Vrije Universiteit Brussel

Proposal bachelor thesis

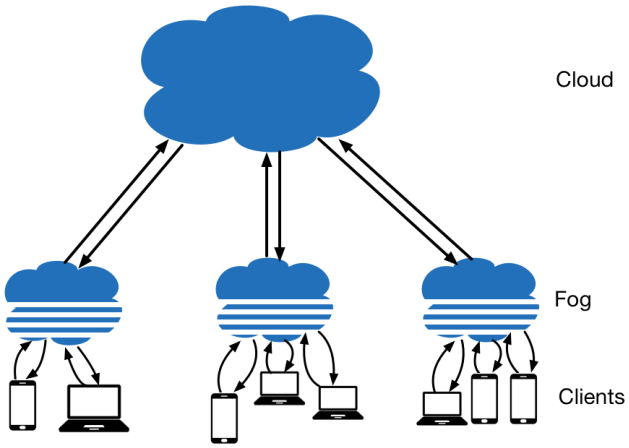
Title: Developing Foggy Applications

Promotor: Wolfgang De Meuter

Includes preparation course: No

Context

The massive adoption of mobile devices, such as smart phones and tablets has given rise to an increasing number of mobile applications that collaborate with the cloud (Google Now, Dropbox, Google Docs, ...).
We also see that the Internet of Things collaborates closely with the cloud. Smart devices constantly upload their state to the cloud which can control these devices.
Although such applications often generates and reason over data that is bound to a geographical area, the data is centralized in the cloud which can be located anywhere in the world.
This means that one cloud is responsible for processing and distributing this data that comes from all over the world as well as for controlling smart devices everywhere. This introduces high latency, which is problematic for applications that rely on real-time information.
In addition, all hardware is centralized which implies very powerful and expensive machines or entire data-centers for such applications. To eliminate these disadvantages, IBM and Cisco introduced the fog computing model.
In this model, clients do not communicate with the cloud, they rather communicate with a fog. The fog consists of a number of access-points that are geographically distributed and serve as an intermediary tier between client and cloud. This architecture serves multiple purposes. The fog can process data that is bound to a geographical area which reduces the latency because clients are communicating with access-points that are nearby. The model also reduces the work-load of the cloud which implies that the processing power and powerful hardware no longer needs to be centralized. The fog can be visualized as follows:



The fog architecture is useful in many scenarios such as collaborative location-driven applications and home automation.
Imagine that every light bulb of every office of the VUB is a *smart device* that can be turned on/off through a mobile application. Every research group could have a fog with which the light bulbs of their offices are connected. The research members of the AI lab are thus not able to switch on/off lights at the SOFT lab and vice versa. Imagine now that the VUB is interested in identifying which research labs are leaving most lights switched on overnight. The VUB has multiple geographically distributed campuses of which the data needs to be compared. This introduces the necessity to have a cloud that keeps track of the information of all the research labs. The fog is in this scenario responsible to aggregate all the data of one particular research lab whereas the cloud compares the data of each fog.

Proposal bachelor thesis

This proposal focuses on developing non-trivial fog-applications with existing technologies. The student can choose which technologies/programming languages (node.js/.NET environment/Java/…) he/she will use to implement the applications.

We expect fog applications to typically have a number of characteristics that make them difficult to implement;

- Certain data needs to be shared and kept consistent between different tiers
- Clients can connect and disconnect at any time with the fog. This should have as little impact as possible for the end user.
- Each tier can spawn events upon which other tiers must respond in near real-time.

Although we have a number of use cases in mind that the student can implement (such as the use case described in the context concerning the light bulbs), we encourage the student to come up with interesting use cases that benefit from the *fog* architecture.

After implementing the use cases, we expect the student to analyze them by extracting recurring patterns, identifying common pitfalls and using established metrics to see what portion of the code is actually concerned with the business logic in comparison to the portion of the code that is concerned with technological complexity.

Contact

Simon Van de Water
svdewate@vub.ac.be