Dynamic Virtual Machine Migration in a Vehicular Cloud

Tarek K. Refaat*, Burak Kantarci+, and Hussein T. Mouftah+
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, ON, Canada
*tarek.k.r@ieee.org
+{kantarci, mouftah}@site.uottawa.ca

***Abstract*—Vehicular clouds are formed by incorporating cloud-based services into vehicular ad hoc networks. Amongst the several challenges in a vehicular cloud network, virtual machine migration (VMM) may be one of the most crucial issues that need addressing. In this paper, a novel solution for VMM in a vehicular cloud is presented. The vehicular cloud is modeled as a small corporate data center with mobile hosts, equipped with limited computational and storage capacities. The proposed scheme is called Vehicular Virtual Machine Migration (VVMM). The VVMM aims to achieve efficient handling of frequent changes in the data center topology, host heterogeneity, all while doing so with minimum Roadside Unit (RU) intervention. Three modes of VVMM are studied. The first mode, VVMM-U uniformly selects the destinations for VM migrations, which will take place shortly prior to a vehicle's departure from the coverage of the RU. The second mode, VVMM-LW aims at migrating the VM to the vehicle with the least workload, and the third mode, VVMM-MA incorporates mobility awareness by migrating the VM to the vehicle with the least workload and forecasted to be within the geographic boundaries of the vehicular cloud. We evaluate the performance of our proposed framework through simulations. Simulation results show that VVMM-MA introduces significant reduction in unsuccessful migration attempts and results in an increased fairness in vehicle capacity utilization across the vehicular cloud system.**

Keywords*—cloud computing; data center management; vehicular ad hoc networks; vehicular clouds; virtual machine migration

I. INTRODUCTION

Cloud computing (CC) is commonly presented as the gateway of future Internet of everything, where billions of objects with computing and communication capabilities are interconnected [1]. Vehicular ad hoc networks (VANETs) offer an interconnected communication and computing platform, utilizing vehicles equipped with embedded computer systems, i.e. computing, storage and networking resources. Although most vehicles nowadays are produced with these facilities, these resources mostly remain underutilized [2]. Future VANET deployments are expected to better utilize these resources by consolidating them in a cloud platform, whereby each connected vehicle will act as a mobile physical host in a data center [3].

As presented in Fig. 1, a vehicular cloud can offer Computing-as-a-Service (CompaaS), Storage-as-a-Service (STaaS), Network-as-a-Service (NaaS) [4], Cooperation-as-a-Service (CaaS) [5], Entertainment-as-a-Service (ENaas), Information-as-a-Service (INaaS) [6], and Traffic-Information-as-a-Service (TIaaS) [7].

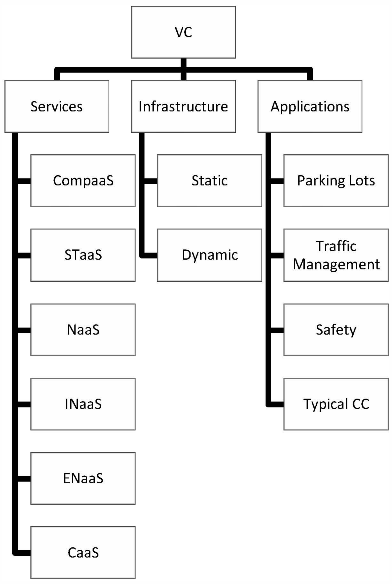

Fig. 1. Overview of a vehicular cloud (VC) system [2].

In a conventional cloud data center, physical servers are partitioned into multiple Virtual Machines (VM), and users are assigned VMs so that they can use the same physical environment by being totally isolated from each other. Resource virtualization enables migrating logical resources, denoted by a VM on a physical server, from one physical host to another for several reasons. Such reasons include enhanced physical resource utilization, improved energy-efficiency and reliability, and hot-spot mitigation [4].

Efficient and reliable VM migration (VMM) in cloud data centers is a crucial problem, which has been widely studied in the literature [5]. However, VMM in a vehicular cloud needs to address further challenges due to the following reasons: As opposed to a cloud data center, a vehicular cloud data center is a more dynamic environment, due to the *mobility* of the physical servers, namely the connected vehicles. Furthermore, due to the mobility of the hosts, the topology of a vehicular cloud data center network changes more rapidly in comparison to the conventional cloud data center. Also, physical hosts and consequently the VMs in a vehicular cloud data center have limited computing and storage capacity when compared to a cloud data center. As such, vehicular clouds call for VANET-specific and cloud-inspired VMM techniques.

