FogFrame: IoT service deployment and execution in the fog

Olena Skarlat,
Kevin Bachmann, Stefan Schulte

KuVS-Fachgespräch Fog Computing 2018

Distributed Systems Group, TU Wien, Austria
http://www.infosys.tuwien.ac.at/
What is a Fog Landscape?

A horizontal, system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum.
Research Questions

**Challenge:** to create and support an execution environment for IoT applications in the fog landscape.

- What are the mechanisms to provide **virtualization** of resources?

- What are the methodologies and tools to realize **software** that manages a fog landscape and executes services?

- How to perform and optimize **resource provisioning** and execute services?
Fog Landscape Resource Model

**Fog cells** control IoT devices (sensors, actuators), can execute services

**Fog control nodes** control fog cells and execute services

A fog control node and connected fog cells form a **fog colony**, acts as a micro data center

**Cloud-fog middleware** manages cloud resources and supports fog colonies
FogFrame: A Fog Computing Framework

Functionality of FogFrame:

• Coordinated control over a fog landscape
• Monitoring and analysis of resources
• Service placement plan
• Deploy and execute applications
• (Re-)configuration of the fog landscape based on runtime events

https://github.com/keyban/fogframe
FogFrame Architecture

Fog cells and fog control nodes
Application Model

Distributed Data Flow* application

• Quality of Service (QoS) requirements, e.g., deadline on deployment and execution time
• Set of services to be deployed
• Each service is characterized by demands in CPU, RAM, and storage,
  its service type (e.g., a certain sensor equipment is needed, or it is a purely cloud service)

Service placement

Fog Service Placement Problem (FSPP)

**Goal:** to produce a service placement plan which maximizes the utilization of fog colonies while satisfying QoS

FSPP is solved by each head fog control node:

- Which services have to be executed in its **own fog colony**?
- Which services have to be executed **locally** on own resources of fog control node?
- Which services have to be propagated to the **closest neighbor colony**?
- Which services have to be propagated to the **cloud**?

FSPP is solved here
Fog Service Placement Problem (FSPP)

**Variables:** decision variables

**Goal:** maximize fog colony resources utilization, while adhering to QoS parameters

\[
\max \sum_{A_k} \ P(A_k) \left( \sum_{a_i} \left( \sum_{f_j} \ x_a^{f_j} + x^{F}_a \right) + |A_k| y_{A_k} \right)
\]

\[
P(A_k) = \frac{1}{D_{A_k} - w_{A_k}}
\]

**Constraints:**

- Resource capacities
- Number of deployed containers
- Adherence to QoS
- Propagation
Solutions of FSPP

• Exact mathematical method
• First-fit heuristic algorithm
• Genetic algorithm
Solutions of FSPP - Genetic Algorithm

**Chromosome:**

<table>
<thead>
<tr>
<th>Application $A_1$ services of $A_1$</th>
<th>Application $A_2$ services of $A_2$</th>
<th>...</th>
<th>Application $A_k$ services of $A_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$a_2$</td>
<td>$a_3$</td>
<td>$a_i$</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

**Fitness:** encourage if a chromosome fulfills constrains, apply penalties upon violations, “death penalties”

\[
F(c) = \sum_{\beta_p \in \Psi} \omega_{\beta_p} (1 - 2\delta_{\beta_p(c)}) + \sum_{\beta_\gamma \in \Gamma} \omega_{\beta_\gamma} (1 - 2\delta_{\beta_\gamma(c)}) - \omega_p D(c)
\]
Solutions of FSPP - Genetic Algorithm

Parameters:
• 80%-uniform crossover
• Tournament selection
• 2% random gene mutation
• 20% elitism rate
• Population size of 1000 individuals

Stopping condition:
• Positive fitness (no death penalties applied)
• Tolerance value: dividing the incremental variance of the fitness values by the maximum fitness value over generations
**Challenge**: how to deploy services in the heterogeneous environment of a fog landscape?

- Deployment in the cloud
- Deployment in the fog colonies

<table>
<thead>
<tr>
<th></th>
<th>Cloud</th>
<th>Fog colonies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resources</strong></td>
<td>VMs</td>
<td>Raspberry Pis</td>
</tr>
<tr>
<td><strong>Processor architecture</strong></td>
<td>64-bit Amazon Machine Image</td>
<td>ARM</td>
</tr>
<tr>
<td><strong>Operating system</strong></td>
<td>CoreOS</td>
<td>Hypriot</td>
</tr>
<tr>
<td><strong>Base Docker image</strong></td>
<td>FROM java:8</td>
<td>FROM hypriot/rpi-java</td>
</tr>
<tr>
<td><strong>Service storage</strong></td>
<td>Docker Hub</td>
<td>Shared storage</td>
</tr>
</tbody>
</table>
Service Deployment in Fog Colonies

Docker Hook instead of Docker-inside-Docker

Fog cells (FC)
• Redis FC container with a local database

Fog control nodes (FCN)
• Redis FCN with a local database
• Redis Shared with the shared repository of service images
Service Deployment in the Cloud

AWS cloud (Openstack cloud)

CoreOS with Docker runtime preconfigured
Evaluation

Metrics:

• Deployment time
• Utilization of resources (deployed containers)

Scenarios:

• Assessment of deployment time
• Different arrival patterns of application requests
• Different service placement algorithms
• Reaction on runtime events
Evaluation: different algorithms, arrival patterns

Constant arrival pattern of application requests: 10 services each 2 minutes

Pyramid arrival pattern: 5, 10, 15, 10, and 5 services each 2 minutes
## Related Work

<table>
<thead>
<tr>
<th>Execution environment</th>
<th>Resource provisioning</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1] de Brito et al. IoT testbed: Docker Swarm + OpenMTC M2M, VMs</td>
<td>Docker labels</td>
</tr>
<tr>
<td>[3] Yigitolu et al. „Foggy“, Raspberry Pis</td>
<td>Orchestration server on every node, first-fit algorithm</td>
</tr>
<tr>
<td>[8] Saurez et al. „Foglets“, simulated</td>
<td>Trigger-based: latency and resources</td>
</tr>
</tbody>
</table>
Conclusions

• Execution environment for IoT applications in a fog landscape
• FogFrame: placement, deployment, and execution of IoT applications
• Optimization problems for resource provisioning and service placement

Future work:

• Automated device discovery, fault tolerance mechanisms
• Availability, reliability of services and devices, cost in the optimization
• Heuristic and exact algorithms
Thank you for attention!

Olena Skarlat, MSc
Research Assistant
Vienna University of Technology
Institute of Information Systems
Argentinierstrasse 8/184-1, 1040 Vienna, Austria
T: +43 1 58801-18459    F: +43 1 58801-18491
E: o.skarlat@infosys.tuwien.ac.at
www.infosys.tuwien.ac.at
## Related Work

<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
</table>
## Related Work (ctnd)

<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
</table>
Our papers

**Resource provisioning for IoT services in the fog**
O. Skarlat, S. Schulte, M. Borkowski, P. Leitner
SOCA 2016

**Towards QoS-aware fog service placement**
O. Skarlat, M. Nardelli, S. Schulte, S. Dustdar
ICFEC 2017

**Optimized IoT service placement in the fog**
O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, P. Leitner
SOCA Journal 2017

**FogFrame: Service placement, deployment, and execution in the fog**
O. Skarlat, K. Bachmann, S. Schulte
FGCS Journal 2018 (under review)