An Approach to Enhance Computational Processing Efficiency through IoT Computing Continuum

Lucas Trombeta, Carlos Alberto Kamienski

UFABC, Av Dos Estados 5001, Santo Andre, Brazil

lucas.trombeta@ufabc.edu.br, cak@ufabc.edu.br

Abstract: This paper suggests an approach to use Iot Computing Continuum concepts to create a flexible distributed computational processing infrastructure based on software defined everything(SDX) strategy

1. Introduction

The emergence of IoT (Internet of Things) along with low power smart devices has changed the computing world with new concepts such as fog and mist computing. This has enabled the engineers and systems developers more flexibility on their creations and the very concept of computing, because the physical limitations of the past have now being surpassed by new handheld devices which are many times more powerful than what was available on the last decade.

It is also important to notice that one feature which also enabled this level of integration were the new high-speed optical networks offering unprecedented speeds, with 40Gb per fiber core being common nowadays [1], and speeds up to 100 Tb being achieved in experimental settings [2].

The combination of both high speed networks and low power devices enables the outsourcing of processing power to a centralized cloud while still retaining the ability to gather local data fast and with some kind of local processing, this has created other challenges such as task orchestration on a hybrid computing system.

Many of those challenges are nowadays manually solved within each system, with programmers and engineers having to consider each individual system specifics.

2. IoTinuum

According to Zyrianoff et al. [3], is possible to gather the IoTecture components (namely L1-Device, L2-Transport, L3-Data, L4-Model, and L5-Service) and treat them using stages(S1-Thing, S2-Mist, S3-Fog, S4-Cloud, and S5-Terminal) as a structure called Iot Computing Continuum (IoTinuum). The index of each stage refers, in general, to the network bandwidth capacity and processing power (e.g. S1 lowest bandwidth capacity and lowest processing power, S4 high bandwidth capacity and high processing power). IoTinuum approach turns its use transparent to project and development members.

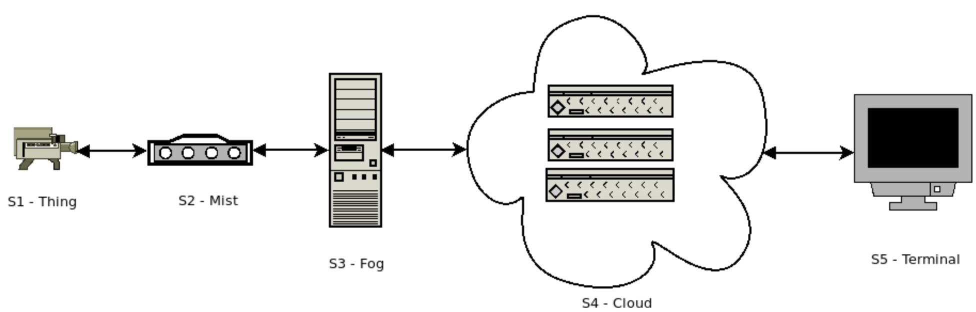

Fig. 1. Example of IoTinuum stages

3. Deployment

In order to maximize the use of computational resources, the idea of IoTinuum might be explored as a flexible distributed computational processing infrastructure. Although we usually have more computational processing capacity at cloud, we might have some different devices inside IoTinuum that can be more efficient for some algorithms and can be shared through different instances or projects that are evolved at the same structure (e.g. small FPGA hybrid devices).

A new orchestration level is required to be able to manage this distribution. This orchestrator should consider some metrics such as capacities, confidence, cost, time processing, network latency, distance and bandwidth among others to allow time processing improvements through this infrastructure.

The strategy designed by this paper is inspired on Software Defined Everything(SDX) [4] which devices capabilities are used according to a controller that orders them how to act.

The main benefit of using this concept is having a flexible distributed computational power processing that enhances computational processing efficiency from a group of devices working together through a bundle of different tasks.

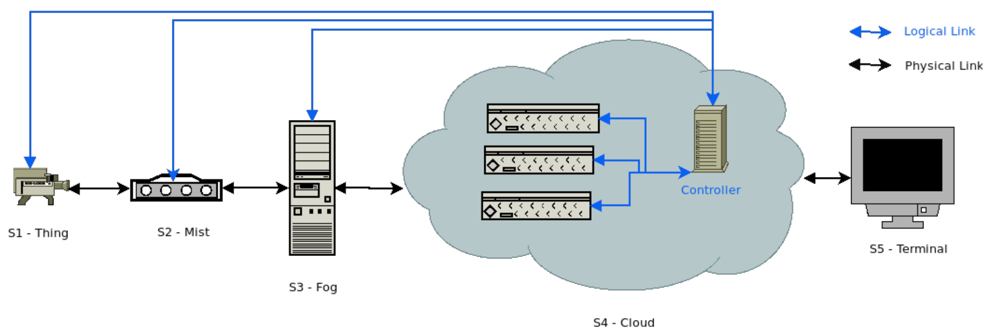

Fig. 2. IoTinuum stages with a device controller and logical control network

On the other hand, a network control layer that evolves most of the devices must be implemented over the same physical structure that already exists, and it will possibly cost some effort and bandwidth from all network links. Once IoTinuum network is made of different kinds of network links, the cost of an additional network layer is different between them and can harm edge devices connections, which are in general the lowest bandwidth and high latency links. It must be considered when the controller raises an instance to run part of the processing on an edge device.

Another important challenge is the processing migration between stages. The previous designed flow for the processing result should be granted. In other words, the final path of the data generated by processing (storage, terminal or entry of another process that is handled by another device) must not be changed.

An approach that might be used is attach a flag to every processing that can be moved between different stages. It might prevent future inconsistencies after migrations, once is possible to check every flexible processing dependency in a processing flow.

4. References

[1] P. J. Winzer, D. T. Neilson, and A. R. Chraplyvy, "Fiber-optic transmission and networking: the previous 20 and the next 20 years," *Optics express*, vol. 26, no. 18, pp. 24190–24239, 2018.

[2] D. J. Blumenthal, H. Ballani, R. O. Behunin, J. E. Bowers, P. Costa, D. Lenoski, P. A. Morton, S. B. Papp, and P. T. Rakich, "Frequency-Stabilized Links for Coherent WDM Fiber Interconnects in the Datacenter," *Journal of Lightwave Technology*, vol. 38, no. 13, pp. 3376–3386, 2020.

[3] I. Zyrianoff, A. Heideker, D. Silva, J. Kleinschmidt, J.-P. Soininen, T. Salmon Cinotti, and C. Kamienski, "Architecting and deploying IoT smart applications: A performance–oriented approach," *Sensors*, vol. 20, no. 1, p. 84, 2020.

[4] H. Mei and Y. Guo, "Toward ubiquitous operating systems: A software-defined perspective," *Computer*, vol. 51, no. 1, pp. 50–56, 2018.