This paper proposes a VMM framework for the vehicular cloud, named Vehicular Virtual Machine Migration (VVMM). Since a vehicular cloud can be considered as a data center with mobile physical hosts, frequent changes in the data center topology are expected. Furthermore, since vehicles are equipped with different on-board systems, heterogeneity of the hosts introduces further challenges in VMM. VVMM aims at addressing these challenges with minimum intervention required by the Roadside Units (RUs). To this end, three modes of VVMM are proposed as follows: VVMM-U uniformly selects the new hosts of the VMs needing migration, due to a vehicle soon leaving the vehicular cloud. VVMM-LW shifts the VMs to the vehicle with the least workload. VVMM-MA incorporates mobility awareness by shifting the VM to the vehicle with the least workload as well as one predicted to be within the geographic boundaries of the vehicular cloud, for the duration of the migration. Simulation results confirm that when VVMM-MA is adopted, the proposed framework can achieve a greater ratio of successful migrations, as well as a higher fairness among the vehicles in the cloud system.

The rest of the paper is organized as follows. Section II presents the related work on Vehicular clouds and VMM. Section III presents the proposed framework along with its three modes. Section IV discusses the numerical results. Finally, Section V concludes the paper and gives future directions.

II. RELATED WORK AND MOTIVATION

A Vehicular Cloud system is a non-conventional cloud environment that aims at efficient utilization of computing and storage resources hosted on the vehicles. Vehicular cloud computing bears several challenges which are mainly energy efficiency, resource mobility, inter-vehicle wireless communication and networking constraints, latency, heterogeneity and limitation of resources, safety, security and privacy services, and standardization issues.

A vehicular cloud system can be implemented by either of the following approaches. A cloud data center can be implemented by consolidating the computing resources in a group of cars that are geographically co-located in the parking position [10]. This type of vehicular cloud system is very similar to the conventional cloud system which consists of virtualized resources in geographically co-located physical hosts in a data center. The second approach could be a cloud linked to an infrastructure or backbone via cellular towers or wired links to the Internet. In this approach, incorporation of the backbone can ease the management and orchestrating the cloud system [11]. A pure vehicular cloud system can be implemented over existing VANET technology between mobile vehicles. This type of implementation is the most challenging. However, it is the most desired type of cloud system for today's connected vehicles [12].

As mentioned before, in a conventional cloud system, virtualization enables cloud users sharing the same physical medium in an isolated manner. During service time, the VMs assigned to particular users can be shifted between physical hosts for several purposes such as enhanced utilization, energy-efficiency and hot-spot mitigation, among others. In [8], a comprehensive survey of the existing VM migration (VMM) solutions in conventional clouds has been presented in detail.

In vehicular clouds, VM hosts may vary based on the application. For instance VM hosts, for traffic data mining for real time navigations, are the central and roadside clouds. Furthermore, roadside clouds are also the VM hosts for distributed storage for video surveillance. On the other hand, VMs for cooperative download of large files are mainly hosted on connected vehicles. These types of applications introduce the most significant challenge in VM management and migrations in vehicular clouds. To the best of our knowledge, VM management and migration has not been well studied in the literature, and the main contribution to this field has been made by the authors in [13] where the usage of roadside clouds as VM hosts are preferred over using connected vehicles. Therefore, *connected vehicles as physical hosts of the VMs in a vehicular cloud system* still has several unaddressed issues. As such, this paper aims at providing initial insights for VM migration in such an architecture, where vehicles are the physical VM hosts.

III. VEHICULAR VIRTUAL MACHINE MIGRATION (VVMM)

Vehicular Virtual Machine Migration is the proposed framework that addresses VMM in the vehicular cloud. As vehicles are VM hosts in this architecture, they are analogous to physical machines or servers in a conventional cloud data center. In contrast to conventional centers, the main need for VMM in the vehicular cloud stems from the dynamic nature of the topology. The physical hosts, in this case the vehicles, are rapidly changing location. Such a change in location may involve frequent handovers between various coverage ranges, as well leaving coverage completely. A vehicle that undergoes a handover between two communication ranges may not necessarily require a migration. This situation occurs if the two communication ranges are both reporting to the same Roadside Unit (RU). Migration will be necessary in two cases:

i. The handover occurs between communication ranges that fall under two different RUs.

ii. The vehicle is leaving the communication range of one RU and will not be in range of any RU.

