ITS-Cloud: Cloud Computing for Intelligent Transportation System

Salim Bitam
LESIA Laboratory
Department of Computer Science
University of Biskra, ALGERIA
salim.bitam@laposte.net

Abdelhamid Mellouk
Image, Signal and Intelligent Systems Laboratory-LISSI
Department of Networks and Telecoms, IUT C/V
University of Paris Est Creteil (UPEC), FRANCE
mellouk@u-pec.fr

***Abstract*—Cloud computing is known as services delivery such as shared resources, platforms, software and data, in the interest of end-users. They are located in distributed datacenters over a network such as the Internet. In this paper a new cloud computing model called ITS-Cloud applied to the Intelligent Transportation Systems (ITS) is proposed to improve transport outcomes such as road safety, transport productivity, travel reliability, informed travel choices, environment protection, and traffic resilience. It consists of two sub-models: the statistic and the dynamic cloud sub-models. In the former, vehicles benefit of the conventional cloud advantages however; the dynamic one which is a temporary cloud is formed by the vehicles themselves which represent the cloud datacenters. To validate our proposal, a simulation study is performed to deal with the load balancing as a NP-Complete problem. The reached results are obtained using Bees Life Algorithm (BLA) applied to ITS-Cloud and compared with those reached by (BLA) applied only to the conventional Cloud.**

Keywords-cloud computing; intelligent transport system; load balancing; bees life algorithm;

I. INTRODUCTION

Intelligent Transport Systems (ITS) are defined as a set of advanced applications aim at applying intelligent technologies of information and communication to provide innovative services of traffic management and transport. They help to avoid traffic congestions and accidents which are increased with the population growth and caused several undesirable effects like long travel time, air pollution, and fuel consumption. Many efforts have been devoted by ITS organizations in order to find ubiquitous solutions by developing vehicular networking and traffic communications. For example, the US Federal Communications Commission (FCC) has allocated 75MHz of spectrum in the 5.850 to 5.925 GHz band for the exclusive use of Dedicated Short Range Communications (DSRC). In addition, some approved amendments have been dedicated to ITS technology such as Wireless Access in Vehicular Environments (WAVE IEEE 802.11p), Worldwide Interoperability for Microwave Access (WiMAX IEEE 802.16) etc.

All these efforts help and allow the ITS users (vehicle drivers) to reach their digital requirements and services but with an expensive cost. In other words, to benefit a service, the end-used should buy the required software, the suitable platform and the appropriate hardware which can be rented and used for one time.

For this reason, we propose to take advantage of the cloud computing to serve drivers and ITS users and then, only one rule is applied which is pay as you go.

Cloud computing is defined as the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility (like the electricity grid) over a network (typically the Internet) [1]. It is worth mentioning that the cloud computing can be divided into two types; the public cloud and the private one. In the public type, services are sold to the public Internet customers. Here, Amazon Elastic Compute Cloud (EC2) [2], Google App Engine [3] are large public cloud providers. The private cloud is known as an owner networks and/or datacenters that provide reserved services to specific end-users or customers. In the cloud computing, the end-users are allowed to execute remotely their applications. This technology is called Software-as-a-Service (SaaS) [4]. Another offered cloud services known as Infrastructure-as-a-Service (IaaS) refers to physical computing resources such as computers, virtual machines, etc. which are made available to the end-users. Along with SaaS and IaaS, there is a third service model of cloud computing called Platform-as-a-Service (PaaS). It helps the customer to develop its own software using tools and libraries provided by the cloud.

In the transportation context, the cloud computing is faced to many challenges such as the production of unified architectural framework that defines a layered model to facilitate its structure and make easy their functionalities. In this paper, we propose a novel architectural and functional model called ITS-Cloud that put the cloud available to vehicular systems. It is based on two sub-models where the conventional cloud services are offered to vehicles in the first sub-model such as software as a service, platform as a service, and hardware as a service. However, the second sub-model consists of vehicles together to form a new temporary cloud. It is designed to expand the permanent cloud in order to increase the capacity of processing, computing and storing in the ITS-

978-1-4673-0921-9/12/$31.00 ©2012 IEEE

Cloud using the vehicles themselves. The interaction between the two sub-models is also suggested to ensure the integration of the entire model. Also, the load balancing problem is studied in this paper and dealt with Bees Life Algorithm (BLA) [5] applied to ITS-Cloud. It is used to validate our proposal after comparison with (BLA) applied only to the conventional cloud.

