edX Load Tests

Introduction And Current Status

Troy Sankey

September 5, 2017

Introduction

- Tests live in the edx/edx-load-tests repo.

Introduction

- Tests live in the edx/edx-load-tests repo.
- Uses the Locust load testing tool.

Why Load Test?

- Performance-related improvements.

Why Load Test?

- Performance-related improvements.
- Performance-sensitive code changes.
- Major version changes of dependencies.

Overview Of Writing Tests

1. Identify endpoints.

Overview Of Writing Tests

1. Identify endpoints.
2. Write tasks for each endpoint.

Overview Of Writing Tests

1. Identify endpoints.
2. Write tasks for each endpoint.
3. Write startup (`on_start()`) function.

Overview Of Writing Tests

1. Identify endpoints.
2. Write tasks for each endpoint.
3. Write startup (`on_start()`) function.
4. Write instructions for seeding test data.

locustfile.py

```
class ChatTasks(TaskSet)
    @task(3)
    def send_message():
        print("sending a message.")
    @task(8)
    def get_emoji():
        print("getting list of emoji.")

class ServiceTasks(TaskSet):
    tasks = {
        ChatTasks: 1,
        SearchTasks: 2,
    }

class ServiceUser(HttpLocust):
    task_set = ServiceTasks
    min_wait = 1000
    max_wait = 5000
```

locustfile.py

```python
class ChatTasks(TaskSet)
    @task(3)
    def send_message():
        print("sending a message.")
    @task(8)
    def get_emoji():
        print("getting list of emoji.")

class ServiceTasks(TaskSet):
    tasks = {
        ChatTasks: 1,
        SearchTasks: 2,
    }

class ServiceUser(HttpLocust):
    task_set = ServiceTasks
    min_wait = 1000
    max_wait = 5000
```

locustfile.py

```python
class ChatTasks(TaskSet)
    @task(3)
    def send_message():
        print("sending a message.")
    @task(8)
    def get_emoji():
        print("getting list of emoji.")

class ServiceTasks(TaskSet):
    tasks = {
        ChatTasks: 1,
        SearchTasks: 2,
    }

class ServiceUser(HttpLocust):
    task_set = ServiceTasks
    min_wait = 1000
    max_wait = 5000
```

User Experience

- No browser, no javascript, no good measure of user experience.
- Focus on the system response, not the users.

Settings

- Each load test is configurable via a YAML file.

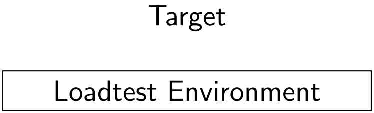
Settings

- Each load test is configurable via a YAML file.
- Use `helpers.settings`.

Settings

- Each load test is configurable via a YAML file.
- Use `helpers.settings`.
- The Jenkins job (run-simple-loadtest) will use settings specified by a job parameter.

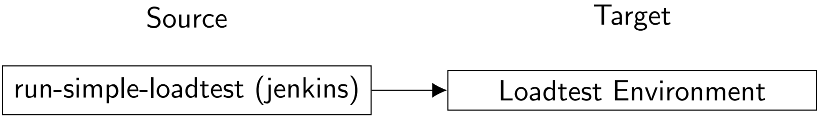
Overview Of Running Load Tests



Target

Loadtest Environment

Overview Of Running Load Tests



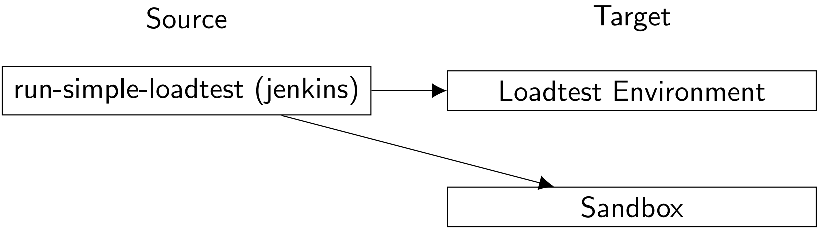
Source

run-simple-loadtest (jenkins)

