Demo Abstract: FogExplorer

Jonathan Hasenburg
TU Berlin
Mobile Cloud Computing
hasenburg@tu-berlin.de

Sebastian Werner
TU Berlin
Information Systems Engineering
werner@tu-berlin.de

David Bermbach
TU Berlin
Mobile Cloud Computing
david.bermbach@tu-berlin.de

ABSTRACT

Fog application design is complex as it comprises not only the application architecture, but also the runtime infrastructure, and the deployment mapping from application modules to infrastructure machines. For each of these aspects, there is a variety of design options that all affect quality of service and cost of the resulting application. In this demo paper, we introduce FogExplorer, an interactive simulation tool for the QoS and cost evaluation of fog-based IoT applications already during the design phase.

CCS CONCEPTS

• **Software and its engineering → Integrated and visual development environments**; • **Computer systems organization** → *n-tier architectures*; Sensors and actuators;

KEYWORDS

Fog Computing, Application Design, Simulation, IoT

ACM Reference Format:
Jonathan Hasenburg, Sebastian Werner, and David Bermbach. 2018. Demo Abstract: FogExplorer. In *19th International Middleware Conference (Middleware '18 Posters and Demos), December 10–14, 2018, Rennes, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3284014.3284015

1 INTRODUCTION

The widespread deployment of connected devices in the Internet of Things (IoT) has substantially increased the amount of data available to developers. Today's IoT applications can make use of this data to enable more sophisticated application scenarios.

When designing an IoT application, the current go-to approach is collecting data at the edge, transmitting it to the cloud for processing, and sending the processed results back to the edge, e.g., to switch on a light in the presence of movement in a smart home scenario [4]. Due to its simplicity, this approach is used by many services, e.g., AWS IoT[1] or the Azure IoT Hub[2]. However, disadvantages include long response times, unnecessary data transmissions and the risk of exposing sensitive data to third parties [1].

[1]aws.amazon.com/iot/
[2]azure.microsoft.com/en-us/services/iot-hub/

Middleware '18 Posters and Demos, December 10–14, 2018, Rennes, France

ACM ISBN 978-1-4503-6109-5/18/12…$15.00
https://doi.org/10.1145/3284014.3284015

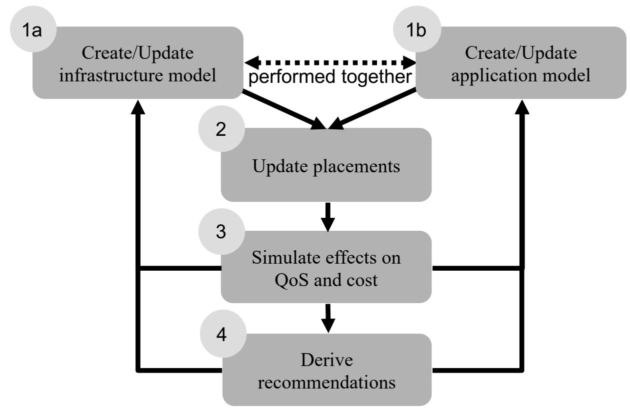

Figure 1: The Iterative Modeling and Simulation Process

Performing some tasks already at the edge, as done by AWS Greengrass[3], can reduce bandwidth consumption and enables the edge to keep operating in the presence of network partitions. However, this approach is limited by available processing capabilities as edge devices are often not powerful enough to run compute-intensive tasks.

An obvious solution to this problem is to leverage the compute power provided by stronger machines such as cloudlets [5, 6] within the core network [1]. This execution environment is commonly referred to as *fog* [1, 2] and consists of edge devices, machines within the core network, and the cloud.

When designing an application for the fog, developers typically have to consider the application architecture, the runtime infrastructure, and the deployment mapping from application modules to infrastructure machines. For each of these three aspects, a number of design options exist and each option can be combined in various ways with options from the other aspects. This leads to a multitude of possible application design options.

Deciding on a particular design should be based on a careful evaluation of effects on quality of service (QoS) and cost. While such an evaluation tends to be complicated, we believe that it is worthwhile as the added benefits of efficiently using the fog can bring significant improvements to IoT applications. Therefore, we developed FogExplorer, an interactive simulation tool for the evaluation of fog-based IoT applications already during the design phase.

2 MODELING AND SIMULATION APPROACH

Our tool is based on an iterative modeling and simulation process which is shown in figure 1: a developer first creates a high level