The rest of the paper is organized as follows. The next section gives the state of the art of the vehicular cloud computing. The description of the proposed ITS-Cloud model is presented in section 3. This proposal is validated in section 4 by solving the load balancing as NP-Complete problem using the Bees Life Algorithm. The experimental results, comparisons and discussions are presented in section 5. Finally, we conclude the paper and we list some future researches.

II. RELATED WORK

In the last years, the cloud computing for vehicular systems has been studied in few research activities because of the novelty of this research discipline. Nevertheless, the following approaches can be cited as related works. First, the researcher can find the work proposed in [6] in which authors proposed a novel V-Cloud architecture included vehicular cyber-physical system (VCPS) as major contribution as well as two V-Cloud layers namely vehicle-to-vehicle network (V2V) layer and vehicle-to-infrastructure network (V2I) layer. They specified a system based on vehicle as cyber-physical system enabled cloud computing entity. In addition, smart phone embedded sensors are added in use along with automobile's health monitoring system.

[7] investigated the extension of the Platform-as-a-Service (PaaS) level of the cloud supporting a large collection of mobile devices to enrich the needed capabilities of the cloud. They simulated a millions of cars on the road per country. In [8], the authors used the cloud services in addition to a schedule calculation system and on-board communication devices installed in the vehicles. Simulations and field tests conducted in this work are dealt with three different-size cities in Japan to validate the developed system. As results, the authors confirmed that the developed calculation algorithm works well as designed and the on-demand bus service using the cloud technology can be practical and cost-effective after travelers' feedbacks.

The authors of [9] developed an intelligent disaster management systems focus on transportation systems by exploiting ICT developments (VANET, ITS, Mobile and Cloud computing). They proposed an intelligent system aiming to collect data from several locations, comprising the source of incident. Moreover, it communicates the necessary information to vehicles in real-time. The system effectiveness has been evaluated by modeling the impact of a disaster on a real city transport environment and comparing it with the case where the proposed disaster management system was in place. Therefore, the authors reported great benefits in terms of improved and balanced traffic flow and smooth evacuation.

The purpose of [10] is to propose a new system called Vehicular Cloud (VC) which profits from underutilized vehicles resources by the aggregation of VC resources. The proposed system contained three levels: the Node Controller, the Cluster Controller and the Cloud Controller. The simulation study aimed the load balancing with delay and retransmission packet rate.

III. ITS-CLOUD

In this section, the proposed Cloud computing model for Intelligent Transportation System is described. First of all, an illustration of its three horizontal layers; from the end-user to the cloud level is presented. After that, structures, functions and interactions between these layers are explained. On the other side, the two parallel sub-models of the ITS-Cloud will be presented namely the permanent and the temporary vehicular clouds. Finally, the main cloud applications provided by ITS-Cloud are presented.

A. ITS-Cloud architecture

We propose a new vehicular cloud architecture called ITS-Cloud that gathers, on the one hand, the advantage of the conventional cloud which is permanent and sustainable. On the other hand, it uses the utilities of the computing resources of the intelligent transportation devices which are temporarily presented in the area such as vehicles, passengers, and transportation servers (taxis and bus companies). Therefore, this architecture is based on two aspects static and dynamic respectively.