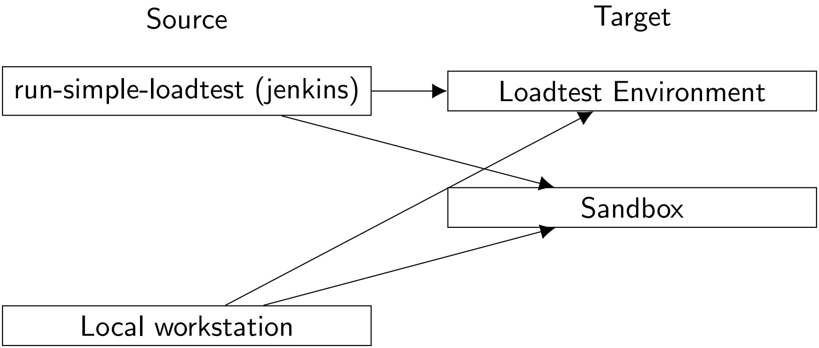
Target

Loadtest Environment

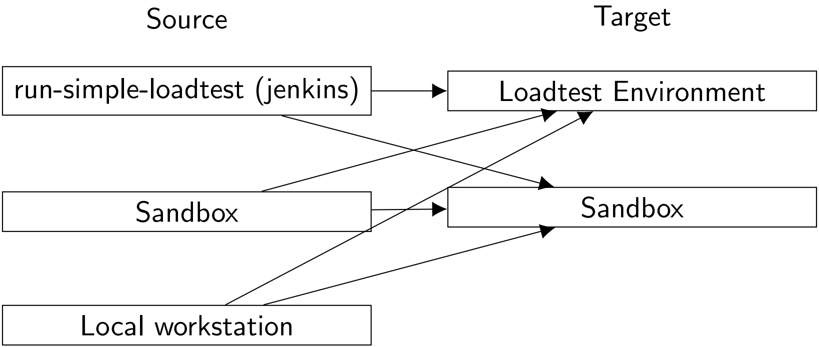
Overview Of Running Load Tests



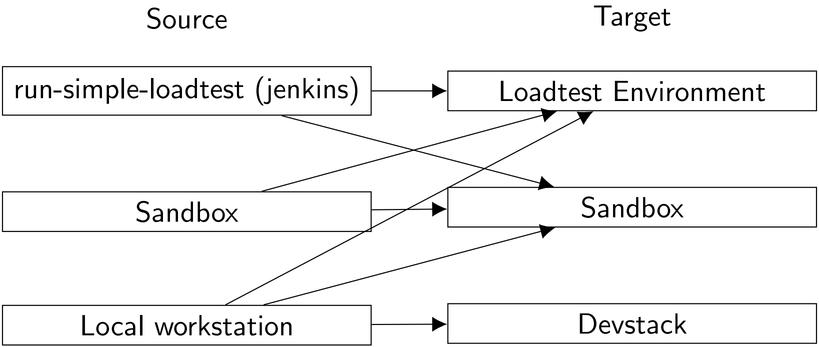
Overview Of Running Load Tests



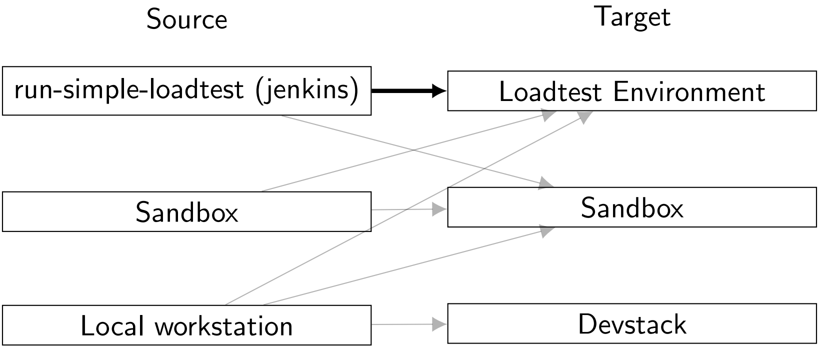
Overview Of Running Load Tests



Overview Of Running Load Tests



Overview Of Running Load Tests



Time Investment For Running Tests

- run-simple-loadtest Instructions:

Time Investment For Running Tests

- run-simple-loadtest Instructions:
  1. Fill in build parameters and click build.

