Peer-reviewed version available at Appl. Sci. 2017, 7, 1325; doi:10.3390/app7121325

Article

Fog over Virtualized IoT: New Opportunity for Context-Aware Networked Applications and a Case Study

Paola G.V. Naranjo [1], Zahra Pooranian [2], Shahaboddin Shamshirband [3,4,*], Jemal H. Abawajy [5] and Mauro Conti [2]

[1] Department of Information Engineering, Electronics, and Telecommunication, Sapienza University of Rome, via Eudossiana 18, 00184 Rome, Italy; paola.vinueza@uniroma1.it
[2] Department of Mathematics, University of Padua, Italy; zahra@math.unipd.it, conti@math.unipd.it
[3] Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[4] Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam; shahaboddin.shamshirband@tdt.edu.vn
[5] Faculty of Science, Engineering and Built Environment, Deakin University, Melbourne, Australia; jemal.abawajy@deakin.edu.au
* Correspondence: shahaboddin.shamshirband@tdt.edu.vn

Abstract: In this paper, we discuss the most significant application opportunities and outline the challenges in performing a real-time and energy-efficient management of the distributed resources available at mobile devices and Internet-to-Data Center. We also present an energy-efficient adaptive scheduler for Vehicular Fog Computing (VFC) that operates at the edge of a vehicular network, connected to the served Vehicular Clients (VCs) through an Infrastructure-to-Vehicular (I2V) over multiple Foglets (Fls). The scheduler optimizes the energy by leveraging the heterogeneity of Fls, where the Fl provider shapes the system workload by maximizing the task admission rate over data transfer and computation. The presented scheduling algorithm demonstrates that the resulting adaptive scheduler allows scalable and distributed implementation.

Keywords: Vehicular Fog Computing (VFC); Big Data Streaming (BDS); Energy-Efficiency; resource management; Infrastructure-to-Vehicular (I2V)

---

1. Introduction

Vehicular Fog Computing (VFC) is a promising paradigm (still unexplored), where the aim is supporting services in order to include the come up Internet infotainment applications. The Cloud Computing (CC) has attracted the big companies (such as VTube, Netflix, Cameleon, etc) [1,2] that exploit the Fog Nodes (FNs) of the Infrastructure-to-Vehicle (I2V) with a single-hop IEEE 802.11 likes wireless links for data dissemination, proving low energy consumption [2].

Furthermore, Fog Computing (FC) is used to design an efficient solution within supporting low-latency [3], geo-distributed Internet of thing (IoT) devices/sensors [4] and energy-efficient as well as self-adaption. It plays as spectrum efficient by means of a strict collaboration among end-devices in the same integrated platform [5]. FC aims at distributing self-powered data center (e.g., FNs) between resource remote clouds and resource-limited smartphone equipped in the vehicles [6], in order to perform context-aware energy-efficient data mining and dissemination for the adoption of smartphone device and clients' heavy dependence mobile applications [3,7,8]. Moreover, being deployed in the proximity of the Vehicular Clients (VCs), the FNs efficiently exploit the awareness of the connection states supporting latency and delay-jitter. Among the contributions that deal with the adaptive scaling of the server resources in similar VCs, the work in [9] focuses on the impact of the

Peer-reviewed version available at Appl. Sci. 2017, 7, 1325; doi:10.3390/app7121325

dynamic scaling of the CPU computing frequencies on the energy consumption experienced by the execution of MapReduce-type jobs.

Fog Data Centers (FDCs) are dedicated to supervising the transmission, distribution and communication networks [10]. All these require a precisely manage grid infrastructure from generation points to consumption ones by using the communication of measured values and transmitted control information more accurately. The IoT applications (real-time requirements, as stream processing) are distributed in different geographical locations [11], with numerous devices 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. FDC (Fog Data Center) services are mainly implemented in the Fog, that use a distributed architecture over moderate-bandwidth to provide high availability operations. The vehicles are attached to one of the (RSU) in the FN [12] with the heterogeneous type of devices consider a physical infrastructure, consisting each FN as a single server or a set of servers building heterogeneous FNs.

FDC has a huge amount of computations and it is distributed and may be more energy-efficient than the centralized Cloud model of computation, being the reduction of energy consumption on FDC an important challenge. Also, this drastically reduces the traffic sent to the Cloud by allowing the placement of filtering operators close to the sources of data. FDC as a vital component over the IoT (vehicular)environmental, 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.

FDC, in 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) is submitted from the remote clients 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.

FDC, roughly speaking, it is composed by 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 load balancer, the related switched Virtual LAN, and iii) an adaptive controller that dynamically manages all the available 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. As we know, many efforts have been made to curtail the energy consumption in data centers. Roughly speaking, FDC consolidation is a popular strategy to further reduce the energy consumption by turning OFF the underutilized VMs and grouping the VMs 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 recent consolidation technologies employed in data centers 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.

1.1. The main contribution of the paper

The technical contribution of this paper focuses on the design of Fog-based VFC architecture for the energy-efficient joint management of the networking and computing resources under hard constraints on the overall tolerated computing-plus-communication latency.

The main contribution of the paper is :

i) To the best of our knowledge, there is no related work that consider FDC across the VFC and presented resource allocation and scheduling for the considered real traffics. This is the first

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 November 2017 doi:10.20944/preprints201711.0126.v2

Peer-reviewed version available at *Appl. Sci.* **2017**, *7*, 1325; doi:10.3390/app7121325

work exploits dynamically scheduler the incoming real-based traffics into the FNs. Besides, this is the first work that study the impact of stream of data on network resource allocation.

ii) Present a distributed architecture Fog-based VFC architecture that over moderate-bandwidth to provide high availability operations of the mobile users.

iii) Define a cases study, named "StreamVehicularFog" (SVF), an Internet assisted peer-to-peer (P2P) service architecture and evaluate the stream of data passed/process over the underutilized networking/computing servers in FDC.

iv) The simulation results show significant reduction in the total energy consumption over FDC with various communication link shapes using real datasets.

The rest of the paper is organized as follow. The background is detailed in Section 2. The description of the FDC architecture is reported in the Section 3. A simple case study of the Fog data center data stream management and its correlated problem/solution are reported in Section 4. The performance evaluation of the optimal formulations is reported in Section 5. Finally, Section 6 concludes our work.