The likelihood of requiring migration is directly dependent on many factors that affect the probability of a vehicular VM host falling under one of the two cases. Such factors include the geographical mapping of the wireless coverage in the area under study, the congestion of the vehicles, the vehicle speeds and the wireless communication technology adopted by the vehicular cloud. The wireless communication overhead on the VVMM performance is not considered in this study, however the transmission delay is set at τ seconds, which would be the maximum time possibly needed to perform a migration. The impact of wireless communication overhead is being addressed in our ongoing work.

Three migration schemes are tested: VVMM-U (i.e., *U* denoting ‘uniform’), VVMM-LW (i.e., *LW* denoting least workload) and VVMM-MA (i.e., *MA* denoting mobility-awareness).

Note that, in all of these schemes, searching a street implies performing the search on both directions of the street. Secondly, an unsuccessful migration denotes one of two scenarios:

i. The VM instance is hibernated within the vehicle until it rejoins a new coverage area. This will cause downtime on the client side.

ii. The VM is migrated to a temporary storage located in the RU, which is the last resort, due to an RU having limited capacity.

Both of these scenarios are referred to as “unsuccessful” migrations or “dropped” VMs. Note also, that for the sake of this study, only 1:1 migrations are considered. A vehicle can only target one destination. It cannot split its load over multiple destinations. Such a scenario is currently still in progress.

Algorithm 1 Generic pseudo-code for VVMM

```
1: {s: source of VVMM}
2: {d: destination of VVMM}
3: Begin
4: v: Set of nearest vehicles detected by s
5: if (|v| >0) then
6:   dest ← Select one destination based on the search criteria;
7:   if (dest has insufficient excess capacity to host the VM) then
8:     return Unsuccessful VVMM
9:   else
10:    if dest will need to migrate within τ seconds then
11:      return Unsuccessful VVMM
12:    else
13:      return Successful VVMM
14:    end if
15:  end if
16: else
17:   Resort to the RU
18:   ℵ ← Set of nearest destinations in the entire grid found by the RU
19:   if (|v| >0) then
20:     Go to line-6
21:   else
22:     return Store at the RU; unsuccessful VVMM
23:   end if
24: end if
25: End
```

Algorithm-1 illustrates the generic steps of VVMM techniques that are proposed here. The techniques differ from each other based on the *search criteria* in the set of potential destinations for the VM to be migrated.

A. *VVMM-U*

VVMM-U (Uniform) involves the vehicle that needs to migrate (source) following this algorithm to select a destination. In fact, this migration scheme does not utilize any intelligence and is considered the benchmark to compare the enhancements presented in VVMM-LW and VVMM-MA.

B. *VVMM-LW*

VVMM-LW (Least Workload) involves the vehicle (source) that needs to shift its workload following this algorithm to select a destination (dest). The vehicle initially searches for the vehicle(s) in same street with sufficient excess capacity to host the source workload. If at least one vehicle is found, the one with the *least workload* and *nearest* (i.e., *search criteria*) is selected out of the set of nearest vehicles. If the destination found will need to migrate the VM within τ seconds from now, the migration is resorted to RU. Otherwise, the migration is said to be successful.

If the vehicle could not find any potential destination in the vicinity, the VVMM is resorted to the RU, and the RU searches entire grid for vehicles with sufficient excess capacity to host the source workload. The RU uses the same search criteria, and aims at selecting the vehicle with the least workload and preferably the nearest one among those with the least workloads. If the VM is to be migrated within τ seconds, the VM is temporarily hosted in the RU and migration is returned to be unsuccessful. Otherwise, the VM is shifted to its new destination successfully.

This migration scheme utilizes further intelligence when selecting destination candidates, taking into account the candidate having least load. However, if the selected candidate will need to migrate within τ seconds from the migration decision, then the migration will fail. This scheme does not proactively avoid dropped migrations due to the selected candidate having to migrate within the next τ seconds. This flaw is what motivates the final scheme VVMM-MA, presented next.

