slides

September 19, 2022

##

Workshop

#

Testing

##

Chris Edsall

1 Why

- reproducability crisis
- retractions
- gives collaborators confidence
- attracts more users => enhanced reputation
- finds bugs earlier in the SDLC ("shift left")

2 What

3 Science Test Cases

- e.g. "Does this formulation of the climatge model, when spun up and run for 20 model years produce the quasi- biannial oscilartion in the right place and time?"
- important, but not the focus of this workshop

4 Integration tests

- Tests the application end to end
- Can be time consuming

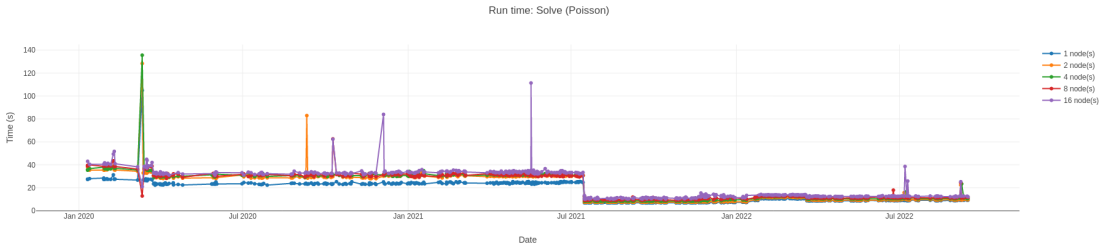
5 Unit Tests

- Test the smallest sensible unit, e.g. a **single function**
- Usually much simpler, constrained test cases
- Easier to reason about the test cases
- (Ideally) **runs quickly**
- Can be used for Test Driven Development
  - When you write the test before you write the code that implements it
  - You know when to stop coding!
- Unit testing implies the code is sufficiently modular

6 Regression Tests

- make sure things *still* work the way they should
- detect
  - regressions in **correctness** (common to reintroduce an old bug when fixing a new one)
  - regressions in **perfromance**

e.g. https://fenics.github.io/performance-test-results/



7 Other types

7.1 Fuzz Testing

- feed the software under test **random input** or **syntacticaly correct but erroneous** data
- mostly used in security

7.2 Black Box

- "Golden outputs"

7.3 Property Based Testing

8 Coverage

- What fraction of the lines of code are run when running a test suite
- Diminishing returns when aiminig for 100%

9 Test Frameworks

- You can use bare asserts
- frameworks provide "affordances"
  - convenience functions like `assert_almost_equal`
  - grouping tests
  - running subsets of tests
  - providing "fixtures"
    - eg setting up and tearing down a populated database

10 Common Test Frameworks

10.1 Python

- pytest
- (`Unittest` is builtin, but don't use it)

10.2 C++

- Catch2
- Google Test

10.3 Fortran

- FIXME: don't have a recomendation

11 Pytest

12 Exercise 0 - Install Pytest

12.1 You might already have it!

```
pytest -v
```

12.2 pip / venv

12.3 conda

13 Exercise 1 - Run Pytest

14 Exercise 2 - Add a test

15 Exercise 3 - Add an expected fail test

16 Exercise 4 - Add a test with a fixture

17 Exercise 5 - Install Catch2

- Catch2 can be used as a header only library, so it is sufficient to download the `.hpp` file and add it to your project
- FIXME: Check licensing

- code coverage (in python)
- forward announce github CI (wednesday morning)
- mention property based testing
- where to get the source to be tested
- hints for pytest
  - `--pdb`
- reframe for HPC testing
- IDEs supporting testing