

Dynamic Resource Discovery Scheme for Vehicular Cloud Networks

Mahantesh G. Kambalimath

Electronics and Instrumentation Engineering Department
Basaveshwar Engineering College (Autonomous), Bagalkot - 587102, Karnataka, India
E-mail: mgkei@becbgk.edu

Mahabaleshwar S. Kakkasageri

Electronics and Communication Engineering Department
Basaveshwar Engineering College (Autonomous), Bagalkot - 587102, Karnataka, India
E-mail: mskec@becbgk.edu

Received: 13 September 2019; Accepted: 15 October 2019; Published: 08 December 2019

Abstract—To discover computing resources available for any application before they are allocated to requests dynamically on demand, developing effective mechanism for resource discovery in Vehicular Cloud Networks (VCN) is very important. Providing the services to the requested vehicle in time is a major concern in the VCN environment. Dynamic and intelligent resource discovery schemes are essential in VCN environment so that services are provided to the vehicles in time. Resource discovery is key characteristic of VCN. VCN requires intelligent algorithms for resource discovery. Creating a mechanism for resource management and search resources is the largest challenge in VCN. There is a need to consider for dynamic way to discover the resources in the VCN. The lack of intelligence in resource handling, less flexible for dynamic simultaneous requests, and low scalability are issues to be addressed for the resource discovery in VCN. In this paper we proposed dynamic resource discovery scheme in VCN. Proposed resource discovery scheme uses Honey Bee Optimization (HBO) technique integrated with static and mobile agents. Mobile agent collects the vehicular cloud information and static agent intelligently identifies the required resources by the vehicle. Dynamic discovery model will take into account different parameters influencing the task execution time to optimize subsequent schedule. To test the performance effectiveness of the scheme, proposed dynamic resource discovery scheme is compared with fixed time scheduling algorithm. The objective of the proposed scheme is to search the resources in VCN with a minimum delay. The simulation results of the proposed scheme is better than the existing scheme.

Index Terms—Vehicular Cloud Networks, Honey Bee Optimization, Queen Mobile Agent, Vehicle Manager Agent.

I. INTRODUCTION

High dynamic topology and predictable mobility are

the unique characteristic of Vehicular Ad hoc Networks (VANETs). VANETs have been quite a hot research area in the last few years. From both academic and industry VANETs attract so much attention [1]. Cloud that is set of several servers and computers everything, is hosted on it instead of placing data and application programs on a personal computer, it can be accessed through Internet. Dedicating the resources to the system requests is the challenge of cloud computing system [2].

A crucial question of distribution is the resource scheduling; it gives the resources in the system, user task execution efficiency, and the performance [3]. The largest challenge in vehicular mobile clouds are the creating a mechanism for management and search resources that does not depend on road side infrastructure. Such vehicular mobile clouds consequently require the dynamic and spontaneous creation of a cloud through the resource shared by vehicles [4].

To improve the transport safety, relieve traffic congestion, reduce air pollution, and enhance the comfort of driving, all components in Intelligent Transportation Systems (ITS) will be connected. The collection and storage of large amounts of traffic related data poses a significant challenge to the vision of all connected vehicles in ITS. The vehicles can share computation resources, storage resources, and band width resources in integrated cloud computing in vehicular networks. Vehicle utilize services for its own purpose and each vehicle access the cloud. The physical resources of vehicles are dynamically scheduled on demand through cooperation in the group. A vehicular cloud has much more resources compared to an individual vehicle hence overall resource utilization is significantly enhanced. Some vehicles directly apply for resources from other vehicles which are assigned as candidate cloud sites. Few vehicles are act as candidate cloud sites assigned by a vehicle. The vehicles become cloud sites, if the application is approved according to the vehicle demand will customize virtual machines (VMs) [5].

A novel way to offer non safety application with QoS

for VANETs are the Road Side Unit (RSU) clouds. Using Software Defined Networking (SDN) the traditional RSUs consist of specialized micro data centers and virtual machines. To migrate or replicate virtual services and reconfigure the data forwarding rules dynamically SDN offers flexibility [6]. The cloud computing services are hosted by the vehicles, to act as mobile cloud servers the vehicles that have sufficient resources. To discover the services and resources the vehicles in a VANET to search for mobile cloud servers, as mobile cloud servers have sufficient resources. RSUs are act as cloud directories with which mobile cloud servers register [7].

It is important to develop optimal policies for resource management in the cloud environment. Researchers strive to developing policies to achieve optimization of resource utilization for the vehicular cloud computing. Machine learning has a major role to play in these efforts [8]. Due to the heterogeneity of resources, variability of the workload and large scale of data centers resource management in cloud infrastructures are the most challenging problem in cloud computing. Efficient management of physical and virtual resources are required in cloud computing. By considering performance requirements of hosted applications and infrastructure costs efficient management of physical and virtual resources can be achieved [9]. The other way of looking vehicles as resource providers. In vehicle resources can assist on road safety and like other services in vehicular network managed by cloud environment. Directional behavior, clustered movement and high mobility are the distinct features of vehicular cloud networks, such environment requires new mechanism for resource management. Vehicular networks environment requires effectively to manage the dynamic resources, as it vary in availability and type [10].

Main objective of the research work is to discovery of resources in VCN environment with minimum response time. Dynamic resource discovery scheme is proposed and implemented. Dynamic discovery is a learning algorithm that is loosely based on linear regression. The proposed algorithm looks the previous performance metric to alter the next cycle time.

There are many generic algorithms analyzed, used and, discovered over the course to apply to this field. Linear regression, which correlates the impact of one variable (input) on the behavior of another (output) has been in existence from centuries. This statistical model is used in many fields to better predict and control system behavior. Linear correlation, however, is not optimal when it comes to complex systems. If there are multiple variables steering the system behavior, we need to apply non-linear models to predict the outcome of relationships between various parameters on input. Recurrent neural network is one such complex technique where dynamic behavior can be modeled. For the scope of this research, we will focus on linear models though.

II. RELATED WORKS

A fuzzy virtual machine workload prediction method

for cloud environment is proposed in [11]. An intuitive way of designing function blocks for intelligent systems is by fuzzy logic mathematically emulates human reasoning. To forecast the resource utilization of Virtual Machine (VM) in cloud environments predication methods such linear regression and neural networks have been used and to estimate required resources and size for each VM. To execute allocated work load resource utilization is required. To control resource utilization, they apply a neural network technique along with defined fuzzy logic.

Fuzzy logic methods are introduced to discover the resources in cloud systems. Neural networks is an another technique to discover the computing resources in cloud computing environment. Combination fuzzy logic and neural networks technique are also used to discover the resources in large systems. Neural network techniques are based on the learning algorithm by providing the set of examples. Our proposed scheme is based on the loosely linear regression. Proposed scheme alter the cycle time based on previous performance metric.