ITS-Cloud is seen as three abstract layers as mentioned in "Fig. 1" and explained below:

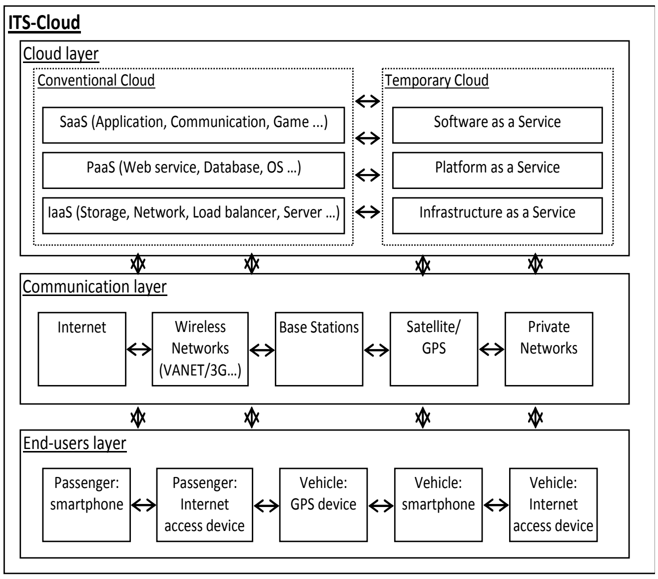


Figure 1. ITS-Cloud architecture

1) End-users layer

The end-users represent the basic entity of this layer and are characterized by a certain level of mobility. The end-user could be either a customer or an ITS-Cloud datacenter. In the former case, the entity represented by a vehicle driver or passenger, can establish its service request through its communication and computing device like Smartphone, Laptop, Onboard computer, GPS, Wi-Fi, Internet access point etc. To this end, the end-user receives its service response through the upper layers.

Furthermore, the end-user can play the cloud datacenter role. In other words, it is considered as computing entity which helps the others to perform their service requests against a service price. As a matter of fact, the end-user belongs to the cloud layer as temporary cloud because of its impermanent presence and function.

2) Communication layer

The purpose of this layer is to ensure the connection between the end-user as customer and the datacenter as cloud server (datacenter). This layer consists of several communication devices and schemes namely the Internet, wireless networks in its various formats (VANETs, WSN, 3G Networks etc), cellular base station, road base station, satellite, private networks etc. At this level, all connection technologies are defined and fixed according to the used communication device such as the physical layer, the data link layer, the media access control, routing protocols etc. The technology choice depends also, on the type of the end-user and the datacenter. For example, if the end-user is vehicle, the LLC/MAC technology may follow the IEEE 802.11p standard.

3) Cloud layer

Cloud layer refers to the software delivered as services and the hardware located in the datacenters over the communication layer. Services made available to the customer, can be divided into three kinds of services: software (application, communication, email, entertainment etc.), platform (database, data warehouse, operating system, web services etc.) and infrastructure (storage devices, networks, load balancer tools, optimization process etc.).

It is worth noting that this layer can be divided into both static and dynamic cloud, as two parallel structures. The static cloud consists of fixed and conventional cloud servers however; the dynamic one is formed by the vehicles' which belong to the cloud in order to contribute to the computing process. They will be explained in the next paragraphs.

B. ITS-Cloud sub-models

In this section, we distinguish the ordinary cloud defined often without mobile entities and vehicles from our proposed ITS-Cloud that gathers the immobile and mobile entities. The motivation of the cloud extension shown in "Fig. 2", is to take advantage of their computing resources especially in the vehicular area namely car parking, vehicles in intersections and road etc. Moreover, vehicle drivers benefit from the allocation price of their resources to the cloud customers.

1) Conventional cloud sub-model

It is a set of interconnected and virtualized computers in parallel and distributed fashion which are seen as an entire computing system. These entities are interconnected through the Internet as public cloud and/or through private networks. The conventional cloud sub-model provides several computing functionalities and services. Software as a service is provided in the interest of the customer as applications run remotely from the datacenters.

Also, there are services provided to customers as platforms. They allow the deployment of applications with a reduced complexity and low cost when renting the underlying hardware or software and, when provisioning hosting capabilities. Mainly, the conventional cloud sub-model offers all of the required facilities, and it supports, built and delivers services entirely available from the Internet.

Infrastructure as a service is ensured by this sub-model. It provides the equipment used to maintain the computing and processing execution like the storage devices, hardware, servers and networking components. All these services allow only the payment of the service by the customer despite buying software, platform or infrastructure.

