Project Proposal: Exploring the space of energy constrained devices using NebulaStream

Laurits Bonde Henriksen (lauh) & Markus Kildebæk Raun Johansen (mkjo)

February 2023

The NebulaStream project is a data management system built for IoT architectures. The project attempts to solve some of the challenges involved with using a conventional data management system in an IoT scenario. The main challenge is that all sensor data must be sent to a central server before it can be queried. Energy wise, transmitting data is more expensive, than performing local computations, making such a system ineffective. NebulaStream tries to optimize energy consumption and resource utilization by propagating the queries down to individual end devices (or as close to the data source as possible), utilizing the computational resources of all devices, while transmitting only the necessary data for the query.

The NebulaStream End Device Runtime is a module written in MicroPython, which runs on end devices and can handle and execute incoming NebulaStream queries. However, this module is limited in multiple ways: MicroPython only runs on specific microcontrollers, such as the Pycom boards, making the code limited in portability. The Python language has significant performance overhead because it supports dynamic types, garbage collection, etc.

The goal for this project is to make the *NebulaStream End Device Runtime* module more portable and reduce performance overhead by implementing it in C. Additionally we will attempt to locate possible performance pitfalls in the current codebase and optimize these. The translated library will then be used in a small project to gather data and analyze possible performance and energy improvements.

Intended Learing Outcomes

- Reason about important factors when dealing with energy-constraint devices.
- Develop an implementation of *NebulaStream End Device Runtime* in the C language
- Compare running time and performance of C and MicroPython code.

- Design, implement and evaluate an example IoT project, using the *NebulaStream End Device Runtime* written in C.

Method

The method is experimental. We will design, implement and evaluate a prototype.

What will be handed in

A report containing system design, results and discussion along with a Git repository containing the project code.