

Cloud and Edge Computing for Developing Smart Factory Models using a iFogSim Wrapper: Transportation Management System (TMS) Case Study

Instructor: Dr. Sabah Mohammed
COMP5112-WC Supervisor
Lakehead University

Karankumar Bhaskarbhair Patel
Student Id: 1111107
Lakehead University

Bhavik Subhashchandra Ray
Student Id: 1111419
Lakehead University

Dhairya Shaileshbhai Patel
Student Id: 1110149
Lakehead University

Vikram Singh
Student Id: 0889607
Lakehead University

Abstract—The given literature focuses on developing a Smart Factory model based on Cloud and Edge computing used to develop Transportation Management System(TMS) using a iFogSim wrapper. Cloud computing identifies data centres for users and offer computer system services on-demand, including data storage and processing power, without direct active user management. In the smart industry, several devices are connected together across the internet, where vast volumes of data are collected during the entire process of output. Thus, to handle this data smart factory based on cloud and edge computing is used. The intelligent cloud-based factory offers some facility like large scale analysis of data. Concepts like fog and edge computing play a significant role in extending data storage and network capacities in the cloud that addresses some challenges, such as over-full bandwidth and latency. The literature also focuses on the implementation of TMS using the iFogSim Simulator. The simulator provides efficient execution of TMS by showing the amount of resources used which gives an idea regarding optimum use of resources. All types of data related to TMS is obtained at cloud by using smart factory like object location, time taken and energy consumption. To implement the TMS we have created a topology which displays various devices connected to the cloud which gives necessary information regarding the ongoing transportation simulation.

Index Terms—Cloud, Edge and Fog Computing, Smart Factory, Supply Chain Management, iFogSim, TMS.

I. INTRODUCTION

The term Smart Factory describes a highly digitalized and connected environment where machinery and equipment are able to improve processes through automation and self optimization[6]. The entirely automated manufacturing system has already been the ultimate objective of industrial companies for several decades. Cloud computing offers a number of advantages and definitely plays an ample role in smart manufacturing[5]. With the collaboration of thin client and cloud computing, the server will perform all the major tasks like launching software programs, performing calculations, data storage etc. Merits of using thin client will enhance

the aspects like data security, accessibility and productivity. Supply chain model in context with cloud provides better services to improve the execution of smart factory procedures. Also, iFogSim is the simulator which provides customized environment to develop simulation with various services. TMS' simulation is being implemented on the iFogSim where various sensors and actuators are used. We aim to provide the sort of quick and virtually unimpeded communication required to establish smart factory with an optimized edge- cloud architecture on the TMS.

Initially, the smart factories were having manual workforce dealing with the maintenance and other handling functions. Various manufacturing process consists of large amount of data which requires storage devices. Network issues created obstruction in some functionalities. Smart factories using transportation management system are deprived of some factors like estimated delivery time, object location, temperature record for goods. Also, their respective Transportation management system are not connected with the sensors and actuators. The supply chain needs to be provided with proper storage devices which stores the supply data accordingly. Selection of proper implementation model is also mandatory. The status of the entire system is not retrieved which leads to inaccurate information. Administration needs to assign various employees to supervise the workflow and take the note of each and process accordingly. Sometimes, this may lead to incorrect observation of the processes which increase the chances of error.

Fig.1 shows the general architecture of smart factory in which Physical Manufacturing System and Cloud Environment are the major components of the architecture. The cloud services are connected with cloud in order to access the services of the cloud. Data capturing, Data management, Real time tracking, Process Planning, Execution control are the major process included in the Cloud Environment.

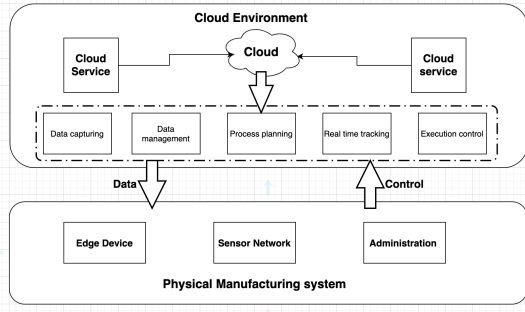


Fig. 1. Smart Factory Architecture

II. CLOUD AND EDGE COMPUTING

Enterprises shift quickly in real time into event-based systems and automated business operations. Moreover, given the various suppliers with their specific features and applications in every region, the big challenge is the management of purely centralised cloud services and corporate data centres. An architecture with edge computing and centralised cloud capabilities with growing interest in new technologies such as automated manufacturing, increased reality and multiple IoT technologies is needed. All data is stored and processed in cloud computing, usually in a data centre, in a centralised location. All devices needing access to or using associated applications must first connect to the cloud. As the entire cloud is centralised, safe and control is easy, though reliable remote access is still possible.

