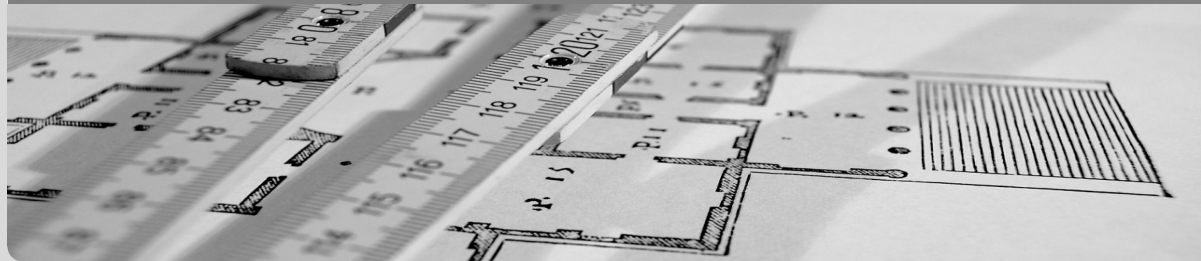


Dockerifizierung von Palladio

Betreuer: M.Sc. Dominik Werle und M.Sc. Stephan Seifermann

Thomas Weber | 25.03.2020

INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION



Reproduzierbarkeit

- zentrale Anforderung an wissenschaftliche Experimente [6]
- häufig aber nur schwer oder überhaupt nicht umsetzbar
 - Abhängigkeiten von äußeren Umständen lassen sich nicht exakt nachstellen
 - Aktualisierungen der Plattform verändern Ergebnisse

Im Kontext der Palladio Experiment Automation [5]

- große Zahl von Einflussfaktoren
 - OS
 - Java
 - eclipse
 - Palladio
 - etc.

Problem

- Durch Einflussfaktoren keine einfache Möglichkeit der Reproduzierbarkeit

Idee

- Erstellen eines Dockerfiles bzw. -images, in welchem alle Konfigurationen und Experimentdaten enthalten sind

Vorteile

- Reproduzierbarkeit der Experimente ist einfach möglich

Aktionen

- Erstellen eines Dockerfiles für allgemeinen headless Betrieb
- Erstellen eines Dockerfiles für Palladio und die Experiment Automation

Palladio [4]

- QoS Vorhersagen für Software-Architekturen
- Testen verschiedener Architekturen

Experiment Automation

- ermöglicht die automatische Ausführung von Simulationen
- unterstützt verschiedene Solver
- ohne die einzelnen Experimente konfigurieren oder starten zu müssen
- besitzt eigenes Konfigurations-Meta-Modell
- läuft (fast) headless

Was ist Docker? [1]

- Werkzeug zur Erstellung von Applikationen in Containern
- dazu verpacken aller nötigen Ressourcen in Image
- Image kann im Container überall ausgeführt werden
- unabhängig vom installierten Host-System
- leichtgewichtiger als eine Virtual Machine(VM)

Begriffe [2]

- Dockerfile
 - 'Bauanleitung' zur Erstellung eines Images
 - hat Basis-Image
 - hinzufügen weiterer Ressourcen (Skripte, Bibliotheken, etc.)
- Docker container
 - Docker Virtual Machine
- Docker image
 - vergleichbar mit 'Snapshot' einer VM
 - nicht veränderbar
 - dadurch bleibt die Ausführbarkeit erhalten
 - verschiedene Schichten

Github

- öffentliche Speicherung des Projekts
- abrufbar durch Docker Hub

Docker Hub

- Bibliothek für Container Images
- stellt Ressourcen zum Bauen von Images bereit
- welche dann auf `https://hub.docker.com/` abrufbar sind
- und wieder als Basis-Image für andere Dockerfiles genutzt werden können

Eclipse mit und ohne Modelling Framework

Eclipse

- Als Basis-Image wird Ubuntu20.04 genutzt
- alpine nicht einfach nutzbar, da eclipse gegen libc kompiliert ist