2. Related Work

In this section we aim to focusing on the resource management of Cloud/Fog-based distributed computing architectures for the energy-efficient support of real-time Big Data Streaming (BDS) applications by resource-limited wireless devices [13–16]. In detail, the S4 and DS-streams management frameworks in [13] and [14] perform dynamic resource scaling of the available virtualized resources by explicitly accounting for the delay-sensitive nature of the streaming workload offloaded by proximate wireless devices. The Time Stream and PLAstiCC resource orchestrator in [15] and [16] integrate dynamic server consolidation and inter-server live VM migration. Overall, like our contribution, the shared goals of [13–15] and [16] are: (i) the provisioning of real-time computing support to BDS applications run by resource-limited wireless devices through the exploitation of the virtualized resources done available by proximate FNs; and, (ii) the minimization of the overall inter/intra-data center computing-plus-networking energy consumption under BDS applications. However, unlike our contribution, we point out that: (i) the resource management techniques developed in [13] and [14] are not capable to self-tune to the (possibly, un-predictable) time fluctuation of the workload to be processed; and, (ii) the management approaches pursued in [15] and [16] do not guarantee hard limits on the resulting per-task execution times.

Focusing on the FC over VFC architecture, FC affords proximity to the end users, a dense geographical distribution and supports the mobility [3], it means, it provides services in local area instead of global and near to the users [3,17,18].

Authors in [19] have explored Fog Computing application in the way that they applied new fall detection algorithms and implements them into a distributed fall detection system in inter-FNs. Goal of [20] is the attainment of an optimized delay-vs.- energy trade-off under bag-of-task type vehicular applications that are executed on Dynamic Frequency Voltage Scaling (DVFS)-enabled data centers.

In Vehicular scenarios that are presented in [21,22], FNs are virtualized and connected to the networked data center (VNetFCs), which are hosted by Road Side Unit (RSU) at the edge of Vehicular Network and the task offloading can be achieved without introducing intolerable delay. In the IoTs era, Foglets (Fls) are defined as large implemented serving demands for the future 5G [23].

3. FDC over VFC Paradigm

In this paper, a basic system model for VFC with multiple Fls is illustrated in Fig. 1. The current practice allocates mobile tasks to the closest Foglet (Fl). The model provides and operates multiples Fls. Moreover, we consider a case study and go in depth in the Fog Data Center (FDC) as a vital component in VFC, present general resource allocation model and propose an adaptive scheduler which covers the hard limited QoSs. FC is a platform that uses a collaborative multitude of near-users edge devices [3] with some important characteristics shown in Table 1. It provides data, computing, storage and application services [3,18] to end users [24]. An example scenario is shown in Fig. 1. Specifically, in the Fig. 1, we adopt a simple three main areas: Vehicles area, RSU area, FDC area [3,17,18].

Table 1. Main Characteristics for Vehicular Fog Computing

FC	FDC	←FN	←RSU	←Vehicles
Access Medium	Fix Wireless	Fix Wireless	Wireless	Wireless
Technologies	WiFi/3G/4G-LTE			
Mobility	No	No	No	Yes
Proximity	Round	Near	Near	Nearly
Connectivity	Multi Hops	Multi Hops	Single Hop	Single Hop
Geographic Distribution	Low	Medium	High	Very High
Heterogeneity	No	Yes	Yes	Yes
Architecture	Centralized	Distributed	Distributed	Mobile
Bandwidth	High	High	High	Low
Latency	Low	Low	Low	Low
Delay Jitter	Low	Very Low	Very Low	Low
Network Area	Core	Edge	Field	Edge
QoS	Improve	High	High	High

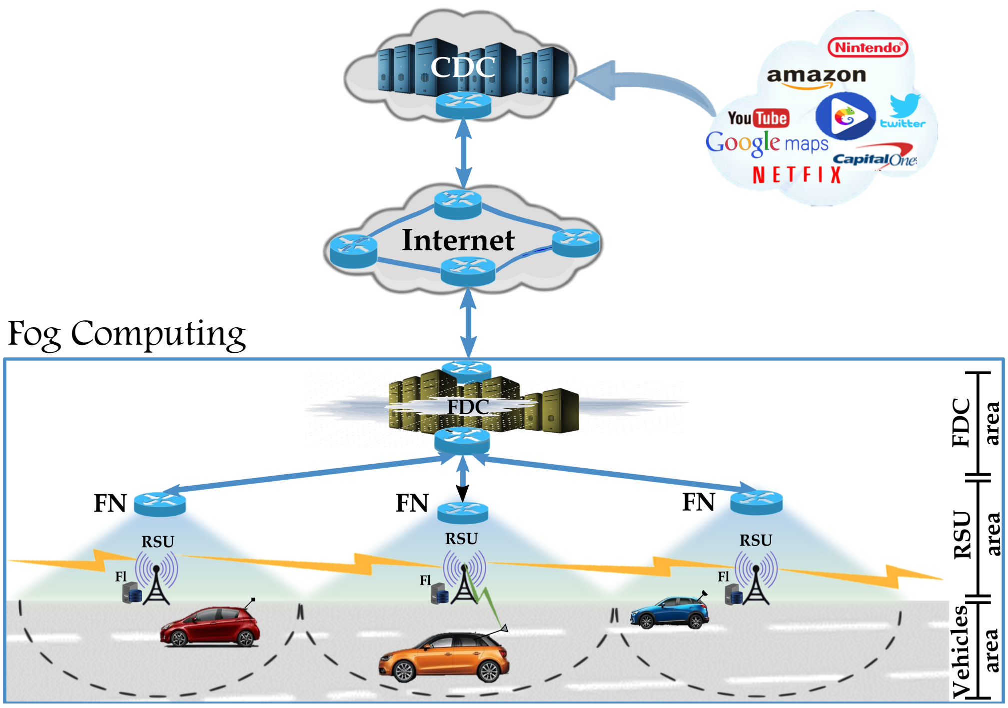


Figure 1. The overall VFC architecture. CDC:= Cloud-Data-Center with the main areas: FN:=Fog-Node, RSU:=Road-Side-Unit, Fl:=Foglet FDC:= Fog-Data-Center.

