How to best deploy your Fog applications, probably

Antonio Brogi        Stefano Forti        Ahmad Ibrahim

1st International Conference on Fog and Edge Computing 2017
14th May 2017
The Cloud alone cannot support the IoT momentum. There is a need for filtering and processing before the Cloud.
Fog Features

**QoS-awareness**
- App deployments dynamically adapt to the **state** of the network.

**Location-awareness**
- **Position** is known so to handle fluid and mobile computation.

**Context-awareness**
- Discover and use available resources, **cooperating** horizontally.
Open Problems

• How to **automatically** decide *where* to deploy each component of an application by exploiting QoS-, location-, and context-awareness?

• How to estimate the **QoS-assurance** of a candidate deployment?
Motivating example

DataStorage → Dashboard

ThingsController

- video
- water
- moisture
- fire

Sat. 3G VDSL

Amazon Web Services
Google Cloud Platform
Microsoft Azure
Concretely...

How many and how powerful Fog nodes do I need to adequately deploy my application?

Should I deploy this component onto the Cloud, onto a Fog-as-a-Service opened in my city or on my premises gateway?

Is there any component I'd better deploy on a different node after this link/node failure?
Concretely...

Is it possible to reduce resource consumption of some Fog nodes, or avoid them?

Do I have to upgrade my infrastructure if the application requirements change?

Which are the eligible deployments that comply most with the required QoS?
Our Solution

**Modelling** of IoT apps and Fog infrastructures

**Algorithms** to determine eligible deployments

**Evaluation** of output deployments via Monte Carlo
Our Prototype

fogtorch

https://github.com/di-unipi-socc/FogTorchPI

di-unipi-socc/FogTorchPI is licensed under the MIT License
QoS Profiles

• A QoS profile is a pair

\[ \langle \ell, \langle b_\downarrow, b_\uparrow \rangle \rangle \]

• They represent latency and bandwidth featured by a link or requested by a software interaction.
Application

\(\langle 160 \text{ ms}, 0.5 \text{ Mbps}, 0.7 \text{ Mbps}\rangle\)

SD video
Infrastructure

98% (70 ms, 6 Mbps, 0.75 Mbps)
2% (70 ms, 0 Mbps, 0 Mbps)

Satellite 7M
A **software component** is **compatible** with a Fog or Cloud node when its software and hardware* can support at least that component.

* Hardware only for Fog nodes.
Things Binding

- Software components may have Things requests.
- Each request is bound to a **specific Thing** before deployment.
Deployment Policy

• A **start-up** sponsored by a specific Cloud provider,
• an **automated industrial** plant,
• an invoked **third party service**...

...may enforce **legal, commercial** or **political** constraints for deploying an application.

• We allow specification of a **whitelist** of nodes permitted for installing each component.
Eligible Deployments

• An **eligible deployment** for an application over a Fog infrastructure ensures Compatibility and deployment policies, Hardware resources, Things binding, and Bandwidth and latency.
NP-hard Problem*

Backtracking strategy to explore the search space.

"I can’t find an efficient algorithm, but neither can all these famous people."

[By reduction from Subgraph Isomorphism. A. Brogi and S. Forti, QoS-aware Deployment of IoT Applications Through the Fog, in IEEE Internet of Things Journal, 2017.]
Bird’s eye view

Fog Infrastructure
QoS Probabilities
Application
Deployment Policies
Things Binding

Monte Carlo simulator

Fog Infrastructure

Eligible deployments

Fog resource consumption

QoS-assurance

https://github.com/di-unipi-socc/FogTorchPI
Monte Carlo Simulator

Repeat a sufficiently large number of times:

1. Sample a **QoS profile** for each link in the infrastructure.
2. Run **backtracking** algorithm.

Compute **QoS-assurance** of generated deployments.
FogTorch II Results

Which are the eligible deployments that comply most with the required QoS?

<table>
<thead>
<tr>
<th>Deployment ID</th>
<th>Things Controller</th>
<th>Data Storage</th>
<th>Dashboard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Δ1</td>
<td>fog2</td>
<td>cloud2</td>
<td>cloud1</td>
</tr>
<tr>
<td>Δ2</td>
<td>fog2</td>
<td>cloud2</td>
<td>cloud2</td>
</tr>
<tr>
<td>Δ3</td>
<td>fog2</td>
<td>cloud1</td>
<td>cloud2</td>
</tr>
<tr>
<td>Δ4</td>
<td>fog2</td>
<td>cloud1</td>
<td>cloud1</td>
</tr>
<tr>
<td>Δ5</td>
<td>fog3</td>
<td>cloud1</td>
<td>fog2</td>
</tr>
<tr>
<td>Δ6</td>
<td>fog2</td>
<td>cloud2</td>
<td>fog2</td>
</tr>
</tbody>
</table>
Is it possible to reduce resource consumption of some fog nodes, or avoid them?

E.g., avoid using fog_3 for deployment.
Is it possible to reduce resource consumption of some fog nodes, or avoid them?

E.g., avoid using fog_3 for deployment.
DO I HAVE TO UPGRADE MY INFRASTRUCTURE IF THE APPLICATION REQUIREMENTS CHANGE?

E.g., deploying HD video streaming without upgrade, leads to same QoS-assurance.
FogTorch Results (2)

Do I have to upgrade my infrastructure if the application requirements change?

Deploying HD video streaming without upgrade, leads to worse QoS-assurance.
Results FogTorch\(\Pi\) (3)

(a) Satellite 14 Mbps upgrade.

(b) 4G upgrade.
Results FogTorchΠ (3)

(a) Satellite 14 Mbps upgrade.

(b) 4G upgrade.
Conclusions

Determine, simulate and compare eligible deployments

QoS- and context-awareness of deployments

Evaluation of QoS variations impact based on links data
Future Work

Design a **cost model** to improve search & evaluation

Include **multiple and multi-tenant** deployments

Assessment over **case studies**
Thanks!

Q&A
Roles and Stakeholders