A Framework for Vehicular Cloud Computing

Hossam S. Hassanein*, Sherin Abdelhamid*, and Khalid Elgazzar†
*School of Computing, Queen's University, Kingston, ON, Canada
{hossam, sherin}@cs.queensu.ca
†School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
elgazzar@cs.cmu.edu

***Abstract*—With the abundant on-board resources in smart vehicles, they have been considered major candidates for providing ubiquitous services. Processing and storage resources provided through on-board units along with a variety of sensing and communication capabilities can turn smart vehicles into potential computing engines that can be pooled to serve as powerful vehicular clouds (VCs). Computing tasks can be assigned to dynamically-formed VCs supporting a wide scope of ICT applications. In this paper, we introduce a framework for vehicular cloud computing. This framework aims at providing an inclusive architecture to manage the whole operation of a VC through different layers/types of controllers. Preliminary results are presented showing some of the benefits of utilizing vehicular cloud computing.**

I. Introduction

Smart vehicles have been considered major players in service provisioning due to the plethora of their on-board resources along with the ubiquitous vehicular availability [1]. A typical smart vehicle is equipped with a variety of sensors, communication capabilities, and abundant computing resources. A vital component of each smart vehicle is the On-Board Unit (OBU), also known as the in-vehicle PC, that works as the service/application hub and interface to the driver. Currently, some OBUs are becoming as powerful as personal computers in terms of storage and processing capabilities [2]. Thanks to the rapid technological advances, it is anticipated that in-vehicle processing capabilities will be considerably boosted, and storage capacity will reach multiples of Terabytes in the future.

Motivated by such abundance of in-vehicle computing resources, we remark that the resources of smart vehicles can be pooled and utilized as a cloud computing environment known as a *vehicular cloud* (VC). Vehicles on roads or at parking lots can be reached for handling computing tasks similar to traditional clouds. VCs present more advantages compared to traditional fixed clouds such as their ability to be mobile, autonomous, survivor in emergencies, and closer to the users [3].

Many ICT applications can be foreseen through the utilization of smart vehicles as VCs. For example, the resources of smart vehicles at the parking lot of a company can be pooled and used by the company for computing purposes instead of outsourcing a computing infrastructure. The employees who own the vehicles participating in the VC can be rewarded for such an access to their vehicular resources, bringing benefits to both the company and employees. The same perspective can be applied on vehicles parked at airports and left unutilized for many days. The storage resources of such vehicles can be pooled and tapped into transforming the airport parking into a dynamic data center with the aid of a scheduling mechanism that keeps track of the travel plans of the participating vehicle owners [4].

In this paper we introduce a framework for vehicular cloud computing. This framework aims at providing all the functionality needed to manage the process of accessing vehicular resources as a part of a VC. To handle such functionality, the framework encapsulates components of three different types. These components collaborate under an inclusive architecture to manage the VC operation.

II. Overview Of the Vehicular Cloud Framework

The proposed framework manages the whole process of vehicular cloud computing through a centralized architecture encompassing three types of components: 1) a cloud controller, 2) node controllers, and 3) cluster controllers. This architecture is depicted in Fig. 1 through a VC formed at a parking garage.

The ***cloud controller*** is the central entity that handles the main management functionality of the VC. It communicates with users/clients seeking resource access and with vehicles working as resource providers. It receives computing requests from users and assigns computing tasks to vehicles accordingly, handling any required control/data messages exchange. To handle such functionality, a cloud controller encompasses three underlying modules: a) a broker, b) a resource manager, and c) a task scheduler.

Our framework follows a client/server model where clients send resource requests to the cloud controller to reach computing resources managed by that controller. The *broker* is responsible for receiving client requests and negotiating their service resolution.

The *resource manager* is the most crucial entity in a VC. It is responsible for:

- monitoring, discovering, and predicting the availability of resources.
- deciding on the sufficiency of the available resources to match the requirements of a computing task.
- allocating resources in cooperation with the task scheduler.
- migrating on-going tasks from vehicles leaving the VC to other available vehicles.
- ensuring that recruitment requirements are met.

The *task scheduler* cooperates with the resource manager to create an access schedule based on the availability span of the candidate vehicles, the task temporal requirements, and the dependency requirements between the tasks.

