A Novel Distributed Fog-based Networked Architecture to Preserve Energy in Fog Data Centers

Zahra Pooranian[1], Mohammad Shojafar[2], Paola G. Vinueza Naranjo[3], Luca Chiaraviglio[2,4], Mauro Conti[1]
[1]University of Padua, Padua, Italy
Email: {zahra, conti}@math.unipd.it
[2]Intel Software Innovator, CNIT, Rome, Italy
Email: {mohammad.shojafar, luca.chiaraviglio}@cnit.it
[3]DIET, Sapienza University of Rome, Rome, Italy
Email: paola.vinueza@uniroma1.it
[4]Department of Electronic Engineering, Tor Vergata University of Rome , Rome, Italy

***Abstract*—The distinguishing feature of the Fog Computing (FC) paradigm is that FC spreads communication and computing resources over the wireless access network, so as to provide resource augmentation to resource and energy-limited wireless (possibly mobile) devices. Since FC would lead to substantial reductions in energy consumption and access latency, it will play a key role in the realization of the Fog of Everything (FoE) paradigm. The core challenge of the resulting FoE paradigm is to materialize the seamless convergence of three distinct disciplines, namely, broadband mobile communication, cloud computing, and Internet of Everything (IoE). In this paper, we present a new IoE architecture for FC in order to implement the resulting FoE technological platform. Then, we elaborate the related Quality of Service (QoS) requirements to be satisfied by the underlying FoE technological platform. Furthermore, in order to corroborate the conclusion that advancements in the envisioned architecture description, we present: (i) the proposed energy-aware algorithm adopt Fog data center; and, (ii) the obtained numerical performance, for a real-world case study that shows that our approach saves energy consumption impressively in the Fog data Center compared with the existing methods and could be of practical interest in the incoming Fog of Everything (FoE) realm.**

***Index Terms*—Fog computing (FC), Internet of Everything (IoE), Fog of Everything (FoE), energy-efficiency, networked computing platform, distributed resource management.**

I. Introduction

Fog computing (FC) is a new computing paradigm that extends Cloud Computing (CC) and services to the edge of the communication network [1]. FC involves various types of applications that run both in the Cloud and in devices, especially billions of devices which are interconnected using the Internet of things (IoT) [1]. IoT-based integrated managing system are able to handle data and energy in complex systems such as electrical grid networks (grid). Such systems will enable us to utilize our energy assets better, by balancing loads and network demands in grids efficiently, and by handling transacting multiple directions, not only for producers of energy but also for consumers of energy. Fog Nodes (FNs) are the architecture that provides resources for services at the edge of the network. Hence, the network of such devices is called *Fog Computing Platform* (FCP). The application of FCP over grids has influenced in several directions that have not been taken into account before in energy-aware systems and it leads these systems to become more adaptive, by dealing with intermittent power and several others challenges. The joint manager of Fog and IoT paradigms will reduce energy consumptions and operating costs of state-of-the-art Fog-based data centers (FDCs). Several energy- /cost-aware works in the literature target fog-supported mobile devices [2], [3], jointly cloud component-mobile user QoSs [4], [5], emerging IoE in FC architectures such as CAPEX/OPEX [6], cost-aware[19] and bandwidth and delay tolerant services [7]. In contrast to them, our work provides a holistic fog-supported architecture that jointly manages the computational and networking cost considering variable network traffic costs. The main features of the proposed work are given as follows: (i) We introduce a novel architecture to perform real-time IoT communications over FCP; (ii) We detail FDC over IoE-based environment and its features; (iii) We propose a mathematical problem to minimize the total energy consumption that includes computing and communication cost of on FDC; (iv) We design a heuristic algorithm to preserve load balancing and resource managements over FDC on a real-time case study; and, (v) We evaluate the proposed algorithm through extensive simulations based on real-world use cases and traces.

The remainder of this paper is organized as follows. Section II describes the reference scenario for our proposal, while Section III describes our model for solving the VEs allocation problem. Section IV describes the experimental results used to validate our model. Finally, Section V concludes the paper.