III. RELATED RESEARCH WORK

Our research was totally inclined towards the accomplishing the goal of successful implementation of smart factory. Thin clients, supply chain management, TMS, implementation of transportation simulation of iFogSim were our major areas of research. Each and every factor mentioned above have their own characteristics in terms of efficiency and working capability.

In Smart Manufacturing for a production with a greater efficiency will require a proper monitoring as well as maintenance. Also, providing manual updates to the devices or a software at a large scale will create a tedious job and also investment will increase to a greater extent. Thus, thin clients are those devices which can be established at a particular location which will function like a PC and perform all the primary tasks required for smart manufacturing.

The basic concept of Supply Chain Management says that it is the process of obtaining the finalized product from the raw material in which the processes like manufacturing, distribution, supplying plays a key role. The best supply chain model which is implemented in the smart manufacturing is the cloud-based supply chain model. The traditional supply chain model takes consideration of cost, software utility, hardware and physical facility which is proved to be least efficient. However, in cloud-based supply chain model only software is taken into consideration. Also, we get extra cloud storage

for data storage which can be used as an alternative if the storage devices are full.

TMS gives a detailed information regarding the transfer of goods from one place to another[9]. Data regarding source and destination, information of goods and their respective qualities and other necessary information can be retrieved using cloud. Moreover, using a iFogSim simulator we have implemented a simulation which shows a basic topology of the TMS used in our smart factory. To study the main functionalities of TMS the classes of iFogSim play a crucial role as the topology of TMS is developed using iFogSim. Fig 2 displays the iFogSim physical topology. Classes are mentioned below.

- FogDevice
- Sensor
- Tuple
- Actuator
- Application

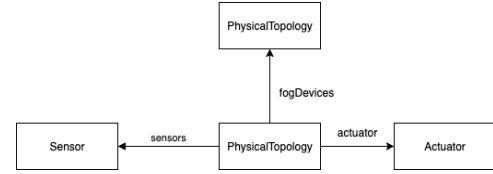


Fig. 2. iFogSim physical topology classes

1) *FogDevice*: The hardware properties of Fog devices and their connections with other Fog devices, sensors and actuators are listed in this section. The key attributes of the FogDevice class are available power, processors, storage capacity, uplink, downlink bandwidths, and extensions from the PowerDatacenter class in CloudSim[10].

2) *Sensor*: The class includes attributes, ranging from connectivity to output characteristics of a sensor. The class includes a reference attribute to the Fog gateway to which the sensor is connected and the contact latency between the sensors. Most significantly, the sensor output properties and the tuple inbound time distribution are specified which specifies the tuple rate of arrival at the gateway[10].

3) *Tuple*: Tuples constitute the fundamental coordination unit between Fog entities. Tuples are seen as Tuple class instances of iFogSim inherited from CloudSim's Cloudlet class. A tuple is defined by its type and its programme modules for the source and destination. The class attributes specify the processing requirements and the length of data encapsularized in the tuple (defined as a million instructions (MI)[10].

4) *Actuator*: By specifying its network contact property, this class models an actuator. The gateway to which the actuator is attached and the latency of the connection relate to an attribute in the class[10]. The class specifies a method by which a tuple from a module may take an action.

5) *Application*: An application is modeled as a directed graph, the vertices of the Directed Acyclic graph(DAG) representing modules that perform processing on incoming data and edges denoting data dependencies between modules[10].

- **AppModule**: Instances of AppModule class represent processing elements of Fog. For each incoming tuple, an AppModule instance processes it and generates output tuples that are sent to next modules in the DAG. The number of output tuples per input tuple is decided using a selectivity model which can be based on a fractional selectivity or a bursty model.
- **AppEdge**: An instance of AppEdge denotes the data dependence between a couple of application modules and represents a directed edge of the application model. Edge is defined by the kind of tuple it contains that the AppEdge class tupleType attribute, the processing conditions and the length of the data encapsulated in these tuples are captured applications. Fig 3 displays the iFogSim classes.

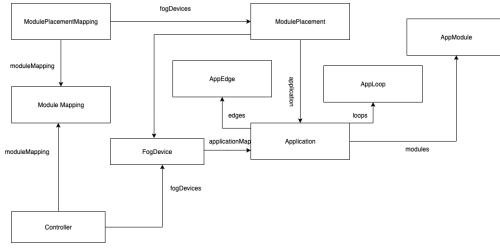


Fig. 3. iFog Classes

IV. METHODOLOGY

A. iFogSim Approach

iFogSim is based on the definition of fog computing that presents it as an infrastructure having similar characteristics as cloud computing but placed close to the edge of the network. It does not support device-to-device communication as it assumes a hierarchical organization of fog devices. Application placement happens in a north-south direction, and it is not possible (at present) to offload modules to another device on the same level of hierarchy. Fig 4 represents the class diagram for the components of iFogSim.

