Modelling Energy Consumption of IoT Devices in DISSECT-CF-Fog

Andras Markus and Attila Kertesz
Department of Software Engineering, University of Szeged, Szeged, Hungary

Keywords: Energy Consumption, Fog Computing, Internet of Things, Simulation.

Abstract: The continuously evolving information technology carries requirements to foster cost, resource and energy-aware systems. The Internet of Things is considered as one of the most trending technology, which is often coupled with Cloud or Fog Computing resources that manage the possibly big data generated by smart devices in an effective way. To reduce the carbon footprint of such IoT-Fog-Cloud infrastructures, planning and optimisation of their energy consumption is necessary to realise sustainable solutions. It is also inevitable to use simulation in the design phase of such complex systems, hence it would by hardly feasible and rather costly to evaluate numerous settings effecting the energy use. In this paper, we design an IoT energy model based on real world measurements, and propose an extension of the energy model of the DISSECT-CF-Fog simulator to enable the energy usage monitoring of complex, IoT-Fog-Cloud infrastructures. We also present a validation of the extension with a weather forecasting use case to exemplify its configuration possibilities that meet the design requirements of the energy sector.

1 INTRODUCTION

The latest complex distributed systems involving thousands of IoT devices promote widely usable services by leveraging the computing and storing capacities of cloud datacenters. To enhance the elasticity of a concrete service, cloud resources are often aided by resource-constrained fog nodes to improve the response time of the IoT application and to disperse the various types and unforeseen amount of data (Mahmud et al., 2018).

Besides scalability, latency and resource management issues, energy consumption of a fog environment and the corresponding smart devices is also a great challenge as stated in (Atlam et al., 2018), therefore it should be considered as one of the key factors of the development of Fog Computing solutions. Energy-efficient solutions also have a significant impact on carbon footprint and on climate change. In order to avoid wasting energy, smart decisions could take into account IoT device motion or corresponding environmental parameters, in order to handle optimally the related equipment – and such analytic evaluation is usually done in the cloud (Motlagh et al., 2020).

In parallel with the spreading of Cloud Computing, the need for energy-aware systems have been increased, which led to the appearance of Green Computing (Garg and Buyya, 2011). The main features to handle requests in an energy-efficient way are the following: (i) dynamic provisioning, (ii) multi-tenancy, (iii) server utilisation and lastly (iv) datacenter efficiency (requiring the usage of energy-efficient technologies). An IoT system generally utilises different types of devices, such as smart phones or microcontrollers, which are responsible for sensing the environment and behave as a gateway between the heterogeneous system components. Microcontrollers typically have low capacity of processing unit and memory. They connect to the Internet using wireless network protocols, such as Bluetooth or WiFi. It is not negligible that the energy consumption of these devices are significantly less compared to the energy usage of a cloud or fog node (Samie et al., 2016).

Analysing complex IoT-Fog-Cloud systems in a simulation environment is a common practise among researchers, because investigating various large-scale network topologies with thousands of IoT devices barely feasible in real world. Though the requirements of such a simulator are straightforward, i.e. to ensure detailed, realistic and fine-grained model of all entities (e.g. datacenters, fog nodes and IoT devices), the realisation of corresponding simulators require great efforts. The survey paper by (Markus and Kertesz, 2020) overviews the current approaches, revealing that they are far not complete, and energy modelling of these systems are rarely studied.

The main contributions of this paper are: (i) de-

Markus, A. and Kertesz, A.
Modelling Energy Consumption of IoT Devices in DISSECT-CF-Fog.
DOI: 10.5220/0010500003200327
In *Proceedings of the 11th International Conference on Cloud Computing and Services Science (CLOSER 2021)*, pages 320-327
ISBN: 978-989-758-510-4

signing an IoT energy model based on real world experiments, (ii) proposing a novel extension of the DISSECT-CF-Fog simulator (Markus et al., 2020) for energy usage monitoring of IoT devices with this model, and (iii) validating of the proposal with a weather forecasting use case to exemplify the configuration possibilities and the use of the extended simulator.

The rest of this paper is organised as follows: in Section 2 we briefly summarise the related works, in Section 3 we introduce our simulator extension and the proposed energy model of microcontrollers. Section 4 presents the evaluation with various IoT use cases, and finally, Section 5 concludes our work.

