SCALING THE GVL

Evaluating Galaxy's scalability characteristics on the cloud (and other adventures)

Nuwan Goonasekera

Overview

- GVL Quality Assurance
  - Automated QA tests
- GVL Scaling
  - Many QA tests in parallel = Scalability test

Quality Assurance

Why?

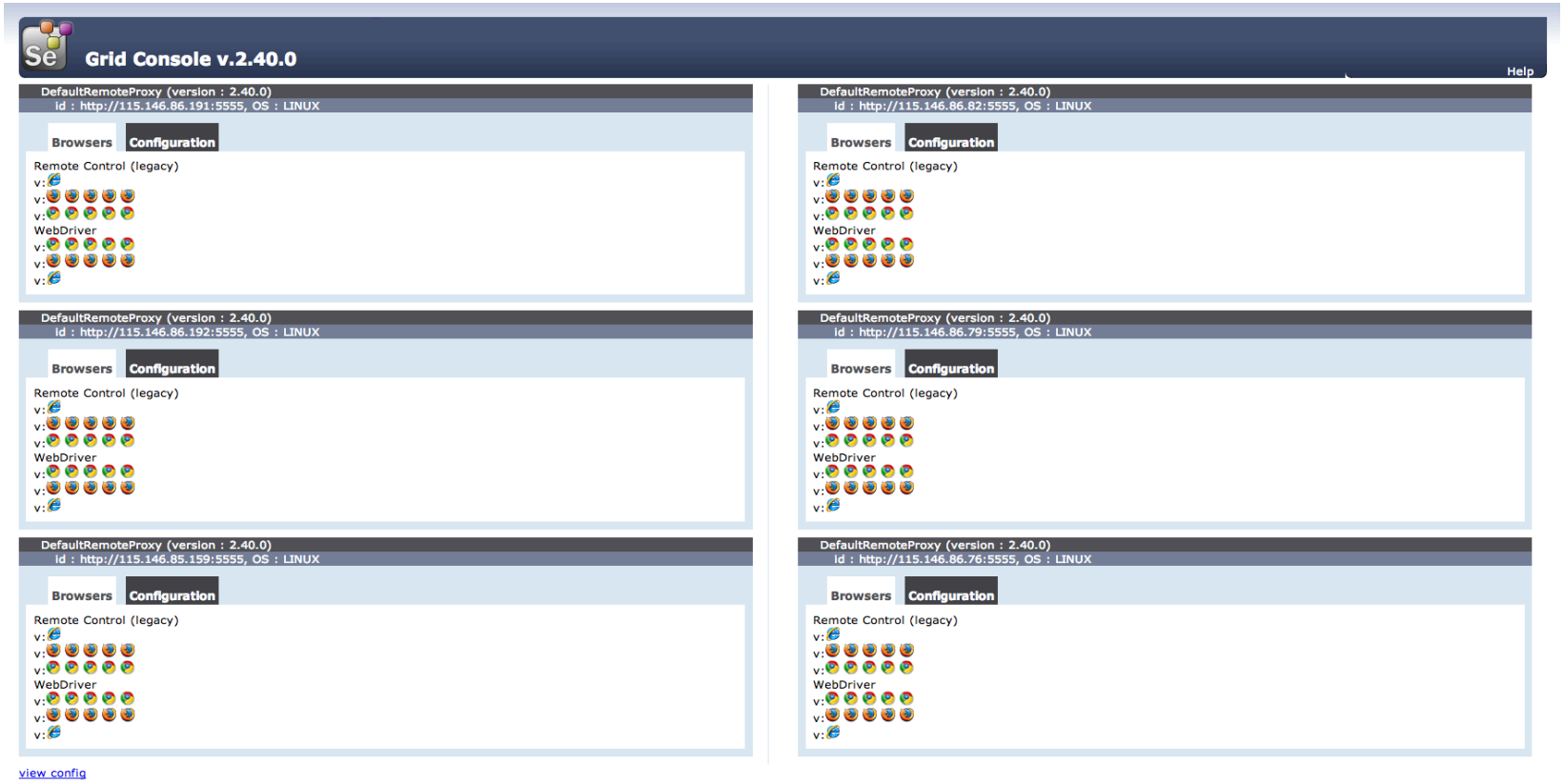
- Tedious and error prone to check manually
- Need a quick way of knowing things are in reasonable shape
- An end-user perspective on how things work

How?

