



A **UT**/ORNL PARTNERSHIP
NATIONAL INSTITUTE FOR COMPUTATIONAL SCIENCES



Regression Testing on Petaflop Computational Resources

Mike McCarty

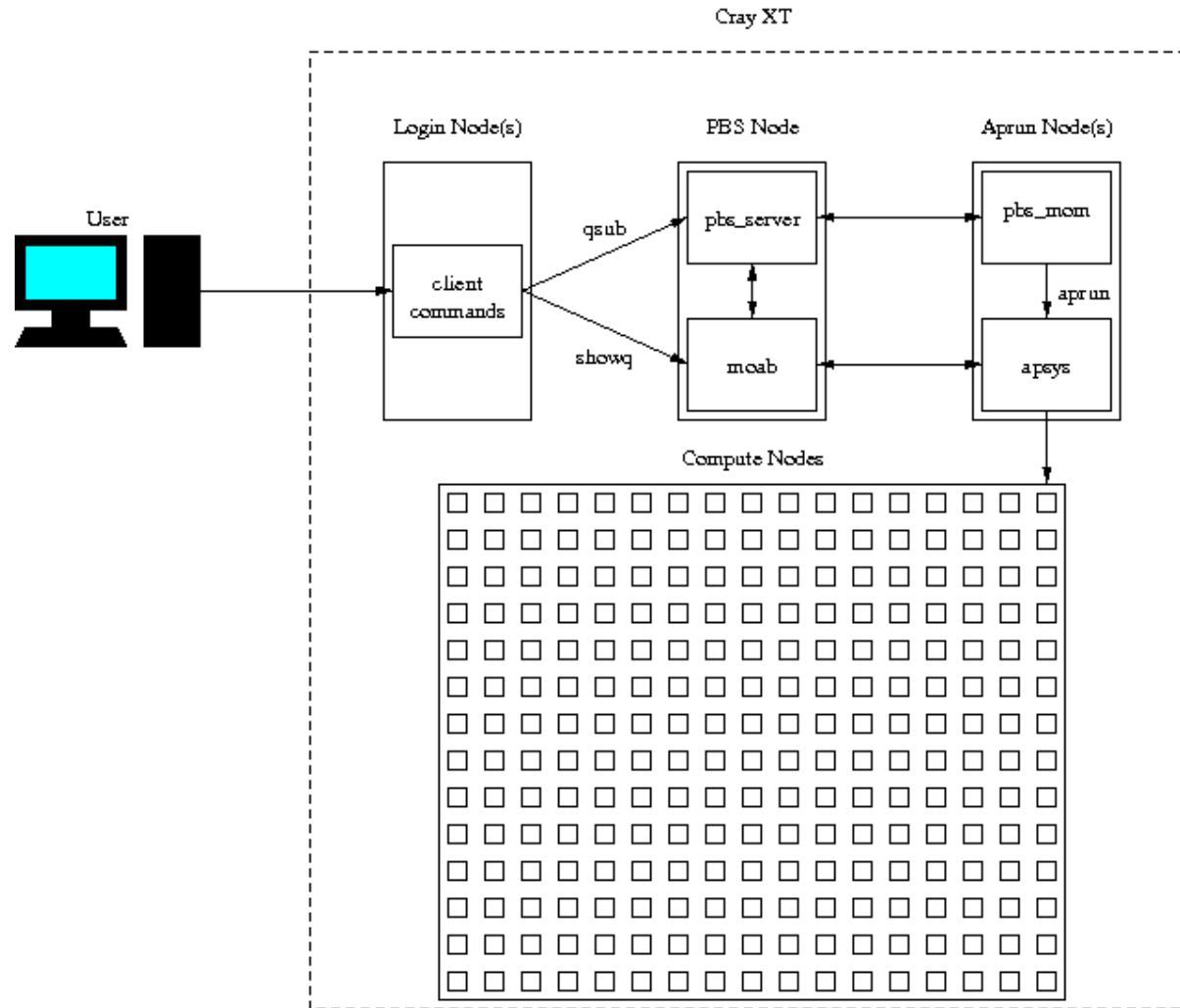
Software Developer

May 27, 2010

What? and Why?

- **What is regression testing?**
 - Regression testing is any type of software testing that seeks to uncover software errors by partially retesting a modified program *or system*.
- **Why should we do regression testing?**
 - To track how the performance of a system changes overtime.
 - To test the system after maintenance to make sure it is ready for production.

Depth of testing (Where?)