II. The considered Fog-based networked computing platform

Fig. 1 reports a scheme of the proposed architecture. More in depth, we combine distributed edge located processing and control applications with intelligent analytics and real-time secure connectivity framework (i.e., FC). Indeed, our architecture enables machine-to-machine, machine to control center (i.e., FDC), and machine to Cloud data connectivity connections. The framework will run in real world power applications and it will interface with the operational equipment of the Smart Grid (SG) core. A critical architecture is a high-speed field data bus that connects devices and FNs. The data bus also interacts with the central station (i.e., the

2155-6814/17 $31.00 © 2017 IEEE
DOI 10.1109/MASS.2017.33

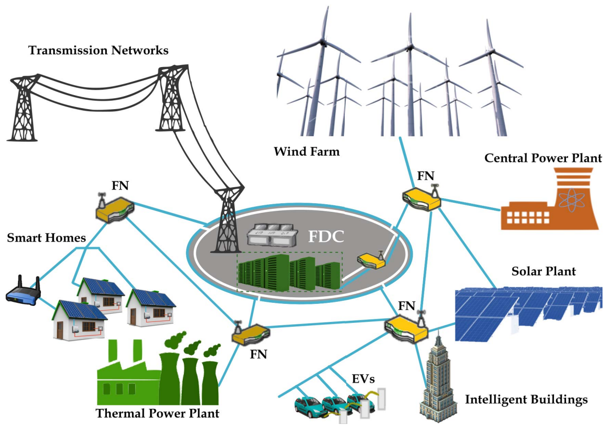


Fig. 1: IoE architecture on Fog Computing, EVs:=Electric Vehicles, FN:=Fog Node.

FDC, oval-shape dark component middle of Fig. 1), that uses an Advanced Distribution Management System (ADMS) to ensure that the SG optimizes circuit flow-voltages (see Fig. 1).

The IoE/SG architecture is represented by wind patterns (the wind farm that provides green power to the grid, most upper part of Fig. 1), a flexible AC transmission system (that connects remotely located renewable generation, increases transmission grid capacity, and helps to stabilize the grid), a solar plant (the solar panels help power to the grid, helping the utility meet renewable portfolio standards, middle right part of Fig. 1), a control power plant, etc. This model involves SGs and intelligent buildings through smart metering. In fact, the energy storage helps to address the variability of renewable generation and can reduce peak demand. The solar panels can charge energy-storage units attached to the vehicles chargers and can recharge before heading back out, or overnight when the energy costs are low. The SG architecture manages the generation parts by remote monitoring and controlling subsystems, that are considered as FNs (see cyan components over Fig. 1).

An FDC is dedicated to supervising the transmission, distribution and communication networks. All these require to precisely manage grid architecture from generation points to consumption ones by using the communication of measured values and transmitted control information more accurately. SG involves bidirectional data communications, e.g., distributed generation, interaction (exchanger's information) collected by many sensors (IoE) of interactive power electronics [8]. Besides, the IoE applications (real-time requirements, as stream processing) are distributed across different geographical locations, with numerous devices with heterogeneous capabilities, able to sense the environment and emit data via gateways (FNs) for further processing and filtering. In addition, the FNs (gateways) are hosting application modules that connect sensors to the Internet. FNs include Cloud resources that are provisioned on-demand from geographically distributed FDC. The data that travels from source (sensors) towards applications deployed in Cloud servers could pass through many devices. Therefore, it is important to take advantage of computational and storage capabilities of those intermediate devices, meet application level QoS requirements, and end-to-end latencies while minimizing resource and energy wastage to ensure QoS, to avoid energy wastage, and to ban resource fragmentation that happens in the integral parts of IoE systems. The Fog devices are discoverable, generic, stateless servers located in the single-hop close proximity of mobile devices, and can operate in a disconnected mode that is VM based, in order to promote flexibility, mobility, scalability and elasticity.

A. Energy Storage Device on IoE/SG architecture

Energy Storage Device (ESD) is the significant part on the Internet of Energy because it improves the quality-of-power (QoP), the quality of the generator output voltage, it increases the reliability of distributed generators, grid run-time and solves the voltage drop, surge and transient power outages by keeping the stability of the system.

As shown in Fig. 1, the architecture is composed of various components. Indeed, the sensors (or ESDs, small blue components on each FN) have capabilities to cover Internet of Energy. This FN-based energy coverage helps the customers to maximize the cost of resource efficiency, minimize the environmental effects, raise correlated performance and borrow the Fog advantages that are applied locally and remotely on the real-time data passing through the network edge by using a data distribution service protocol. Hence, ESDs provide power to users when the distributed power generation devices are not exploited, e.g., solar power at night, wind power on days without the wind. ESD plays a crucial role in the transition and supplies continuous power to the user. Moreover, ESD improves the economic efficiency of distributed power generation unit by selling electricity to the power company in order to maximize economic benefits [4].