Dockerfile

```
FROM ubuntu:20.04
RUN apt-get clean && \
    apt-get update --fix-missing && \
    apt-get install -y --fix-missing openjdk-11-jre-headless wget
RUN /usr/bin/wget 'http://ftp.fau.de/eclipse/technology/epp/downloads/
release/2019-12/R/eclipse-java-2019-12-R-linux-gtk-x86_64.tar.gz' && \
    tar -xzf eclipse-java-2019-12-R-linux-gtk-x86_64.tar.gz -C /usr && \
    rm eclipse-java-2019-12-R-linux-gtk-x86_64.tar.gz
```


Example Application

- Headless Ausführung generell interessant
- Erstellen einer Beispiel-Applikation für eclipse
- welche headless in einem container ausgeführt werden kann

Dockerfile

```
FROM thomasweber/eclipse
COPY RunApplication.sh /usr/RunApplication.sh
RUN chmod a+rx /usr/RunApplication.sh
COPY edu.kit.sdq.exampleapplication_1.0.0.202002111755.jar /usr/eclipse/
  dropins/plugins/edu.kit.sdq.exampleapplication_1.0.0.202002111755.jar
RUN /usr/RunApplication.sh
```

Provisionierung von eclipse

- Nutzung der p2 Provisioning Plattform [3]
- Installation bequem über features.txt parametrisierbar

Dockerfile

```
FROM thomasweber/eclipsemodelingtools
COPY InstallFeature.sh /usr/InstallFeature.sh
COPY InstallLocalUpdates.sh /usr/InstallLocalUpdates.sh
COPY features.txt usr/features.txt
RUN chmod a+rx usr/InstallLocalUpdates.sh
RUN chmod a+rx usr/InstallFeature.sh
RUN usr/InstallLocalUpdates.sh
```

Bibliotheken und weitere Skripte

- xvfb und libgtk-3-0 zur Ausführung der Palladio Experiment Automation
- Skripte zum Erstellen der .experiments Dateien und zur Ausführung der Experimente

RunExperimentAutomation.sh

```
#!/bin/bash
# first argument is the path to the original experiments file , second is the path where to
  store the generated file
Xvfb :99 —screen 0 1920x1080x16 &
export DISPLAY=:99
/usr/ModifyExperimentsFile.sh "$1" "$2"
/usr/eclipse/eclipse \
  —clean \
  —application org.palladiosimulator.experimentautomation.application \
  —consoleLog "$2" \
  —data "/usr/workspace"
```

Änderung des Palladio Experiment Automation Metamodells

Änderungen

- Hinzufügen eines Enums ExportOption
- Hinzufügen eines Attributs exportOption zu Experimenten

ModifyExperimentsFile.sh

```
#!/bin/bash
# modifies the file given with the first parameter and saves it to the file
# given with the second parameter
sed -E 's/<datasource [^>]*\|>/<datasource xsi:type="ExperimentAutomation.
Experiments.AbstractSimulation:FileDatasource" location="\|result\"
exportOption="CSV"\|>/' "$1" > "$2"
```

Beschaffung der Experiment-Daten

- mit in das Image packen (wie im Teilprojekt PalladioExperimentImage)
- über docker als mount (PalladioRunExperiment)

RunExperimentAutomation.sh

```
SET SRC_PATH=%cd%
SET IMAGE_NAME=thomasweber/palladioexperimentautomation:latest
SET CONTAINER_PATH=/usr
SET EXPERIMENT_FILE_NAME=Capacity . experiments
SET EXPERIMENT_GEN_FILE_NAME=Generated . experiments
SET EXPERIMENTS_FILE_DIR=/ExperimentData/model/Experiments/
docker pull %IMAGE_NAME%
docker run -it -v "%SRC_PATH%\Output":/result -v "%SRC_PATH%\
ExperimentData:/usr/ExperimentData" %IMAGE_NAME% /usr/RunExperimentAutomation.sh
%CONTAINER_PATH%%EXPERIMENTS_FILE_DIR%%EXPERIMENT_FILE_NAME%
%CONTAINER_PATH%%EXPERIMENTS_FILE_DIR%%EXPERIMENT_GEN_FILE_NAME%
```

