Planning and Execution of System Adaptations in Cloud-Based Environments

Master’s Thesis
Lars Erik Blümke
Motivation
Motivation
Motivation

Sun, 17.00: Germany vs. Mexico
Motivation

Sun, 17.00: Germany vs. Mexico
Motivation

Sun, 17.00: Germany vs. Mexico
Motivation

Sun, 17.00: Germany vs. Mexico

Sun, 17.45 TV-ad during halftime
Motivation

Sun, 17.00: Germany vs. Mexico
Sun, 17.35: Scale up system
Sun, 17.45 TV-ad during halftime
Motivation

Sun, 17.00: Germany vs. Mexico
Sun, 17.35: Scale up system
Sun, 17.45 TV-ad during halftime
Sun, 17.46 System scales up automatically
Motivation

Sun, 17.00: Germany vs. Mexico
Sun, 17.35: Scale up system
Sun, 17.45 TV-ad during halftime
Sun, 17.46 System scales up automatically
iObserve

Image from: https://github.com/research-iobserve

iObserve Overview [Hasselbring et al. 2013]
iObserve Overview [Hasselbring et al. 2013]
Goals

G1: Architectural Integration of an Existing Architecture Optimization Approach

G2: Improvement of an Existing Computation Method for Execution Plans to Address the Availability of the Observed System During Execution

G3: Evaluation of the Approach
The Palladio Component Model (PCM)

- 5 submodels representing different system domains:
  - Repository Model
  - System Model
  - Resource Environment Model
  - Allocation Model
  - Usage Model
- Used in iObserve to represent the observed system
- Implemented with Eclipse Modeling Framework (EMF)
- Serializable

Image from: https://sdqweb.ipd.kit.edu/wiki/File:Pcm-logo-stilisiert.png
Planning
1st Phase of Planning

We have:

- Present-architecture-model (PAM)
- Optimization goals (e.g. better response times, lower operating cost, ...)

We want:

- Candidate-architecture-model (CAM) which improves with respect to the specified goals
PerOpteryx

- Eclipse plug-in
- Evolutionary algorithm
- Uses PCM models
  - Inputs:
    - Present architecture model
    - Optimization goals (degrees of freedom)
  - Output:
    - Candidate architecture model

Using PerOpteryx with iObserve

1st Attempt: Integration as a Dependency

- As an Eclipse plug-in PerOpteryx consists of JARs
- Reference these JARs from iObserve

2nd Attempt: Creating a Standalone RCP-Application

- Export executables manually from Eclipse
  or
- Use Goomph Gradle plugin for automatized export
Architecture Overview
Pipe-and-Filter Architecture of iObserve’s Planning Service
2nd Phase of Planning

We have:

- Present-architecture-model (PAM)
- Candidate-architecture-model (CAM) which improves with respect to the specified goals

We want:

- **Execution plan**
  - consisting of adaptation actions
  - actions transform the real system from its current state (represented by PAM) to its intended state (represented by CAM)
Rule Based Detection of Adaptation Actions

- Drools Business Rule Management System
  - Can be embedded into Java Code
  - Java Objects act as facts
  - Rules are defined in their own language in a separate file
- Define rules to detect adaptation actions from model difference
- Insert PAM and CAM into working memory
- Receive adaptation actions
Allocation and Deallocation

For resource container $r$:

$\neg r \in PAM \land r \in CAM \rightarrow \text{Allocation of } r$

$r \in PAM \land \neg r \in CAM \rightarrow \text{Deallocation of } r$
Replication and Dereplication

For allocation context $c$:

\[ \neg c \in \text{PAM} \land c \in \text{CAM} \rightarrow \text{Replication of } c \]

\[ c \in \text{PAM} \land \neg c \in \text{CAM} \rightarrow \text{Dereplication of } c \]
For allocation context $c$, $c'$:

$c \in \text{PAM} \land c' \in \text{CAM} \land c = c'$

$\land c\.getRC \neq c'\.getRC \rightarrow \text{Migration of } c \text{ to } c'\.getRC$
Atomic Adaptation Actions

- The detected adaptation actions are composed
- Example: Dereplication
  - Block requests
  - Finish running tasks
  - Disconnect
  - Undeploy
- Replace each composed action by atomic actions
- Add atomic actions to execution plan
Architecture Overview
P&F Architecture of iObserve’s Adaptation Service
Execution
Execution

We have:

- Execution plan

We want:

- Apply the execution plan to the real system
  - Technology dependent implementations needed
  - Kubernetes in our case
Architecture Overview
P&F Architecture of iObserve’s Execution Service
Evaluation

We evaluate three tasks in a feasibility study:

**T1:** Creation of execution plan

**T2:** Application of execution plan to the observed system

**T3:** Expected behavior of the service based architecture
Evaluation Scenarios

We evaluate three adaptation scenarios

**S1:** Replication of the account component to an existing resource container

**S2:** Dereplication of the account component

**S3:**
- Allocation of a new resource container
- Migration of the account component onto this resource container
- Deallocation of the old resource container
## Results

Did we receive the expected results?