- Using Selenium
- Run full workflows that exercise a complete set of tools
- Check whether tool output == expected output
- Also exercise typical use cases in UI

Demo

- Selenium in action



Writing a simple test

```
1  from gvl_test_base import GVLTestBase
2  from selenium_snippets.galaxy.history import History
3  from selenium_snippets.galaxy.get_data import GetData
4  from selenium_snippets.galaxy.rna_analysis import RNAAnalysis
5  from selenium_snippets.galaxy import snippet_base
6
7  class SimpleTest(GVLTestBase):
8
9      def __init__(self, galaxy_test_context):
10         super(SimpleTest, self).__init__(galaxy_test_context)
11
12     @snippet_base.ui_action()
13     def execute_gvl_testcase(self):
14         history = History(self.context)
15         history.create_new_history("Hello world")
16         file = "https://swift.rc.nectar.org.au:8888/v1/AUTH_377/public/RNAseqDGE_BASIC/C2_R1.chr4.fq"
17         GetData(self.context).run_upload_file('url', file, 'fastqsanger')
18         history.wait_for_datasets_to_finish()
19         RNAAnalysis(self.context).run_tophat(file, "D melanogaster (dm3)")
20         history.wait_for_datasets_to_finish()
21
```

Notes

- Some elements have no ids = brittle xpaths

```
@snippet_base.ui_action()
def edit_history_item(self, name):
    driver = self.driver
    self.switch_to_galaxy_history_frame()
    driver.find_element_by_xpath("//div[@class='dataset-primary-actions'][following-sibling::div[contains(., '"
                                 + name
                                 + "')]]//a[contains(@data-original-title, 'Edit attributes')]").click()
    self.switch_to_galaxy_outer_frame()
    self.wait_for_galaxy_content_frame()
```

- Similar to Galaxy's Twill based tests of internal API
- Bioblend has its own (similar) API
- Would it make sense to converge these into shared tests?
  - E.g. Write one test case in bioblend, switch implementation to run against each layer?
- http://buildbot.genome.edu.au/

Performance Testing

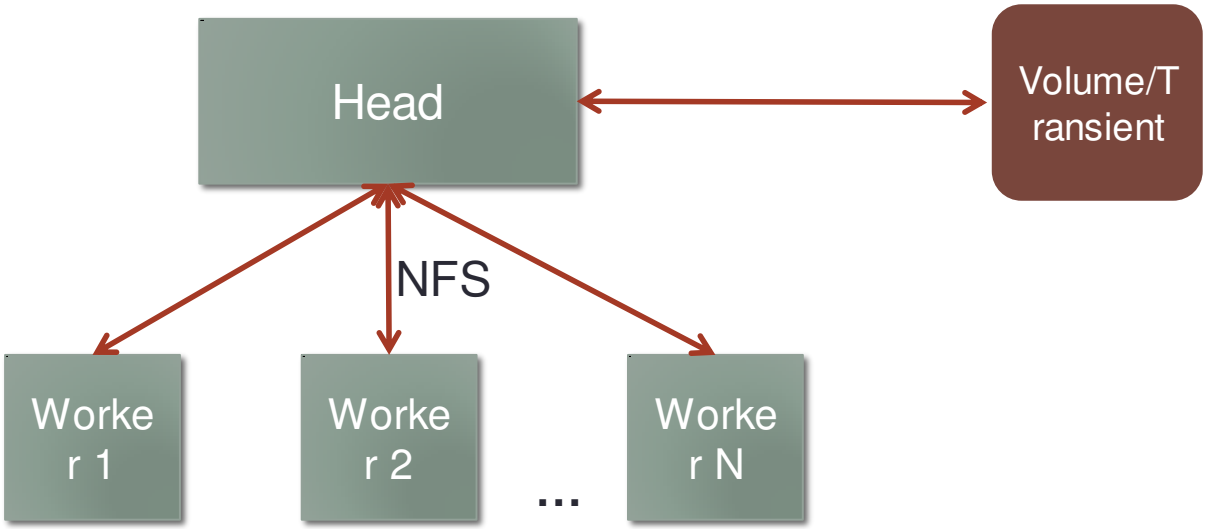
Why?