[3]aws.amazon.com/greengrass/

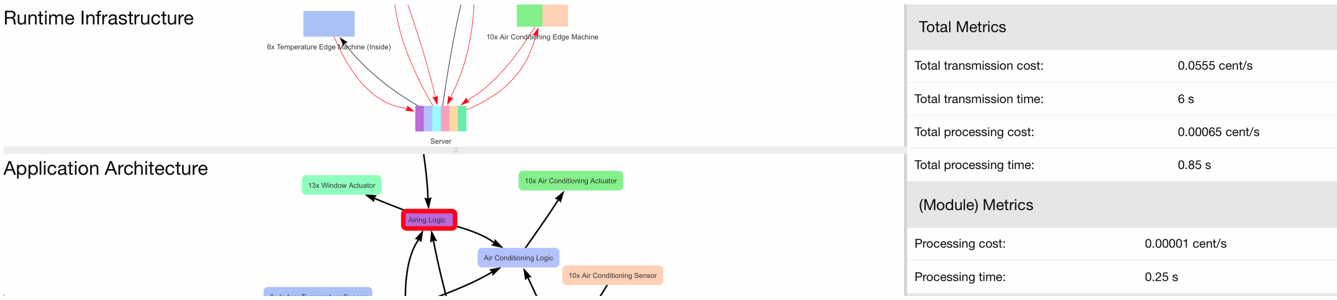

Figure 2: Evaluating a Smart Building Application with the FogExplorer Prototype: Developers can assign application modules (bottom) to infrastructure machines (top) and directly receive feedback on how this affects QoS and cost.

infrastructure model (1a) and application model (1b). The infrastructure model describes machines and their interconnections; the application model defines application modules and all inter-module data streams. Both models support a high level, abstract description of their respective assets, as developers only have a limited amount of information available early during the design phase.

Thus, the properties of the infrastructure model are a performance indicator, which is a rough estimate on the performance in relation to a reference machine, memory amount and price, latency of connections between machines, and bandwidth amount and price for each connection.

For the application model, we identified three general types of application modules: sources, services, and sinks. Sources produce data, services process data and forward results, and sinks receive data. The module properties are the amount of memory required to run properly, the estimated processing time on the reference machine, and three more properties used to determine data output rates per module. If necessary, our model can be extended, but as we explicitly target the application design phase, we opted for the minimum amount of required information possible. Further details can be found in our research paper [3].

Based on a first infrastructure and application model, developers can then start to place application modules on infrastructure machines (2) to create a deployment mapping. Every placement update affects QoS as well as cost and should, hence, trigger a new simulation run (3). By studying these effects, recommendations on how to optimize placements and the infrastructure or application model can be derived (4). This information allows developers to iteratively improve their design and compare different design solutions.

3 FOGEXPLORER

FogExplorer is an interactive simulation tool for the evaluation of fog-based IoT applications during the design phase. The interface of FogExplorer is shown in figure 2, we also prepared a smart building demo[4].

With the current prototype, users can describe infrastructure and application models, create deployment mappings, and study resulting effects on QoS and cost metrics. When using FogExplorer, developers do not have benchmarking results on the available performance of machines or detailed information on the application design available, as it targets the design phase. Thus, we can only require a set of high-level properties for the description of the runtime infrastructure and application architecture for our simulations. These collected properties are used by FogExplorer to simulate *processing cost* and *processing time* for individual application modules, as well as *transmission cost* and *transmission time* for individual data streams.

Based on these individual metrics, our tool also calculates aggregated metrics that enable an easy comparison of design variants. Furthermore, it highlights invalid module placements (a missing infrastructure connection disrupts data streams) and underprovisioned infrastructure resources (a machine has insufficient memory or the connection between two machines lacks bandwidth).

FogExplorer is available as open source[5] and built as a frontend javascript application that utilizes ECMAScript 2015 features. It runs without a backend or web server, so only a modern web browser is required for the execution. We also implemented a node.js[6] package that can be used for automatic model evaluations without the visual front end.

4 CONCLUSION

In this demo abstract, we introduced FogExplorer, our interactive simulation tool for the QoS and cost evaluation of fog-based IoT applications already during the design phase.

REFERENCES

[1] David Bermbach, Frank Pallas, David García Pérez, Pierluigi Plebani, Maya Anderson, Ronen Kat, and Stefan Tai. 2018. A Research Perspective on Fog Computing. In *2nd Workshop on IoT Systems Provisioning & Management for Context-Aware Smart Cities (ISYCC)*, Vol. 10797. Springer, 198–210.
[2] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog Computing and its Role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing (MCC'12)*. ACM Press, 13 pages.
[3] Jonathan Hasenburg, Sebastian Werner, and David Bermbach. 2018. Supporting the Evaluation of Fog-based IoT Applications During the Design Phase. In *Proceedings of the 5th Workshop on Middleware and Applications for the Internet of Things (M4IoT '18)*. ACM, 6 pages.
[4] Lukas Reinfurt, Uwe Breitenbücher, Michael Falkenthal, Frank Leymann, and Andreas Riegg. 2016. Internet of Things Patterns. In *Proceedings of the 21st European Conference on Pattern Languages of Programs - EuroPlop '16*. ACM Press, 1–21.
[5] Mahadev Satyanarayanan, Paramvir Bahl, Ramon Caceres, and Nigel Davies. 2009. The Case for VM-based Cloudlets in Mobile Computing. (2009), 9.
[6] Mahadev Satyanarayanan, Grace Lewis, Edwin Morris, Soumya Simanta, Jeff Boleng, and Kiryong Ha. 2013. The Role of Cloudlets in Hostile Environments. 12, 4 (2013), 40–49.

[4] https://openfogstack.github.io/FogExplorer/

[5] https://github.com/OpenFogStack/FogExplorer

[6] https://nodejs.org/en/about/