2) Temporary (vehicular) cloud sub-model

It consists of a set of interconnected vehicles and/or passengers placed initially in the vehicular area as customers, and then, they accept to allocate their computing resources to the other cloud customers. We call them ITS datacenters. These computing resources vary from simple Smartphone to an integral computer installed onboard of the vehicles. The interconnection between ITS datacenters is realized by the communication layer according to the vehicular networks like GPS, GSM, VANETs, 3G networks or others.

They will be integrated with the conventional cloud to form the whole cloud: ITS-Cloud. Therefore, any customer may be served by any datacenter in a virtualization fashion. Consequently, the temporary cloud provides the same services as the conventional within their resources capacities. It serves as storage device, as processing and computing devices, as virtual platform etc.

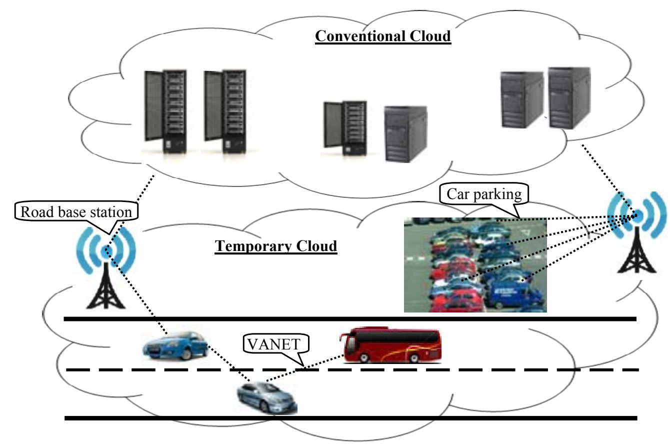


Figure 2. ITS-Cloud sub-models

C. Cloud applications provided by ITS-Cloud

ITS-Cloud can provide several services through the following main applications:

a) Correspondence application: ITS-Cloud offers the personal information and knowledge management such as email transmission and archiving, appointments and meeting organization, project management features, files sharing, chat etc.

b) Web services application: ITS-Cloud allows also services over the Internet that are made available from business's web server for cloud end-user.

c) Cloud backup: it is the cloud application which ensures storing data in the cloud datacenters with a regular integrity validation to facilitate its recoverability.

d) Business application: it concerns the possibility of e-commerce and its features via the cloud in the interest of the end-user with the option to pay as he goes.

e) Research application: ITS-Cloud can help the scientist in their experimental studies to overcome the computing complexity using this new technology.

f) Load balancing capability: it helps spreading a service workload between the cloud datacenters to reach several advantages like the scalability, avoiding the servers overloading, ensuring reliability of the computing in the case of starting multiple processing copies in different datacenters, minimising the response time etc. This ITS-Cloud application is studied, as example to validate our theoritical model in the following section.

IV. LOAD BALANCING PROBLEM IN ITS-CLOUD

In order to validate the benefit of the ITS-Cloud, a load balancing study is proposed in this paper. It is considered as an infrastructure service offered by the cloud to schedule the job tasks faced to the cloud datacenters. This study is performed in an ITS-Cloud and the results are compared with those obtained in a conventional cloud in which the ITS datacenters are not presented.

In the following sub-section, the studied problem is initiated. After that, the load balancing is solved using bee life algorithm (BLA) [5] as metaheuristic to overcome this NP-Complete problem. To do that, BLA is briefly explained.

A. Problem description

In the Cloud computing, load balancing is the assigning of the different tasks of jobs which are expected to be executed over datacenters. This distribution should ensure the minimum execution time of the overall tasks called also makespan. This problem can be formulated as follows.

We denote by:

$Jobs = \{J_1, J_2, ..., J_i, ..., J_n\}$, a set of *'n'* jobs to be scheduled.