B. Fog networking and inter-Fog communication

The Fog devices (FNs), including switches, routers, embedded servers, are naturally nearer to things that generate and process IoE data. The function of the FN is to manage information and energy, perform customization, and meet the needs of the system operations. Also, an FN is able to analyze top time-sensitive data nearby things by producing data. It also ensures a realistic flow of energy to store the right amount of energy and data that flow to the load properly. FNs monitor the QoS and quality-of-energy (QoE) flow in real-time and adjust safely the energy flow.

In an SG distribution network, the verification, protection of the control loops, as well as ensuring that they are operating properly, is the top time-sensitive activity. Therefore, the FNs near the grid sensors can monitor filters and then advise them by sending control commands to actuators. Typically, each substation in an SG may possibly be equipped with a node that aggregates and reports the operational status of each downstream feeder and lateral.

III. THE FOG DATA CENTER: PROPOSED ARCHITECTURE AND PERFORMANCE EVALUATION

In order to specify the structure of the FDC appliance over the IoE-based environment we go in depth on the FC architecture (Fig. 1) that involves organized FNs throughout the SG network between sensors and the Cloud at the core of the network (or Fog Data Center). Fig. 2 reports the main scheme of the considered FDC. Specifically, the FDC provides a platform for filtering and analyzing the data generated by sensors utilizing resources of FNs. Fog environment entails the deployment of a vast number of FNs and the correlation of this network with the FDC. Therefore, the FDC has a huge amount of computations and. Also, FDC is distributed and may be more energy-efficient than the centralized Cloud model of computation, so, being the reduction of energy consumption on FDC an important challenge. Moreover, this drastically reduces the traffic sent to the Cloud by allowing the placement of filtering operators close to the sources of data. The FDC, as a vital component of the IoE environment, is capable of filtering and processing a considerable amount of incoming data on edge devices, by making the data processing architecture distributed and thereby scalable. Hence, it is an important task to give a neat simulated scenario or case study, in order to detail the analytic structure of the FDC and the traffic injected to the engaged servers to make the presented model more efficient and interesting.

In our case study, we define FC as a distributed computing paradigm that extends the services provided by the Cloud to the edge of the network. It enables the seamless leveraging of Cloud and edge resources along with its own architecture (see Fig. 2). FC facilities the management and programming of computing, networking and storage services between FDCs and end devices. FC also supports mobility, resources and interface heterogeneity. Finally, we evaluate the resulting energy-efficient schedule that jointly performs the minimum-energy dispatching of the admitted traffic and the consolidation of VMs hosted by Fog-supported servers [9].

A. FDC design

In an FDC, each processing unit executes the currently assigned task by self-managing own local virtualized storage/computing resources. When a request for a new job (i.e., it is transferred through data network from heterogeneous components, e.g., solar plants, intelligent buildings or smart homes) is submitted from the remote clients (see the most upright cloud-shape component in Fig. 2) and transferred through the Internet to the FNs and FDC, the resource controller dynamically performs both admission control and allocation of the available virtual resources [10]. According to the typical architecture recently presented in [4], Fig. 2 reports the main blocks pf the considered FDC, namely: *i)* an Access Control Server and Router (ACSRs or Adaptive load dispatcher); *ii)* a reconfigurable computing Cloud managed by the Virtual Machine Manager (VMM), called *Dynamic load balancer*, *iii)* the related Switched Virtual LAN, and *iv)* an adaptive controller that dynamically manages all the available

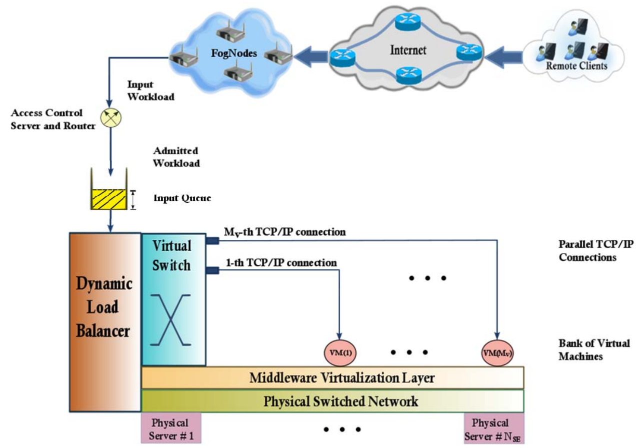


Fig. 2: The envisioned FDC virtualized architecture: It operates at the Middleware layer of the corresponding protocol stack.

computing-communication resources and also performs the admission control of the input/output traffic flows out to the ACSRs and reaches the processed information to the FNs.