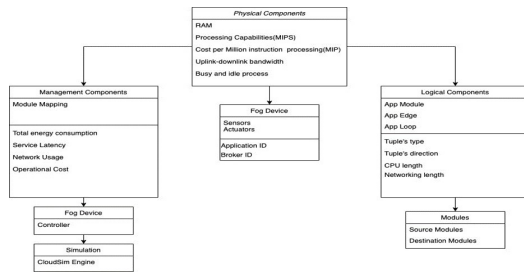


Fig. 4. Components of iFogSim class representation

Relying on rapid advancement of hardware and communication technology, Internet of Things (IoT) is consistently promoting every sphere of cyber-physical environments. Consequently, different IoT-enabled systems such as smart healthcare, smart city, smart home, smart factory, smart transport

and smart agriculture are getting significant attention across the world. Simulation toolkits not only provide frameworks to design customized experiment environment but also assist in repeatable evaluation. There exists a certain number of simulators such as Edge cloudsim, Simple IoT Simulator and iFogSim for modelling Fog computing environment and running experiments. Since iFogSim does not resolve the network issues at low levels, such as the interference management of highly positioned devices, users must compare these low-level problems with high-level characteristics such as latency or communicating bandwidth between IoT devices and gateways. Comprehensive profiling helps the user to create a wireless component behaviour model of IoT devices that can be plugged into iFogSim and simulated[2].

1) *Components of iFogSim*: In a significant amount of research works, there are 3 major iFogSim components[3]:

- Physical components
- Logical components
- Management components

a) *Physical components*: Physical components include Fog devices (Fog nodes). The Fog devices are orchestrated in hierarchical order. The lower level Fog devices are directly connected with associated sensors and actuators. Fog devices act like the datacentres in Cloud computing paradigm by offering memory, network and computational resources. Each Fog device is created with specific instruction processing rate and power consumption attributes (busy and idle power) that reflects its capability and energy efficiency.

b) *Logical components*: Application modules (AppModules) and Application edges (AppEdges) are the logical components of iFogSim. In iFogSim applications are considered as a collection of interdependent AppModule that consequently promotes the concept of distributed application. The dependency between two modules are defined by the features of AppEdges.

c) *Management components*: Management component of iFogSim is consist of Controller and Module Mapping objects. The Module Mapping object according to the requirements of the AppModule, identifies available resources in the Fog devices and place them within it. By default, iFogSim support hierarchical placement of the modules. If a Fog device is unable to meet the requirements of a module, the module is sent to upper level Fog device.

2) *Interaction between components of iFogSim*: The interaction between the components of the iFogSim is shown in the fig 1. The Physical components have the sensors and actuators whereas management components deal with mapping and the controller. Logical components contain application modules. The interaction between all of them is necessary because the simulator's implementation can't be possible without the presence of any of the components. If Physical components would be absent then the location of the particular vehicle won't be retrieved. So, if a Global Positioning System(GPS) sensor won't be there then location can't be identified.

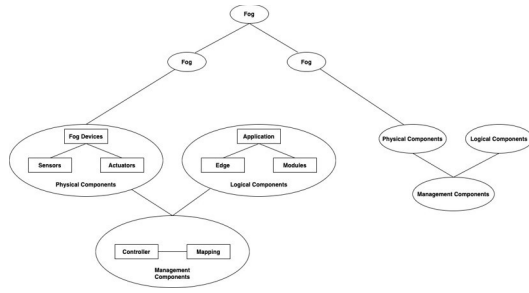


Fig. 5. Interaction between components of iFogSim

B. Importance of Simulators

Simulation modeling solves real-world problems safely and efficiently. It provides an important method of analysis which is easily verified, communicated, and understood. Across industries and disciplines, simulation modeling provides valuable solutions by giving clear insights into complex systems[1]. Below mentioned are some advantages of using a simulator

- Risk Free Environment
- Visualization
- Increased Accuracy
- Handle Uncertainty
- Saves Money and Time

C. TMS based on Smart factory

In TMS the goods are being transported from source to destination. This is the basic concept of TMS. Our domain is Cloud and Edge computing and we are developing a smart factory using a iFogSim Wrapper. By providing TMS some more features using smart factory. Traditional TMS system were deprived of having advanced features like obtaining each and every detail of the transportation at the server. Fig 6 displays the Graphical User Interface of the system.