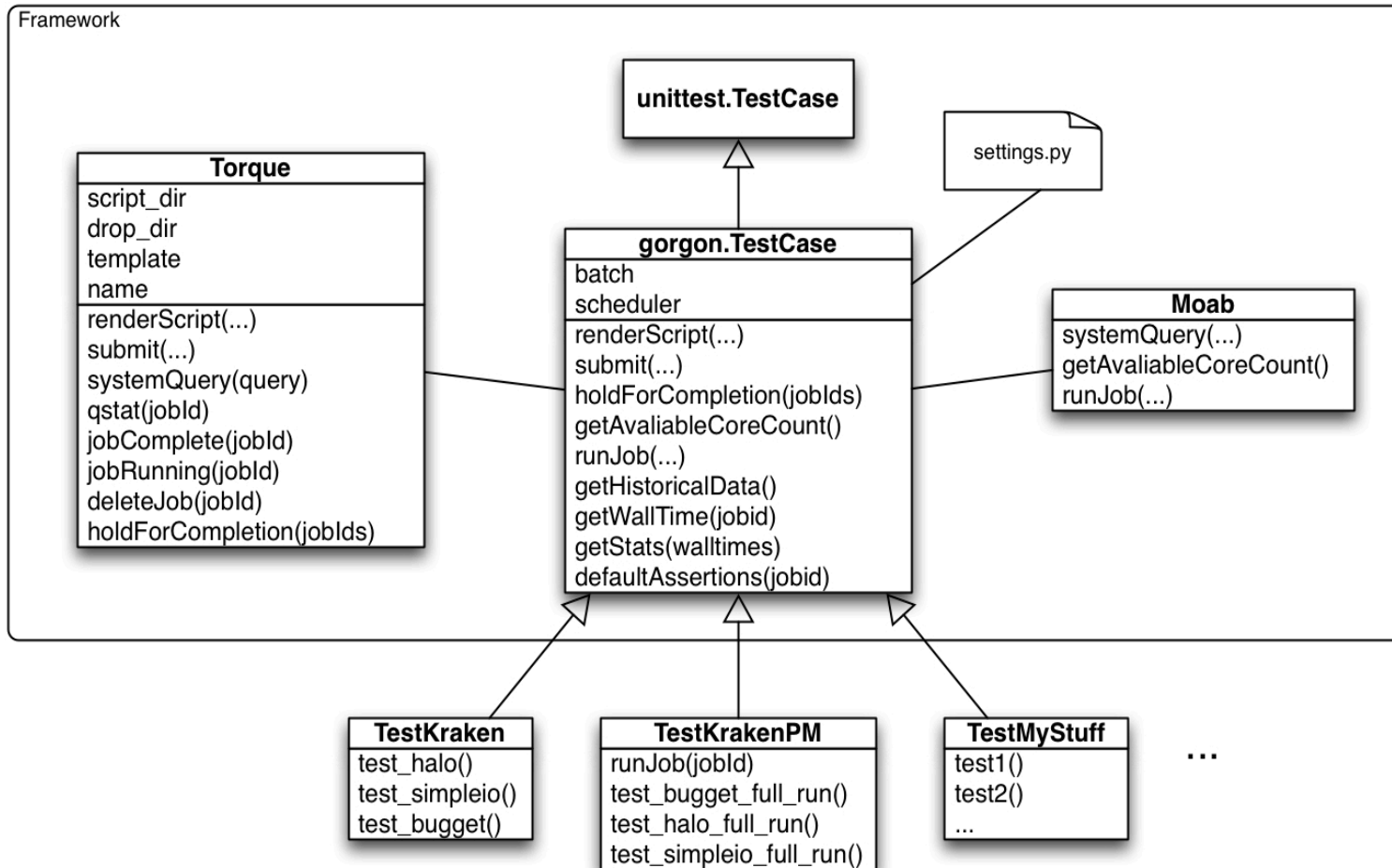
Automate Testing

- **Regression testing can be a labor intensive task.**
 - Testing how scaling curves change for various applications over time
 - Comparing recent test results to previous results; while performing maintenance on a system.
- **Rerunning test periodically to build up regression data over time.**
- **Store result data in a convenient format for reporting.**
- **Automatically generate plots and reports.**

The Framework

- **Automates the role of the user**
 - Test specification
 - PBS script rendering
 - Job(s) submission
 - Test assertions based on output or end state
- **Post processing and report generation automation less trivial since they are application specific.**