The second type of components in a VC is the ***node controller***. Each vehicle interested to be a part of a VC has a

978-1-5090-0264-1/15/$31.00 ©2015 IEEE DOI 10.1109/ICCVE.2015.51

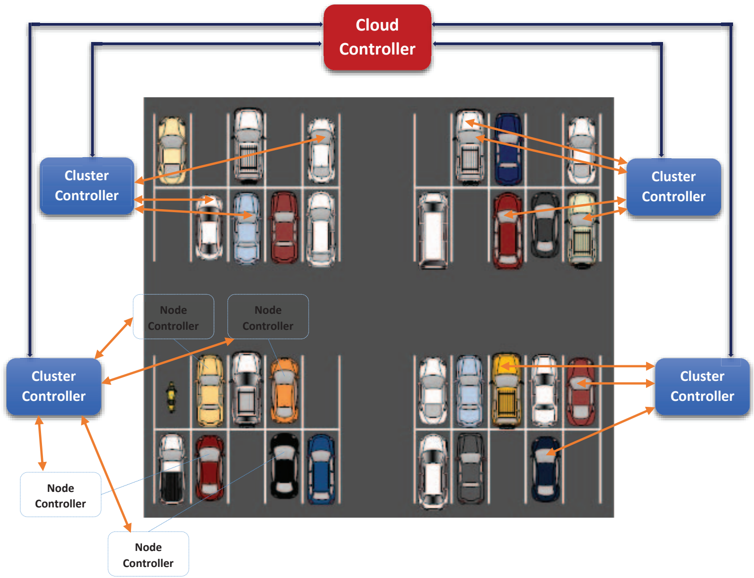

Fig. 1. The architecture of the proposed framework.

node controller which works as the interface between the on-board computing resources and the higher controlling layer. It works as a local resource controller reporting on the vehicle availability and controlling access to its resources.

The third controlling component of a VC is the ***cluster controller***. When a VC is big in size (i.e., the number of participating vehicles), vehicles can be grouped into clusters, each of which is managed by a cluster controller that connects the node controllers and the cloud controller. The motivation behind having this mid-layer of control is to reduce the load on the cloud controller through offloading a part of its management load to the cluster controllers.

III. A Use Case and Preliminary Results

We assess the introduced framework through a use case elicited from the work presented in [5], where a framework is proposed for cloud-assisted mobile service provisioning. The framework in [5] supports resource-constrained mobile service providers via computation offloading to a cloud. We consider a use case where a resource-intensive image processing task has been initiated on a mobile device. Computing assistance has been sought from different resource providers including a vehicular cloud connected to the mobile device over a one-hop high speed WiFi connection. The prototype is implemented in Python. The service initiating the processing task is deployed on a Nexus 6 (Quad-core 2.7 GHz and Android OS v5.1), connected to a WiFi network and is LTE-enabled.

Preliminary results are shown in Fig. 2. We measure the task response time over different resource providers. The results show the benefit, in terms of the response time, gained through offloading computation to a vehicular cloud compared to other resource providers. The low latency is attributed to the close proximity of the VC to the client (i.e., the mobile device) compared to the other computing resources.

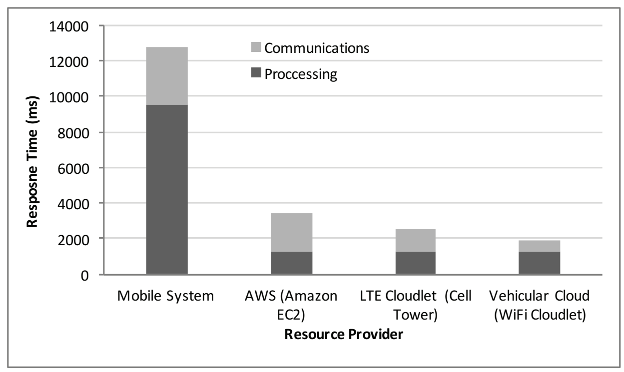

Fig. 2. The response time of an image processing application on different computing resources.

IV. Conclusions

In this paper, we introduced a framework for vehicular cloud computing targeting providing an inclusive architecture to manage the whole process of forming and accessing vehicular clouds. The framework consists of three control layers with a centralized architecture. We discussed the functionality of these layers and their underlying modules. In addition, we discussed a use case of utilizing vehicular clouds and presented preliminary results showing potential benefits of such utilization.

References

[1] S. Abdelhamid, H. S. Hassanein, and G. Takahara, "Vehicle as a resource (VaaR)," *IEEE Network Magazine*, vol. 29, no. 1, pp. 12–17, 2015.

[2] "SINTRONES In-Vehicle Computing," http://www.sintrones.com/applications/in-vehicle_computer.php, accessed: 24-08-2015.

[3] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *Journal of Network and Computer Applications*, vol. 40, pp. 325–344, 2014.

[4] S. Olariu, M. Eltoweissy, and M. Younis, "Towards autonomous vehicular clouds," *ICST Transactions on Mobile Communications and Applications*, vol. 11, no. 7-9, pp. 1–11, 2011.

[5] K. Elgazzar, P. Martin, and H. S. Hassanein, "Cloud-assisted computation offloading to support mobile services," *IEEE Transactions on Cloud Computing*, 2014.