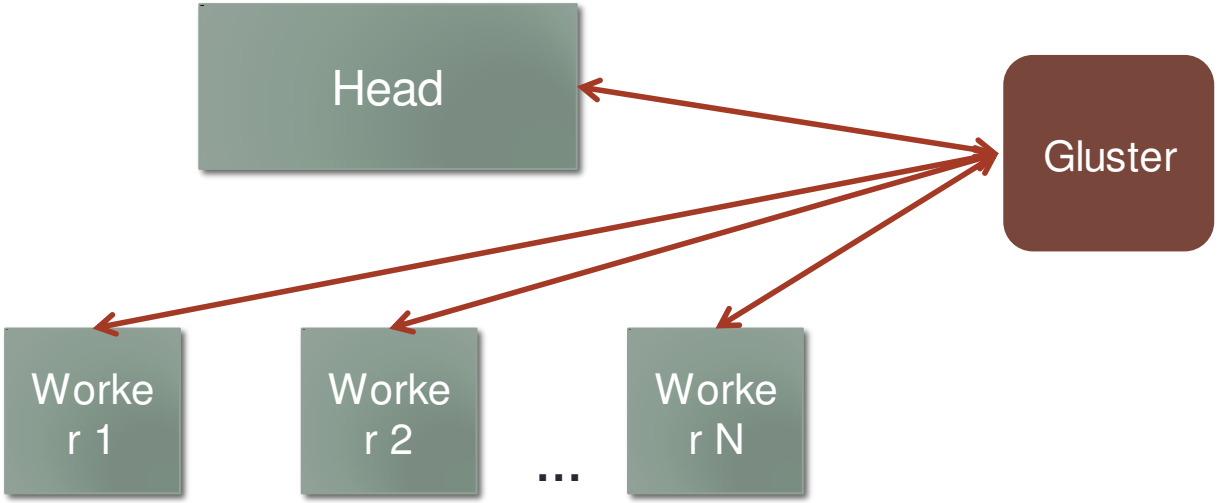
- How does the GVL scale for different workloads?
- What combinations of storage, instance types, workers etc. are recommended?

How?

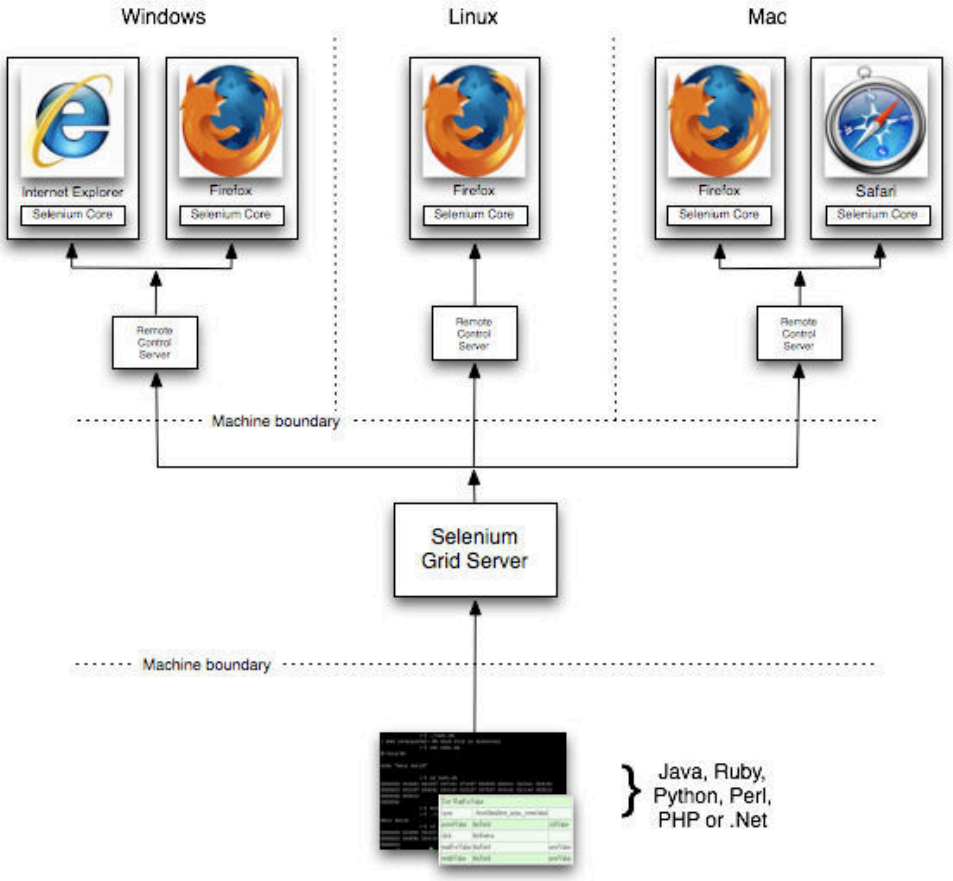
- 1 thread = 1 user = QA
- Many threads = Multiple Users = Performance
- Use Selenium Grid

Bottlenecks





Selenium Grid



Combinations

- storage_type = { gluster, transient, volumes etc. }
- machine_type = { m1.small, m1.medium, m1.large, m1.xlarge, m1.xxlarge }
- workers = { 1 to x workers }
- workloads = { rnaseq_basic tutorial, deseq basic tutorial etc. }
- simultaneous users = { 1, 5, 10, 20, 30 }

Combinations (contd..)

and more…

- Gluster vs NFS vs PVFS/OrangeFS vs …?
- Amazon/EC2 vs NeCTAR/Openstack?
- No. of web runners?
- Job Handlers?
- Nginx workers?