2 RELATED WORK

The literature has numerous studies covering different aspects of energy-aware approaches related to the interoperation of Cloud and Fog Computing and IoT. Since our goal is to propose a detailed energy model for these complex systems, including both computational resources and sensor devices, we restrict ourselves to focus on existing simulation tools.

Nevertheless the monitoring of energy consumption entails significant challenges for IoT-Fog-Cloud systems. A research paper by (Sun et al., 2020) aims to resolve the task offloading and resource allocation of an IoT-Fog-Cloud architecture as a minimisation problem of energy and cost. The authors consider different energy consumption for a given task separately executed on IoT device, fog or cloud node. A slightly detailed approach was proposed in (Oma et al., 2018), where the authors differentiate computation, storing and sending processes of the data, however they applied a simple power consumption model, which means that the energy consumption is set to the maximum, if at least one process is executed, otherwise the computing resource consumes minimal energy. (Ahvar et al., 2019) shows an elaborated model, in which static and dynamic consumption components are defined, thus each of them takes into account the computational and network (i.e. routers and switches) parts of the system considering idle power, cores power, which is proportional to the CPU usage of a resource, and lastly, energy consumption of a switch, when storing or transmitting data.

Such formal models are sufficient for the analysis of specific IoT and fog related problems, however considering only the extreme values of energy consumption, which may lead to a distortion of the results. Nevertheless, more sophisticated approaches take into consideration the diverse types of constants and factors. For example, the energy consumption of a computational node is represented by the power usage effectiveness (PUE) coefficient, which makes the model more punctual. In general, a simulation-based approach may conduct to a more configurable and realistic model, for instance the energy consumption of various entities (i.e. CPU, network or storage) connected to state transition of those leads a more manageable and extensible realisation.

Numerous studies offer and compare IoT and fog simulations based on different properties and features - without claiming completeness -, such as code quality, architecture, VM management, resource and energy consumption, addressed in (Perez Abreu et al., 2020) and (Markus and Kertesz, 2020).

The CloudSim-based solutions, such as iFogSim (Gupta et al., 2017) or EdgeCloudSim (Sonmez et al., 2018) inherit the energy model, which defines the consumption as follows: a static constant power (e.g. a switched on machine) as the idle power, and dynamic components (using linear function) as the busy/max power are summed. IoT devices and fog nodes are handled equally in this model.

FogNetSim++ associates the energy consumption to certain tasks as the combination of device and fog node consumption (Qayyum et al., 2018). The device energy is calculated based on the transmission power weighted by the ratio of the task size and the bandwidth, whilst the fog node energy takes in consideration the idle power with weight of the task size and the computing power. This approach also considers the residual energy of mobile nodes.

YAFS also ensures a minimal model for monitoring the energy consumption (Lera et al., 2019), however it is represented by only a static consumption value of computing and IoT entities. IoTSim-Edge targets to model smart devices using low energy protocols (Jha et al., 2020). The authors also consider the simulation of battery power by using a predefined drainage rate.

DISSECT-CF-Fog can be used to simulate cloud, fog and IoT infrastructures, and their combined, hybrid solutions. As a result, three types of elements can appear in it simulations: cloud datacenters, fog nodes and IoT devices. The initial energy model of DISSECT-CF published in 2015 (Kecskemeti, 2015) covered cloud datacenters, by introducing resource consumption modelling for CPU, disk and network energy utilisation. This approach consider minimum, idle and maximum power values as well during the calculation of energy consumption based on linear interpolation. As a summary, our current extension of DISSECT-CF-Fog has the most detailed energy model using dynamic consumption values for the