Framework Architecture

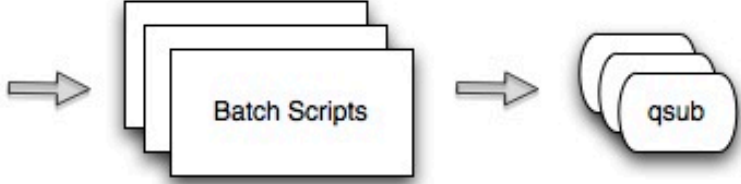


PBS Script Rendering Parameters

Name	Description	Default
size	The number of processors for the job to run on.	12
machine	The name of the machine that the test will be ran on.	None
name	The name of the executable.	None
project	The name of the project to charge the job too.	None
walltime	The walltime limit for the job.	00:10:00
pbs_additions	A raw string for specifying custom PBS variables.	Empty String
env_vars	A raw string for specifying environment variables.	Empty String
preamble	A raw string for specifying a preamble of code that is inserted into the PBS script before the aprun command is issued.	Empty String
aprun_options	Options for the aprun command.	"-n \$PBS_NNODES"
options	Options for the application's executable.	
profile	Boolean that controls whether or not to look for a profile from FPMPI.	True

How does it work?

```
class TestKraken(unittest.TestCase):  
  
    def setUp(self):  
        self.batch = Torque(script_dir, drop_dir)  
  
    def test_halo(self):  
        jobids = []  
        for size in range(1056, 16000, 1056)[:2]:  
            self.batch.render_script(size = size  
                                    , name = "halo"  
                                    , walltime = "00:10:00"  
                                    )  
  
            jobids.append(self.batch.submit())  
        self.batch.hold_for_completion(jobids)
```



```
mmccarty@krakenpf6(XT5):/lustre/scratch/mmccarty/regression_tests> python ~/sandbox/gorgon/regression_tests/trunk/test.py  
Running tests...  
.  
-----  
Ran 1 test in 153.490s  
  
OK  
mmccarty@krakenpf6(XT5):/lustre/scratch/mmccarty/regression_tests> █
```

```
mmccarty@krakenpf6(XT5):~> qstat | grep mmccarty  
400937.nid00016      Halo      mmccarty      0 R medium  
400938.nid00016      Halo      mmccarty      00:00:00 R medium
```


Assertions

```
def test_bugget(self):
    jobids = []
    for size in range(1056, 16000, 1056)[:2]:
        self.batch.renderScript(size = size
                                , name = "bugget"
                                , walltime = "00:15:00"
                                , options = "-f"
                                , profile = False
                                )

        jobids.append(self.batch.submit())
    self.batch.waitForCompletion(jobids)

    jobs_failed = []
    verbose = False
    for jobid in jobids:
        output = open("Bugget.o" + jobid, 'r')
        fail = False
        for l in output.readlines():
            if "3.0-sigma" in l:
                line_data = l.split()
                if line_data[1] != "0:":
                    jobs_failed.append(jobid)
                    fail = True
                    print "JobID %s: %s processes outside of 3.0 sigma." % (jobid, line_data[1].replace(":", ''))
            if "rank" in l and fail and verbose:
                print l.replace('\n', '')
    self.assertTrue(len(jobs_failed) == 0)
```

Assertion from command line

```
mmccarty@krakenpf6(XT5):/lustre/scratch/mmccarty/regression_tests> python ~/sandbox/gorgon/regression_tests/trunk/test.py
Running tests...
JobID 400918: 1 processes outside of 3.0 sigma.
F..
=====
FAIL: test_bugget (__main__.TestKraken)
-----
Traceback (most recent call last):
  File "/nics/a/home/mmccarty/sandbox/gorgon/regression_tests/trunk/test.py", line 62, in test_bugget
    self.assertTrue(len(jobs_failed) == 0)
AssertionError

-----
Ran 3 tests in 554.473s

FAILED (failures=1)
mmccarty@krakenpf6(XT5):/lustre/scratch/mmccarty/regression_tests> █
```