In the architecture presented in Fig. 1, the VFC provides a solid platform for applications used to provide services to users. VFC services are mainly implemented in the Fog: these use a distributed architecture over moderate-bandwidth to provide high availability operations. As we see, vehicles are attached to one of the RSU in the FN [12], with the heterogeneous type of devices considered as a physical infrastructure. Indeed, each FN consists of a single server or a set of servers that builds

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 November 2017 doi:10.20944/preprints201711.0126.v2

Peer-reviewed version available at Appl. Sci. 2017, 7, 1325; doi:10.3390/app7121325

heterogeneous requests. On the other hand, vehicles equipped with a smart device (e.g., smartphones) [7,17] that are wirelessly connected to the RSU. The vehicles in mobility are arbitrarily entering in the field of RSU and may leave the coverage area of it. FNs could be interconnected and each of them is linked to the FDC. Vehicles could request a service to RSU (i.e, then transfer this request to FN and finally FDC for managing the request) while they are freely moving across the FDC and RSU areas and FDC can provide required services to them in order to facilitate the connectivity between service providers and users. They provide application services for near-users that are located in this area (FDC area) and direct access to the context parameters with the main characteristics of VFC showing in the Table 1. The management architecture of FNs should include challenge strategies for resource allocation, edge devices management jointly with computing and storage resources. Also, it is important to remind that, it should be managed by Software Defined Networks (SDNs) [3] and Network Functions Virtualization (NFV) [24]. The correlated explanations are out of the topic of this paper. Motivated by these considerations, we aim to focus on the Fog computing environment and FDC component which are detailed in the next section.

4. A Case Study: StreamVehicularFog

We then move our attention to a more formal modeling of the presented case study and their correlated models. Motivated by the aforementioned consideration and based on the self-configuring framework as a prototype which is recently presented in [25], we define a cases study, named "StreamVehicularFog" (SVF), as a paradigm which is covered in an Internet assisted peer-to-peer (P2P) service architecture. The SVF aim is to minimize the total energy cost in mobile devices and data centers with an adaptive synchronization of the resource management operations that guarantee the stream of data passed/process over the underutilized networking/computing servers in FDC. In this scenario, FC equipped with RSUs to broadcast locally processed data to smartphones in the vehicles through point-to-point TCP/IP based I2V connections with only one hop wireless link. In order to evaluate the performance of our model, we consider a real-time streams workload as is presented in Fig. 2 being the arrival rate for the challenging task that sent to the FNs and should be processed through FDC to the available servers. Moreover, FDC adopts some resource allocation and scheduler techniques to analyze the data and setup frequently ON/OFF transitions of the underutilized computing/communication servers with the aim of energy provisioning. Indeed, in turn, Fig. 2 presents the trace of the I/O arrival data for the smart-phones on each vehicle of the VFC model in Fig. 1. Remind that, in this case study, we assume the processing data are gathered from the vehicular application service requested.

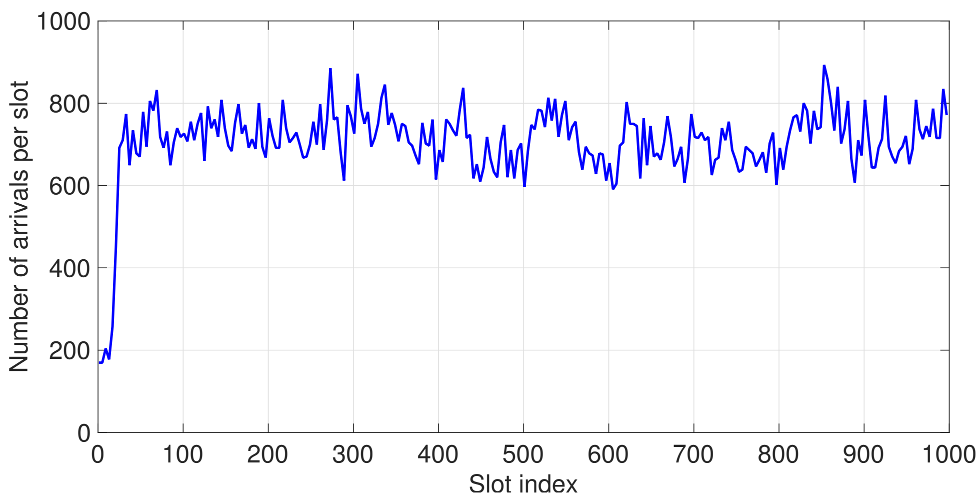


Figure 2. Trace sample of I/O arrival from enterprise cluster in WorldCup98 Workload [1:1000] [26]. The corresponding average arrival rate and PMR (Peak Mean Rate) are 1.56 and 19.65.

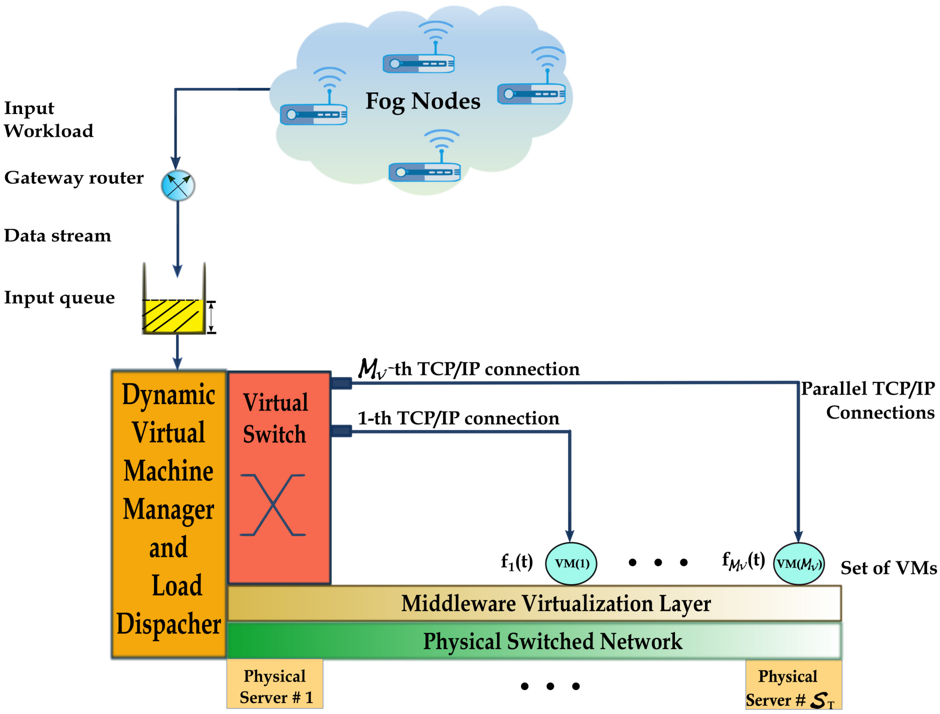


Figure 3. FDC Structure over StreamVehicularFog scenario.