Direction aware resource discovery service in large scale grid and cloud computing is proposed in [12]. Based on an unstructured overlay the proposed model consider the decentralized resource discovery. Without global knowledge about sharing resource information the major challenge is to locate desired resources. Network overhead may incur in the scheme as more nodes are involved in the resource discovery. Route a query over the network to conduct resource discovery, it considers the unstructured information system. If there are no local resources satisfying the requirement, it quires local information for the required resource and the node forwards the request to other neighbors. The characteristic attributes, such as the system load, memory capacity, etc are composed of each resource. Resource management in cloud computing is presented in [13]. Cloud resource management process is very complex in nature. Cloud resource management process is divided in to different segments. Resource management in cloud computing as a sequential process is described. It is concluded that efficient cloud resource management should meet criteria like efficient utilization of resources, cost reduction, energy and power. A cluster based vehicular cloud architecture with learning based resource management is proposed in [14]. To group vehicles and to provide resources cooperatively the proposed model makes use of clustering technique. It uses the fuzzy logic in the cluster head selection procedure. They also use q-learning technique to improve the resource management. Agent based resource discovery in cloud computing using bloom filters is proposed in [15]. The proposed model consists of bloom filter. Bloom filters are used to store resource information. The model also consists of broker agent. The broker agent sends the information to the related database and broker agent receives the information from the bloom filter. The model consists of hash values. User make use of these hash values to send them to related broker agent. The bloom filters are used in the model for search and discovery information and

then achieved bloom filter is transferred. The amount data transmitted is reduced significantly by using this technique.

Load balancing approach for cloud computing is proposed in [16]. They proposed Throttled Modified Algorithm (TMA) for improving the response time of VMs on cloud computing to improve performance for end use. They compare the TMA with round robin and throttled algorithm for load balancing. The performance of cloud computing using TMA algorithm is improved compared to the round robin and throttled algorithm. Dynamic task scheduling in cloud computing based on the availability level of resources is proposed in [17]. The proposed algorithm allocates the incoming task on the best resource. Based on measuring the current situation of each resource with respect to its availability level the incoming task is scheduled. For scheduling the incoming task, the proposed scheme considers the different parameter such as processing power, cost etc. Algorithm is to allocate new coming tasks on the existed resources while some existed tasks are running.

Vehicular cloud network a new challenge for resource management based system is proposed in [18]. The relevant issues related to the concepts of resource management in the vehicular cloud are addressed. The resource management techniques for the vehicular cloud are also discussed. Towards efficient resource management in cloud computing, survey is described in [19]. The research in the resource management in cloud computing are analyzed. The most important criteria for evaluating the effectiveness of the approach are examined and they also examine the monitored values and supported application.

Load balancing algorithm based on estimating finish time of services in cloud computing is proposed [20]. To provide the resources work load efficiently the load balancing is an important technique. The techniques optimize resource utilization, maximizes throughput and minimizing response time. The technique also avoids overloading of any single resources. Novel load balancing algorithm is based on the method of estimating the end service time.

Distributed cloud resource management framework for high performance applications is proposed in [21]. The multiple cloud resources and large number of user application can be handled by the proposed system in an inter operable manner. By submitting a large number of real world high performance computing applications the proposed system is evaluated. Application of honey bee mating optimization algorithm is proposed in [22]. Many search algorithm is inspired by honey bee mating behavior. Honey bee mating behavior is considered as a typical swarm based approach for optimization in search of resources in many systems.

An on line fuzzy decision support system for resource management in cloud environment is proposed in [23]. On line Resource Management Decision Support System (ORMDSS) addresses both resource management optimization and task scheduling. ORMDSS contains a fuzzy method as well as neural networks approach. For

predicting VMs workload patterns fuzzy methods are used. They applied neural networks and fuzzy expert systems for predicting the VMs migration.

Tera Scaler ELBE (Elastic Load Balancing) algorithm of Prediction for balancing resource management in cloud computing is proposed in [24]. The processing capacity of back end server cluster with the applied load is dynamically by Tera Scaler ELB algorithm. Real time resource prediction engine for cloud management is proposed in [25]. Genetic algorithm based scheduler for computational grids is proposed in [26]. Highly scalable distributed resource management architectures for grid computing are presented. The scheduler schedules the job for grid computing is based on the genetic algorithm. To use the available resources efficiently, schedulers are used while it satisfying competing and mutual conflicts goals. It minimizes make-span, idle time of the available computational resources, turnaround time. The model also considers the deadline provided by the user.

A Cluster based vehicular cloud architecture with learning based resource management is proposed in [27]. To deploy new applications in a mobile vehicle is a significant challenge, as resources are very limited in mobile vehicles. To group vehicles and provide resources cooperatively they propose clustering technique. To select a service, provide among participant vehicles q-learning techniques are employed. For cluster head selection fuzzy logic techniques are used. Bees life algorithm for job scheduling in cloud computing is proposed in [28]. Task are scheduled based on a bee swarm optimization algorithm called Bees Life Algorithm (BLA). The BLA algorithm efficiently schedule the computation job. Among processing resources in cloud data centers the proposed algorithm is to schedule competition jobs (task). In an optimal fashion to reduce the total execution time of jobs it aims at spreading the workloads among the processing resources.

Task scheduling optimization in cloud computing based on heuristic algorithm is proposed in [29]. Particle Swarm Optimization (PSO) algorithm is presented for scheduling in the proposed system. Small position value rules are used in the algorithm. Based on these rules tasks are formulated and tasks scheduled. Cost parameters are used to measure the performance. Self-adaptive resource management system in Infrastructure as Service (IaaS) clouds is proposed in [30]. Based on a hierarchical multi agent they proposed self-adaptive resource management system. Novel adaptive utilization threshold mechanisms are used in the system. They also employ reinforcement technique. Reinforcement technique dynamically adjust CPU and memory thresholds for each of physical machine. To keep the total resource utilization of each physical machine within given thresholds it periodically runs a virtual machine placement optimization algorithm. Using this approach, they are improving Service Level Agreement (SLA) compliance.

Cooperative resource management in cloud enabled vehicular networks proposed in [31]. They proposed a coalition game model for resource management in cloud enabled vehicular networks. For cooperation among

cloud service providers are based on the two sided matching theory in the model to share their idle resources.

A survey on resource scheduling in cloud computing: issues and challenges are presented in [32]. The different techniques for resource scheduling in cloud computing are compared. Standard methodological literature analysis techniques are used in the survey. Virtual machine migration in road side cloud let based vehicular cloud is proposed in [33].

Mobility prediction for efficient resources management in vehicular cloud computing is proposed in [34]. Artificial Neural Network (ANN) techniques are adopted in the system. Based on the output of ANN, this method enables the vehicular cloud to take preplanned procedures. By this technique they reduce the effect of resource mobility and hence achieve efficient resource management.

Vehicles as connected resources concept is described in [35]. The evolution leading to the Internet of Vehicles (IoV) are described. Reliable adaptive resource management for cognitive cloud vehicular networks is proposed in [36]. The proposed model has distributed and adaptive resource management controller for cognitive cloud vehicular network. Optimal exploitation of cognitive radio is achieved in the design model. It also includes the soft-input/soft-output data fusion in vehicular access networks. The currently available bandwidth energy resources are quickly able to acquire context information. In the state of the vehicular network it adapts to the mobility induce due to abrupt changes. Vehicular cloud network resource management based systems is proposed in [37].