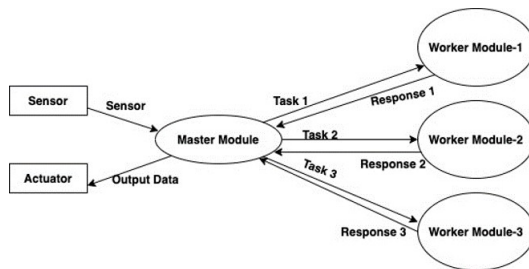


Fig. 6. Graphical User Interface

1) *Using smart factory on TMS:* As mentioned earlier our domain is Smart Factory based on Cloud and Edge Computing. So, using cloud computing the data of a particular transportation event can be retrieved on the cloud server. In our model, we have used the sensors and the actuators in the TMS. We have used the GPS sensor to locate the vehicle's position. We will also get the data regarding source and destination and details about the time taken to finish the transportation

as well. Due to the help of the sensors and actuators we will get the exact amount of temperature of some commodities which are being transported. Data regarding consumption of energy, total delay and object detection are retrieved by using cloud and edge computing. These are the some advantages of using smart factory based on cloud and edge computing on TMS. These features will gradually improve the feasibility and efficiency in terms of execution to a great extent.

2) *Modeling of TMS:* We have created a topology on TMS using iFogSim. The topology on TMS follows a hierarchy. It consists of various levels. To create topology first we need add uplink bandwidth and downlink bandwidth, Million Instructions per second, location and RAM. We have bifurcated our topology into three levels. The level 1 includes the cloud whereas the level 2 has the mobile devices which are interconnected to the cloud itself. Subsequently, level 3 consists of Sensors and actuators which are connected through mobile devices. Cloud is the start node whereas the mobile devices are targeted nodes as they are linked with the cloud. Similarly, the mobile devices are considered as the start node as the sensors and actuators are connected with them. Sensors and actuators are the target node of the mobile devices. Initially, when the start node is created edges are assigned which are connected to the target node forming a topology. Fig 3 shows the topology of TMS.

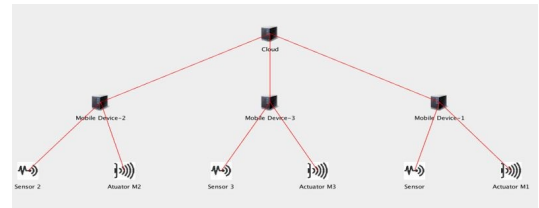


Fig. 7. Topology of TMS

3) *Benefits of Simulation on TMS:* The real world implementation of Fog's research system is very expensive, as Fog's computing system combines IoT, Fog Nodes and Cloud datacenters with vast amounts of IoT data and distributed application. In addition, changes in every organisation in the Fog system in the real world would be repetitive. Simulation of the Fog computing environment may be very useful in this case. The toolkits provide not only structures for the design of personalised experimental environments but also support repeat evaluations[3]. As mentioned earlier, topology is created of TMS using iFogSim because we can get a clear view of working of TMS by using this approach. In the implementation of TMS we have used the GPS sensor. So we can get the location of the object as well as simulation will give a proper flow of the transportation started from source to destination.

REFERENCES

- [1] <https://www.anylogic.com/use-of-simulation/>
- [2] Gupta, Harshit, Amir Vahid Dastjerdi, Soumya K. Ghosh, and Rajkumar Buyya. "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments." *Software: Practice and Experience* 47, no. 9 (2017): 1275-1296.

- [3] <https://github.com/Cloudslab/iFogSimTutorials>
- [4] Rahbari, Dadmehr, and Mohsen Nickray. "Scheduling of fog networks with optimized knapsack by symbiotic organisms search." In 2017 21st Conference of Open Innovations Association (FRUCT), pp. 278-283. IEEE, 2017.
- [5] Shi, Weisong, and Schahram Dustdar. "The promise of edge computing." *Computer* 49, no. 5 (2016): 78-81.
- [6] Li, Xiaomin, Jiafu Wan, Hong-Ning Dai, Muhammad Imran, Min Xia, and Antonio Celesti. "A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing." *IEEE Transactions on Industrial Informatics* 15, no. 7 (2019): 4225-4234.
- [7] Schneider, Dean. *An Introduction to Smart Manufacturing: Lighting the Path to the Value of your Data*. Univ. of California, Los Angeles, CA (United States), 2019.
- [8] Uddin, Mohammad Nashir, He Lie, and Hao Li. "Hybrid Cloud Computing and Integrated Transport System." In 2017 International Conference on Green Informatics (ICGI), pp. 111-116. IEEE, 2017.
- [9] Benotmane, Zineb, Ghalem Belalem, and Abdelkader Neki. "A cloud computing model for optimization of transport logistics process." *Transport and Telecommunication Journal* 18, no. 3 (2017): 194-206.
- [10] Gupta, Harshit, Amir Vahid Dastjerdi, Soumya K. Ghosh, and Rajkumar Buyya. "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments." *Software: Practice and Experience* 47, no. 9 (2017): 1275-1296.