In order to understand the FDC structure over SVF scenario, we modeled it in Fig. 3. We understand that each FN sends the coming traffic through the router gateway to the load dispatcher and transfer the traffic to the available virtual machines (VMs) running on the servers for the processing. We need to minimize the computation cost in each server as well as decreasing the inter-communication TCP/IP connection cost between VMs. In this model, virtual machine manager is responsible for the resource allocation and VM management. Therefore, we need to propose an efficient scheduling method for this resource management problem. To cope with this problem, firstly, we formulate a stochastic programming problem with the considerations of uncertainties in FNs. We propose a general algorithm to solve the simple problem engaged in FDC and give some results. Due to the possible high number of involved VCs, RSU, and hosted VMs, we require that the resulting adaptive scheduler allows scalable and distributed implementation in the emerging paradigm of the cognitive computing.

4.1. StreamVehicularFog virtualized TCP/IP communication model

Reporting Fig.3, the master functional block of our considered FDC is the intra-communication part. FDC operates at Middleware layer and is composed by i) an Admission Control Server (ACS) that is considered also as gateway Internet router, ii) an input buffer of the admitted workload (where our corresponding resource scheduler adaptively performs the control and allocation admission of the available virtual resources) shared by the VMs, iii) the Load Dispacher that reconfigure and consolidate the available computing/networking physical resources, dispatching the workload buffered by the input/queue to the turned ON VMs that processes the currently assigned task by self-managing guarantee the demand of the local virtualized computation and communication virtual resources, iv) the Virtual Switch that control/manage the TCP/IP (TCPNewReno congestion (by the switches) avoidance state) protocol as an effective means to manage the end-to-end transport connections, allowing the internal VM communication.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 November 2017 doi:10.20944/preprints201711.0126.v2

Peer-reviewed version available at Appl. Sci. 2017, 7, 1325; doi:10.3390/app7121325

4.2. StreamVehicularFog computing model

We consider the most general case composed by multiple reconfigurable VMs per physical server interconnected by a switched rate-adaptive virtual LAN (VLAN), managed by the controller, named StreamVehicularFog Manager (SVFM), with the following parameters:

- $\mathcal{S}_T \stackrel{\triangle}{=}$ total number of physical server with $\mathcal{S}_T \geq 1$;
- $\mathcal{VM}_{max}(s) \stackrel{\triangle}{=}$ maximum number of VMs (possibly, heterogeneous) hosted by the $s-$th physical server with
  $\mathcal{VM}_{max}(s) \geq 1,\ s = 1, \ldots, \mathcal{S}_T$;
- $\mathcal{VM}(s;v) \stackrel{\triangle}{=} v-$th VM hosted by the $s-$th server, $1 \leq v \leq \mathcal{VM}_{max}(s),\ 1 \leq s \leq \mathcal{S}_T$;
- $\mathcal{M}_{sv} \stackrel{\triangle}{=}$ total number of VMs in the data center that is calculated as: $\sum_{s=1}^{\mathcal{S}_T} \mathcal{VM}_{max}(s) \equiv \mathcal{S}_T \times \mathcal{VM}_{max}$. Formally, $\mathcal{M}_{sv}$ is the set of the available VMs which may be hosted by the FDC.

Moreover, the engaged internal parameters for the server of FDC is presented as follows:

- $\mathcal{M}(t) \stackrel{\triangle}{=}$ set of the VMs which are turned-ON at (t) time slot
  $(t) \subseteq \{(s;v);\ 1 \leq s \leq \mathcal{S}_T;\ 1 \leq v \leq \mathcal{VM}_{max}(s)\}$
- $\overline{\mathcal{M}}(t) \stackrel{\triangle}{=}$ set of the VMs which are turned-OFF at $(t) \subseteq \{(s;v);\ 1 \leq s \leq \mathcal{S}_T;\ 1 \leq v \leq \mathcal{VM}_{max}(s)\}$

so that's it

- $\overline{\mathcal{M}}(t) \equiv \{(s;v);\ 1 \leq s \leq \mathcal{S}_T;\ 1 \leq v \leq \mathcal{VM}_{max}(s)\} / \mathcal{M}(t)$;
- $f_{s,v}(t) \stackrel{\triangle}{=}$ processing rate of VM(s;v) at $t \in [0,\ f_{s,v}^{max}](IU/s),\ 1 \leq s \leq \mathcal{S}_T,\ 1 \leq v \leq \mathcal{VM}_{max}(s)$;
- $\mathcal{L}_{s,v}(t) \stackrel{\triangle}{=}$ workload processed by VM(s;v) at $t \in [0,\ \mathcal{L}_{s,v}^{max}](IU),\ 1 \leq s \leq \mathcal{S}_T,\ 1 \leq v \leq \mathcal{VM}_{max}(s)$;
- $f_{s,v} \stackrel{\triangle}{=}$ maximum processing rate of VM(s;v), $(IU/s),\ 1 \leq s \leq \mathcal{S}_T,\ 1 \leq v \leq \mathcal{VM}_{max}(s)$;
- $\mathcal{L}_{s,v}^{max} \stackrel{\triangle}{=} t_{slot} f_{s,v}^{max} \stackrel{\triangle}{=}$ maximum workload processed by VM(s;v) during each time slot (t); in $(IU),\ 1 \leq s \leq \mathcal{S}_T,\ 1 \leq v \leq \mathcal{VM}_{max}(s)$;
  $\mathcal{L}_{s,v}^{max} \leq t_{slot} f_{s,v}^{max}$.

Let $t \in \mathcal{T}$ be a consolidation instant. Each slot t can be considered as the following states:

1. turn-ON some physical servers which were turned-OFF at slot t;
2. turn-OFF some physical servers which were turned-ON at slot t;
3. turn-ON or turned-OFF some VMs hosted by servers which were turned-ON at slot $(t-1)$ and are still turned-ON at slot t;
4. scaling up/down the processing rates and workloads of some VMs which are turned-ON at slot t.

According to the aforementioned engaged parameters, the following actions should be made to manage the problem applicable in the real environment: Let: $f_{ON}(IU/s)$ be the minimum practical processing rate allowed a turned-ON VMs and $\mathcal{M}(t) \equiv \{(s;v):\ f_{s,v}(t) > f_{ON};\ 1 \leq s \leq \mathcal{S}_T;\ 1 \leq v \leq \mathcal{VM}_{max}(s)\}$ and $\overline{\mathcal{M}}(t) \equiv \{(s;v):\ f_{s,v}(t) \leq f_{ON};\ 1 \leq s \leq \mathcal{S}_T;\ 1 \leq v \leq \mathcal{VM}_{max}(s)\}$.