<table>
<thead>
<tr>
<th>Task/Scenario</th>
<th>S1 (Replication)</th>
<th>S2 (Dereplication)</th>
<th>S3 (De-/Allocation + Migration)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>T1</strong> (Executionplan)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td><strong>T2</strong> (Execution)</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td><strong>T3</strong> (iObserve service)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
## Migration Problem

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>account-546cddf8f-4fjlp</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>56s</td>
</tr>
<tr>
<td>catalog-67c66d6c6f-ss7s4</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>56s</td>
</tr>
<tr>
<td>frontend-69b8d9c7f6-268Br</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>56s</td>
</tr>
<tr>
<td>order-59d6d9bdf4-z42l5</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>56s</td>
</tr>
<tr>
<td>account-546cddf8f-4fjlp</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>57s</td>
</tr>
<tr>
<td>account-59f54677b-qdvf6</td>
<td>0/1</td>
<td>ContainerCreating</td>
<td>0</td>
<td>8s</td>
</tr>
<tr>
<td>catalog-67c66d6c6f-ss7s4</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>57s</td>
</tr>
<tr>
<td>frontend-69b8d9c7f6-268Br</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>57s</td>
</tr>
<tr>
<td>order-59d6d9bdf4-z42l5</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>57s</td>
</tr>
<tr>
<td>account-546cddf8f-4fjlp</td>
<td>1/1</td>
<td>Terminating</td>
<td>0</td>
<td>58s</td>
</tr>
<tr>
<td>account-59f54677b-qdvf6</td>
<td>0/1</td>
<td>ContainerCreating</td>
<td>0</td>
<td>1s</td>
</tr>
<tr>
<td>catalog-67c66d6c6f-ss7s4</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>58s</td>
</tr>
<tr>
<td>frontend-69b8d9c7f6-268Br</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>58s</td>
</tr>
<tr>
<td>order-59d6d9bdf4-z42l5</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>58s</td>
</tr>
<tr>
<td>account-546cddf8f-4fjlp</td>
<td>1/1</td>
<td>Terminating</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>account-59f54677b-qdvf6</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>3s</td>
</tr>
<tr>
<td>catalog-67c66d6c6f-ss7s4</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>frontend-69b8d9c7f6-268Br</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>order-59d6d9bdf4-z42l5</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>account-59f54677b-qdvf6</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>33s</td>
</tr>
<tr>
<td>catalog-67c66d6c6f-ss7s4</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>frontend-69b8d9c7f6-268Br</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>order-59d6d9bdf4-z42l5</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
</tbody>
</table>
### Migration Problem

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>account-546cdddf8f-4fjlp</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>56s</td>
</tr>
<tr>
<td>catalog-67c6d6c6f-ss7s4</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>56s</td>
</tr>
<tr>
<td>frontend-69b8d9cfc76-26h8r</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>56s</td>
</tr>
<tr>
<td>order-59d6d9bd4f-42264</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>56s</td>
</tr>
<tr>
<td>account-546cdddf8f-4fjlp</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>57s</td>
</tr>
<tr>
<td>account2-59f54f677b-qdfv6</td>
<td>0/1</td>
<td>ContainerCreating</td>
<td>0</td>
<td>6s</td>
</tr>
<tr>
<td>catalog-67c6d6c6f-ss7s4</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>57s</td>
</tr>
<tr>
<td>frontend-69b8d9cfc76-26h8r</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>57s</td>
</tr>
<tr>
<td>order-59d6d9bd4f-42264</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>57s</td>
</tr>
<tr>
<td>account-546cdddf8f-4fjlp</td>
<td>1/1</td>
<td>Terminating</td>
<td>0</td>
<td>58s</td>
</tr>
<tr>
<td>account2-59f54f677b-qdfv6</td>
<td>0/1</td>
<td>ContainerCreating</td>
<td>0</td>
<td>1s</td>
</tr>
<tr>
<td>catalog-67c6d6c6f-ss7s4</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>58s</td>
</tr>
<tr>
<td>frontend-69b8d9cfc76-26h8r</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>58s</td>
</tr>
<tr>
<td>order-59d6d9bd4f-42264</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>58s</td>
</tr>
<tr>
<td>account-546cdddf8f-4fjlp</td>
<td>1/1</td>
<td>Terminating</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>account2-59f54f677b-qdfv6</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>3s</td>
</tr>
<tr>
<td>catalog-67c6d6c6f-ss7s4</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>frontend-69b8d9cfc76-26h8r</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>order-59d6d9bd4f-42264</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>account2-59f54f677b-qdfv6</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>33s</td>
</tr>
<tr>
<td>catalog-67c6d6c6f-ss7s4</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>frontend-69b8d9cfc76-26h8r</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>order-59d6d9bd4f-42264</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
</tbody>
</table>

- Pod **account** already terminates before pod **account2** is running
- Account component unavailable for at least 2 seconds

**Fix:** Deployment action has to wait until the deployed component has actually become available
Conclusions

- We failed to integrate PerOpteryx into iObserve
- We succeeded in providing
  - a rule-based approach for the computation of composed adaptation actions
  - a refinement into atomic adaptation actions to address component dependencies
  - an execution mechanism for system adaptations on Kubernetes cluster
  - a service-based architecture for iObserve
Future Work

- PerOpteryx integration
  - with Goomph
  - on a code basis

- Runtime evaluation of execution results
  - no feedback on success of execution so far
  - conformance between CAM and observed system after execution

- Additional rules for specific scenarios
  - e.g. migrate closely coupled components closer after each other

- Compatibility to different cloud infrastructures
Sources