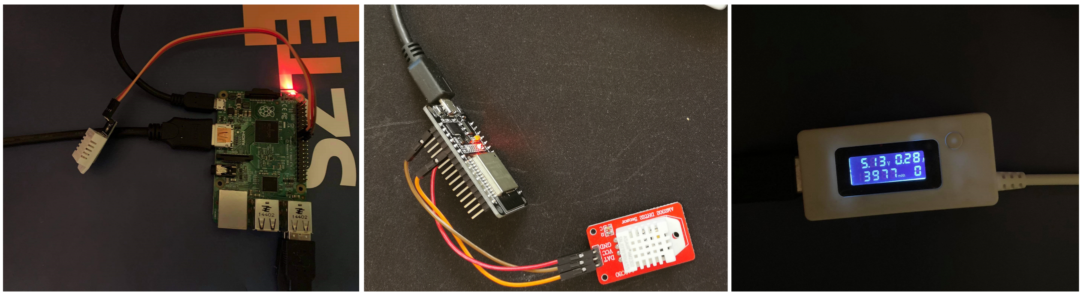

Figure 1: The utilisation of Raspberry Pi (left) and ESP32 (middle) microcontrollers and KCX-017 meter (right).

widest variety of resources (cloud, fog and IoT).

3 MODELLING ENERGY CONSUMPTION IN DISSECT-CF-FOG

In this paper we build on the original model of DISSECT-CF and use its earlier proposed extension for fog and IoT systems. To ensure the required level of system granularity, the simulator mimics the behaviour of infrastructure clouds by predefined states of physical machines (PM), virtual machines (VM), storage and disks. For instance, a PM can be in the following states: turned off, switching on, running and switching off. As a result, the basic concept of energy saving can be easily realised by turning off the unused machine. Besides, this refined model supports the mapping of certain energy consumption values to the predefined states, which ensures the fine-granularity of the simulator.

As we discussed earlier, the energy model takes into account: the minimum (min) power (e.g. the machine/device is turned off, but still plugged into the energy source), the maximum (max) power (e.g. if the CPU is fully utilised), and the idle power (e.g. when the PM is running without executing computational tasks). At this moment the simulator has two power models: (i) dynamic power draining behaviour applies linear interpolation between idle and max power values, whilst (ii) constant power draining behaviour can consider any power value (e.g. min). By default, the dynamic model is applied in case of states with high energy consumption (e.g. running state of a PM), and it handles the idle power with min power values, and the consumption range can be get by subtracting the idle power value from the max power value.

In the simulator, the *PhysicalMachineEnergyMeter* class can be utilised to monitor the physical cloud resources concerning energy consumption (introduced in DISSECT-CF-Fog in 2015), and we proposed a fog extension (with DISSECT-CF-Fog in 2020) by introducing the *ComputingAppliance* class to simulate fog nodes (possibly having additional parameters). With this extension we arrived to a unified energy model for fogs and clouds, since the *PhysicalMachineEnergyMeter* class is encapsulated in the *ComputingAppliance* class, and can be used to simulate both fog and cloud nodes.

In this work we take a step forward, and cover IoT devices with our proposed extended model, to enable complex energy utilisation analysis of IoT-Fog-Cloud systems. First, we started to analyse real power consumption of microcontrollers, which is detailed in the next subsection.

3.1 Analysis of Real Microcontrollers

In order to determine a fine-grained energy model for microcontrollers, we measured and collected energy consumption values of real devices. In our experiments we chose ESP32[1] (WROOM-32) and Raspberry Pi[2] (1 Model A+) microcontrollers for further analysis. Both devices were equipped with DTH22 temperature and humidity sensors, and a KCX-017 meter was applied to display the voltage and the current of the connected through USB port. The assembly of the used gadgets can be seen in Figure 1.

To measure the general power consumption of IoT applications, we developed a typical and simple program written in Python/MicroPython covering the following functionalities: sensor data reading (temperature and humidity values in our current case), message creation and sending as an IoT client device by using the MQTT protocol. We scheduled sensor value sampling every minute by default, and connected the devices to thee Internet via WiFi. The data application running on the microcontrollers forwarded the sensor

[1] The official website of ESP32 is available at: https://www.espressif.com/en/products/socs/esp32/

[2] The official website of Raspberry Pi (RPi) is available at: https://www.raspberrypi.org/

Table 1: Uniform sampling of microcontrollers.