4.3. Energy models

Let $\mathcal{E}_c^{idle}(s)[J]$, and $\mathcal{E}_c^{max}(s)[J]$ $1 \leq s \leq \mathcal{S}_T$, be the energies wasted by the $s-$th physical server in the idle (inactive) state and we have $\mathcal{E}_c^{idle}(s) \leq \mathcal{E}_c^{max}(s)$

and the TCP/IP intra-connection inside the FDC can be considered under Fast/Giga Ethernet with the practical ranges of values for $\Omega_{s,v}^c$ as bandwidth power coefficient of $[J/(IU)^\gamma]$, and R_{max} as end-to-end band-witch rate in $([IU/sec])$ as follows:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 November 2017 doi:10.20944/preprints201711.0126.v2

Peer-reviewed version available at Appl. Sci. 2017, 7, 1325; doi:10.3390/app7121325

- Fast Ethernet: $1.1 \times 10^{-3} \leq \Omega^c_{s,v} \leq 1.8 \times 10^{-3}$, R_{max}=100
- Giga Ethernet: $1.8 \times 10^{-3} \leq \Omega^c_{s,v} \leq 2.5 \times 10^{-2}$, R_{max}=1000.

The corresponding communication energy $\mathcal{E}^{LAN}_{s,v}[J]$ wasted by the s-the virtual link is: $\mathcal{E}^{LAN}_{s,v} \equiv \mathcal{E}^{LAN}_{s,v}(\mathcal{L}_{s,v}) = \Omega^c_{s,v}(\mathcal{L}_{s,v})^{\gamma}[J], 1 \leq v \leq \mathcal{VM}_{max}(s); \quad s = 1, \ldots, \mathcal{S}_T$.

When the VMs hosted per physical server is perfectly isolated, the resulting model for the overall wasted computing energy simplifies to the following formula:
$\mathcal{E}_T(t) = \mathcal{E}_C(t) + \mathcal{E}_S(t) + \mathcal{E}_N(t)$ $[J]$: where, $\mathcal{E}_C(t)$ is the SVF computing energy; $\mathcal{E}_S(t)$ is the SVF energy switching; and, $\mathcal{E}_N(t)$ is the SVF TCP/IP communication energy. Therefore, the $\mathcal{E}_T$ for slot t is defined as

$$\mathcal{E}_T(t) = \sum_{s=1}^{\mathcal{S}_T} \left\{ \left[\mathcal{E}^{idle}_c(s) \left(\sum_{v=1}^{\mathcal{VM}_{max}(s)} f_{s,v}(t) \right) + \left(\frac{\mathcal{E}^{max}_{c,v}(s) - \mathcal{E}^{idle}_{c,v}(s)}{\mathcal{VM}_{max}(s)} \right) \left(\sum_{v=1}^{\mathcal{VM}_{max}(s)} \left(\frac{f_{s,v}(t)}{f^{max}_{s,v}} \right)^w \right) \right] + k_e \left(\sum_{v=1}^{\mathcal{VM}_{max}(s)} (f_{s,v}(t) - f_{s,v}(t-1))^2 \right) + \left(\sum_{v=1}^{\mathcal{VM}_{max}(s)} \Omega^c_{s,v}(\mathcal{L}_{s,v})^{\gamma}(t) \right) \right\}, \tag{1}$$

where $t \in T, t \geq 1$; γ is the powered model for the per-transport layer connections energy holds for energy instant t; $\Omega-$ is the common energy consumptions of all LAN (TCP/IP) connection measured $[J/(IU)^{\gamma}]$ and $w \geq 2$ does strictly convex in $f_{s,v}(t)$. Therefore, the general optimization problem can be simplified as

$$\min\{\mathcal{E}_T(t)\}, \tag{2.1}$$

subjected to:

$$\mathcal{L}_T(t) - \left(\sum_{s=1}^{\mathcal{S}_T} \sum_{v=1}^{\mathcal{VM}_{max}(s)} \mathcal{L}_{s,v}(t) \right) = 0, \tag{2.2}$$

$$\mathcal{L}_{s,v}(t) - t_{slot} f^{max}_{s,v} \leq 0, \qquad \forall s, v \in \mathcal{VM}(s; v), \tag{2.3}$$

$$f_{s,v}(t) - f^{max}_{s,v} \leq 0, \qquad \forall s, v \in \mathcal{VM}(s; v), \tag{2.4}$$

where t_{slot} is the hard deadline considered to process each slot, in fact, it can be considered as the service level agreement for the mobile application requested from VFC. The workload processed should not be more than input workload in each time slot (t) (see Eq.(2.2)), and the frequency of each VMs should not be more than the maximum frequency (see Eqs.(2.3) and (2.4)).

In order to resolve the problem, we can consider the problem as a *bin packing penalty-aware* problem. **Algorithm 1** presents the pseudocode of the proposed solution. Indeed, in turn, for each incoming workload $\mathcal{L}_T(t)$ we calculate how many VMs can be allocated for t-th slot. After that, we consider servers as bins and VMs as packs that must be served distributively based on their time limitations and frequency limitations. For each iteration *it*, we give a penalty to each server based on their energy characteristics (idle energy, maximum energy, maximum frequency) and look for the best servers to be allocated. The penalty is considered in order to decrease the fatality cases (increasing the energy consumptions exponentially). It means the server will be punished and banned to be used again in next *it*. When the freezing iterations n-th for the server passed, the server will back to the list of available servers for the next *it*. Note that, server S can serve VMs when it does not pass the maximum number of VMs $\mathcal{VM}_{max}(s)$. Otherwise, server S puts out from the list of processing servers. The maximum interactions are defined $\mathcal{M}_{s,v}$. The process is done for 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 t.

Algorithm 1 Pseudo-code Heuristic Solution

.

