VRIJE
UNIVERSITEIT
BRUSSEL

Graduation thesis submitted in partial fulfilment of the requirements for the degree of bachelor in de Wetenschappen: Computerwetenschappen

PYTHON ON THE EDGE

Gérard Lichtert

May 14, 2024

Promotors: Prof. Dr. Joeri de Koster and Prof. Dr. Wolfgang de Meuter.
Advisor: Mathijs Saey

Sciences and bioengineering sciences

Contents

1 Introduction

In a world of electronics and machines where power consumption is always increasing, we must do our best to optimize power consumption. This because electricity is expensive and while there is research being done to make energy generation more efficient, we can still do our part in optimizing the energy consumption to put less strain on the power generation. In the world of distributed computing, according to Pesce, 2021 cloud computing takes 1% of worldwide energy consumption. While this may not seem like much, it is still a significant amount of energy, and we want to do our part to decrease this percentage. While the end goal is to reduce the energy consumption for distributed programs, or more specifically IoT systems, where the generated data gets aggregated on the server in the cloud. We will be looking at reducing the energy consumption of the network traffic caused by transmitting data from the IoT devices to the cloud.

In IoT systems, the IoT devices, or sensors, are responsible for generating data. This data is then sent to the server for processing. Naturally this means that there is extra network traffic caused by the IoT devices. Grossetete, 2020 predicted that "IoT devices will account for 50% of all networked devices by 2023". This also means that there is an increase in energy consumption due to the necessary infrastructure required for the increase in data volume, as stated in J. and Klarin, 2021. What if we could reduce the network traffic by applying the edge-computing principle to the IoT systems? Meaning that prior to the IoT device sending the data to the cloud, it processes it locally first and then sends the preprocessed data to the cloud. This would mean that not all the data has to be sent but rather a subset of it. Logically this should reduce the total network traffic and the required energy. But what if this is not the optimal configuration? What if it is more beneficial to have some parts processed on the IoT devices and some part processed on the server? Or what if it is more beneficial to have some sensors process the data and then send it to the cloud and other sensors to send the data directly to the cloud? To try these different configurations we require a tool that allows us to easily deploy the distributed system and easily change the configuration of the deployment as well as configure where what gets processed. We will be focusing on this deployment tool in this thesis. For now our goals will be:

1. Easily deploy the distributed system
2. Easily allow us to change the configuration of the deployment
3. Easily allow us to declare which data gets processed where

1.1 Distributed computing

To start with distributed computing, or our IoT systems we have to look for a suitable distributed computing paradigm that allows our IoT devices to be deployed from the cloud without much manual configuration. We have several options, but not limited to:

1. Message Passing Interface (MPI) as presented in "MPI: A message passing interface", 1993

2. Remote Procedure Call (RPC) as presented in Tay and Ananda, 1990
3. Shared Memory Model as presented in Herlihy et al., 2013
4. The Actor Model as presented in Hewitt, 2015
5. MapReduce
6. Bulk Synchroneous Parallel (BSP)
7. Publish/Subscribe (Pub/Sub)

First we need to get acquainted with the system and its environment that we will be creating a tool for. The system is an actor-based distributed system that runs actors on different machines. For the time being it runs actors on the sensors and server. The actors on the sensor are responsible for sending data to the server actors, which in turn are responsible for processing and storing the data.

Furthermore, our language of implementation will be Python. This because the company that is involved in the project already has most of the infrastructure in Python and thus tasked us with creating the tool in Python as well.

1.2 Goals

The goal of this bachelor thesis is to create a tool that allows us to statically deploy actors at pre-configured locations. We want to try the configuration of moving part of the processing to the sensors but would also like to try other configurations such as having some preprocessing done on some sensors, and some preprocessing done on the server. We want to be able to easily change the configuration and see the effects of the changes on the network traffic and energy consumption. Note that monitoring the network traffic and energy consumption is outside the scope of this thesis. We will only be focusing on the static deployment of the actors.

2 Conclusion

References

Grossetete, P. (2020). *Iot and the network: What is the future?* https://blogs.cisco.com/networking/iot-and-the-network-what-is-the-future

Herlihy, M., Rajsbaum, S., & Raynal, M. (2013). Power and limits of distributed computing shared memory models [Structural Information and Communication Complexity]. *Theoretical Computer Science*, *509*, 3–24. https://doi.org/https://doi.org/10.1016/j.tcs.2013.03.002

Hewitt, C. (2015). Actor model of computation: Scalable robust information systems.

J., L., & Klarin, Z. (2021). *How trend of increasing data volume affects the energy efficiency of 5g networks.* https://doi.org/https://doi.org/10.3390/s22010255

Mpi: A message passing interface. (1993). *Supercomputing '93:Proceedings of the 1993 ACM/IEEE Conference on Supercomputing*, 878–883. https://doi.org/10.1145/169627.169855

Pesce, M. (2021). *Cloud computing's coming energy crisis.* https://spectrum.ieee.org/cloud-computings-coming-energy-crisis

Tay, B. H., & Ananda, A. L. (1990). A survey of remote procedure calls. *SIGOPS Oper. Syst. Rev.*, *24*(3), 68–79. https://doi.org/10.1145/382244.382832