The relevant issues to resource management in the vehicular cloud are addressed. Resource management model based on cloud computing environment is proposed in [38]. The dynamic model makes use Hadoop and HBase. For computing resources based on Hadoop and HBase the model regularly stores the usage data. An effective recommendation regarding the consumption of computing resources are analyzed in the model. It also includes the analysis of raw data for virtual machine.

Rescheduling for reliable job completion with the support of clouds is presented in [39]. The novel technique which supports for rescheduling in the cloud environment and archives reliable completion of tasks in the cloud environment. It investigates the effectiveness of rescheduling to increase the reliability of job completion. For only rescheduling to cope with a delay in job completion, the cloud resources (relatively costlier) are used. Job scheduling algorithm based on berger model in cloud environment is presented [40]. Dual fairness constraint in the job scheduling process are established in algorithm.

Recent internet protocol is the IPV6. IPV6 is the communication protocol and intend to replace IPV4. The pit falls in the IPV4 such as address exhaustion is addressed in IPV6. Internet engineering task force (IETF) has proposed the IPV6 as current version of IPV4 is rapidly running out of IP address. IPV6 has many advantage over IPV4 such as header filed, related to the

security aspects, large address space etc [41].

VANETs is an Intelligent Transportation System, two kinds of messages are transferred in VANETs safety related messages and non-safety related messages. Broadcasting based on the location and density of network has huge attention in VANET. False hindering are reduces in VANETS, if the messages are prioritized [42].

Cloud computing has multitenancy feature that enables the sharing of resources. The other features of cloud such as scalable of resources and elasticity makes the user to access the services and resources also location independent. The key requirement of IoT platform is the sharing of resources, hence combination of IoT and cloud for future technologies provides the better services [43].

III. DYNAMIC RESOURCE DISCOVERY SCHEME IN VEHICULAR CLOUD NETWORKS

In today's world, where computing resources are virtualized, building user applications to scale resource needs seamlessly has become very easy. Developers need not worry about how the servers are configured and how to increase capacity on demand. Infrastructure services are provided on the internet (cloud) with quick and easy requisition. This is fundamental to distributed deployment of application on the cloud and anywhere access. Whether we are dealing with virtual or physical resources, we need to discover computing resources available for any application before they are allocated to requests dynamically on demand. Discovery of resources may seem random but once the repository is built, we can employ various intelligent algorithms to allocate best possible virtual computing machine for a request. It is understood that resource repository is kept up to date with regularly scheduled discovery.

We will assume all resources will be assigned unique IP address and same will be used between agent/manager architecture for discovery. An agent will be built to run on each resource. This agent will respond to any request from discovery manager. We will also assume discovery manager will get resource details using a range of IP addresses. For instance, when discovery manager initializes, it will not know exactly how many resources are in the network.

In discovery phase, after IP address range expansion, simplest way to discover resources is for the server to schedule discovery of each resource using multiple threads in a round robin fashion. Example: if there are 100,000 resources to be discovered and assuming we have configured the server to run with 10 threads, 10 discovery tasks will be scheduled each second. This means, discovery of all 100,000 resources will take approximately 2 hrs 46 minutes ($100,000/10 = 10,000$ seconds) if each resource is discoverable within one second.

In this model, there is no optimization in the way resources are discovered and how the system behaves. If threads take more than one second to discover a resource, subsequent discovery will be pushed out as it is scheduled

only when the thread is available after timeout. However, if discovery is completed within one second, subsequent resource discovery is not scheduled as server will wait for timeout to run next schedule. As it is evident, this is not efficient. Hence proposing dynamic discovery model. In this model resources are discovered using a dynamic and optimized algorithm. Server will begin with a preconfigured timeout for the first set of schedules. Following the above example, 10 threads are used to discover every sets of 10 of 100,000 resources. The difference in this model, however, is apart from recording resource configuration in the repository during the discovery process, each thread will also track and store processing parameters like time taken for processing, memory consumption of the resource, CPU utilization and soon.

During subsequent scheduling, server will look up previous performance metrics to alter the timeout efficiently so that cycle is run optimally. If VMs responded to discovery process well within one second (say within 100 milliseconds), subsequent cycle will try to discover 10 resources within next 100 milliseconds rather than the next second. With this model if the system performance remains constant, scheduling remains constant. In contrast, if response (of either the network or target systems) varies from cycle to cycle, scheduling will adjust accordingly to discover resources with varying degree of optimization. This is much more efficient way to discover resources with optimal use of resources

A. Network environment

We consider a three tier Vehicular Cloud Network (VCN) architecture as depicted in figure 1. Vehicular Cloud (VC) is considered as tier-1, where the resources are shared among vehicles using vehicle-to-vehicle (V2V) communication only. Tier-2 consists of Infrastructure Cloud (IC) formed by Road Side Unit (RSU), where the vehicles have access to cloud services. Traditional Cloud (TC) exists in the internet domain. TC has many resources, which are accessed by vehicles. TC has spread over the large geographical area. Having defined a way to identify resource in the system, as depicted in the figure 1. Each module can be deployed independently in a distributed environment. Every component interacts with the other with a standard protocol. We will have a manager application running on a central server. It will communicate with agents running on virtual machines and other hardware (vehicle models through radio tower proxy) to receive requests and distribute them for processing. Response will be relayed back to requester. This server component will act as discovery and allocation manager. In the discovery phase, it will build the repository of all available computing resources.

B. Preliminaries

This section presents some of the terms used in the proposed scheme.

- Resource management: Resource management is technique that provides the available resources to

user's by providing proper resource identification, allocation and sharing.

- Resource discovery: Resource discovery or location is process in which application that need to locate the available (either hardware or software) resources
- Scheduling of task: To decide which task to execute first and which task to execute last to achieve maximum performance

C. Multi agent system model

Proposed multi agent model comprises of static and mobile agent. Components of agency and their interactions are as shown in figure 2. Agency consists of knowledge base, static agent (Vehicle Manager Agent, VMA) and mobile agent (Queen Mobile Agent, QMA).

- Knowledge Base (KB): It comprises of information of vehicle IDs, cloud IDs, available bandwidth for communication, vehicle status (connected/disconnected to network), available cloud services, KB is read or updated by VMA and QMA.
- Vehicle Manager Agent: It is a static agent based on Honey Bee Optimization mechanism that runs in a node (vehicle and cloud server), creates QMA and knowledge base, controls and coordinates activities of agency. This agent triggers QMA for discovering resources in vehicular cloud, infrastructure cloud and traditional cloud.
- Queen Mobile Agent (QMA): It is a mobile agent, employed to discover the resources available in vehicular cloud, infrastructure cloud and traditional cloud. It is triggered by VMA that travels around VCN by creating its clones (a clone is a similar copy of agent with different destination addresses). For each visited cloud service provider, it updates knowledge base (KB) in coordination with VMA

Honey bees (QMA) is characterized as swarm and possess swarm intelligence. Honey bee colony consists of three types of bees, namely queen, drone (male bee), and workers (female worker). Queen is the mother of colony. In the biased random walk, the inherent direction is given by the parameter in equation (1)

$$\beta = \beta_{base} + \mu L \quad (1)$$

Where β_{base} is the natural error in the honey bee's navigation, μ is the configurable parameter, L is the toxic level in the bee.