Microcontroller	ESP32			Raspberry Pi (RPi)		
Sampling (min.)	V	I	P	V	I	P
1	5.13	0.07	0.3591	5.12	0.28	1.4336
2	5.17	0.02	0.1034	5.12	0.26	1.3312
3	5.19	0.02	0.1038	5.12	0.31	1.5872
4	5.17	0.02	0.1034	5.13	0.26	1.3338
5	5.19	0.02	0.1038	5.13	0.28	1.4364
6	5.17	0.02	0.1034	5.12	0.28	1.4336
7	5.16	0.05	0.258	5.13	0.28	1.4364
8	5.19	0.02	0.1038	5.13	0.26	1.3338
9	5.21	0.02	0.1042	5.12	0.28	1.4336
10	5.17	0.02	0.1034	5.13	0.28	1.4364
11	5.18	0.02	0.1036	5.13	0.31	1.5903
12	5.17	0.02	0.1034	5.13	0.28	1.4364
13	5.20	0.05	0.26	5.12	0.28	1.4336
14	5.21	0.02	0.1042	5.13	0.28	1.4364
15	5.18	0.02	0.1036	5.13	0.26	1.3338

Table 2: Mapping the benchmark and measured values to the model power values in DISSECT-CF-Fog.

Data Source		Research Papers		Websites		Our Experiments		IoT Energy Model	
Microcontroller		ESP32	RPi	ESP32	RPi	ESP32	RPi	ESP32	RPi
Power Cons. (W)	min	N.A.	N.A.	0.01	0.1	N.A.	N.A.	0.01	0.1
	idle	0.17	0.94	0.04	1.1	0.1	1.33	0.1	1.1
	max	0.28	1.57	0.42	2.1	0.36	1.59	0.35	1.75

data to an IoT analytics platform called Thingspeak[3], where it could be visualised.

To determine the electric power (P measured in watts) in the SI system, we multiplied the metered voltage (V measured in volts) with the metered electric current (I measured in amps) values:

$$P = V * I, e.g. 1W = 1V * 1A \quad (1)$$

Finally, we can determine the energy usage (J measured in joules/watt-second/kilowatt-hour) by:

$$J = P * t, e.g. 1J = 1W * 1s \quad (2)$$

3.2 The Energy Model for IoT Devices

Finally, in order to utilise monitored data of real IoT devices in DISSECT-CF-Fog, we executed our sampling application five times on both microcontrollers for 15 minutes, while measuring the power consumption each millisecond.

Table 1 presents the average values of the uniform sampling of the metering device for each one minute periods. Based on the monitored values, we calculated the electric power. The results show that our typical IoT monitoring application consumed 0.1 to 0.36 *W* per minute in average with ESP32, and 1.3 to 1.59 *W* with RPi. In the next subsection we show how we applied these measured values to our proposed IoT energy model.

Concerning power consumption of IoT resources, we had to build up the energy model from scratch. In this work, we had to extend the *Device* class of DISSECT-CF-Fog, which represents any smart objects, and responsible for power consumption metering for IoT devices during simulations.

In the previous version of DISSECT-CF-Fog, only the elements of the IoT layer lacked energy metering functions. To resolve this issue, we decided to create the *Microcontroller* class for implementing our energy model of microcontrollers. Such realisation keeps the already existing functionalities (e.g. data sensing of IoT sensors, temporary data storing and data forwarding to fog or cloud nodes), and introduces predefined states for microcontrollers, which allow to map a certain power consumption to a certain state.

Besides our real measurements of a typical use of a microcontroller, we gathered information from the following works. (Maier et al., 2017) and (Kaup et al., 2014) focus on the comparative analysis and the monitoring of ESP32 and Raspberry Pi devices,

[3] The official website of Thingspeak is available at: https://thingspeak.com/

while detailed online benchmark results for their energy consumption can also be found on websites[4] [5]. After studying these sources, we The collected and measured numbers are shown and compared in Table 2. It also shows the predefined values (for min, idle and max) we chose to be the base for our IoT Energy Model. We arrived to these values by counting the median for the concrete values gathered from the research papers, websites and by our measurements.

Our findings and experiments revealed that the power consumption values of microcontrollers are highly dependant on their actual behaviour and their use cases. Typical modifying circumstances may be the usage of wired connection instead of wireless, and/or different types of power supply cable or converter. As we mentioned it earlier, during our experiments we used an online service for retrieving and storing the generated data by the DTH22 sensor. Table 1 also shows that in a few cases (i.e. after every fifth sampling), the consumption doubled in case of ESP32. To handle such extreme cases and to be able to simulate uncertainty, we introduce three different states of a microcontroller in our model (that have been implemented in the *Microcontroller* class of our simulator).