C. *VVMM-MA*

VVMM-MA (Mobility Aware) involves the vehicle that needs to migrate (source) following this algorithm to select a destination (dest) and attempting to migrate. VVMM-MA sets the search criteria as follows: *Search for vehicle(s) in same street with τ seconds guarantee.*

VVMM-MA is the most intelligent of the three VVMM schemes, whereby the candidate is initially only considered if it will not be migrating for another τ seconds, due to its departure from the grid in the nearby future. Thus, the ‘mobility awareness’ incorporates prediction of the vehicle’s future path. If the candidate is considered to be viable based on the mobility awareness, only then is it evaluated based on least workload, and finally proximity.

IV. CASE STUDY

In order to test the three migration schemes, a simulation environment was built on MATLAB. The details of the initialization process prior to applying the scheme are described in what follows.

1) Model Topology:

The area used to model vehicular mobility is a generic grid, composed of nine one-way street intersections. This is shown in Fig. 2. The vertices in Fig. 2 represent the intersections and the transitions represent the streets. Note that the grid is made up of three vertical streets and three horizontal streets. For instance, when the links between '1-2' and '2-3' are separate, they represent two portions of the same street. All transitions from one intersection to another are of equal length: 0.5Km. The number of vehicles modeled is 500. The potential entry point (start) $S \in \{1, 2, 6, 7, 9\}$ and the potential exit point (destination) $D \in \{1, 3, 4, 8, 9\}$. The trajectory of the i^{th} vehicle can be represented in the form $T_N = \{N_1, N_2, \ldots N_k\}$, where k is the number of intersections (N_i) along the path. Another representation is $T_L = \{L_1, L_2, \ldots L_{k-1}\}$, where L_i is the segment (or leg) of the path, from one intersection to the next.

For the i^{th} vehicle V_i, the rules are:

1. $N_1 = S_i$, $N_k = D_i$, $S_i \neq D_i$. Vehicles may *not* start and end at the same point.
2. Vehicles may revisit intersections.
3. $L_i \neq L_j \; \forall \, i, j, i \neq j$. Vehicles may *not* revisit the same segment multiple times.

Following these rules results in a total of 39 possible trajectories. In some cases there are multiple valid paths for the same start and destination pair.

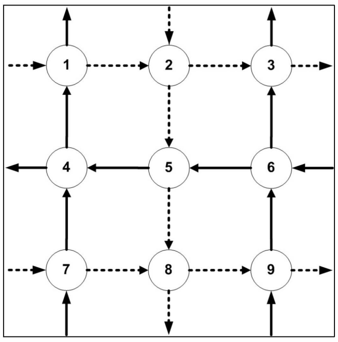

Fig. 2. Grid Layout

2) Trajectory Setup:

A vehicle trajectory is determined in the following process:

i. Select start, uniformly random.

ii. Select destination, uniformly random.

iii. Select path (from possible multiple alternative paths), uniformly random.

iv. Select start time and speeds.

v. Determine migration time.

In order to model the effect of congestion, vehicles enter the grid at various times and are restricted to a maximum possible speed based on the degree of congestion. Vehicles move at a speed of 10, 20, 30, 40 or 50Km/hr. Three levels of congestion are modeled: High, Medium and Low. The maximum speed is restricted to 30, 40 and 50Km/hr for high, medium and low congestions respectively.

The start time determined for the vehicles is based on a 'five-car-time-window'. This time-window is 10, 30 and 60 seconds for high, medium and low congestions respectively. For example, in a high congestion model, every 10 seconds, a total of five new cars enter the grid from random points as selected using the above process. These five new cars each select a start time from 1 to 10 seconds after the last window, uniformly random. So for example:

1. Window 1: Vehicles 1 to 5 would select start times between 1 and 10 seconds.
2. Window 2: Vehicles 6 to 10 would select start times between 11 and 20 seconds…

Once all the vehicles trajectories, speeds and start times are determined, the total duration of the model is known. With an observation interval of 300 seconds (far less than the total duration), centered at the middle of the total duration, there is a continuous car inflow and outflow throughout the observation interval. No vehicle re-enters the grid after leaving.

Based on the vehicle's assigned trajectory, its speed is realistically selected along the path. Speeds are incremented or decremented by 10 km/hr steps, over 1-second intervals. Furthermore, gradual deceleration is modeled prior to reaching an intersection. Along the path, the car randomly selects a speed and progresses to that speed in 10 km/hr increments/decrements every second. Once the selected speed is reached, the random selection of a successive speed is repeated, and the vehicle either maintains its speed or proceeds with a new speed value. This introduces dynamicity to the mobility model.