Time Investment For Running Tests

- run-simple-loadtest Instructions:
  1. Fill in build parameters and click build.
  2. Wait 1 minute for an ec2 instance to launch and run your test.

Time Investment For Running Tests

- run-simple-loadtest Instructions:
  1. Fill in build parameters and click build.
  2. Wait 1 minute for an ec2 instance to launch and run your test.
- However, there's more to it than just running the test:

Time Investment For Running Tests

- run-simple-loadtest Instructions:
  1. Fill in build parameters and click build.
  2. Wait 1 minute for an ec2 instance to launch and run your test.
- However, there's more to it than just running the test:
  - How difficult is it to seed your test data?

Time Investment For Running Tests

- run-simple-loadtest Instructions:
  1. Fill in build parameters and click build.
  2. Wait 1 minute for an ec2 instance to launch and run your test.
- However, there's more to it than just running the test:
  - How difficult is it to seed your test data?
  - Do you need to create/modify a locust task?

Time Investment For Running Tests

- run-simple-loadtest Instructions:
  1. Fill in build parameters and click build.
  2. Wait 1 minute for an ec2 instance to launch and run your test.
- However, there's more to it than just running the test:
  - How difficult is it to seed your test data?
  - Do you need to create/modify a locust task?
  - How many tests do you need to run?

Time Investment For Running Tests

- run-simple-loadtest Instructions:
  1. Fill in build parameters and click build.
  2. Wait 1 minute for an ec2 instance to launch and run your test.
- However, there's more to it than just running the test:
  - How difficult is it to seed your test data?
  - Do you need to create/modify a locust task?
  - How many tests do you need to run?
  - How long do the tests need to run?

Time Investment For Running Tests

- run-simple-loadtest Instructions:
  1. Fill in build parameters and click build.
  2. Wait 1 minute for an ec2 instance to launch and run your test.
- However, there's more to it than just running the test:
  - How difficult is it to seed your test data?
  - Do you need to create/modify a locust task?
  - How many tests do you need to run?
  - How long do the tests need to run?
  - How easy is it to infer test outcomes?

Time Investment For Running Tests

- run-simple-loadtest Instructions:
    1. Fill in build parameters and click build.
    2. Wait 1 minute for an ec2 instance to launch and run your test.
- However, there's more to it than just running the test:
    - How difficult is it to seed your test data?
    - Do you need to create/modify a locust task?
    - How many tests do you need to run?
    - How long do the tests need to run?
    - How easy is it to infer test outcomes?
    - Is there loadtest environment contention?

Time Investment For Running Tests

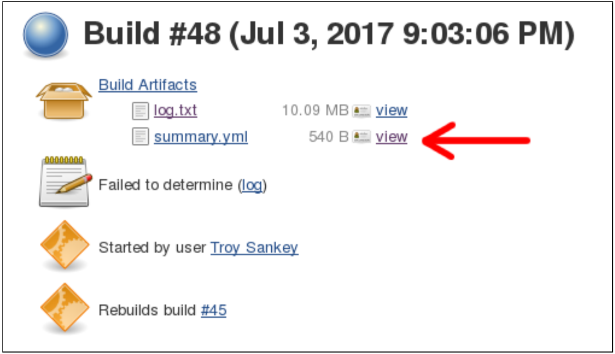
- run-simple-loadtest Instructions:
    1. Fill in build parameters and click build.
    2. Wait 1 minute for an ec2 instance to launch and run your test.
- However, there's more to it than just running the test:
    - How difficult is it to seed your test data?
    - Do you need to create/modify a locust task?
    - How many tests do you need to run?
    - How long do the tests need to run?
    - How easy is it to infer test outcomes?
    - Is there loadtest environment contention?
    - Do you need to block on devops?