The probability of traveling in the correct vertical direction (towards infrastructure cloud i.e., IC and traditional cloud i.e., TC) is given in equation (2)

$$p_v = \frac{dv}{dv + dh} (1 - \beta) + \frac{\beta}{4} \quad (2)$$

The probability of traveling in the correct horizontal

direction is given in equation (3)

$$p_h = \frac{dh}{dv + dh} (1 - \beta) + \frac{\beta}{4} \quad (3)$$

where dv is the discrete step towards vertical direction, dh is the discrete step towards horizontal direction and β is the navigation error in four quadrangles.

D. Multi agent system for resource discovery

Discovery of resources, routing of task request and allocation of tasks execution through cloud is as shown in fig. 3. The resource discovery manager will have a list of resources to discover. It is responsible for collecting server resources such as information of processor (kind of processor, its capacity, clock rate in GHz) memory, storage and network band width. When it comes to optimal utilization of computing resources it is imperative

that we have a real time map of resources. It is fairly simple to discover available resources in the network if every node in the network is reachable. It is equally easy to model such systems programmatically, if computing systems follow a predictable behavior pattern, which is far from reality. Hence, it becomes extremely complex if we need to keep the data up to date as utilization of systems is very dynamic in nature. IPv4 is not suitable for VCN, it is appropriate to use IPv6. This can either pre-configured as a list of unique address locations (like IPv4) at which resources can be reached or can be done through a range sweep. Controller then starts with a time slot to schedule first set of discovery tasks. A network system like vehicular cloud which must be extremely efficient and optimal in using resources. It is important in such real time networks to use optimal algorithms to keep repository up to date and also allocate resources for task execution. Repository is used to store data.

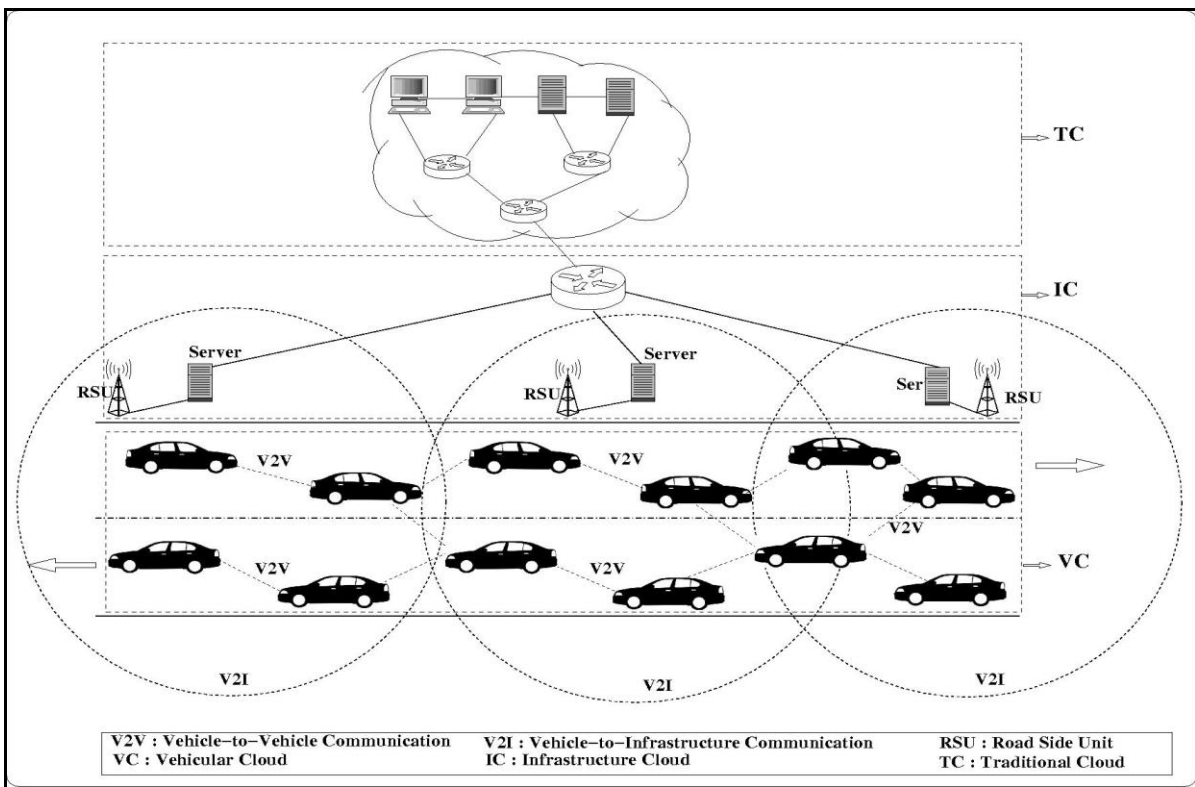


Fig.1. Vehicular Cloud Network Environment

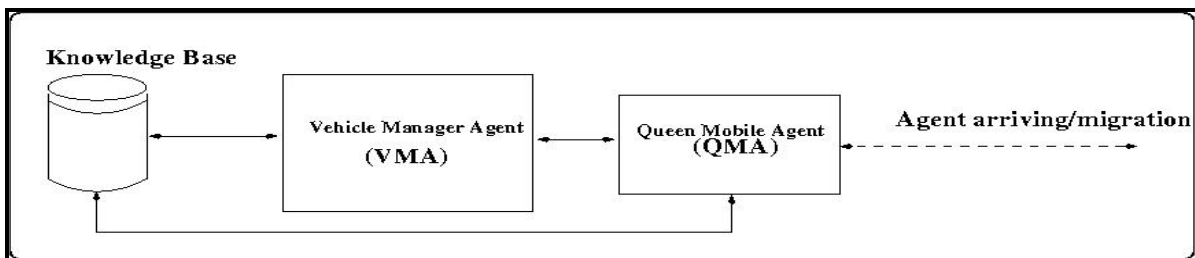


Fig.2. Multi agent System Model

E. Algorithm

Algorithm for the proposed resource discovery scheme is depicted as shown in algorithm 1.