After the vehicles are initialized, the migration time of each vehicle is determined as 25 seconds prior to its departure from the grid. This is based on the assumption that the maximum time taken to complete a migration is τ = 25 seconds.

3) Vehicle Workload Initialization

There are three VM sizes or types:

1. Light = 1 unit
2. Medium = 2 units
3. Heavy = 4 units

Vehicle capacities are all identical = 8 units

All vehicles are assumed to have an identical capacity (8 units). The vehicles are assumed to initially host VMs of one type only, but can host multiple types if selected as a destination for a migration.

As shown in Fig. 3, first the type of VM is selected, then the number of units. So for example, a vehicle selects type: Heavy. Then the number of Heavy VMs that it can host are 0, 1 or 2. Using Fig. 3, it would mean the actual workload on the vehicle can take a value 0, 4 or 8 units. The same process extends for all vehicles' initial workload.

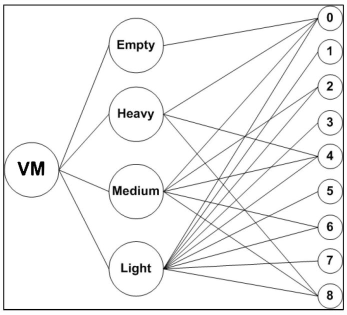

Fig. 3. VM Initialization Steps

The following section discusses the various runs utilizing the initializations described.

V. PERFORMANCE EVALUATION

The model as described in Section IV was run on MATLAB for multiple scenarios. It is important to note that each scenario involves its own initialization process, re-run for each seed. VVMM-U, VVMM-LW and VVMM-MA have been tested under light, medium and heavy congestion scenarios where each scenario has been generated by 30 different seeds.

Each scenario is assumed to last 300 seconds, and the following metrics are used to evaluate the proposed schemes:

A_{PD} = Average Percentage of Dropped VMs is formulated in Eq. 1 as follows:

$$A_{PD} = \frac{\sum_{i=1}^{n} D_i}{\sum_{i=1}^{n} D_i + \sum_{i=1}^{m} S_i} \quad (1)$$

where A_{PD} is the total percentage dropped D_i is the i^{th} dropped migration value (VM units) S_i is the i^{th} successful migration value (VM units) n is the total number of instances of a dropped migration m is the total number of instances of a successful migration.

The value of A_{PD} is calculated for each of the three schemes, for each congestion level and is presented in Fig. 3. Note that the percentage improvement over VVMM-U, in A_{PD}, for VVMM-LW and for VVMM-MA is shown in Fig. 4. Indeed, this metric is a conservative indicator of the scheme performance since dropped migrations do not necessarily denote failed migrations as stated earlier in the paper.

A_{UF} = the Average Utilization Fairness (using Jain's Fairness Index) [14] as shown in Eq. 2

$$A_{UF} = \frac{\sum_{t=1}^{300} \frac{\left(\sum_{i=1}^{n} WL_{ti}\right)^2}{n \cdot \sum_{i=1}^{n} {WL_{ti}}^2}}{T} \quad (2)$$

UF is the average fairness of the capacity utilization of the vehicles (averaged over the whole observation window) where WL_{ti} is the i^{th} car's workload (VM units) at time t, n is the total number of cars in the grid at time t, t is the time instant (seconds), and T denotes the total simulation duration.

Upon analyzing the average percentage drop and the improvement from one scheme to another in Fig. 4 and Fig. 5, respectively, there are several trends that can be observed.

