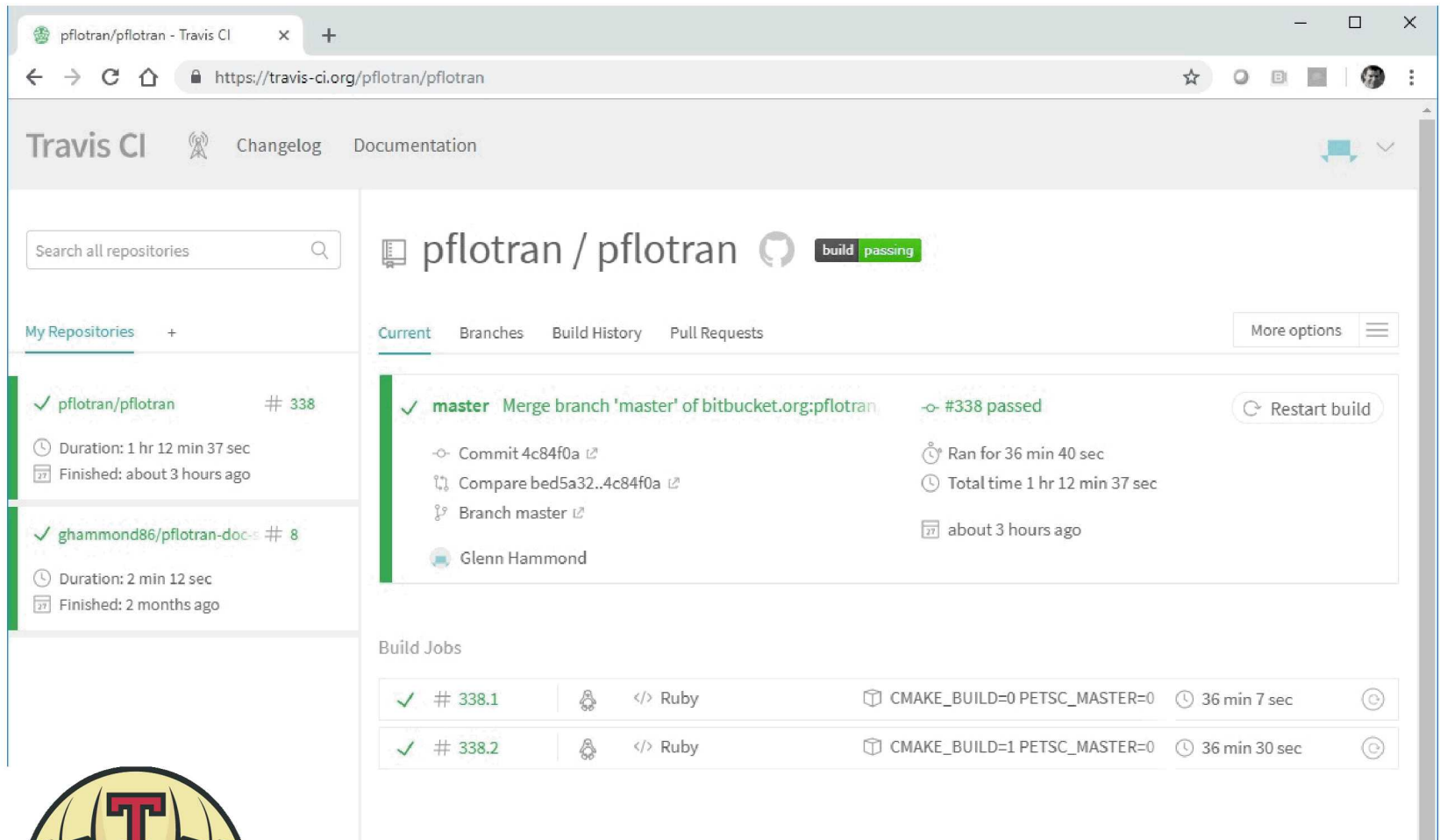
# Testing (cont.)

- Example regression test failure
  - Revert back to original value

```
subroutine EOSWaterDensityIFC67()
```

```
    tc1 = H2O_CRITICAL_TEMPERATURE      ! K  
-   pc1 = H2O_CRITICAL_PRESSURE + 1.d-10*H2O_CRITICAL_PRESSURE    ! Pa  
+   pc1 = H2O_CRITICAL_PRESSURE        ! Pa  
    vc1 = 0.00317d0    ! m^3/kg
```

# Bitbucket → Github → Travis CI



The screenshot shows the Travis CI web interface for the repository `pflotran/pflotran`. The page displays a search bar, navigation links for Changelog and Documentation, and a sidebar with "My Repositories" listing `pflotran/pflotran` (# 338) and `ghammond86/pflotran-docs` (# 8). The main content area shows the current build status as "passing" for the `master` branch. The build details include a commit hash `4c84f0a`, a merge of the `master` branch from Bitbucket, and a duration of 36 minutes and 40 seconds. The build jobs section shows two jobs, # 338.1 and # 338.2, both using Ruby and CMAKE, with durations of 36 minutes and 7 seconds, and 36 minutes and 30 seconds, respectively.