Zusammenfassung

- Generelle Ausführung von headless Applikationen in einem Docker Container
- Reproduzierbarkeit von Palladio Experiment Automation Experimenten

Zukünftige Arbeiten

- Abhängigkeiten von Palladio auf UI-Pakete finden und entfernen
- für umfassenden headless-Betrieb von Palladio
- Example Application erweitern um die Nutzung von ins Image gegebenen Parametern
- Nutzung einer festen Version von Palladio



Docker. <https://opensource.com/resources/what-docker>. Aufgerufen: 18.03.2020.



Docker Image.
<https://searchitoperations.techtarget.com/definition/Docker-image>.
Aufgerufen: 18.03.2020.



P2 Provisioning Platform. https://help.eclipse.org/2019-12/index.jsp?topic=%2Forg.eclipse.platform.doc.isv%2Fguide%2Fp2_overview.htm.
Aufgerufen: 18.03.2020.



Palladio. https://www.palladio-simulator.com/about_palladio/. Aufgerufen: 18.03.2020.



Palladio Experiment Automation.

[https://sdqweb.ipd.kit.edu/wiki/Palladio_Experiment_Automation.](https://sdqweb.ipd.kit.edu/wiki/Palladio_Experiment_Automation)

Aufgerufen: 18.03.2020.



Reproduzierbarkeit. [https://de.wikipedia.org/wiki/Reproduzierbarkeit.](https://de.wikipedia.org/wiki/Reproduzierbarkeit)

Aufgerufen: 18.03.2020.

RunApplication.sh

```
#!/bin/bash
/usr/eclipse/eclipse \
  -clean \
  -application edu.kit.sdq.exampleapplication.application \
  -consoleLog
```

InstallFeature.sh

```
echo installing "$1"  
echo from server "$2"  
/usr/eclipse/eclipse \  
  -application org.eclipse.equinox.p2.director \  
  -repository https://download.eclipse.org/releases/2019-12/,"$2" \  
  -installIU "$1"
```

InstallLocalUpdates.sh

```
sed '/^#/ d' /usr/features.txt | sed 's/\([^%]*\) %\([^%]*\) / "\1" "\2"/' |  
xargs -r -n 2 /usr/InstallFeature.sh
```

feature.txt - 1

```
#package1,package2,etc%updatesite1,updatesite2,etc  
#currently all packages needed to run the palladio experiment automation  
org.palladiosimulator.edp2.feature.feature.group,  
org.palladiosimulator.pcm.feature.feature.group,  
org.palladiosimulator.simucom.feature.feature.group,  
org.palladiosimulator.solver.feature.feature.group,  
org.palladiosimulator.recorderframework.feature.feature.group,  
org.palladiosimulator.analyzer.feature.feature.group,  
org.palladiosimulator.monitorrepository.feature.feature.group,  
org.palladiosimulator.simulizar.feature.feature.group,  
org.palladiosimulator.simulation.abstractsimengine.desmoj.feature.feature.group  
%https://updatesite.palladio-simulator.com/palladio-build-updatesite/releases/4.2.0/
```

feature.txt - 2

```
# comment this line and uncomment the following one to add the export function in the nightly
release
# add the feature back to the list above once a new release with export is available
org.palladiosimulator.experimentautomation.application.feature.feature.group,
org.palladiosimulator.experimentautomation.feature.feature.group,
org.palladiosimulator.experimentautomation.application.tooladapter.simulizar.feature.feature.group
%https://updatesite.palladio-simulator.com/palladio-addons-
experimentautomation/branches/csvExportRelease/,
https://updatesite.palladio-simulator.com/palladio-build-updatesite/releases/4.2.0/
# org.palladiosimulator.experimentautomation.application.feature.feature.group
%https://updatesite.palladio-simulator.com/palladio-build-updatesite/releases/nightly/
```