Running tests

```
mmccarty@krakenpf7(XT5):/lustre/scratch/mmccarty/regression_tests> python -/sandbox/gorgon/regression_tests/trunk/TestKrakenPM.py
Running tests...
.EF
=====
ERROR: test_halo_full_run (__main__.TestKrakenPM)
-----
Traceback (most recent call last):
  File "/nics/a/home/mmccarty/sandbox/gorgon/regression_tests/trunk/TestKrakenPM.py", line 37, in test_halo_full_run
    raise
TypeError: exceptions must be classes, instances, or strings (deprecated), not NoneType

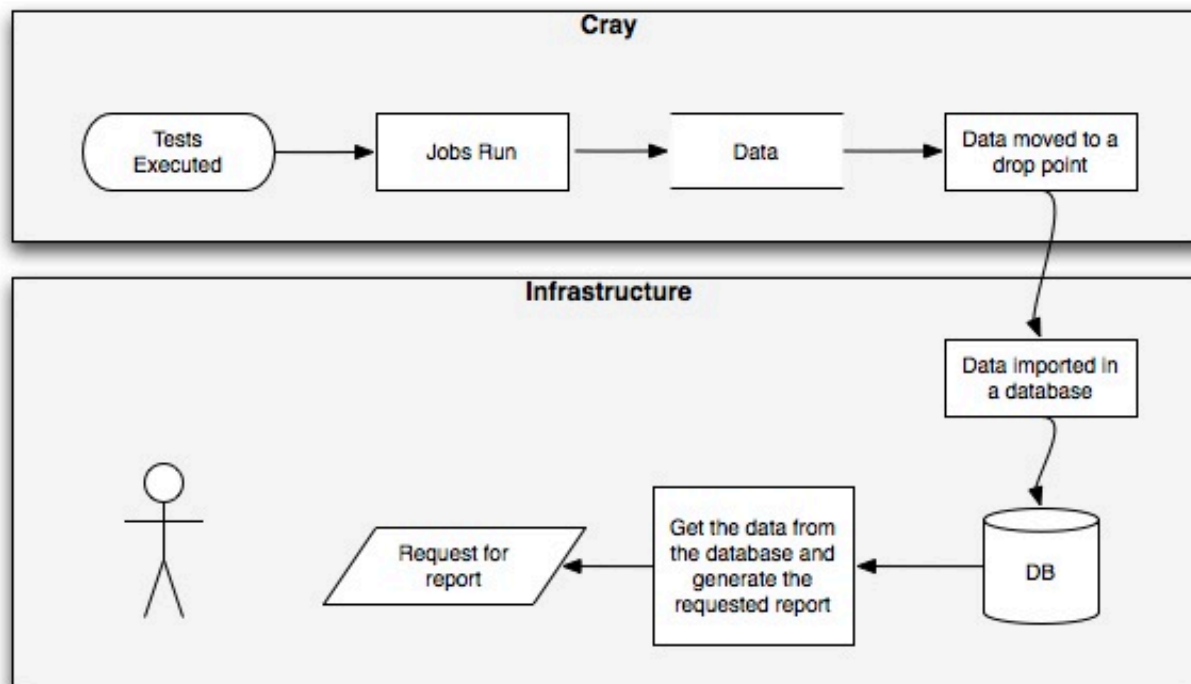
=====
FAIL: test_simpleio_full_run (__main__.TestKrakenPM)
-----
Traceback (most recent call last):
  File "/nics/a/home/mmccarty/sandbox/gorgon/regression_tests/trunk/TestKrakenPM.py", line 52, in test_simpleio_full_run
    self.assertTrue(False)
AssertionError

-----
Ran 3 tests in 2.922s

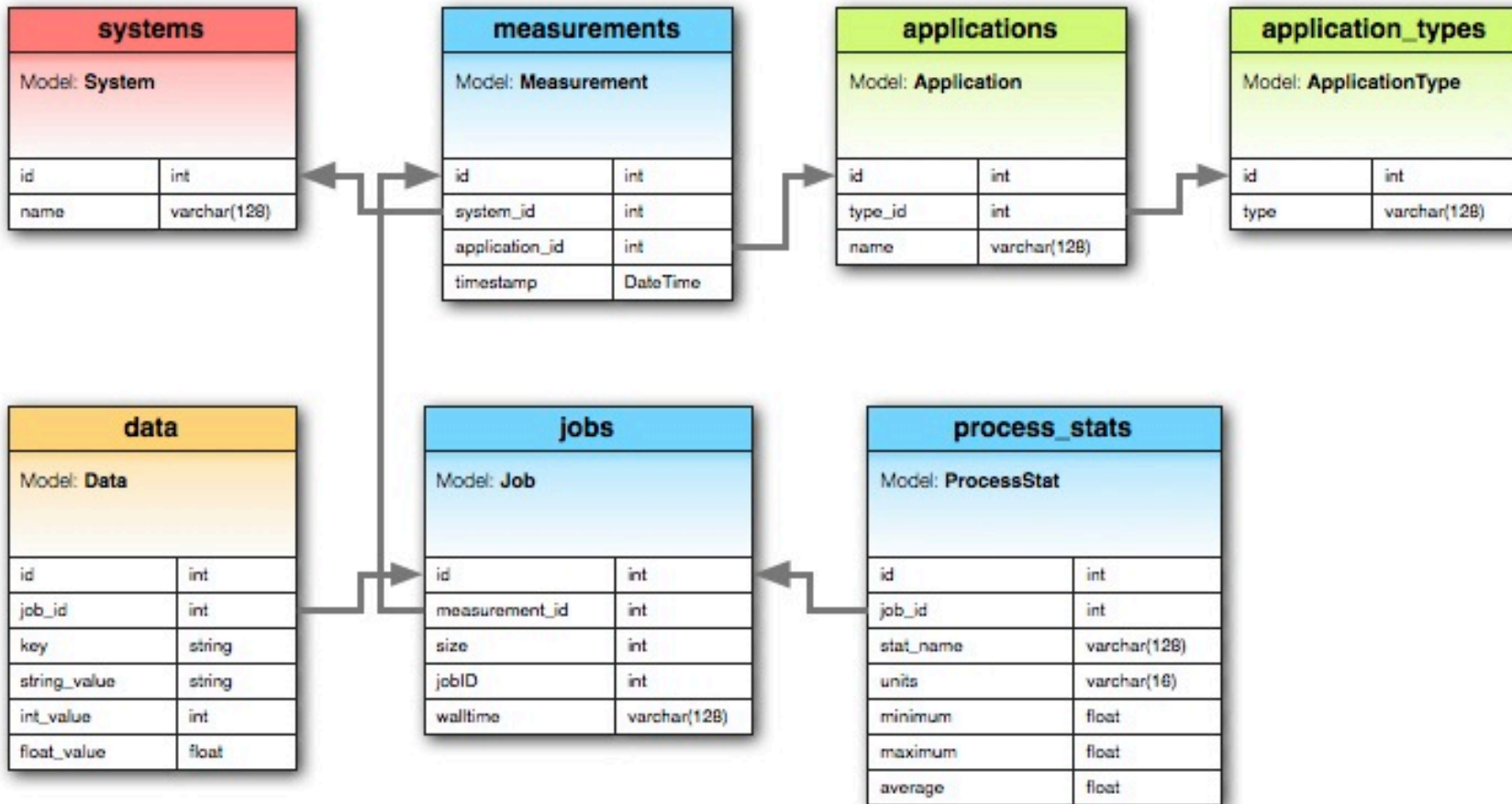
FAILED (failures=1, errors=1)
mmccarty@krakenpf7(XT5):/lustre/scratch/mmccarty/regression_tests> █
```