Moreover, we consider for each job *'i'*:

$Job_iTasks = \{JTask_{i1}, JTask_{i2}, ..., JTask_{im}\}$, as a set of *'m'* task partitions of *'J_i'* disseminated among *'m'* cloud datacenters *(DCs)* in order to be executed. Consequently, each cloud datacenter can carry out a disjoint subset of the decomposed jobs set. Each datacenter *'DC_j'* runs its assigned tasks:

$DC_jTasks = \{JTask_{aj}, JTask_{bj}, ..., JTask_{rj}\}$

The overall disjoint and ordered subsets *'DC_jTasks'* are equal to the various jobs. For example, *'DC_j'* carries out:

$DC_jTasks = \{JTask_{3j}, JTask_{6j}, ..., JTask_{9j}\}$ which are tasks of jobs J_3, J_6 ..., and J_9 respectively.

Therefore, the total execution time of all job tasks (*'r'* tasks) assigned to DC_j would be:

$$Makespan(DC_jTasks) =$$

$$Max\ (JTask_{kj}.StartTime + JTask_{kj}.ExeTime)$$

Where $JTask_{kj}.StartTime$ is the time when job task *'k'* $JTask_{kj}$ starts executing on DC_i and $JTask_{kj}.ExeTime$ is the execution time of $JTask_{kj}$ at DC_j.

More formally, the problem can be described as finding:

$DCTasks = \{DC_1Tasks, DC_2Tasks, ..., DC_mTasks\}$

each $DC_jTasks = \{JTask_{aj}, JTask_{bj}, ..., JTask_{rj}\}$

with: $0 < r \leq n$, in order to reduce all: $Makespan(DC_jTasks)$

To evaluate the quality of the requested solution *(DCTasks)*, a fitness function is defined as follows (it is used to calculate the above *makespan*):

$Fitness(DCTasks) = \Sigma\ (Fitness(JTask_{ij}, DC_j))$ where $(1 \leq j \leq m)$ and $Fitness(JTask_{ij}, DC_j) = JTask_{ij}.TimeToExe$

where, $JTask_{ij}.TimeToExe$ is the execution time of a task of job *'i'* needs to run in DC_j.

B. Bees Life Algorithm

Bees Life Algorithm (BLA) presented in "Fig. 3", starts with the initialization of the bee population which contains *(N)* bees (individuals) chosen randomly from the search space. Next, it is the second step in which the population fitness is evaluated. Each population consists of one *(1)* queen which is the fittest bee, *'D'* bees as drones which are the fittest following bees and, *'W'* remaining bees as workers. Consequently, the sum of the different bee individuals *(1, D and W)* equal to the population size *(N)*. After that, BLA performs a cycle of a bee life. It includes reproduction and food foraging behaviors.

In marriage behavior or reproduction step, the queen starts mating with the drones in the space. The future offsprings are generated as result of the crossover and mutation operators. Afterwards, queen starts breeding *'N'* broods in step 4. Then, the evaluation of the brood fitness is performed in step 5.

Steps 6 to 8 sort the offsprings. If the fittest brood is fitter than the queen, it will be considered as the new queen for the next population. Furthermore, among the *'D'* fittest following broods and the previous drones, *'D'* best bees are selected as the next population drones. Moreover, the *'W'* next population workers are chosen as best bees among the *'W'* fittest remaining broods and the previous workers.

Steps 9 to 11 explain the food foraging step in BLA. In *'W'* regions of flowers, each worker among the last *'W'* workers searches food source. For each region, new bees are recruited to search the best food source between several sources in this region. These new bees consist of neighbor solutions in the search space. This step represents a local search aiming to take advantage of the neighborhood search. It is worth noting that BLA recruits more bees for the *'B'* best regions among *'W'* regions which are probably better than the remaining *'W-B'* regions. Thus, only the bee with the highest fitness will be selected to form the next bee population.

In step 12, the new population fitness is evaluated. Then, a new bees' life cycle is rerun if the stopping criterion is not satisfied, and so on. In this paper, *'ItMax'* as an iteration number is statically selected as stopping criterion.

