The JPetStore Suite: A Concise Experiment Setup for Research

SSP 2018, Hildesheim

Reiner Jung & Marc Adolf

9th November 2018
Experimental Evaluation

Evaluation of

- methods
- approaches
- techniques
- tooling

for software qualities
Experimental Evaluation

Evaluation of

- methods
- approaches
- techniques
- tooling

for software qualities
Lack of a complete experiment suites

SPECjbb Benchmark

CoCOME

NetflixBieker
Requirements

1. Experiment documentation
2. Experiment setup
   - Software System & setup instructions
   - Workload drivers & documentation
   - Working scenarios which are easy to setup up
3. Tests to support setup
4. Instrumentation examples
5. Different deployments/architectures
What to do?

• Extend a big widely used case study?
  • too complicated for a first example
  • there is no widely used case study

• Use a small open source application?
  • but small examples are only single service architectures
Distributed JPetStore

- **Experiment Host**
  - Event Collector
  - Workload Driver

- **jpetstore**
  - Frontend

- **Database**
  - Account
  - Catalog
  - Order

- **Event Collector**

- **Workload Driver**

- **Database**
Distributed JPetStore

- **<<host>>** Experiment Host
  - **<<service>>** Event Collector
  - **<<service>>** Workload Driver
- **<<container>>** jpetstore
  - **<<service>>** Frontend
- **<<container>>**
  - **<<container>>** Account
  - **<<container>>** Catalog
  - **<<container>>** Order
- **<<container>>** Database

Reiner Jung
Distributed JPetStore
Distributed JPetStore

Experiment Host

Event Collector

Workload Driver

jpetstore Frontend

Account

Catalog

Order

Database
Distributed JPetStore
Distributed JPetStore

Experiment Host

Event Collector

Workload Driver

jpetstore

Frontend

Account

Catalog

Order

Database
Distributed JPetStore
activityDelay: 1

webDriverConfiguration:
  baseUrl: http://172.18.0.5:8080/jpetstore-frontend/
  type: org.iobserve.selenium.behavior.ChromeDriverFactory
  driver: /home/reiner/iObserve/experiments/tools/chromedriver
  timeout: 60000

workloads:
...

behaviors:
...

Reiner Jung
**Workload Driver - Setup**

```yaml
workloads:
- name: AccountManager
  intensity:
    type: org.iobserve.selenium.configuration.ConstantWorkloadIntensity
    name: AccountManager
    spawnPerSecond: 0.1
    durations: [100, 50]
    delays: [50, 50]
```
Workload Driver - Setup

behaviors:
AccountManager:
  name: AccountManager
  activityDelay: 2
  subbehaviors:
    - name: LoginJPetStoreTask
      parameters:
        username: "j2ee"
        password: "j2ee"
    - name: ChangeAccountInformationTask
      repetition: { min: 1, max: 10 }
      parameters:
        attribute: ADDRESS2
        value: "Christian-Albrechts-Platz 4"
    - name: ViewOrderTask
      repetition: { min: 1, max: 20 }
Workload Driver - Usage & Extension

Run a workload

```
bin/selenium-workloads.sh -c workload-file.yaml \
    -d /home/reiner/iObserve/experiments/tools/chromedriver \
    -u http://172.17.0.2:8080/jpetstore-frontend/
# -d and -u are optional and overwrite workload settings
```

Extending workloads

1. Configuration class IWorkloadIntensity
2. Workload model IWorkloadBalance
Running JPetStore & Observations

- **Execute JPetStore**
  - Local docker `execute-jpetstore.sh $WORKLOAD_FILE`
  - Kubernetes `execute-kube-jpetstore.sh $WORKLOAD_FILE`

- **Execute Observation**
  - `execute-observation.sh $WORKLOAD_FILE`
Experiment Configuration

# docker repositoriy
DOCKER_REPO="blade1.se.internal:5000"

# logger IP
LOGGER=192.168.48.213

# workload driver
WORKLOAD_RUNNER="$TOOLS/bin/selenium-experiment-workloads"

# webdriver
WEB_DRIVER="$TOOLS/chromedriver"

# collector
COLLECTOR="$TOOLS/bin/collector"
Conclusion

Features
- Workloads
- JPetStore variants
- Cloud setups
- Documentation
- PCM complete models

Upcoming
- Models for privacy
- Other intensity models
- Additional probes

Sources
- Version snapshots on Zenodo.org
  DOI 10.5281/zenodo.1292788
- All else on github
  https://github.com/research-iobserve