The state *OFF* indicates a fully turned off device with static minimal energy consumption using the min power preset value. The *RUNNING* state represents a high energy consumption state, where the actual power consumption can change dynamically wrt. the actual CPU utilisation. The minimal and maximal consumption values in this state are set by the predefined idle and max power values. To simulate specific events when high power spikes appear (caused by e.g. activating a previously unused port of the device), we introduce the *ACTIVE* state. It also represents a high energy consumption state allowing dynamic changes, but its minimal value should be higher than in the *RUNNING* state; by default it is set to the double of the idle power value.

According to our observation, we experienced such behaviour in 20% of the sampling process, therefore we decided that the *ACTIVE* state will be set during the sensing process of IoT sensors until the data is saved into the local storage of the IoT device. In this way, each simulated IoT device enters the *OFF* state when it is created, the *RUNNING* state, when it is started, and it is periodically switches between the *ACTIVE* (performing sensor data generation) and *RUNNING* states till it is stopped (*OFF* state) or terminated.

We would also note that DISSECT-CF-Fog provides a transparent and easily usable interface to create additional, new states, and hence multiple energy models, and it is up to the researcher, where to use such new states during the simulation.

In the next section we continue with the evaluation of the proposed, extended energy model on two typical and widespread IoT use cases.

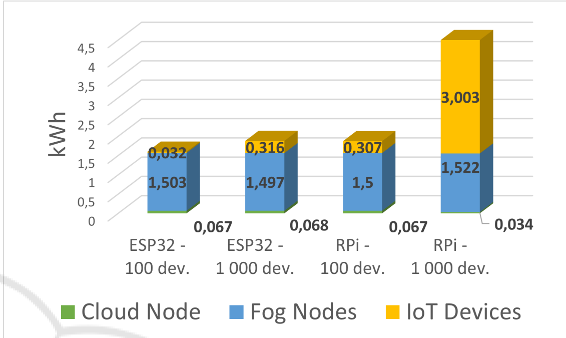

Figure 2: Cumulative energy consumption of cloud, fog nodes and IoT devices.

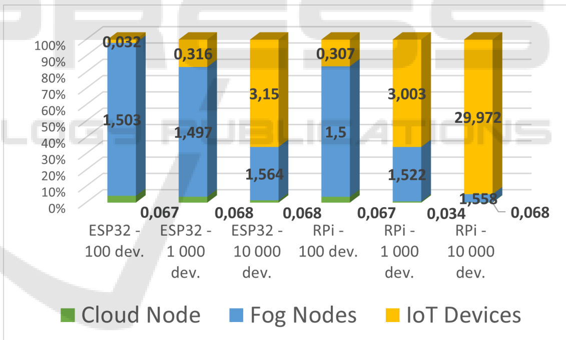

Figure 3: Energy consumption percentage of cloud, fog nodes and IoT devices.

4 EVALUATION

In this section we illustrate the use of the extended, unified energy model for IoT-Fog-Cloud architectures in DISSECT-CF-Fog. For this purpose, we model one of the typical IoT use cases, which represents a weather forecasting scenario with numerous weather stations (run by IoT microcontrollers with special sensors). These devices can communicate with a fog layer directly, which contains three different nodes with the equal amount of resources, utilising 40 CPU cores and 40 GB of memory in total. On the top of the fog topology, there is one cloud datacenter having

[4]Raspberry Pi benchmark values are available at: https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/

[5]ESP32 benckmark values are available at: https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/

Table 3: Comparison of the final results of the simulated scenarios.