1. Initialize population (*N* bees) at random

2. Evaluate and sort the population bees (fittest bee is the queen, *D* fittest following bees are drones, *W* fittest remaining bees are workers)

3. ***While*** not (stopping criteria) ***do*** (generation of a new population)

/* **reproduction phase** */

4. Generate ***N*** broods by **crossover** and **mutation**

5. Evaluate fitness of broods

6. If the fittest brood is fitter than the queen then replace the queen for the next generation

7. Choose *D* best bees among *D* fittest following broods and drones of current population (Forming next generation drones)

8. Choose *W* best bees among *W* fittest remaining broods and workers of current population (to ensure food foraging)

/* **food foraging phase** */

9. Search of food source in *W* regions by *W* workers

10. Recruit bees for each region for **neighborhood search** (more bees *(FBest)* for the best *B* regions and *(FOther)* for remaining regions)

11. Select the fittest bee from each region

12. Evaluate fitness of population (fittest bee is the queen, *D* fittest following bees are drones, *W* fittest remaining bees are workers)

13. ***End while***

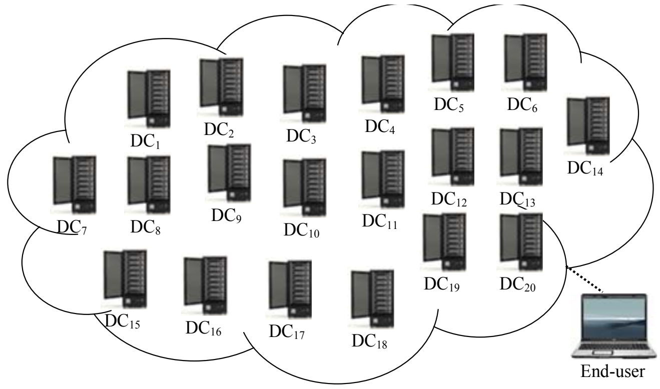
Figure 3. Bees Life Algorithm pseudo-code

V. EXPERIMENTAL STUDY

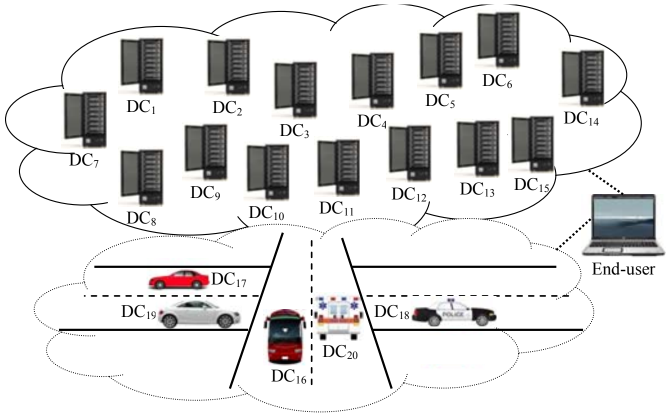
A. Experimental settings

In this study, a set of tests are carried out on the two kinds of a virtual clouds namely conventional cloud and ITS-Cloud. Each one contains 20 datacenters. The cloud datacenters are assumed to perform 5 jobs.

In the conventional cloud "Fig. 4(a)", all datacenters are stationary nevertheless; in ITS-Cloud "Fig. 4(b)", datacenters 16 to 20 are considered as ITS datacenters. They move according to the Manhattan mobility model [11]. Each job can be divided in tasks to be executed by a set of datacenters. Job 1 and 2 contain 5 tasks each one, job 3 includes 6 tasks, job 4 and 5 are divided into 7 tasks each one. All tasks have the same execution time which is equal to 8 milliseconds.



(a): Conventional cloud



(b): ITS-Cloud

Figure 4. Conventional cloud versus ITS-Cloud used in the experimental study

To deal with BLA, the user parameters are chosen as follows: the population size equal to 10 individuals with 4 drones and 5 workers. We fix the maximum number of iteration *'ItMax'* to 200. Also, we fix the crossover and mutation probabilities at 0.95 and 0.01 respectively, as practical values. The crossover operator used is the 1-point however; the permutation operator is used for the mutation.

