Behavior-driven Load Testing Using Contextual Knowledge – Approach and Experiences –

Henning Schulz, Dušan Okanović, André van Hoorn @andrevanhoorn
Vincenzo Ferme, Cesare Pautasso

10th ACM/SPEC Int. Conf. on Performance Engineering (ICPE 2019)
Mumbai, India – April 11, 2019
Load Testing – Approach and Challenges

- High manual effort for maintaining load tests
- Load tests need much time to execute
- There are no suitable load tests

BenchFlow – Declarative Load Testing

Background

Example BenchFlow specification

```
configuration:
goal:
type: load_test
observe:
  workload:
    browse: [response_time]
services:
  catalogue WS: [ram, cpu]
  dbms: [ram, cpu]
load_function:
users: 1000
ramp-up: 2m
steady-state: 10m
ramp-down: 2m
```

Manual workload specification needed.

• Generation of executable artifacts
• SUT deployment
• Load test execution
• Collection of test results

ContinuITy – Continuous Load Testing in DevOps

Background

- Workload model extraction
- Workload model evolution
- Workload prediction
- Workload model selection

Monitoring data

Contextual information

Schulz et al., LTB@ICPE 2018

Workload – prediction and context


Limited load test execution automation.
Behavior-driven (Functional) Testing – Workflow

Background

1. Write story

Scenario: A trader is alerted of status

Given a stock and a threshold of 15.0
When stock is traded at 5.0
Then the alert status should be OFF

2. Map steps to Java

```java
public class TraderSteps {
    private TradingService service; // Injected
    private Stock stock; // Created

    @Given("a stock and a threshold of \$threshold")
    public void aStock(double threshold) {
        stock = service.newStock("STR", threshold);
    }

    @When("the stock is traded at price \$price")
    public void stockIsTraded(double price) {
        stock.tradeAt(price);
    }

    @Then("the alert status is \$status")
    public void alertStatusIs(String status) {
        assertEquals(stock.getShared().name(), equalTo(status));
    }
}
```

3. Configure Stories

```java
public class TraderStories extends JUnitStories {
    public Configuration configuration() {
        return new MostTestableConfiguration()
            .useStoryLoader(new LoadFromClasspath(this.getClass())
            .useStoryReporterBuilder(new StoryReporterBuilder()
            .withCodeLocationFromClass(this.getClass())
            .withFormats(Console, Text, HTML, XML));
    }

    public List<ScenarioStep> candidateSteps() {
        return new InstanceStepsFactory(new TraderSteps().getSteps());
    }

    protected List<String> storyPaths() {
        return new StoryFinder().findPaths(codeLocationFromClass(this.getClass()),
            "**/*.story");
    }
}
```

4. Run Stories

5. View Reports

Scenario: A trader is alerted of status

Given a stock and a threshold of 15.0
When stock is traded at 5.0
Then the alert status is OFF
Behavior-driven *Load Testing* (BDLT) – Example

**Approach**

Given the next Black Friday,
when varying the CPU cores between 1 and 4,
then run the experiment for 1h and ensure the maximum CPU utilization is less than 60%.

**BDLT example definition**

**Given**
- *initial context*
  - date (range)
  - event
  - app config

**when**
- changes
  - num. users
  - event
  - app config

**then**
- outcome
  - quality gates
  - run time
  - metrics
Behavior-driven Load Testing (BDLT) – Workflow

Approach

BDLT definition

Given

When

Then

SUT definition

BenchFlow

Load driver

Workload specification

(declarative)

BDLT definition

Continuity

Monitoring data

Contextual information

Workload specification

Test results

System under Test

LOAD

SUT
EBNF-based BDLT grammar with extension points for events
Mapping from BDLT to ContinuITy and BenchFlow

BDLT
- date range
- next event
- user number
- assignment
- varying
- event happens
- run
- break
- ensure
- collect

ContinuITy

BenchFlow
- users
- workload
- exploration_space
- steady_state
- termination_criteria
- quality_gates
- observe

Evaluation – Research Questions

RQ1: How expressive is the BDLT language in regards to load test concerns of industrial use cases?

RQ2: How would BDLT be used in industrial contexts?

RQ3: What are the benefits and limitations of using BDLT in comparison to defining load test scripts?
Case Study – System and Method

- System under study
  - IoT system from automotive sector
  - IoT endpoint migrating to Cloud
- Methodology
  - Workshops
  - We iteratively developed test specifications based on expert feedback
  - Incorporation of production monitoring data (per-device behavior and intensity)
- Four scenarios

<table>
<thead>
<tr>
<th>Name</th>
<th>goal</th>
<th>quality_gates</th>
</tr>
</thead>
<tbody>
<tr>
<td>configuration exploration</td>
<td>exhaustive_exploration</td>
<td>CPU load, message latency</td>
</tr>
<tr>
<td>continuous quality assurance</td>
<td>load</td>
<td>number of instances, cost</td>
</tr>
<tr>
<td>recovery spike</td>
<td>load</td>
<td>queue length</td>
</tr>
<tr>
<td>more devices</td>
<td>load</td>
<td>CPU load</td>
</tr>
</tbody>
</table>
Given the next three months and the number of users set to the maximum when varying the CPU cores between 0.5 and 4 in steps of 0.5 and varying the number of instances between 1 and 5 and varying the RAM among (1GB, 2GB, 4GB) then run each experiment for 1 hour and ensure the average CPU load is less than 15% and ensure the message latency is less than 2 seconds
Given 2018/10/15 9:00
when an outage happened from
2018/10/15 7:00 to 2018/10/15 9:00
then run the experiment for 2 hours
and ensure the final queue length is less than 100
<table>
<thead>
<tr>
<th>RQ1</th>
<th>How expressive is the BDLT language in regards to load test concerns of industrial use cases?</th>
<th>Could express all use cases</th>
<th>Custom extensions needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ2</td>
<td>How would BDLT be used in industrial contexts?</td>
<td>Could replace manual tests</td>
<td>Natural language helps understanding and communication (with non-experts)</td>
</tr>
<tr>
<td>RQ3</td>
<td>What are the benefits and limitations of using BDLT in comparison to defining load test scripts?</td>
<td>“The approach has potential“</td>
<td>Some concerns hard to express</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Too many (technical) details in the Then clause?</td>
<td>Current (technical) limitation to HTTP</td>
</tr>
</tbody>
</table>
Summary

Future Work

• Extending the language
• Improving tool support
• Further evaluation
• Supporting regression queries
• Integration into agile development methods, e.g., user stories from Scrum tickets
• Integration into Declarative Performance Engineering landscape

+ Additional (laboratory) case study
+ Supplementary material
Behavior-driven Load Testing Using Contextual Knowledge – Approach and Experiences –

Henning Schulz, Dušan Okanović, André van Hoorn
Vincenzo Ferme, Cesare Pautasso

10th ACM/SPEC Int. Conf. on Performance Engineering (ICPE 2019)
Mumbai, India – April 10, 2019