One of the major problems of IT is the uncontrolled proliferation of physical servers, which causes a quick increasing of architecture and management costs. For example, adding and removing a service or application involves the (de)installation of a new server over the existing architecture. In order to cope with this issue, we need to power on/off some VMs to satisfy the FDC demands in each time period and respect the server limitations. FDC consolidation is a popular strategy to further reduce the energy consumption by powering off the underutilized VMs and grouping them onto the smallest number of physical servers. The effectiveness of FDC consolidation in driving costs out of IT is shown by the popularity of this strategy. The IT organizations apply consolidation to minimize their assets through an efficient technology utilization. The recent consolidation technologies employed in FDCs [11], [4], [2] encompass server and storage virtualization, as well as deploying tools for process automation. In this case study, we use the server virtualization as a dynamic control to improve energy efficiency in FDC. In this scenario, remote clients exploit Internet core connections for submitting their workload to the serving FC and proposing a power-efficient distributed resource scheduling technique, including allocation and consolidation in order to select the minimum number of servers and energy consumption. The FDC is virtualized: clients workloads are submitted in form of demands for VM processing/storage quanta, and reliable inter-VM communication is attained through end-to-end virtualized TCP/IP connections.

In the FDC of Fig. 2, time is slotted: T^{tot} (s) is the slot duration, t is the discrete-time slot index and the t-th slot spans the semi-open time interval $[tT^{tot}, (t+1)T^{tot}), t \geq 0$. In sake of keeping the generality, we define the main characteristics and components engaged in this model as follows: *i)* $\{\mathcal{S}(s),\ s = 1, \ldots, \mathcal{S}\}$ is the set of physical servers in the FDC with $\mathcal{S} \geq 1$; *ii)* $\{\mathcal{VM}(v),\ v = 1, \ldots, \mathcal{VM}_{max}\}$ is the set of VMs on the s-th server with maximum number $\mathcal{VM}_{max}$ of VMs in the FDC; *iii)* $\mathcal{VM}(s;v) \triangleq v$-th VM hosted by the s-th server, $1 \leq v \leq \mathcal{VM}_{max}(s)$, $1 \leq s \leq \mathcal{S}$; *iv)* $\mathcal{M}_{s,v} \triangleq$ total number of VMs in the FDC $\mathcal{S} \times \mathcal{VM}_{max}$ is the total number of the VMs which may be hosted by the FDC; *v)* $\mathcal{M}(t) \triangleq$ set of the VMs which are turned-ON at slot t, and, *vi)* $\overline{\mathcal{M}}(t) \triangleq$ set of the VMs which are turned-OFF at slot t, formally, $\overline{\mathcal{M}}(t) \equiv \{(s;v);\ 1 \leq s \leq \mathcal{S};\ 1 \leq v \leq \mathcal{VM}_{Max}(s)\}/\mathcal{M}(t)$. Moreover, we can define the following FDC parameters: *i)* $f_{s,v}(t) \triangleq$ processing rate of $\mathcal{VM}(s;v)$ at slot t; *ii)* $\mathcal{L}_{s,v}(t) \triangleq$ workload processed by $\mathcal{VM}(s;v)$ at slot t; *iii)* $f_{s,v}^{max} \triangleq$ maximum processing rate of $\mathcal{VM}(s;v)$ at slot t; *iv)* $\mathcal{L}_{s,v}^{max} \triangleq \tau f_{s,v}^{max} \triangleq$ maximum workload processed by $\mathcal{VM}(s;v)$ during each time slot t.

Modeling the FDC computing energy consumption: Let $\mathcal{E}_{c,v}^{idle}(s)$ and $\mathcal{E}_{c,v}^{max}(s)$, $1 \leq s \leq \mathcal{S}$, be the energies wasted in the idle (inactive) state and the maximum energy of the v-th VM over the s-th physical server, respectively. Indeed, $\mathcal{E}_{c,v}^{idle}(s) \leq \mathcal{E}_{c,v}^{max}(s)$. According to [12], the computing energy $\mathcal{E}_{s,v}^{cpu}(t)$ of the v-th VM over the s-th physical server at the t-th slot is defined as

$$\mathcal{E}_{s,v}^{cpu}(t) = \mathcal{E}_{c,v}^{idle}(t) + \left(\frac{f_{s,v}(t)}{f_{s,v}^{max}}\right)^2 \left(\mathcal{E}_{c,v}^{max}(t) - \mathcal{E}_{c,v}^{idle}(t)\right)\ (J), \quad (1)$$

where $f_{s,v}(t)$ is the optimum processing rate of $\mathcal{VM}(s;v)$ at t.