During the foraging phase, 30 foragers *(FBest)* are proposed to exploit 3 regions *(B)* represented by the 3 best workers among 5 since they are the most promising regions. Nevertheless, only 15 foragers *(FOther)* are disseminated to the 2 remainder regions. Here, the cheapest insertion algorithm [12] is used as greedy approach for the neighborhood search.

Due to the absence of the benchmarks in this field, we propose in table 1 the following experimental data. It indicates the maximum execution time performed by a task in the datacenter. It is measured in milliseconds.

TABLE I. EXECUTION TIME OF ONE TASK INTO DATACENTER

DC_j	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
Exe.time	***10***	***11***	***12***	***10***	***12***	***10***	***10***	***10***	***12***	***12***

DC_j	11	12	13	14	15	16	17	18	19	20
Exe.time	***12***	***11***	***8***	***10***	***12***	***12***	***11***	***10***	***10***	***10***

In the ITS-Cloud, each BLA iteration is executed with the new ITS datacenters configuration. In other words, the ITS datacenters (DC_{16} to DC_{20}) can move and modify their position and then, the BLA reacts with this new situation.

B. Experimental results

Using BLA in both conventional and ITS clouds, the best solutions found (tasks repartition) in terms of makespan are listed in table 2.

TABLE II. TASKS REPARTITION USING BLA IN CONVENTIONAL CLOUD AND ITS-CLOUD

Cloud context	Task repartition between datacenters
Conventional Cloud	$Job_1Tasks=\{JTask_{1\ 1}, JTask_{1\ 4}, JTask_{1\ 6}, JTask_{1\ 7}, JTask_{1\ 8}\}$ $Job_2Tasks=\{JTask_{2\ 1}, JTask_{2\ 6}, JTask_{2\ 7}, JTask_{2\ 8}, JTask_{2\ 9}\}$ $Job_3Tasks=\{JTask_{3\ 1}, JTask_{3\ 6}, JTask_{3\ 7}, JTask_{3\ 8}, JTask_{3\ 13}, JTask_{3\ 14}\}$ $Job_4Tasks=\{JTask_{4\ 1}, JTask_{4\ 6}, JTask_{4\ 7}, JTask_{4\ 8}, JTask_{4\ 13}, JTask_{4\ 18}, JTask_{4\ 19}\}$ $Job_5Tasks=\{JTask_{5\ 1}, JTask_{5\ 6}, JTask_{5\ 7}, JTask_{5\ 8}, JTask_{5\ 13}, JTask_{5\ 18}, JTask_{5\ 20}\}$
ITS-Cloud	$Job_1Tasks=\{JTask_{1\ 1}, JTask_{1\ 4}, JTask_{1\ 6}, JTask_{1\ 7}, JTask_{1\ 13}\}$ $Job_2Tasks=\{JTask_{2\ 1}, JTask_{2\ 6}, JTask_{2\ 7}, JTask_{2\ 8}, JTask_{2\ 13}\}$ $Job_3Tasks=\{JTask_{3\ 1}, JTask_{3\ 6}, JTask_{3\ 7}, JTask_{3\ 8}, JTask_{3\ 13}, JTask_{3\ 14}\}$ $Job_4Tasks=\{JTask_{4\ 1}, JTask_{4\ 6}, JTask_{4\ 7}, JTask_{4\ 8}, JTask_{4\ 13}, JTask_{4\ 18}, JTask_{4\ 19}\}$ $Job_5Tasks=\{JTask_{5\ 1}, JTask_{5\ 6}, JTask_{5\ 7}, JTask_{5\ 8}, JTask_{5\ 13}, JTask_{5\ 18}, JTask_{5\ 20}\}$

The function below is applied to calculate the fitness:

$$Fitness(DCTasks) = \Sigma\ (Fitness(JTask_{ij}, DC_j)) = Fitness(JTask_{1\ 1}, DC_1)+Fitness(JTask_{1\ 5}, DC_5)+\ldots+ Fitness(JTask_{5\ 19}, DC_{19})+ Fitness(JTask_{5\ 20}, DC_{20})$$

where:

$$Fitness(JTask_{ij}, DC_j) = JTask_{ij}.TimeToExe$$

For conventional cloud, we have: $Fitness(DCTasks) = (50) + (52) + (58) + (68) + (68) = 296ms$