INPUT: $\mathcal{S}_T, \hat{\mathcal{S}}, \mathcal{M}_{sv}, \mathcal{L}_T, f_{s,v}^{max}, \mathcal{E}_{c,v}^{idle}, \mathcal{E}_{max}, \mathcal{T}$
OUTPUT: $\mathcal{E}_T, \hat{\mathcal{S}}(t), \mathcal{M}(t)$

for $t \geq 1$ **do** $\quad t \in T$
    Check feasibility of $input$
    for $it = 1 : \mathcal{M}_{sv}$ **do**
        if $\mathcal{L}(t) \leq f_{s,v}^{max}\, t_{slot}$ **then**
            $f_{s,v}(t) = \mathcal{L}(t)/t_{slot};$
                ▷ Update set of servers $\hat{\mathcal{S}}(t)$ with 0 penalty;
            Find the minimum $\mathcal{E}_{c,v}^{idle}(t)$ in $\hat{\mathcal{S}}(t)$;
            if $(flag(t) == 1)$ **then**
                ▷ decrease the penalty (-1) in all the servers with $penalty \geq 1$;
            end if
                ▷ add new VM into the $\hat{\mathcal{S}}(t)$;
        end if
    end for
    Update $\hat{\mathcal{S}}(t)$ and $\mathcal{M}(t)$ through $\mathcal{VM}(s;v)$;
    Calculate $\mathcal{E}(t)$ as in (1);
end for
return $\mathcal{E}_T, \hat{\mathcal{S}}(t), \mathcal{M}(t)$

5. Performance Evaluation

In this section, we aim to test the proposed method on FDC.

5.1. Test Scenario

In order to guarantee the accuracy and timeliness of information transmission of the VCs. It is necessary to analyze and act on the data in less than a second. Data are sent to the Cloud for historical analysis, BD analytics, and long-term storage with less latency. For example, each of thousands or hundreds of thousands of FNs might send periodic summaries of the data grid to the Cloud for historical analysis and storage.

In carried out test, the FNs are assumed placed over a spatially limited Intranet. We consider a physical topology with four heterogeneous FNs. In order to model the CPU consumptions of the FNs, we refer to the Intel R-CoreTM 2 CPU Q6700 with 2.67 GHz frequency rate and 4 GB of RAM memory. In order to confirm the correctness of the proposed infrastructure in Fig. 1, we require an evaluation for exploring several resource management and scheduling techniques such as operator, application and task placement and consolidation. We make some evaluations of resource management policies applicable to Fog environment with respect to their impact on latency, network usage, and energy consumption. In this framework, the VCs publish data to SVF, applications run on Fog devices (FN) to process data coming from VCs. Indeed, it is important to test the solution with real-world input workload and compare it with the corresponding ones of the recent FDC-based techniques. So in the next subsection, we will move to this issue and we will specify it with several simulated testbed cases.

5.2. Test Result

In this subsection, we aim to test our scheduler on 100 incoming data stream ($T = 100$) for Fast Ethernet communication channel and calculate the on-line instantaneous overall energy consumption and energy savings that are shown in Fig. 4. From this figure, we conclude that the total energy cost in changing by the time and even it has the effect on its correlated saving energy per time.

On the other hand, in order to evaluate the instantaneous energy consumption for 1000 slots using two different scheme of the communication network (Fast and Giga), we evaluate the energy

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 November 2017 doi:10.20944/preprints201711.0126.v2

Peer-reviewed version available at Appl. Sci. 2017, 7, 1325; doi:10.3390/app7121325

consumptions of Fast/Giga Ethernet of SVF scenario. The results of these comparisons are presented in Figs. 5. We understand that when the $\Omega^{c}_{s,v}$ increases the correlated energy consumption rises with the same scale due to rising the communication constant and this changing continues due to time.

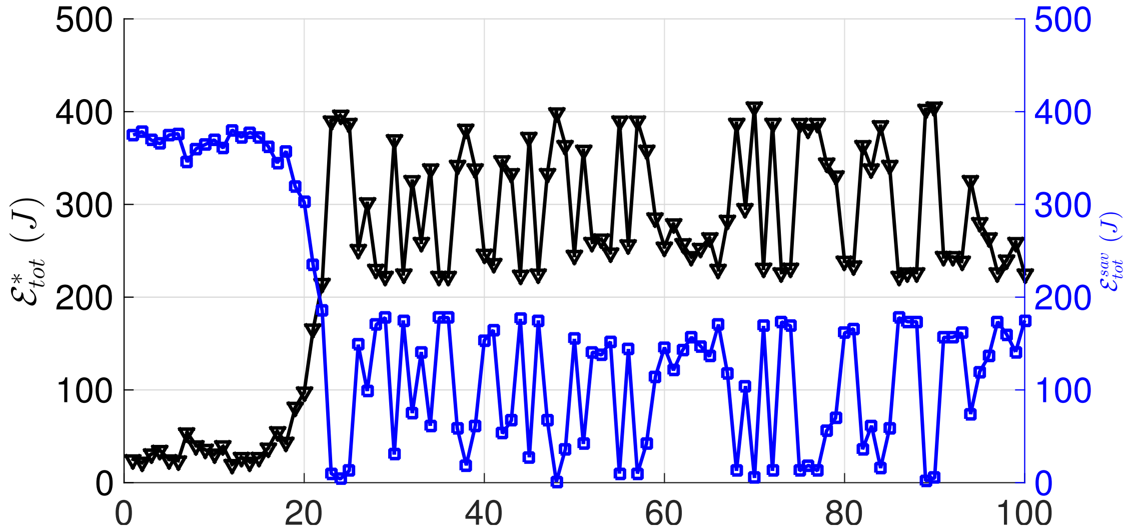


Figure 4. Energy saving of SVF (dashed plot) and energy consumption of the SVF (continuous plot) for the Fast Ethernet LAN for $T = 100$.