Algorithm 1 Dynamic Resource Discovery

Nomenclature:

Resource list: L with resource type R

Resource: R contained within the list L

Default discovery threshold: θ , constant value

Discovery performance list: E with performance of each discovery $\epsilon \leftarrow \theta$

Discovery time allocated for each schedule: t

Input: Resource list, collection of resources to be discovered

Output: Resource repository, L updated with latest attribute information for each resource R and performance data collected. E for each discovery run with ϵ . Used for future discovery run

Keep running discovery of resource every T minutes to refresh the list

While true do

 Prepare list of resources to be discovered, L

For each R each resource to discovery in L do

 Apply discovery scheduling logic;

if E is empty **then**

 Prepare and schedule threads to discover resources in parallel with default schedule;

$L \leftarrow R$ where $t = \theta$;

 Collect actual discovery performance data and update repository for future use;

 Generate ϵ ;

 Update $E(\epsilon)$;

end

else

 Schedule resource discovery based on

last

 performance;

 Apply learning model to calculate next

 Discovery allocation time based on historic data;

$L \leftarrow R$ where $t \leftarrow \text{median}(E(\epsilon) - \text{error factor})$;

 Update $E(\epsilon)$;

end

end

wait(T)

end

F. Example Scenario

Dynamic discovery starts with a time slot to schedule first set of discovery tasks. However, it differs from fixed algorithm from the very next step. Once nodes start responding, the discovery algorithm keeps track of response time, network latency and controller (system scheduling tasks) resource utilization. Subsequent tasks are scheduled by calculating the predicted response time of nodes using different parameters and by adding a small buffer to take into account system overhead.

For example, if we start with scheduling time as 100

ms as the interval between server (resource) discovery and if the node responds quickly (lets assume 45 ms), next task is scheduled to discover the resources is to be

adjusted more than 45 ms. Let us assume that it is 58 ms and for the next step time take for the discovery is 65 ms then the interval between discovery of servers will adjusted of more than 65 ms. In this way our propose scheme adjusts the scheduling dynamically to discover the server contrast to fixed time scheduling where scheduling time is fixed irrespective of the response time.

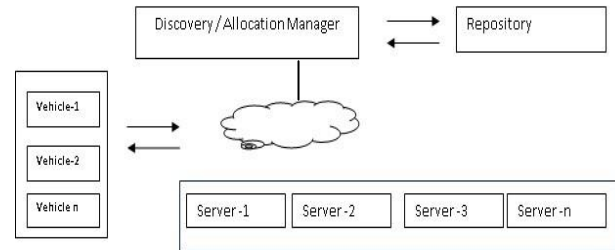


Fig.3. Resource discovery block diagram

IV. SIMULATION

This section presents performance parameters considered, simulation procedure, simulation inputs and result analysis

A. Performance Metrics

Some of the performance metrics evaluated are response time, CPU utilization, memory utilization, CPU overhead.

- **Response time:** This is the time taken by a resource to receive a request, process it and respond back to the requester. It is expressed in terms of milli seconds.
- **CPU utilization:** A task is scheduled to run on a CPU core. CPU Utilization is the part of CPU used during the execution of task(s). It is expressed in terms of percentage
- **Memory utilization:** A task needs system resources like CPU cycles, dynamic memory and storage. Memory utilization refers to amount of dynamic memory used during the execution of task(s). It is expressed in terms of percentage.
- **Bandwidth utilization:** Bandwidth utilization is the amount of network capacity used during the execution of task(s). It is expressed in terms of bits per second.
- **Scheduling overhead:** Switching of resources between tasks usually referred as context switching. It is expressed in terms of percentage

B. Simulation Procedure

Simulation is performed using JAVA language. Simulation procedure for the proposed resource discovery scheme is as follows:

- Begin
- Create virtual machine (or server) farm with different configurations (CPU size, clock speed, RAM and Storage).
- Create vehicle objects.
- Build an agent on virtual machine and vehicle to interact with controller.
- Assign dummy IP address to each virtual machine and vehicle.
- Create controller software to interact with agent on virtual machine and vehicle.
- Configure list of IP addresses to discover in controller.
- Controller schedules and sends discovery request to agent on virtual machine.
- Agent responds with resource configuration of the virtual machine.
- Controller stores discovered information in repository for allocation.
- Controller also stores task performance details to refine discovery schedule

C. Simulation Inputs

Simulation parameters for resource discovery:

Steady state:

- Resource manager for discovery and allocation: A single system is configured as controller
- Resource manager CPU: 4 core CPU
- Resource manager dynamic memory: 16 GB
- Number of vehicles:

5,10,20,35,50,65,75,85,100 (max)

- Number of server resources in the cloud:

3, 5, 7 10 (max)

- Network bandwidth: 500 mbps

Load condition study:

- Number of resources to discover: 1000, 2000, 5000, 10000, 15000, 20000, 250000, 30000, 50000
- Resource manager CPU: 8 core, 4 CPU
- Resource manager dynamic memory: 128 GB

D. Result Analysis

Memory, CPU and bandwidth consumption is constantly changing in large system like VCN environment, it is a challenging task to predict the available resources and optimally allocate the available resources. Low utilization of the resources are major concern in the VCN systems. VCN demands the optimum usage of available resources. Significant performance degradation in the system can occur when the application cannot utilize the resources fully. In VCN environment

with significant workload fluctuation fails to provide the expected services to the vehicle in time.

This section presents the results obtained during simulation we compare results of the proposed work with fixed time scheduling [44]

As shown in the figure 4, each node will be discover in 100 msec, assuming no network latency compare to performance of dynamic algorithm. As it is learning from each task execution experience, it keeps adjusting the task run time based on actual execution time of previous task and other influencing variables. Although dynamic scheduling seems to fluctuate with scaling of nodes, the behavior smoothens out.

The graph shown in figure 5 depicts the total time taken for 50 node discovery.

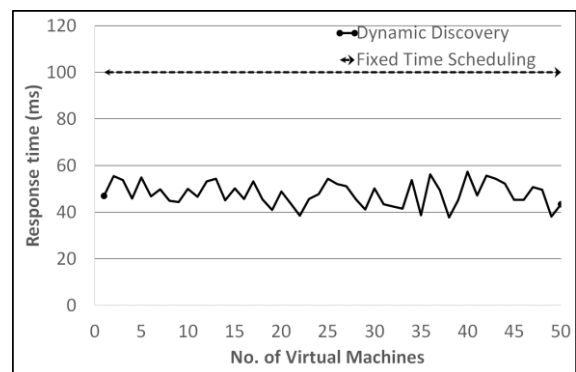


Fig.4. Response time Vs. No. of virtual Machines

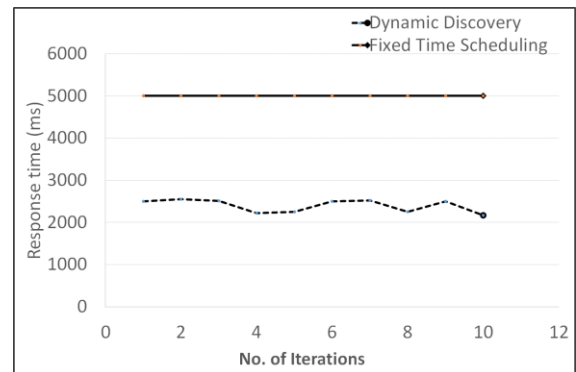


Fig.5. Response time Vs. No. of iterations

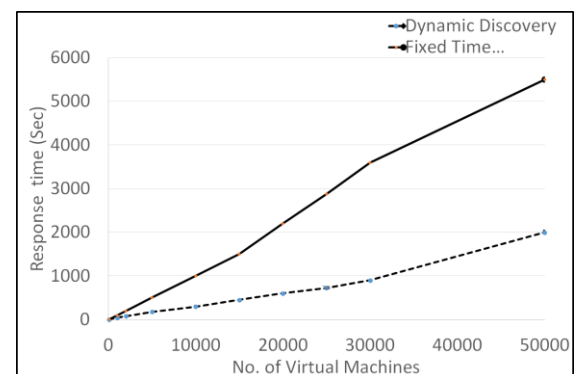


Fig.6. Effect of increasing computation resources on total discovery time of nodes