Modeling the FDC communication energy consumption: In FDC, we resort the TCP New Reno protocol under Fast/Giga Ethernet [5] to model the managed end-to-end intra-FDC transport connections. Under the congestion avoidance state, the communication energy drained by each server connection for time t may be evaluated as in

$$\mathcal{E}_{s,v}^{com}(t) = \mathcal{E}_{s,v}^{com}\left(\mathcal{L}_{s,v}\right)\ (t) = \Omega_{s,v}^{c}\left(\mathcal{L}_{s,v}\right)^{\gamma}(t) \quad (J), \quad (2)$$

where γ-powered model for the per-transport layer connections energy holds for energy instant t, Ω-common energy consumptions of LAN (TCP New Reno protocol) measured in $(J/(IU)^{\gamma})$ and $w \geq 2$ makes strictly convex in $f_{s,v}(t)$, respectively [11], [4].

Hence, the resulting model for the overall wasted computing energy simplifies to the following formula:

$$\mathcal{E}_{s,v}^{tot}(t) \triangleq \mathcal{E}_{s,v}^{cpu}(t) + \mathcal{E}_{s,v}^{com}(t) \quad (J). \quad (3)$$

In order to calculate the overall energy consumption of the whole FDC in t-th time slot, we define $\mathcal{E}^{tot}(t) \triangleq \sum_{s=1}^{\mathcal{S}} \sum_{v=1}^{\mathcal{VM}_{max}(s)} \mathcal{E}_{s,v}^{tot}(t)$.

The FDC energy minimization: The FDC energy minimization problem is presented as follows:

$$\min \mathcal{E}^{tot}(t), \quad (4.1)$$

subject to:

$$\mathcal{L}_{tot}(t) - \left(\sum_{s=1}^{\mathcal{S}} \sum_{v=1}^{\mathcal{VM}_{max}} \mathcal{L}_{s,v}(t)\right) = 0, \quad (4.2)$$

$$\mathcal{L}_{s,v}(t) - \tau f_{s,v}^{max} \leq 0, \qquad \forall s, v \in \mathcal{VM}(s;v), \quad (4.3)$$

$$f_{s,v}(t) - f_{s,v}^{max} \leq 0, \qquad \forall s, v \in \mathcal{VM}(s;v), \quad (4.4)$$

$$f_{s,v}(t) + \mathcal{L}_{s,v}(t) - \tau f_{s,v}(t) \leq 0, \forall s, v \in \mathcal{VM}(s;v). \quad (4.5)$$

where the constraints in (4.2) and (4.3) guarantee that the overall input workload $\mathcal{L}_{tot}(t)$ offered to the FDC at slot t is partitioned over the available VMs in a feasible manner. Eq. (4.4) and Eq. (4.5) limit the processing rate of $\mathcal{VM}(s;v)$ of Fig. 2 and emphasize that it should not be more than the maximum frequency $f_{s,v}^{max}$.

Solution: In order to solve this problem, it is essential to propose mathematically (see [4]) or heuristics methods (see [13], [9]) to find the optimal allocation of frequency and workload for each VM over each server, in each time slot t. In the following, we present a simple algorithm to solve this problem. We consider the problem as a *bin packing penalty-aware* problem. Indeed, in turn, for each incoming workload $\mathcal{L}_T(t)$, we calculate how many VMs can be allocated for the t-th slot and, after that, we consider servers as bins and VMs as packs that must be served distributively-based on their time limitations and frequency limitations. This process continues in some iterations. For each iteration, we give penalty or reward for each server, based on their energy characteristics (idle energy, maximum energy, maximum frequency). The penalty and reward policy is considered in order to decrease the fatality cases (exponentially increasing the energy consumptions). The penalty means the server will be punished and banned to be used again for some iterations. When the freezing iterations for the server passed, it will be back added to the list of available servers for the remaining iterations. Note that the server s can serve VMs when it does not pass the maximum number $\mathcal{VM}_{max}(s)$ of VMs. Otherwise, the server s is put out from the list of processing servers. The maximum interactions are defined by $\mathcal{M}_{sv}$. In a nutshell, in each iteration, we are looking for the best servers to be allocated. This process is done in each time slot t until all the incoming workloads will be served. At last, we calculate the energy consumption components of the optimization problem for each slot.

IV. Performance Evaluation

In this section, we explain the simulation setup and present the results. The proposed algorithm is evaluated thoroughly in a simulated Fog model and results are compared with two heuristic algorithms: MBFD (Modified Best Fit Decreasing) [14] and MDC (Maximum Density Consolidation) [15]. The experimental results show significant improvements in all performance measures, which include resources, the number of select servers, energy consumption, and response time on different scales of FNs.