Combinations to test

```
for storage_type in storage_types: (4)
        for machine_type in machine_types: (5)
                for worker in workers: (5)
                        for workload in workloads: (5)
                                for user in number_of_users: (5)
                                        time_stuff()
```

Can get out of control.

Trimmed to a little less than a 1000 combinations.

What it records

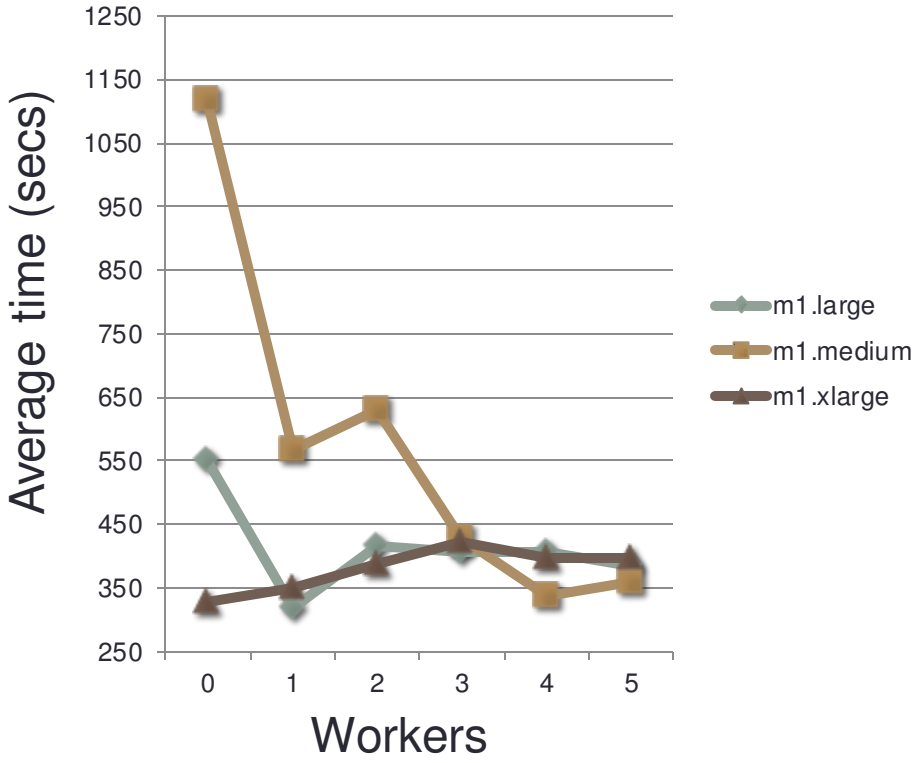
- Time taken for each segment of the test


```
{
    "timing": [{
        "child_actions": [{
            "child_actions": [],
            "action_start": 1395453178.595498,
            "action_type": 1,
            "action_end": 1395453186.418102,
            "action_name": "register_and_login"
        }, {
            "child_actions": [{
                "child_actions": [{
                    "child_actions": [],
                    "action_start": 1395453186.418553,
                    "action_type": 1,
                    "action_end": 1395453189.653837,
                    "action_name": "create_new_history"
                }, {
                    "child_actions": [],
                    "action_start": 1395453189.654075,
                    "action_type": 1,
                    "action_end": 1395453191.343987,
                    "action_name": "run_upload_file"
                }, {
                    "child_actions": [],
                    "action_start": 1395453191.344252,
                    "action_type": 1,
```


- Records atop logs at 10 second intervals
  - Provides snapshot of CPU, memory, process and network usage

Results

Average of time	Machine Size		
Workers	m1.large	m1.medium	m1.xlarge
0	551.41	1117.37	326.14
1	317.53	567.29	349.18
2	416.61	629.45	385.85
3	406.30	428.62	420.55
4	405.38	335.77	397.55
5	385.21	358.32	397.56



Repository

https://bitbucket.org/gvl/gvl-stress-test