For ITS-Cloud, we have: $Fitness(DCTasks) = (48) + (48) + (58) + (68) + (68) = \mathbf{290ms}$

Table 2 shows that the best fitness is that obtained by BLA in the ITS-Cloud (***290ms***) compared with the fitness given by BLA in the conventional cloud (*296ms*). It represents the best execution time of the overall tasks of all jobs ($J_1 \ldots J_5$) performed by the different datacenters ($DC_1 \ldots DC_{20}$).

This ITS-Cloud improvement is due to the ITS datacenters movements which have an influence on the applied BLA operators especially, the neighborhood search approach. Consequently, there is more diversity of the selected solutions and then, a better configuration will have more chance to be chosen as optimal solution.

These results confirm the efficiency of the cloud computing based on both cloud datacenters and vehicular datacenters (ITS-Cloud) using BLA to solve load balancing problem.

VI. CONCLUSION

In this paper, cloud computing model for intelligent transportation system (ITS-Cloud) has been proposed. It is based on two sub-models: conventional cloud and vehicular cloud. In order to validate the benefit of this proposal, load balancing problem has been solved within and without ITS-Cloud using Bees Life Algorithm (BLA). The reached results showed that the makespan in the case on ITS-Cloud is reduced against the makespan obtained by BLA without ITS datacenters.

As perspective, we propose to conduct an experimental study of a realistic ITS-Cloud with load balancing problem as IaaS and other problems belong SaaS and PaaS. We propose also, to realize a practical study to fix a set of benchmarks in the cloud computing.

REFERENCES

[1] Wikipedia. http://en.wikipedia.org/wiki/Cloud_computing (accessed on March 28, 2012)

[2] Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2/ (accessed on March 28, 2012)

[3] Google App Engine. http://code.google.com/appengine/ (accessed on March 28, 2012).

[4] B. Furht, and A. Escalante, “Handbook of cloud computing,” Cloud computing fondamentals chapter writen by B. Furht, Springer, 2010.

[5] S. Bitam, and A. Mellouk, “Bee life-based multi constraints multicast routing optimization for vehicular Ad hoc networks,” Journal of Networks and Computer Applications (JNCA), Elsevier, 2012.

[6] H. Abid, L.T.T. Phuong, J. Wang, S. Lee, and S. Qaisar, “V-Cloud: Vehicular Cyber-Physical Systems and Cloud Computing,” 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL), Barcelona, Spain, 2011.

[7] D. Bernstein, N. Vidovic, and S. Modi, “A Cloud PAAS for High Scale, Function, and Velocity Mobile Applications - With Reference Application as the Fully Connected Car,” 5th International Conference on Systems and Networks Communications (ICSNC), Nice, France, 2010.

[8] K. Tsubouchi, H. Yamato, and K. Hiekata, “Innovative on-demand bus system in Japan,” Journal. IET. Intelligent Transport Systems, Volume 4, Issue 4, pp. 270-279, 2010.

[9] Z. Alazawi, S. Altowaijri, and R. Mehmood, “Intelligent disaster management system based on cloud-enabled vehicular networks,” 11th IEEE International Conference on ITS Telecommunications (ITST), St. Petersburg, Russia, 2011.

[10] S. Olariu, T. Hristov, and G. Yan, “The Next Paradigm Shift: From Vehicular Networks to Vehicular Clouds,” chapter of book: Developments in Wireless Network Prototyping, Design and deployment: Future Generations.Eds, IGI Global, 2012.

[11] F. Bai, N. Sadagopan, and A. Helmy, “The IMPORTANT framework for analyzing the impact of mobility on performance of routing protocols for adhoc networks,” AdHoc Networks Journal, volume 1, pp: 383-403, 2003.

[12] G.A.P. Kindervater, J.K. Lenstra and D.B. Shmoys, “The Parallel Complexity of TSP Heuristics,” Journal of Algorithms volume 10, pp. 249-270, 1989.