Post Processing

- **Some tests may require additional post processing**
- **“Standard” post processing can be automated**

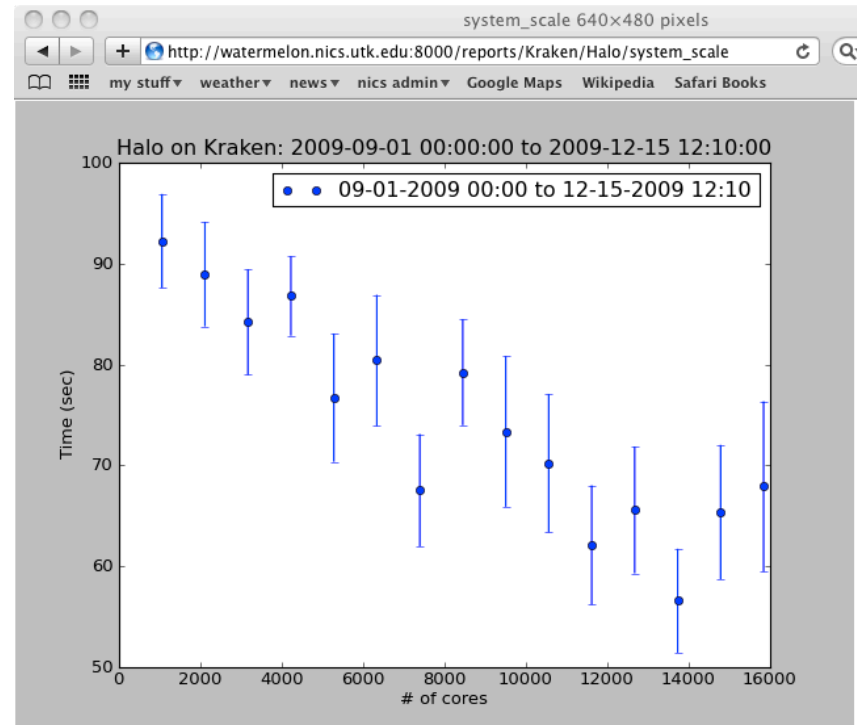


Database Structure

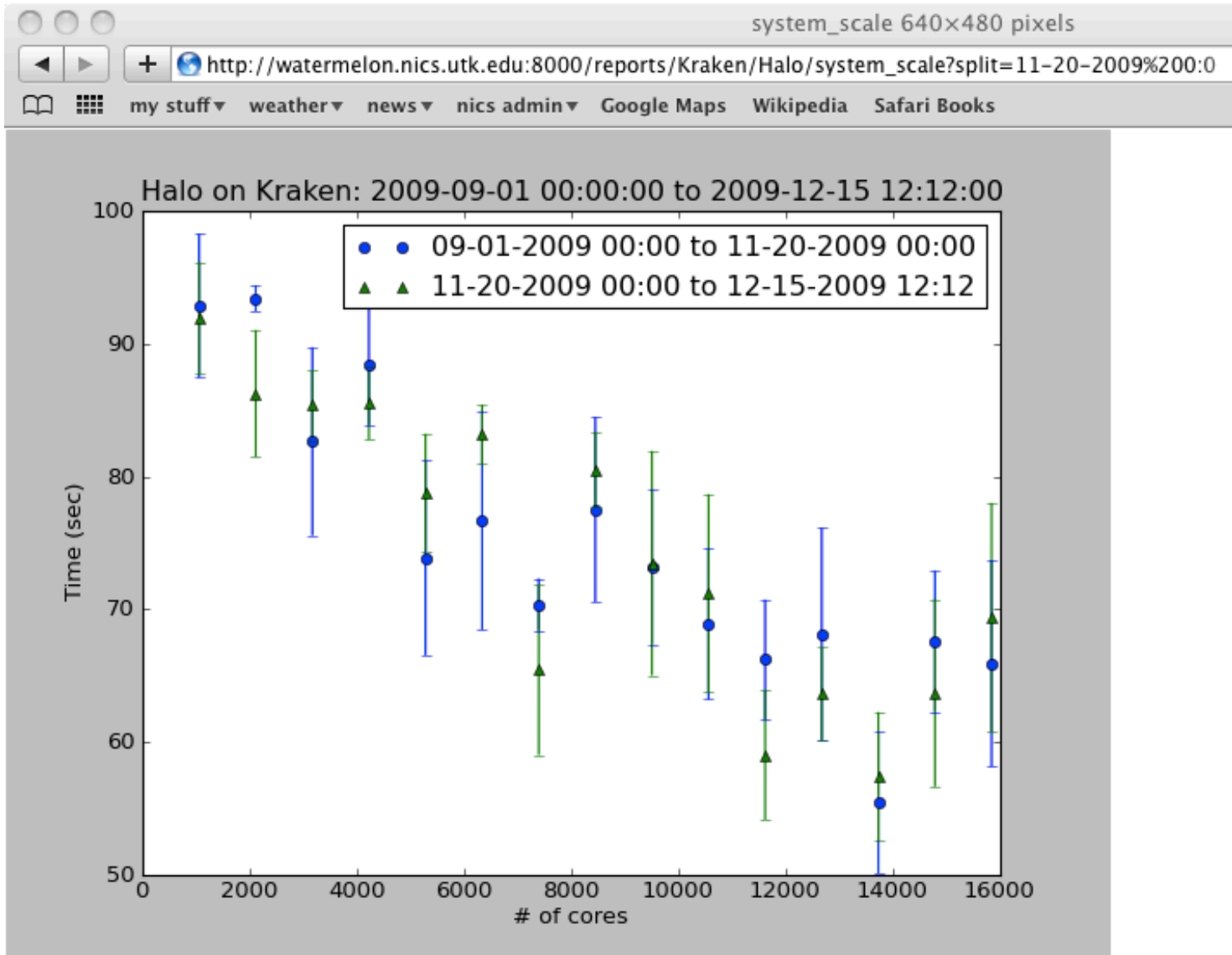


Plotting results

- Plots viewed in a RESTful web interface written in Django
- Plots are generated on the fly using data stored in the database



Plotting (continued)



Thank You