Evaluating Load Test Runs

- Locust provides response time breakdown.

Evaluating Load Test Runs

- Locust provides response time breakdown.
- Click on the summary artifact.



Build #48 (Jul 3, 2017 9:03:06 PM)

Build Artifacts

log.txt 10.09 MB view

summary.yml 540 B view

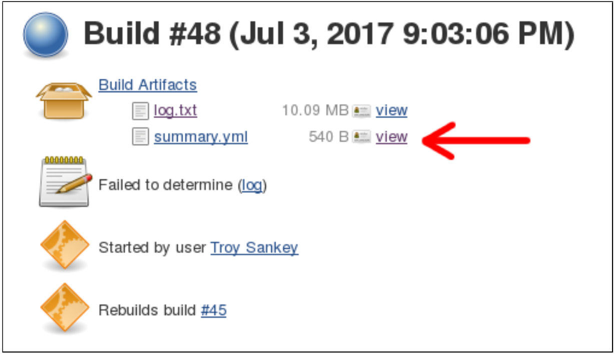
Failed to determine (log)

Started by user Troy Sankey

Rebuilds build #45

Evaluating Load Test Runs

- Locust provides response time breakdown.
- Click on the summary artifact.



Build #48 (Jul 3, 2017 9:03:06 PM)

Build Artifacts

log.txt 10.09 MB view

summary.yml 540 B view

Failed to determine (log)

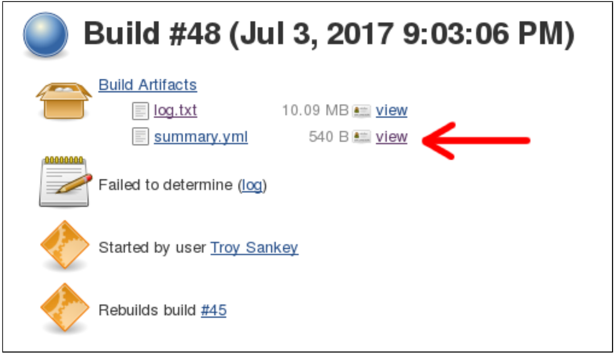
Started by user Troy Sankey

Rebuilds build #45

- NewRelic provides deeper application insight.

Evaluating Load Test Runs

- Locust provides response time breakdown.
- Click on the summary artifact.



Build #48 (Jul 3, 2017 9:03:06 PM)

Build Artifacts

log.txt 10.09 MB view

summary.yml 540 B view

Failed to determine (log)

Started by user Troy Sankey

Rebuilds build #45

- NewRelic provides deeper application insight.
- Custom metrics in edx-platform: make use of the custom metrics middleware for peering into application behavior.

Record Your Findings

- Create a wiki page before testing.

Record Your Findings

- Create a wiki page before testing.
- Caution: Beware of NewRelic data atrophie. NR gets hungry and eats your old dots.

Record Your Findings

- Create a wiki page before testing.
- Caution: Beware of NewRelic data atrophie. NR gets hungry and eats your old dots.
- Continuous deployment ⇒ continuous deletion of system metrics!

Distributed load testing

- Just run multiple load tests simultaneously in Jenkins.

Distributed load testing

- Just run multiple load tests simultaneously in Jenkins.
- You may need to prime N workers first.

Distributed load testing

- Just run multiple load tests simultaneously in Jenkins.
- You may need to prime N workers first.
- If you want/need real distributed load testing, demand it!

Maintaining load tests

- Task ratios will rot.

Maintaining load tests

- Task ratios will rot.
- Tasks themselves will rot.

Maintaining load tests

- Task ratios will rot.
- Tasks themselves will rot.
- New endpoints will need to be added as new tasks.

For fun: Real browser load testing

- `pip install realbrowserlocusts`, then subclass PhantomJSLocust in your locustfile.
- Could make certain tests more realistic (AJAX calls actually happen).
- Potentially far more resource intensive on the client side