Dockerfile

```
FROM thomasweber/eclipsepalladio
RUN apt-get clean && \
    apt-get update --fix-missing && \
    apt-get install -y --fix-missing xvfb libgtk-3-0
COPY RunExperimentAutomation.sh /usr/RunExperimentAutomation.sh
RUN chmod a+rx usr/RunExperimentAutomation.sh
COPY ModifyExperimentsFile.sh /usr/ModifyExperimentsFile.sh
RUN chmod a+rx usr/ModifyExperimentsFile.sh
```

ModifyExperimentsFile.sh

```
#!/bin/bash
# modifies the file given with the first parameter and saves it to the file
# given with the second parameter
sed -E 's/<datasource [^>]*\/>/<datasource xsi:type="ExperimentAutomation.
Experiments.AbstractSimulation:FileDatasource" location="\result\"
exportOption="CSV"\/>/' "$1" > "$2"
```

RunExperimentAutomation.sh

```
#!/bin/bash
# first argument is the path to the original experiments file , second is the
  path where to store the generated file
Xvfb :99 -screen 0 1920x1080x16 &
export DISPLAY=:99
/usr/ModifyExperimentsFile.sh "$1" "$2"
/usr/eclipse/eclipse \
  -clean \
  -application org.palladiosimulator.experimentautomation.application \
  -consoleLog "$2" \
  -data "/usr/workspace"
```


PalladioExperimentImage - RunDockerImage.sh

```

SET SRC_PATH=%cd%
SET IMAGE_NAME=thomasweber/palladioexperimentimage:latest
SET CONTAINER_PATH=/usr
SET EXPERIMENT_FILE_NAME=Capacity.experiments
SET EXPERIMENT_GEN_FILE_NAME=Generated.experiments
SET EXPERIMENTS_FILE_DIR=/ExperimentData/model/Experiments/
SET EXPERIMENTS_FILE_DIR_WINDOWS=\ExperimentData\model\Experiments\
docker pull %IMAGE_NAME%
docker run -it -d %IMAGE_NAME%
FOR /F "tokens=*" %%g IN ('docker ps -q') do (SET CONTAINER_ID=%%g)
FOR /F "tokens=*" %%g IN ('docker ps --format "{.Names}"') do (SET
    CONTAINER_NAME=%%g)
docker exec -it %CONTAINER_NAME% /usr/RunExperimentAutomation.sh %
    CONTAINER_PATH%%EXPERIMENTS_FILE_DIR%%EXPERIMENT_FILE_NAME% %
    
```

PalladioRunExperiment - RunDockerImage.sh

```
SET SRC_PATH=%cd%
SET IMAGE_NAME=thomasweber/palladioexperimentautomation:latest
SET CONTAINER_PATH=/usr
SET EXPERIMENT_FILE_NAME=Capacity.experiments
SET EXPERIMENT_GEN_FILE_NAME=Generated.experiments
SET EXPERIMENTS_FILE_DIR=/ExperimentData/model/Experiments/
docker pull %IMAGE_NAME%
docker run -it -v "%SRC_PATH%\Output" :/result -v "%SRC_PATH%\
ExperimentData:/usr/ExperimentData" %IMAGE_NAME% /usr/
RunExperimentAutomation.sh %CONTAINER_PATH%/EXPERIMENTS_FILE_DIR%/
EXPERIMENT_FILE_NAME% %CONTAINER_PATH%/EXPERIMENTS_FILE_DIR%/
EXPERIMENT_GEN_FILE_NAME%
PAUSE
```