MicroController		ESP32			Raspberry Pi		
Number of Devices		100	1 000	10 000	100	1 000	10 000
Number of VMs		7	7	29	7	7	29
Cloud/Fog Cost ($)		0.84	0.85	2.79	0.84	0.84	2.79
IoT Cost ($)		0.009	0.90	83.2	0.009	0.90	83.2
Delay (min.)		1.03	1.25	3.50	1.03	1.25	3.50
Runtime (sec.)		0	4	95	0	4	118
Energy Consumption (kWh)	*cloud1* Consumption	0.067	0.068	0.068	0.067	0.034	0.068
	fog1 Consumption	0.456	0.456	0.456	0.456	0.456	0.456
	fog1 Device Consumption	0.006	0.053	0.525	0.053	0.500	5.000
	fog2 Consumption	0.436	0.431	0.531	0.433	0.456	0.492
	fog2 Device Consumption	0.011	0.106	1.050	0.105	1.003	10.000
	fog3 Consumption	0.611	0.611	0.578	0.611	0.611	0.611
	fog3 Device Consumption	0.016	0.158	1.575	0.150	1.500	14.972
Total Consumption by Nodes		1.570	1.569	1.636	1.569	1.558	1.628
Total Consumption by Devices		0.032	0.316	3.150	0.307	3.003	29.972

56 CPU cores and 40 GB of memory, furthermore, the devices are not allowed to send messages (unprocessed sensor data) directly to the cloud (they are connected only to the fog). We considered two types of virtual machine images simulating existing Amazon Cloud (AWS) instances. The *cloud1* node can utilise VMs with 8 CPU and 4 GB of memory, their hourly prices were set to 0.202$, while the *fog1*, *fog2* and *fog3* nodes can deploy VMs with 4 CPU, 2 GB of memory with 0.101$ hourly price. We also set the IoT side pricing by applying the IBM Cloud pricing schema, which charges the consumer after the amount of data exchanged (in MB).

In our simulation, the microcontrollers can use either ESP32 or Raspberry Pi energy models, and they are equipped with a temperature-humidity sensor (similarly to our real world measurements). In our weather forecasting use case, we defined three different scenarios by scaling up the number of operating devices. In the first case, we utilised 100 IoT devices, then we increased the number of devices to 1000, finally in the last case the maximum device number was 10000, operated for 60 minutes within the experiments. The microcontrollers measured the environmental parameters every 60 seconds, similarly to the real device evaluation (shown in Section 3.1), hence our goal was to map the real monitoring execution in the DISSECT-CF-Fog simulation environment.

The evaluation process is the following in each scenarios: (i) the IoT microcontrollers monitor the environment based on their sampling frequency, (ii) the generated data are forwarded to the less loaded fog node (using the default scheduling algorithm), (iii) a node allocates a task (i.e. collection of 256 Kilobytes of data) to a VM to be processed, or requests a new one, if there is no free VM available, in case the current resource capacity allows it. Otherwise, the unallocated task will be moved to a less loaded node (in the fog or to the cloud layer).

During the evaluation, we modelled a European-wide scenario, where the cloud was located in Frankfurt, whilst the the three fog nodes were positioned in London, Budapest and Vienna. The latency between them was determined based on online ping statistics[6]. The delay between a device and a fog node was set to an average 50 ms weighted with the actual physical distance, and the positions of the devices were randomly generated across Europe.

In order to highlight the energy consumption of the nodes, the number of VMs were scaled up and down dynamically according to the actual load caused by the tasks. To be as realistic as we can, each computational resource dealt with different energy models based on the resource schema of LPDS Cloud of MTA SZTAKI[7]. The exact values we used to set the energy model parameters are summarised in Table 2 and Table 4.

[6] WonderNetwork website is available at: https://wondernetwork.com/pings/

[7] LPDS Cloud of MTA SZTAKI website is available at: https://www.sztaki.hu/en/science/departments/lpds/

Table 4: The chosen values of the energy model for nodes and microcontrollers.

Resource Type	Min Power	Idle Power	Max Power
cloud1	20	398	533
fog1	20	296	493
fog2	20	296	533
fog3	20	398	493

The comparison of the results can be seen in Table 3. We listed the number of VMs utilised by all nodes (*Number of VMs*), and the cost of both the cloud/fog and IoT sides (*Cloud/Fog cost, IoT cost*). The *Delay* value reflects to the makespan of the IoT application, whilst the *Runtime* indicates the elapsed time in the execution environment required by the actual simulation. The *Power Consumption* ensures consumption information detailed for each computational node (e.g. *cloud1* Consumption denotes the consumed energy by the *cloud1* node), and we also counted the summed consumption values of IoT devices related to an actual node (e.g. *fog1* - IoT Device Consumption denotes the total consumed energy by all simulated microcontrollers connected to *fog1*). Lastly, total energy usage of both nodes and devices are presented by *Total Consumption of Nodes* and *Total Consumption of Devices*.