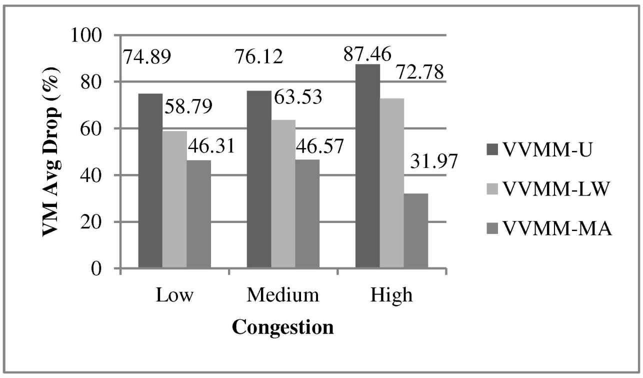

Fig. 4. Average Percentage of Dropped VMs over all Iterations

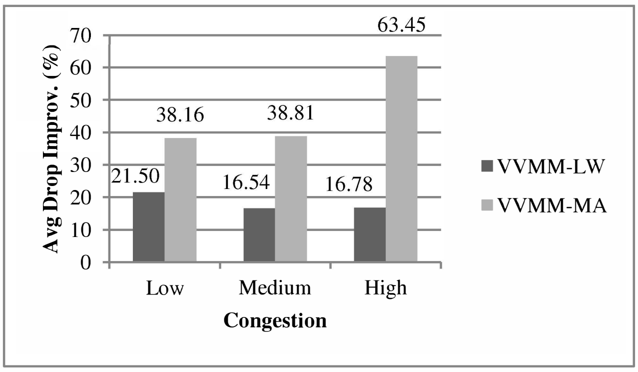

Fig. 5. Percentage Improvement Compared to VVMM-U over all Iterations

Firstly, there is a clear and significant improvement in performance of the VVMM-LW scheme and the VVMM-MA scheme over the VVMM-U scheme. The improvement is also, logically, greater for the VVMM-MA scheme than the VVMM-LW. This is logical due to considering the predicted future path of a potential new host of a VM prior to attempting a migration. Therefore, the VM can still be hosted in the grid without being stored in the RU. Furthermore, the impact of congestion on the performance of the scheme is also noteworthy. There is slight change between low and medium congestions under VVM-U and WWM-LW. However, the behavior of these two schemes under high congestion confirms the need for an intelligent VVMM scheme. In case, there is an increase in the number of migrations over the observation window, more candidates are present in the grid to receive the migrated VM. Selecting the best candidate among that set requires consideration of their current workload, as well as future location which will denote if the migrated VM will still be within the boundaries of the grid. Therefore, VVMM-MA scheme performs best, in the most congested scenario.

The average fairness, A_{UF}, has also been calculated for the three schemes, at the three congestions levels. This is presented in Fig. 6. Instantaneous changes in U_F under medium congestion are also shown in Fig. 7 as a function of time.

Based on the illustrations in these figures, it can be concluded that the fairness in vehicle capacity utilization significantly improves, especially for VVMM-MA, taking the average fairness from 0.5 to as high as 0.92. This is a highly favorable result which indicates that the VVMM-MA scheme also improves the general distribution of workloads over the grid. Furthermore, the fairness introduced by VVMM-MA is more stable when compared to VVMM-U and VVMM-LW throughout the time; hence no vehicle will be penalized by additional workload for its speed or location.