Through simulation, we have established the benefits

of dynamic discovery logic over fixed time scheduling. As shown in figure 6 we increased the number of nodes for discovery and computation resource (higher CPU, more memory and bandwidth) to evaluate the correlation. Through simulation results that dynamic scheduling logic is much more efficient in discovering scaled nodes with increased computing power.

Fixed time technique saturates at every step as it is limited by its very logic of fixed time slots to better utilize scaled computation resources.

On careful analysis of controller (CPU, Memory or I/O) and network resource (bandwidth) utilization, we can observe that dynamic discovery logic adds significant overhead to resource load. As shown in figure 7, we compare the CPU utilization overhead for dynamic and fixed time task scheduling logics.

For fixed time, overhead is minimal whereas for dynamic logic overhead is substantial. Overhead is taken as overall percentage of total utilization. On further analysis of overhead for dynamic discovery logic, we find that overhead is directly influenced by the variance of node response time, a key parameter for the algorithm. From resource utilization perspective, dynamic discovery is much more efficient than fixed time. However, if not controlled with scaling of tasks (more nodes to discover), resources will overload to a point system can crash. Hence the overhead of monitoring.

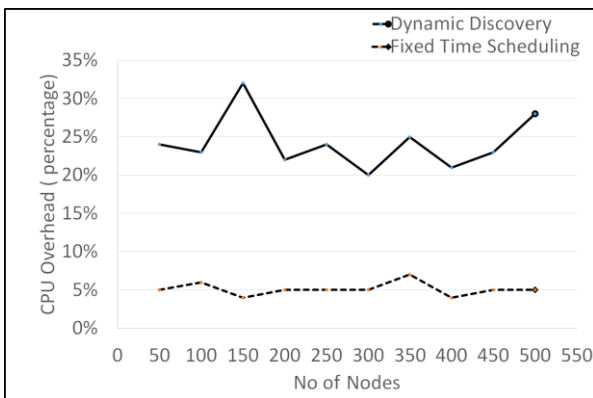


Fig.7. scheduling overhead

Efficiency that dynamic discovery achieves in terms of better Overall time is offset by overhead of complex scheduling logic and load on controller resources. As shown in figure 8 graphs shows CPU utilization controller system during fixed time and dynamic discovery. It shows the controllers (system running discovery tasks) CPU utilization during discovery of set of nodes. As is evidenced, CPU utilization fluctuates for fixed time logic because once nodes respond, controller is idling to run out the time slot before lining up next task. Contrast this to Dynamic logic, CPU utilization stays up throughout the discovery process as controller is scheduling tasks with very little idling in between. If we plotted memory (RAM) utilization, it will follow the same trend.

Compare that with lower overhead when actual response time hovers near mean value as shown in figure

9. Although we have considered only node response time to study the impact on overhead, we can hypothesize similar impact of other parameters, like network load on the overhead of system resource utilization. We can conclude with above study that we need more computing resources if we are to derive benefits of dynamic discovery logic. As the number of resources (nodes) to discover scales, load on discovering system (controller) increases as a direct result. Because more the number of nodes to discover, more is the possibility of erratic response time, thus higher degree of fluctuations from mean.

Figure 10 illustrates the causes for node response time deviation. One of the primary reason for nodes responding slowly is because their management plain is built for lower priority than their control plain.

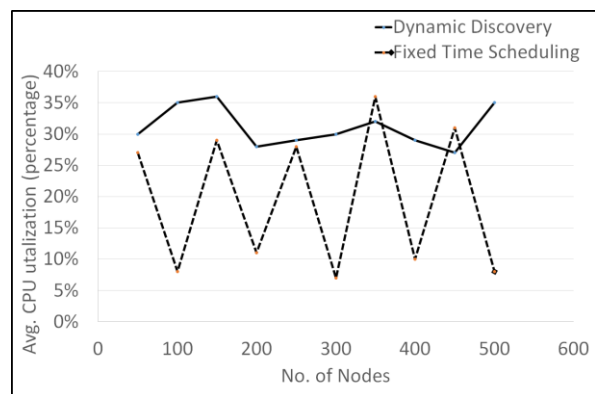


Fig.8. CPU utilization of controller during resource discovery

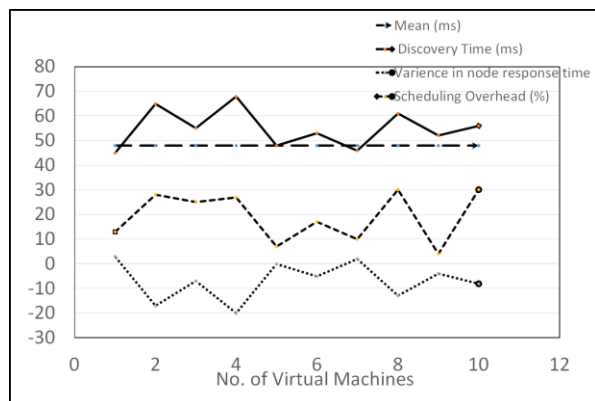


Fig.9. Effect of discovery time variance on scheduling overhead

Primary goal of any computing resource is to execute tasks to accomplish a goal. While that is very simplistic, every node is built to handle different tasks with different priority. The load on the system has a direct correlation with response time for a discovery task. As it is difficult to simulate actual load on the system, we will simulate network load to study the impact on node response time for discovery tasks. Network load or bandwidth utilization is a good measure of how busy the network is, thus simulating load on nodes in the network. Through simulation, we study the correlation between load on network (nodes) and response time variance for dynamic discovery technique.

Simulation for fixed time logic and observed the effect of network load on response time as shown in figure. 11. As observed from the graph, network load has almost no influence on the node discovery time. This because in fixed time logic, there is enough buffer (or idling time) to absorb the fluctuation in response time. Actual node response time still varies from the mean during clogged network. However, fixed time logic has fixed time slots, which ensure any effect of spike smoothen out.