As we can see from Table 3, the cloud resource utilisation is basically the same in all six simulation cases, because they had to deal with around the same amount of unprocessed data/tasks (coming from the fog layer). Nevertheless, it also shows that in case of 1 000 devices, seven VMs could easily handle the scheduled amount of tasks for both microcontrollers. The more data a task contains, the more time it takes for the task to be processed, and additional incoming tasks may trigger new VMs to be deployed (depending on the applied task scheduling policy threshold).

In the third cases having 10 000 devices, the number of VMs is dramatically increased to 29, for both device types.

Since the IBM Cloud pricing is independent of the actual device type, only the transmitted data counts, and the cost of the computational nodes is proportional to the number of utilised VMs, therefore the corresponding costs are the same in case of ESP32 and RPi. It can also be observed that the timeout delay (i.e. application makespan minus the set operation interval of the IoT devices (60 minutes in these scenarios)) is less than 90 seconds for 100 and 10 000 devices. As we can see for the third, 10 000 devices cases, the throughput of the system decreased, hence the delay increased to 3.5 minutes. The execution time (*runtime*) of the simulations for all cases remains within two minutes for all cases, which points out that DISSECT-CF-Fog can manage thousands of entities on a single PC (for the evaluation we used a PC with Windows 10 OS, i5-4460 CPU and 8 GB memory).

Figure 2 and Figure 3 highlight the results by comparing the energy consumption ratio of the utilised cloud node, fog nodes and IoT devices (i.e. microcontrollers). Figure 2 depicts the total energy used in kWh for each categories, while Figure 3 depicts their ratio in percentage. As we can see from the diagrams, the cloud consumption takes only a small part of the total energy consumption in all six scenarios. The fog nodes are mostly capable of handling the vast amount of data with their own resources generated by the IoT layer, and there is no need to involve cloud resources drastically. Nevertheless, when we scale up the number of microcontrollers in the IoT layer, our results show a significant increase in the total energy consumption, caused by only exclusively the operation of the IoT devices. For the case of using 10 000 RPi devices, we can see that the energy consumed by the IoT layer takes up almost 95% of the total consumption, as shown in Figure 3. For smaller scales, we can observe that 100 ESP32 devices caused only 2% of the total energy consumption. This ratio goes up to about 16%, in case of 1 000 ESP32 and 100 RPi devices, and we experienced around the same ratio in case of 10 000 ESP32 and 1 000 RPi devices (with $\tilde{6}6\%$).

5 CONCLUSIONS

In this paper we presented a novel extension of the energy model of the DISSECT-CF-Fog simulator to enable the energy monitoring of its simulated IoT components. In this way, we realised a unified energy model capable of analysing the power consumption of complex, IoT-Fog-Cloud infrastructures. We evaluated the extension and exemplified its utilisation with a weather forecasting use case. Our results showed that detailed energy consumption values can be gathered by the extended DISSECT-CF-Fog, and the proposed solution enables detailed configuration for evaluating various power settings for all system elements.

Our future work will address battery draining simulation of smart devices, and sophisticated energy-aware scheduling algorithm development for task offloading for fog nodes.

The IoT energy model extension of DISSECT-CF-Fog is available online on GitHub:
https://github.com/andrasmarkus/dissect-cf/tree/energy/

ACKNOWLEDGEMENTS

The research leading to these results was supported by the Hungarian Government and the European Regional Development Fund under the grant number GINOP-2.3.2-15-2016-00037 ("Internet of Living Things"), and by the Ministry for Innovation and Technology, Hungary under the grant number NKFIH-1279-2/2020.

REFERENCES

Ahvar, E., Orgerie, A., and Lébre, A. (2019). Estimating energy consumption of cloud, fog and edge computing infrastructures. *IEEE Transactions on Sustainable Computing*, EA:1–12, DOI: 10.1109/TSUSC.2019.2905900.

Atlam, H. F., Walters, R. J., and Wills, G. B. (2018). Fog computing and the internet of things: A review. *Big Data and Cognitive Computing*, 2(2):10, DOI: 10.3390/bdcc202001.