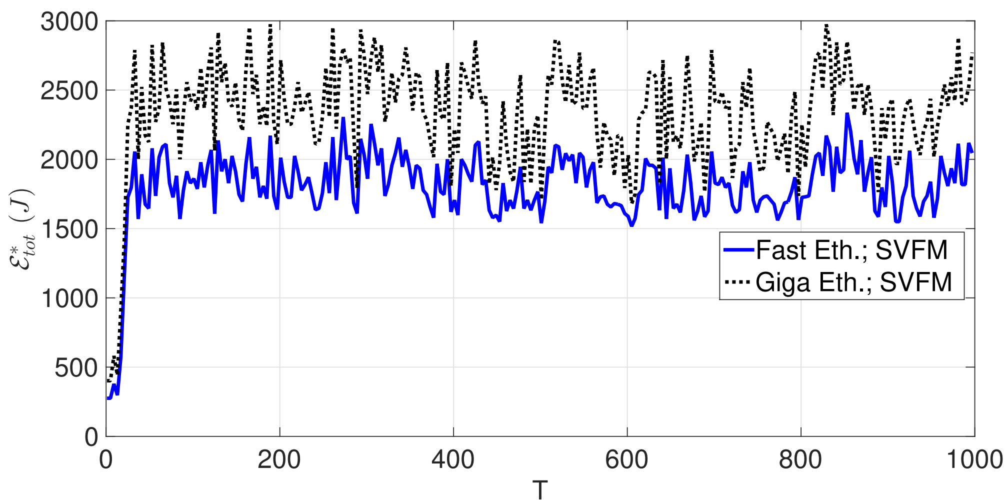
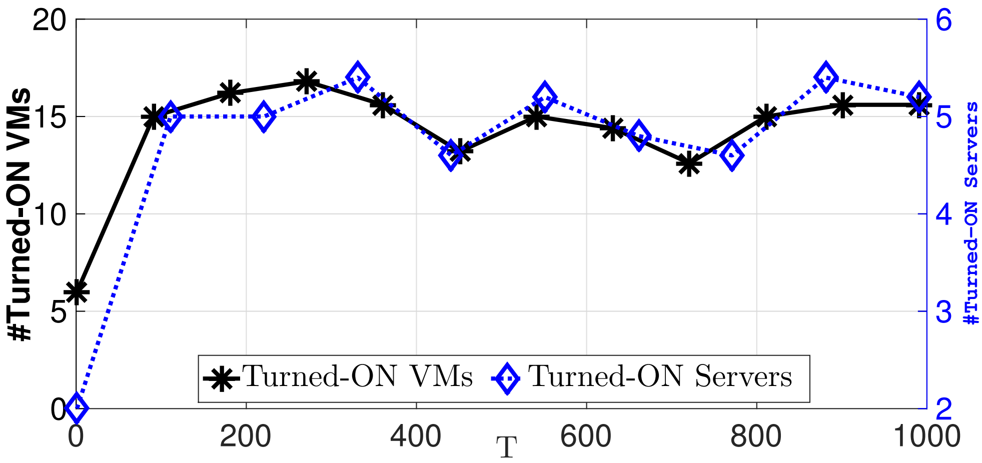


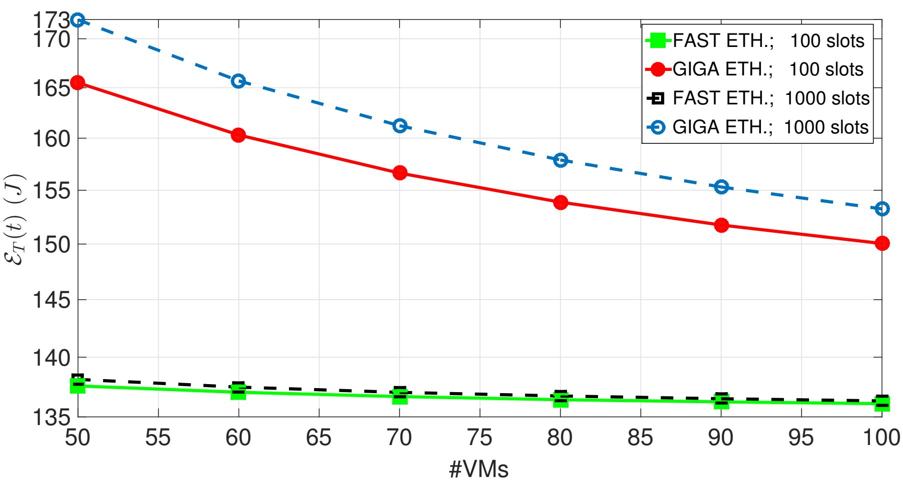
Figure 5. Fast/Giga Ethernet LANs on the energy consumption of the SVF for $T = 1000$.

Finally, in the last test scenario, the plots of Fig. 6a report the time-behaviors of the numbers of VMs and physical servers that are dynamically turned ON by our algorithm scheduler under the real-world workload of Fig. 2. A comparison of the plots of Fig. 6a with the arrival trace of Fig. 2 supports the conclusion that the consolidation action performed by the proposed algorithm is capable to track the abrupt (and unpredicted) changes of the input workload with a delay of 1 slot period. An examination of Fig. 6a points out that: i) The number of Turned-ON VMs/servers increases for increasing values of timeslots (T) (i.e., the network state pushes the scheduler to keep more FNs and corresponding servers to be ON and response the injected traffic over the Fls); and, (ii) The rate of Turned-ON servers in each timeslot is more than the Turned-ON VMs and it confirms that the servers are consuming more energy than VMs and when we use VMs we can control the server energy consumption. Also, this rate depends on the adopted Ethernet technology applied. Focusing on Fig.

2, it is interesting to note that the numerically evaluated average energy consumption over 100 VMs (10 servers residing in 4 heterogeneous FNs) of the proposed heuristic for 100 slots is about 136.5 (J) and 138.5 (J) and this value for 1000 slots is about 155 (J) and 160 (J) or GIGA and FAST Ethernet communication cases, respectively. This confirms that even by increasing the slots 10 times the average energy consumption in the proposed scheduler wont increase a lot and at most 23% for the worst case.



(a) VMs turned ON-vs.- Timeslot (incoming traffic).



(b) Total energy consumption on the FDC-vs.- number of running VMs per FN.

Figure 6. Fig. 6a: Behavior time of number of turned-ON VMs and physical servers, and Fig. 6b: Average total energy in $T = 1000$, respectively.

6. Conclusion and Future Developments

The ability of Fog Computing is never-ending. It has the potential to solve many problems in different fields that are still unexplored such as context-aware data produce/transfer in vehicular networks. Fog computing over the vehicular networks is a platform that offers real-time services to the clients. In this paper, we study one of the most significant issues of FC which aims to focus on the Fog data center data streams which are requested from the vehicular clients. We proposed a holistic VFC architecture, focus on the FDC data streams and formulate the problem analytically and resolve the presented resource allocation and scheduling algorithm over a simplified cases study

Peer-reviewed version available at Appl. Sci. 2017, 7, 1325; doi:10.3390/app7121325

called StreamVehicularFog (SVF). In this regard, the SVF simulations in both small and large scales confirmed the considerable reduction in the average energy cost that can be achieved by scheduling in mobile edge fog of SVF. We test the presented case study with various test scenarios and confirm the energy efficiencies. The energy-efficient adaptive management of the delay-vs.-throughput the trade-off of the SVF TCP/IP mobile connections becomes an additional topic for further research.