A. Simulation Setup

In this test, the Fog devices are assumed to be placed in an Intranet that is bounded in a geographical area. Hence, the devices closer to each FN would have rapid access to the services deployed on Fog devices. As we know, Fog device could store data temporarily for the processing. Clearly, when the results are obtained, the temporary data is deleted. We create CDC and FDC with different characteristics as is presented in Table I. We consider a physical topology with

TABLE I: Main parameters of the simulated Fog devices.

Device Type	CPU (GHz)	RAM (GB)	Latency (ms)
Cloud VM	2.67	4	100
Fog VM (FDC)	2.67	4	50
FNs (WiFi Gateway)	2.67	4	30

TABLE II: Energy *(J)* profile of the simulated FDCs at various computing frequencies *(GHz)* and workloads *(MIPS)*.

	Frequencies				
FNs (GHz)	**1.60**	**1.867**	**2.133**	**2.40**	**2.67**
$\% f_{s,v}^{max}$	59.93	69.93	79.89	89.89	100
$\mathcal{E}_{s,v}^{idle}$	119.86	139.86	159.78	179.78	200.00
$\mathcal{E}_{s,v}^{max}$	239.72	279.72	319.56	259.56	400.00

four Fog devices (FNs) with different configurations modeled in iFogSim [13] via FN and PhysicalTopology (see Fig. 1). Attaining energy efficiency is a central target in the Fog paradigm [16]. Moreover, we assume that each FN supports smart metering devices, which are deployed by energy suppliers to analyze power consumption at the home level (over the FDC component of the IoE-based framework). We can easily exploit applications' APIs and utilize the programming to heavily manage the incoming data and power data. Indeed, Table I illustrates the configurations of the different types of Fog devices used in Fig. 1. This requires a low latency communication between the FDC and the set of FNs control strategies. FNs handle a large amount of data and an intensive computational analysis without burdening the network into a state of congestion. In order to model the CPU rate of the FNs (or Fog devices) we use the DVFS technique for each CPU of the FN as an Intel® Core™ 2 CPU Q6700 with 2.67 GHz frequency rate with 4 GB of RAM memory, running Ubuntu 10.4 LTS (Linux kernel 2.6.32). The power values given by the FNs at $(0\%-100\%)$ of CPU utilization, called $\mathcal{E}_{s,v}^{idle}$ and $\mathcal{E}_{s,v}^{max}$ by each frequency, are measured and shown in Table II.

B. Results

We tested energy performance of the proposed scheduler under a set of real-world web-based trace: the data extracted from World Cup 98 [17]. Results are then compared with two heuristic algorithms: the modified best fit decreasing algorithm (MBFD) [14] and the Maximum Density Consolidation (MDC) [15] with several test cases.

1) Tested energy performance of the proposed heuristic: In the first test scenario, we run the proposed scheduler and evaluate the resulting average total consumed energy $\overline{\mathcal{E}}_{tot}$ for each FN (i.e., we define 10 FLs for each FN and $\#VMs >= 5$ is increasing for each FN) under World Cup 98 traces [17] for various communication costs (Fast: $\Omega_{s,v}^{c} = 1.8 \times 10^{-3}$; Giga: $\Omega_{s,v}^{c} = 2.5 \times 10^{-}2$; Tera: $\Omega_{s,v}^{c} = 1.1 \times 10^{-}2$ [11]) (see Fig. 3). Note that, these FNs can be integrated or distributed all over the presented framework of Fig. 1. This examination points out that: (i) $\overline{\mathcal{E}}^{tot}$ increases for increasing values of $\Omega_{s,v}^{c}$ (the network energy cost) and (ii) $\overline{\mathcal{E}}^{tot}$ grows for increasing values of VMs and this rate is different for each Ethernet cases.

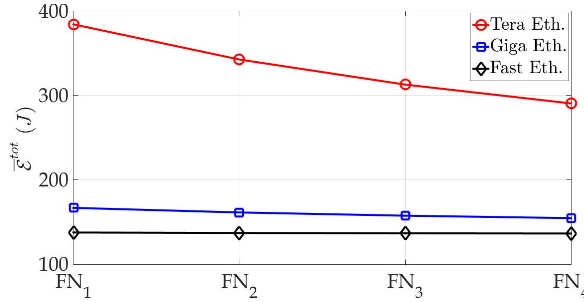


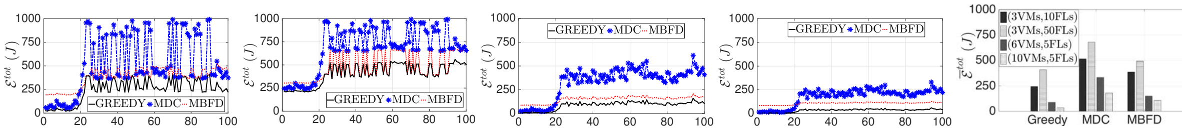
Fig. 3: FNs average total energy consumptions $\overline{\mathcal{E}}^{tot}$ (J) under World Cup 98 traces [17].