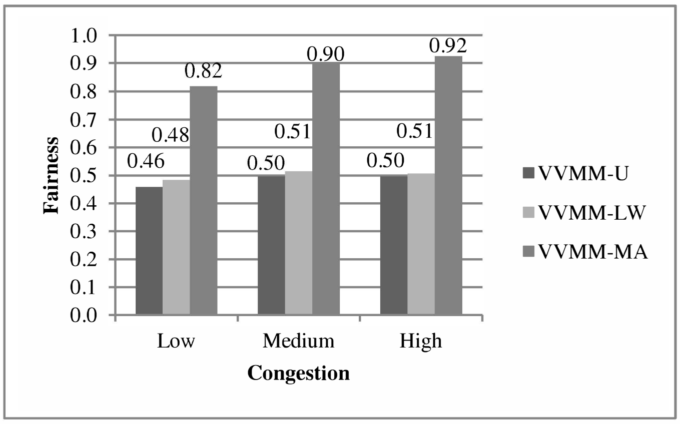

Fig. 6. Average Utilization Fairness over all Iterations

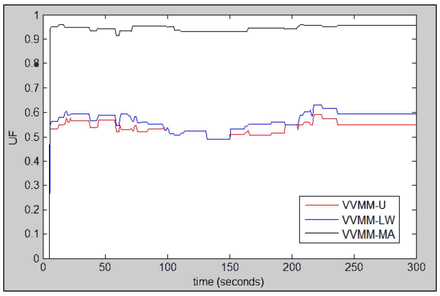

Fig. 7. Average Utilization Fairness vs. Time, for 1 Iteration

VI. Concluding Remarks and Ongoing Work

The future Internet is considered to be dominated by virtualized cloud services. Adopting cloud-based services in Vehicular Ad Hoc Networks (VANETs) has recently been discussed in the context of a cloud system, composed of roadside units and/or access to the cloud services via roadside units. In this paper, a study is presented regarding the virtual machine migration problem in a vehicular cloud system, where certain applications are hosted by, and migrated between, the connected vehicles. To this end, Vehicular Virtual Machine Migration (VVMM) is proposed, which can operate in three modes, namely uniform VVMM (VVMM-U), VVMM aiming at the vehicle with least workload (VVMM-LW) and VVMM aiming at the least loaded vehicle with a prediction of its prospective location, i.e., mobility-awareness (VVMM-MA). The proposed methods are evaluated via MATLAB simulations. Simulation results confirm that VVMM-LW can enhance the performance of VVMM-U by up to 21.5% in terms of successful VM migrations. Furthermore, incorporating mobility-awareness with vehicle workload-awareness, the improvement over VVMM-U can be increased by up to 63.5%. Moreover, both VVMM-LW and VVMM-MA have been shown to introduce enhanced fairness to the mobile hosts (i.e., vehicles) in terms of workload. In conclusion it was shown that VVMM-MA exhibits the highest performance of all three schemes.

This study is currently being extended to include communication challenges in the *vehicular cloud data center,* where communication with the roadside unit, and between the vehicles, is being modeled as a wireless intra-data-center network. Furthermore assumptions such as migration time limit and grid layout are being justified by the ongoing work.

References

[1] K. Hwang, G. Fox and J. J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Waltham, MA: Morgan Kaufmann, 2011.

[2] M. Whaiduzzamana, M. Sookhaka, A. Gania and R. Buyyab, "A Survey on Vehicular Cloud Computing," Journal of Network and Computer Applications, 2013.

[3] R. Hussain, J. Son, H. Eun, S. Kim and H. Oh, "Rethinking Vehicular Communications: Merging VANET with Cloud Computing," in Proc. of the 4th IEEE International Conference on Cloud Computing Technology and Science, Taipei, Taiwan, 2012.

[4] S. Arif, S. Olariu, J. Wang, G. Yan, W. Yang and I. Khalil, "Datacenter at the Airport: Reasoning about Time-Dependent Parking Lot Occupancy," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 11, pp. 2067-2080, 2012.

[5] H. Dinh, C. Lee, D. Niyato and P. Wang, "A Survey of Mobile Cloud Computing: Architecture, Applications and Approaches," Wireless Communications and Mobile Computing, vol. 13, no. 18, pp. 1587-1611, 2013.

[6] H. Mousannif, I. Khalil and H. al Moatassime, "Cooperation as a Service in VANETs," Journal of Universal Computer Science, vol. 17, no. 8, pp. 1202-1218, 2011.

[7] R. Hussain, F. Abbas, J. Son and H. Oh, "TIaaS: Secure Cloud-assisted Traffic Information Dissemination in Vehicular Ad Hoc Networks," in Proc. of the 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Delft, Netherlands, 2013.

[8] R. Boutaba, Q. Zhang and M. F. Zhani, "Virtual Machine Migration in Cloud Computing Environments: Benefits, Challenges, and Approaches," in Communication Infrastructures for Cloud Computing. H. Mouftah and B. Kantarci (Eds.). IGI-Global, USA. pp. 383-408, September, 2013

[9] M. Mishra, A. Das, P. Kulkarni, A. Sahoo, "Dynamic resource management using virtual machine migrations," IEEE Communications Magazine, vol.50, no.9, pp.34,40, September 2012.

[10] M. Eltoweissy, S. Olariu and M. Younis, "Towards Autonomous Vehicular Clouds," Ad Hoc Networks, Springer, 2010, pp. 1-16.

[11] S. Olariu, T. Hristov and G. Yan, "The Next Paradigm Shift: From Vehicular Networks to Vehicular Clouds," in Mobile Ad Hoc Networking, Hoboken, NJ, John Wiley & Sons, Inc., 2013.

[12] S. Olariu, I. Khalil and M. Abuelela, "Taking VANET to the Clouds," International Journal of Pervasive Computing and Communications, vol. 7, no. 1, pp. 7-21, 2011.

[13] R. Yu, Y. Zhang, S. Gjessing, W. Xia and K. Yang, "Toward Cloud-Based Vehicular Networks with Efficient Resource Management," IEEE Network, vol. 27, no. 5, pp. 48-55, 2013.

[14] R. Jain, C. Dah-Ming and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," Eastern Research Laboratory, Digital Equipment Corp., 1984.