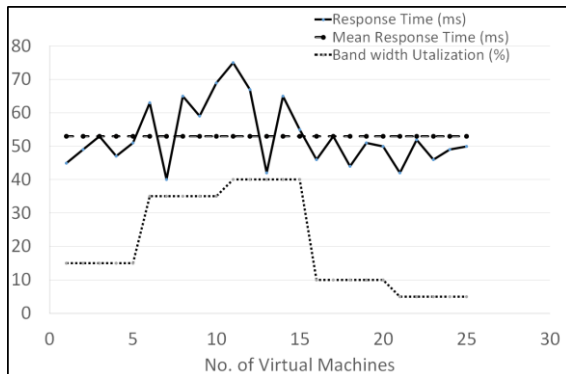


Fig.10. Relation between network load and response time for dynamic discovery

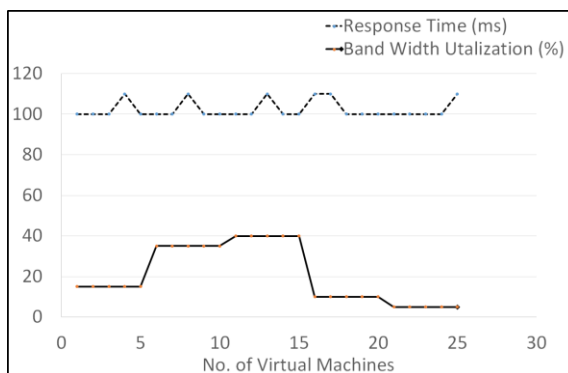


Fig.11. Relation between network load and response time for fixed time scheduling

V. CONCLUSION

Resource discovery scheduling algorithm discovers the resources in VCN environment and should take minimum processing time so that services can be provided to the vehicles in time. Resource discovery mechanism in VCN also demands the optimal solution. In this paper, dynamic resource discovery scheme is proposed and implemented by integrating static and mobile agents. The proposed work is simulated for processing time, CPU utilization, scheduling overhead and memory utilization to validate its performance. Mean response time for dynamic resource discovery scheme is 56 ms whereas for fixed time scheduling mean response time is 101 ms. The performance of dynamic resource discovery scheme is better than fixed time scheduling in terms of response time, CPU utilization and memory utilization. Dynamic resource discovery scheme yields much better results in terms of response time although at the cost of more

resource consumption and additional overheads that need to be closely monitored and controlled. The disadvantage of the proposed method is increase in scheduling overhead compared to fixed time scheduling. Fixed time scheduling, by its very nature is better suited for discovering nodes where discovery time is not critical factor in static and legacy networks where network/node configurations will not change rapidly. In dynamic and ever changing vehicular cloud network, dynamic resource discovery scheme is more suitable for scalable computing resources.

Future extension of this work is to validate the simulation results on real time environment and compare the simulation results with other dynamic scheduling algorithms.

REFERENCES

- [1] Wenshuang Liang, Zhuorong Li, Hongyang Zhang, Yunchuan Sun, Rong fang Bie, "Vehicular ad hoc networks: architectures research issues, challenges and trends", Proc. 9th International Conference on Wireless Algorithms, Systems, and Applications, vol. 8491, pp. 102-113, 2014
- [2] Nasrin Hesabiana, Hamid Haj Seyyed Javadi, "Optimal scheduling in cloud computing environment using the bee algorithm", International Journal of Computer Networks and Communications Security, vol. 3, no. 6, pp. 253-258, June 2015
- [3] Linen Zhu, qingshui Li, Lingna He, "Study on cloud computing resource scheduling strategy based on the ant colony optimization algorithm", International Journal of Computer Science Issues (IJCSI), vol. 9, no. 2, September. 2012
- [4] Rodolfo I. Meneguette, Azzedine Boukerche, Robson de Grande, "SMART: an efficient resource search and management scheme for vehicular cloud connected system", proc. IEEE Global Communications Conference (GLOBECOM), 2016
- [5] Rong Yu, Yan Zhang, Stein Gjessing, Wenlong Xia, Kun Yang, "Towards cloud Based vehicular networks with efficient resource management", IEEE Journal, vol. 27, no. 6, pp. 48-55, 2013
- [6] Mohammad A. Salahuddin, Ala Al-Fuqaha, Mohsen Guizani, Soumaya Cherkaoui, "RSU cloud and its resource management in support of enhanced vehicular applications", IEEE Globecom Workshops, 2014
- [7] Khaleel Mershad, Hassan Artail, "Finding a star in a vehicular cloud", IEEE Intelligent Transportation Systems Magazine, vol. 5, no. 2, pp. 55- 68, 2013
- [8] Mehmet Demirci, "A survey of machine learning applications for energy efficient resource management in cloud computing environments", Proc. 14th International Conference on Machine Learning and Applications (ICMLA), 2016
- [9] Fahimeh Farahnakian, Rami Bahsoon, Pasi Liljeberg, Tapio Pahikkala, "A survey of machine learning applications for energy efficient resource management in cloud computing environments", Proc. 9th International Conference on cloud computing, 2016
- [10] Mohd Umar Farooq, Mohammad Pasha, Khaleel Ur Rahman Khan "A data dissemination model for cloud enabled vanets using in vehicular resources", Proc. 8th International Conference on Computing for Sustainable Global development, pp. 458-462, 2014
- [11] Fahimeh Ramezani, Mohsen Naderpourl, "A fuzzy virtual