2) Performance comparisons: In the second scenario, we compare the energy performance of the proposed solution (Greedy Scheduler, GS) against the MBFD in [14] and MDC in [15]. The main aim of our numerical tests is to compare the reductions in the overall average energy consumption of GS with various network shapes. In order to evaluate the energy reduction induced by scaling up/down the intra-Fog processing and communication rates of GS, we present the instantaneous total energy $\mathcal{E}^{tot}$ and energy saving $\mathcal{E}_{sav}^{tot}$ for 100 slots in four use-cases (two fixed VMs and two fixed FLs) in Figs. 4 and 5, respectively. According to Figs. 4 and 5, they report the instantaneous energy consumptions and savings of the Greedy solution for the submitted real workload: it is representative of a 1-h HTTP-type session arrival process actually measured at the Web servers of the 1998 Soccer World Cup site (see [17] and referenced therein), for various VMs and FLs, for the size $\mathcal{L}_{tot}$ of the job submitted at the beginning of each t-slot. Furthermore, it is worthwhile to note that the maximum average energy (consumptions, savings) of Greedy algorithm are of about (81.55%, 63.9%) and (33.8%, 47.8%) (see the Fig. 4(e) and 5(e)) less than the MDC and MBFD, respectively.

In the last simulation, we aim at evaluating the average energy consumption $\overline{\mathcal{E}}^{tot}$ of our solution compared with MBFD [14] and MDC [15] for different numbers of VMs and FLs. Fig. 6 reports the obtained energy consumptions averaged over the corresponding average numbers of actually turned-ON VMs. Interestingly, since $\mathcal{E}^{tot}(t)$ in (4.1) is, by definition, the minimum requested energy when up to $\mathcal{VM}_{Max}$ VMs may be instantiated, at fixed $\mathcal{E}_{s,v}^{com}$ by Fast and Giga Ethernets, $\overline{\mathcal{E}}^{tot}$ of Fig. 6 decreases for increasing $\mathcal{VM}_{Max}$ and, then, approaches to a minimum value that does not vary when $\mathcal{VM}_{Max}$ is further increased (see the semi-flat segments of the two lowermost plots of Fig. 6). Furthermore, the two uppermost plots of Fig. 6 point out that the optimal value $\mathcal{VM}_{Max}$ needed for driving $\overline{\mathcal{E}}^{tot}$ to the minimum decreases. However, as it could be expected, the lowermost curve of Fig. 6 confirms that, when no communication costs are present, $\mathcal{E}^{tot}$ strictly decreases for increasing $\mathcal{VM}_{Max}$ and vanishes at large $\mathcal{VM}_{Max}$.

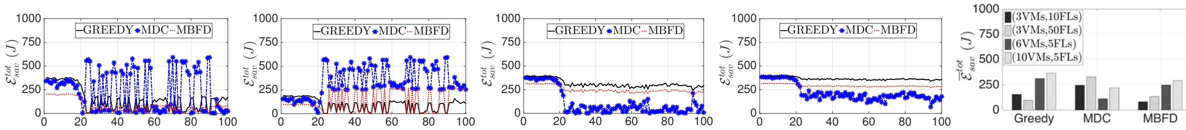
V. Conclusions

This paper explored the Fog platform and its characteristics over IoT-based architecture FDC as a challenging component,



(a) (3VMs, 10FLs). (b) (3VMs, 50FLs). (c) (6VMs, 5FLs). (d) (10VMs, 5FLs). (e) Average comparisons.

Fig. 4: $\mathcal{E}^{tot}$ (J): Total energy consumptions between Greedy, MBFD [14] and MDC [15] at various FNs and FLs.



(a) (3VMs, 10FLs). (b) (3VMs, 50FLs). (c) (6VMs, 5FLs). (d) (10VMs, 5FLs). (e) Average comparisons.

Fig. 5: $\mathcal{E}_{sav}^{tot}$ (J): Total energy saving between Greedy, MBFD [14] and MDC [15] at various FNs and FLs.