7. Acknowledgement

This work has been partially supported by the PRIN2015 project with grant number 2015YPXH4W_004: ”A green adaptive FC and networking architecture (GAUChO)”, funded by the Italian MIUR and by the project ”Vehicular Fog energy efficient QoS mining and dissemination of multimedia Big Data streams (V-Fog)” funded by Sapienza University of Rome. Moreover, 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. Karagiannis, G.; Altintas, O.; Ekici, E.; Heijenk, G.; Jarupan, B.; Lin, K.; Weil, T. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE Communications Surveys & Tutorials* **2011**, *13*, 584–616.
2. Naranjo, P.G.V.; Pooranian, Z.; Shojafar, M.; Conti, M.; Buyya, R. FOCAN: A Fog-supported Smart City Network Architecture for Management of Applications in the Internet of Everything Environments. *arXiv preprint arXiv:1710.01801* **2017**.
3. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog computing and its role in the internet of things. Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012, pp. 13–16.
4. Mahmud, R.; Kotagiri, R.; Buyya, R. Fog computing: A taxonomy, survey and future directions. In *Internet of Everything*; Springer, 2018; pp. 103–130.
5. Tajiki, M.M.; Salsano, S.; Shojafar, M.; Chiaraviglio, L.; Akbari, B. Joint Energy Efficient and QoS-aware Path Allocation and VNF Placement for Service Function Chaining. *arXiv preprint arXiv:1710.02611* **2017**.
6. Shojafar, M.; Cordeschi, N.; Baccarelli, E. Energy-efficient adaptive resource management for real-time vehicular cloud services. *IEEE Transactions on Cloud computing* **2016**, *PP*, 1–14.
7. Büsching, F.; Schildt, S.; Wolf, L. DroidCluster: Towards Smartphone Cluster Computing–The Streets are Paved with Potential Computer Clusters. 2012 32nd International Conference on Distributed Computing Systems Workshops. IEEE, 2012, pp. 114–117.
8. D'Andreagiovanni, F.; Mett, F.; Nardin, A.; Pulaj, J. Integrating LP-guided variable fixing with MIP heuristics in the robust design of hybrid wired-wireless FTTx access networks. *Applied Soft Computing* **2017**, *61*, 1074–1087.
9. Wirtz, T.; Ge, R. Improving mapreduce energy efficiency for computation intensive workloads. Green Computing Conference and Workshops (IGCC), 2011 International. IEEE, 2011, pp. 1–8.
10. Pooranian, Z.; Shojafar, M.; Naranjo, P.G.V.; Chiaraviglio, L.; Conti, M. A Novel Distributed Fog-Based Networked Architecture to Preserve Energy in Fog Data Centers. 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). IEEE, 2017, pp. 604–609.
11. Dabbagh, M.; Hamdaoui, B.; Guizani, M.; Rayes, A. An energy-efficient VM prediction and migration framework for overcommitted clouds. *IEEE Transactions on Cloud Computing* **2016**.
12. Aazam, M.; Huh, E.N. Fog computing and smart gateway based communication for cloud of things. Future Internet of Things and Cloud (FiCloud), 2014 International Conference on. IEEE, 2014, pp. 464–470.
13. Neumeyer, L.; Robbins, B.; Nair, A.; Kesari, A. S4: Distributed stream computing platform. Data Mining Workshops (ICDMW), 2010 IEEE International Conference on. IEEE, 2010, pp. 170–177.

Peer-reviewed version available at *Appl. Sci.* **2017**, *7*, 1325; doi:10.3390/app7121325

14. Zaharia, M.; Das, T.; Li, H.; Shenker, S.; Stoica, I. Discretized Streams: An Efficient and Fault-Tolerant Model for Stream Processing on Large Clusters. *HotCloud* **2012**, *12*, 10–10.
15. Qian, Z.; He, Y.; Su, C.; Wu, Z.; Zhu, H.; Zhang, T.; Zhou, L.; Yu, Y.; Zhang, Z. Timestream: Reliable stream computation in the cloud. Proceedings of the 8th ACM European Conference on Computer Systems. ACM, 2013, pp. 1–14.
16. Kumbhare, A.G.; Simmhan, Y.; Prasanna, V.K. Plasticc: Predictive look-ahead scheduling for continuous dataflows on clouds. Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on. IEEE, 2014, pp. 344–353.
17. Tang, B.; Chen, Z.; Hefferman, G.; Wei, T.; He, H.; Yang, Q. A hierarchical distributed fog computing architecture for big data analysis in smart cities. Proceedings of the ASE BigData & SocialInformatics 2015. ACM, 2015, p. 28.
18. Skala, K.; Davidovic, D.; Afgan, E.; Sovic, I.; Sojat, Z. Scalable distributed computing hierarchy: Cloud, fog and dew computing. *Open Journal of Cloud Computing (OJCC)* **2015**, *2*, 16–24.
19. Cao, Y.; Chen, S.; Hou, P.; Brown, D. FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation. Networking, Architecture and Storage (NAS). 2015 IEEE International Conference on, 2015.
20. Kim, K.H.; Buyya, R.; Kim, J. Power Aware Scheduling of Bag-of-Tasks Applications with Deadline Constraints on DVS-enabled Clusters. CCGrid, 2007, Vol. 7, pp. 541–548.
21. Bitam, S.; Mellouk, A.; Zeadally, S. VANET-cloud: a generic cloud computing model for vehicular Ad Hoc networks. *IEEE Wireless Communications* **2015**, *22*, 96–102.
22. Mtibaa, A.; Fahim, A.; Harras, K.A.; Ammar, M.H. Towards resource sharing in mobile device clouds: Power balancing across mobile devices. ACM SIGCOMM Computer Communication Review. ACM, 2013, Vol. 43, pp. 51–56.
23. Chen, M.; Zhang, Y.; Li, Y.; Mao, S.; Leung, V.C. EMC: emotion-aware mobile cloud computing in 5G. *IEEE Network* **2015**, *29*, 32–38.
24. Vaquero, L.M.; Rodero-Merino, L. Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM Computer Communication Review* **2014**, *44*, 27–32.
25. Barbera, M.V.; Kosta, S.; Mei, A.; Perta, V.C.; Stefa, J. Mobile offloading in the wild: Findings and lessons learned through a real-life experiment with a new cloud-aware system. IEEE INFOCOM 2014-IEEE Conference on Computer Communications. IEEE, 2014, pp. 2355–2363.
26. Urgaonkar, B.; Pacifici, G.; Shenoy, P.; Spreitzer, M.; Tantawi, A. Analytic modeling of multitier internet applications. *ACM Transactions on the Web (TWEB)* **2007**, *1*, 2.