- machine workload prediction method for cloud environments", Proc. International Conference on Fuzzy Systems (FUZZ-IEEE), 2014
- [12] Wu Chun Chung, Chin Jung Hsu, Kuan Chou Lai, Kuan Ching Li, Yeh Ching Chung, Yeh Ching Chung, "Direction aware resource discovery service in large scale grid and cloud computing", The Journal of Supercomputing, vol. 66, pp. 229-248, 2013
- [13] Swapnil M Parikh, Narendra M Patel, Harshadkumar B Prajapati, "Resource management in cloud computing: classification and taxonomy"
- [14] Hamid Reza Arkian, Reza Ebrahimi Atani, Atefe Pourkhalili, "A cluster based vehicular cloud architecture with learning based resource management", 6th International Conference on Cloud Computing Technology and Science, pp. 15-18, 2014
- [15] Rojia Nikbazm, Mahmood Ahmadi, "Agent based resource discovery in cloud computing using bloom filters", Proc. 4th International Conference and Computer Knowledge Engineering, 2014
- [16] Nguyen Xuan Phi, Cao Trung Tin, Luu Nguyen KyThu, Tran Cong Hung, "Load balancing algorithm to reduce response time and processing time on cloud computing", International Journal of Computer Networks and communications(IJCNC), vol. 10, no. 3, May 2018
- [17] Elhossiny Ibrahim, Nirmeen A. El-Bahnasawy, Fatma A. Omara, "Dynamic task scheduling in cloud computing based on the availability level of resources", International Journal of Grid and Distributed Computing, vol. 10, No. 8, pp. 21-36, 2017
- [18] Azzedine Boukerche, Rodolfo I. Meneguette, "Vehicular cloud network: a new challenge for resource management based systems", Proc. 13th International Wireless Communications and Mobile Computing Conference (IWCMC), 2017
- [19] Markus Ullrich, Jorg Lassig, Martin Gaedke, "Towards efficient resource management in cloud computing: a survey", Proc. 4th International Conference on Future Internet of Things and Cloud, 2016
- [20] Nguyen Khac Chien, Nguyen Hong Son, Ho Dac Loc, "Load balancing algorithm based on estimating finish time of services in cloud computing", Proc. 18th International Conference on Advanced Communication Technology (ICACT), 2016
- [21] Kannan Govindarajan, Vivekanandan Suresh Kumar, Thamarai Selvi Somasundaram, "A distributed cloud resource management framework for high performance computing applications", 8th Eighth International Conference on Advanced Computing(ICoAC), 2016
- [22] Kaspar Bienfelda, Klaus Ehrhardt, Friedrich Reinhardt, "Genetic evaluation in the honey bee considering queen and worker effects, a blup animal model approach", EDP Sciences, 2007
- [23] Fahimeh Ramezani, Jie Lu, Farookh Hussain "An online fuzzy decision support system for resource management in cloud environments", Proc. Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013
- [24] He Sheng WU, Chong Jun WANG, Jun Yuan XIE "TeraScaler elb an algorithm of prediction based elastic load balancing resource management in cloud computing", Proc. 27th International Conference on Advanced Information Networking and Applications Workshop and Cloud, pp. 649- 654, 2016
- [25] Christofer Flinta, Andreas Johnsson, Jawwad Ahmed, Farnaz Moradi, Rafael Pasquini Rolf Stadler, "Real time resource prediction engine for cloud management", Proc. IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2017
- [26] Mona Aggarwal, Robert D. Kent, Alioune Ngom, "Genetic algorithm based scheduler for computational grid", Proc. 19th International Symposium on High Performance Computing Systems and Applications (HPCS), 2005
- [27] Hamid Reza Arkian, Reza Ebrahimi, Atani Atefe Pourkhalili, "A cluster based vehicular cloud architecture with learning based resource management", Proc 6th International Conference on Cloud Computing Technology and Science, 2014
- [28] Nasrin Hesbian, Hamid HajSeyyed Javadi, "Optimal scheduling in cloud computing environment using the bee algorithm", Proc. International Journal of Computer Networks and Communications Security, vol. 3, no. 6, pp. 253-258,
- [29] Shuguang Zhao, Shigen Shen, Changyuan Jiang, Lizheng Guo "Task scheduling optimization in cloud computing based on heuristic algorithm", Journal of Networks, vol.7, no.3, 2012
- [30] Fahimeh Farahnakian, Rami Bahsoon, Pasi Liljeberg, Tapio Pahikkala "Self adaptive resource management system in IaaS clouds", Proc. 9th International Conference on Cloud Computing, 2016
- [31] Rong Yu, Xumin Huang, Jiawen Kang, Jiefei Ding, Sabita Maharjan, Stein Gjessing, Yan Zhang "Cooperative resource management in cloud enabled vehicular networks", Proc. IEEE Transactions on Industrial Electronics, vol. 62, pp. 7938-7951, 2015
- [32] Sukhpal Singh, Inderveer Chana "A Survey on resource scheduling in cloud computing: issues and challenges", Journal on Grid Computing, 2016
- [33] Hong Yao, Changmin Bai, Deze Zeng, Qingzhong Liang, Yuanyuan Fan, "Migrate or not exploring virtual machine migration in roadside cloudlet based vehicular cloud", Willey On line Library
- [34] Ahmad M. Mustafa, Omar M. Abubakr, Omar Ahmadien, Ahmed Ahmedin, Bassem Mokhtar "Mobility prediction for efficient resources management in vehicular cloud computing", Proc. 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (Mobile Cloud), pp. 53-59, 2017
- [35] Soumya Kanti Datta, Jerome Haerri, Christian Bonnet, Rui Ferreira Da Costa "Vehicles as connected resources opportunities and challenges for the future", Proc. IEEE Vehicular Technology Magazine, vol. 12, pp. 26- 35, 2017
- [36] Nicola Cordeschi, Danilo Amendola, EnzoBaccarelli "Reliable adaptive resource management for cognitive cloud vehicular networks", Proc. IEEE Transactions on Vehicular Technology, vol.64, pp. 2528-2537, 2014
- [37] Azzedine Boukerche, Rodolfo I. Meneguette, "Vehicular cloud network: A new challenge for resource management based systems", Proc. 13th International Wireless Communications and Mobile Computing Conference(IWCMC), vol. 64, pp. 2528-2537, 2017
- [38] Ahyoung Kim, Junwoo Lee, Muechol Kim, "Resource management model based on cloud computing environment", Proc. International Journal of Distributed Sensor Networks, vol. 12 (11), 2016
- [39] Young Choon Lee, Albert Y. Zomaya "Re scheduling eliable job completion with the support of clouds", Proc. Journal on Future Generation Computer Systems, 2010
- [40] Baomin Xu, Chunyan Zhao, Enzhao Hu, Bin Hu "Job scheduling algorithm based on berger model in cloud environment", Proc. Journal of Advances in engineering

software, vol. 42, pp. 419-425, 2011

- [41] S. Praptodiyono, R. K. Murugesan, R. Budiarto and S. Ramadass, "Handling transmission error for IPv6 packets over high speed networks," First International Conference on Distributed Framework and Applications, Penang, 2008, pp. 159-163
- [42] M. Selvi and B. Ramakrishnan, "An Efficient Message Prioritization and Scheduled Partitioning Technique for Emergency Message Broadcasting in VANET," 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2018, pp. 776-781
- [43] A. R. Biswas and R. Giaffreda, "IoT and cloud convergence: Opportunities and challenges," 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, 2014, pp. 375-376
- [44] Samir Elmougy, Shahenda Sarhan1, Manar Joundy "A novel hybrid of shortest job first and round robin with dynamic variable quantum time task scheduling technique", Proc. Journal of Cloud Computing: Advances, Systems and Applications, 2017

Authors' Profiles



Mr. Mahantesh G Kambalimath. received his B. E (Instrumentation Technology) from Karnataka University, and M. Tech (CSE) from Visvesvaraya Technological University (VTU), Belgaum. Presently working as Assistant Professor in the Department of Electronics and Instrumentation Engineering, Basaveshwar Engineering College, Bagalakot. His research interests are in the area of Vehicular Ad hoc networks and cloud computing.



Dr. Mahabaleshwar S. Kakkasageri received his Ph.D from Visvesvaraya Technological University (VTU), Belgaum. Presently working as Associate Professor in the department of Electronics and Communication Engineering, Basaveshwar Engineering College, Bagalakot. His area of research interests includes Vehicular Ad hoc networks, Wireless Sensor Networks, Multi-Agent Systems. He is member of IETE. He presented more than 50 technical papers in national, international conferences and scientific journals

How to cite this paper: Mahantesh G. Kambalimath, Mahabaleshwar S. Kakkasageri, "Dynamic Resource Discovery Scheme for Vehicular Cloud Networks", International Journal of Information Technology and Computer Science(IJITCS), Vol.11, No.12, pp.38-49, 2019. DOI: 10.5815/ijitcs.2019.12.04