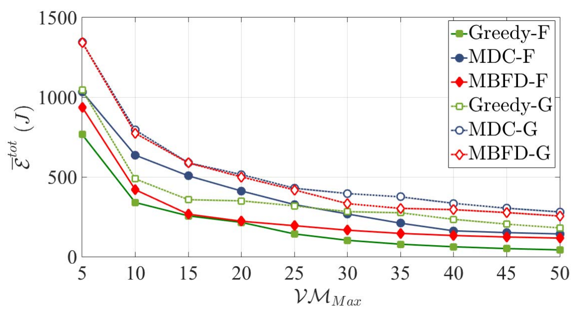


Fig. 6: $\overline{\mathcal{E}}^{tot}$ with various cases using Worldcup98 workload [17] on Fast Ethernet (–F), Giga Ethernet (–G) compared with MBFD [14] and MDC [15].

whose task is to preserve energy by using several modeling techniques, as well as by responding the demands due to the SLA and QoS. Indeed, we propose a distributed Fog-supported IoE-based framework and extend their engaged components that aim at focusing on SG applications. We use an FDC case study and propose a mathematical model for the FDC, concentrated on the computing and communication of the FDC components (i.e., servers and VMM) distributed, and we solve the problem by using some greedy techniques. Finally, we point out that, since FoE is a result of two emerging paradigms, e.g., Fog computing and Internet of Everything, it is in the infancy and, then, it is continuously evolving.

Acknowledgement

Mauro Conti is supported by a Marie Curie Fellowship funded by the European Commission (agreement PCIG11-GA-2012-321980). This work is also partially supported by the EU TagItSmart! Project (agreement H2020-ICT30-2015-688061), the EU-India REACH Project (agreement ICI+/2014/342-896), and by the projects "Physical-Layer Security for Wireless Communication", and "Content Centric Networking: Security and Privacy Issues" funded by the University of Padua. This work is partially supported by the grant n. 2017-166478 (3696) from Cisco University Research Program Fund and Silicon Valley Community Foundation.

References

[1] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014, pp. 169–186.

[2] H. Dou, Y. Qi, W. Wei, and H. Song, "Carbon-aware electricity cost minimization for sustainable data centers," *IEEE Transactions on Sustainable Computing*, 2017.

[3] Z. Su, F. Biennier, Z. Lv, Y. Peng, H. Song, and J. Miao, "Toward architectural and protocol-level foundation for end-to-end trustworthiness in cloud/fog computing," *IEEE Transactions on Big Data*, 2017.

[4] M. Shojafar, N. Cordeschi, and E. Baccarelli, "Energy-efficient adaptive resource management for real-time vehicular cloud services," *IEEE Transactions on Cloud Computing*, vol. PP, pp. 1–14, 2016.

[5] O. Tamm, C. Hermsmeyer, and A. M. Rush, "Eco-sustainable system and network architectures for future transport networks," *Bell Labs Technical Journal*, vol. 14, no. 4, pp. 311–327, 2010.

[6] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, and X. Yao, "Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things," *IEEE Internet of Things Journal*, 2017.

[7] M. Shojafar, Z. Pooranian, P. G. V. Naranjo, and E. Baccarelli, "Flaps: bandwidth and delay-efficient distributed data searching in fog-supported p2p content delivery networks," *The Journal of Supercomputing*, 2017.

[8] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid-the new and improved power grid: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 14, no. 4, pp. 944–980, 2012.

[9] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog computing: Principals, architectures, and applications," *arXiv preprint arXiv:1601.02752*, 2016.

[10] T. Lu, M. Chen, and L. L. Andrew, "Simple and effective dynamic provisioning for power-proportional data centers," *IEEE Trans. on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1161–1171, 2013.

[11] E. Baccarelli, P. G. V. Naranjo, M. Scarpiniti, M. Shojafar, and J. H. Abawajy, "Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study," *IEEE Access*, 2017.

[12] M. Portnoy, *Virtualization essentials*. John Wiley & Sons, 2012.

[13] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments," *Software: Practice and Experience (SPE)*, 2017.

[14] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755–768, 2012.

[15] G. Lovász, F. Niedermeier, and H. de Meer, "Performance tradeoffs of energy-aware virtual machine consolidation," *Cluster Computing*, vol. 16, no. 3, pp. 481–496, 2013.

[16] R. Suryawanshi and G. Mandlik, "Focusing on mobile users at the edge of internet of things using fog computing," *International Journal of Scientific Engineering and Technology Research*, vol. 4, no. 17, pp. 3225–3231, 2015.

[17] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi, "Analytic modeling of multitier internet applications," *ACM Transactions on the Web (TWEB)*, vol. 1, no. 1, p. 2, 2007.