Garg, S. and Buyya, R. (2011). Green cloud computing and environmental sustainability. *Harnessing Green It: Principles and Practices*, 16:315–339, DOI: 10.1002/9781118305393.ch16.

Gupta, H., Dastjerdi, A., Ghosh, S., and Buyya, R. (2017). 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:Pages 1275–1296, DOI: 10.1002/spe.2509.

Jha, D. N., Alwasel, K., Alshoshan, A., Huang, X., Naha, R., Battula, S., Garg, S., Puthal, D., James, P., Zomaya, A., Dustdar, S., and Ranjan, R. (2020). Iotsim-edge: A simulation framework for modeling the behavior of internet of things and edge computing environments. *Software: Practice and Experience*, 50:844–867, DOI: 10.1002/spe.2787.

Kaup, F., Gottschling, P., and Hausheer, D. (2014). Powerpi: Measuring and modeling the power consumption of the raspberry pi. *Annual IEEE Conference on Local Computer Networks*, pages 236–243, DOI: 10.1109/LCN.2014.6925777.

Kecskemeti, G. (2015). Dissect-cf: A simulator to foster energy-aware scheduling in infrastructure clouds. *Simulation Modelling Practice and Theory*, 58:188–218, DOI: 10.1016/j.simpat.2015.05.009.

Lera, I., Guerrero, C., and Juiz, C. (2019). Yafs: A simulator for iot scenarios in fog computing. *IEEE Access*, 7:91745–91758, DOI: DOI: 10.1109/ACCESS.2019.2927895.

Mahmud, R., Kotagiri, R., and Buyya, R. (2018). Fog computing: A taxonomy, survey and future directions. *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*, pages 103–130, DOI: 10.1007/978–981–10–5861–5_5.

Maier, A., Sharp, A., and Vagapov, Y. (2017). Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things. *International Conference on Internet Technologies and Applications*, pages 143–148, DOI: 10.1109/ITECHA.2017.8101926.

Markus, A., Gacsi, P., and Kertesz, A. (2020). Develop or dissipate fogs? evaluating an iot application in fog and cloud simulations. *International Conference on Cloud Computing and Services Science*, pages 193–203, DOI: 10.5220/0009590401930203.

Markus, A. and Kertesz, A. (2020). A survey and taxonomy of simulation environments modelling fog computing. *Simulation Modelling Practice and Theory*, 101:102042, DOI: 10.1016/j.simpat.2019.102042.

Motlagh, N. H., Mohammadrezaei, M., Hunt, J., and Zakeri, B. (2020). Internet of things (iot) and the energy sector. *Energies*, 13(2):494, DOI: 10.3390/en13020494.

Oma, R., Nakamura, S., Duolikun, D., Enokido, T., and Takizawa, M. (2018). An energy-efficient model for fog computing in the internet of things (iot). *Internet of Things*, 1-2:14–26, DOI: 10.1016/j.iot.2018.08.003.

Perez Abreu, D., Velasquez, K., Curado, M., and Monteiro, E. (2020). A comparative analysis of simulators for the cloud to fog continuum. *Simulation Modelling Practice and Theory*, 101:102029, DOI: 10.1016/j.simpat.2019.102029.

Qayyum, T., Malik, A., Khan, M., Khalid, O., and Khan, S. (2018). Fognetsim++: A toolkit for modeling and simulation of distributed fog environment. *IEEE Access*, 6:63570–63583, DOI: 10.1109/ACCESS.2018.2877696.

Samie, F., Bauer, L., and Henkel, J. (2016). Iot technologies for embedded computing: A survey. *International Conference on Hardware/Software Codesign and System Synthesis*, pages 1–10, DOI: 10.1145/2968456.2974004.

Sonmez, C., Ozgovde, A., and Ersoy, C. (2018). Edgecloudsim: An environment for performance evaluation of edge computing systems. *Transactions on Emerging Telecommunications Technologies*, 29, DOI: 10.1002/ett.3493.

Sun, H., Yu, H., Fan, G., and Chen, L. (2020). Energy and time efficient task offloading and resource allocation on the generic iot-fog-cloud architecture. *Peer-to-Peer Networking and Applications*, 13:548–563, DOI: 10.1007/s12083–019–00783–7.