International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.3, No.9, September 2014 DOI:10.15693/ijaist/2014.v3i9.35-44

Towards Resource Discovery Framework Using Honey Bee Swarm Model for Multi-Provider Cloud Environments

M.N.Faruk
Research Scholar
Computer Science and Engineering Department
Bharath University, Chennai-600073, India

Dr.D.Sivakumar*
Professor
Information Technology Department
Easwari Engineering College, Chennai-600089, India
*Corresponding author

***Abstract*—Cloud computing is growing paradigm concerned with computing resources, offered over the internet platform which is scalable and on-demand (web) services. The Honey bee agent based model will discover the cloud services on multiple levels. The original contributions of this effort includes: i) developing an employee forager bee agent search model for private and dedicated cloud resource discovery, ii) deploying the scout bee agent based on discovery mechanism for unknown cloud sources, and iii) The knowledge incurred through 'waggle dancing mechanism' the onlooker bee agents will find the unknown cloud sources and collected resources are clustered by resource cluster methods. The Honey bee agent has cooperative work nature can be efficiently deployed for mechanize cloud service discovery. At cloud user query processing, the user query was analyzed and compared via similarity reasoning and compatibility reasoning which derived from the usage level each service was rated using cloud database. The anticipated resolution is significant: instead of spending more time and effort on discovering, estimating and learning about cloud clusters, users are able to effortlessly discover, select and use the required services. Additionally, Cloud Service Providers (CSP) can certainly publish information about their services.**

Index terms - Cloud Computing, Service discovery, Bee agents, Query Processing Unit, Hive.

I. INTRODUCTION

In recent years the Cloud Service Providers (CSP) has targeted their efforts on the enhancement of performance, resource usage and scalability; however the Cloud Service Providers (CSP) handles complex problems while publishing their services that render resource availability. Other hand cloud users are not capable on discovery and automatic service selection. Moreover, a cloud is collections of independent computers are virtually integrated for storage and computational purposes. The variety of applications and data on the cloud has processed via the internet by the broad user groups across multiple enterprise and platform [1]. Cloud systems are dynamically arranged as one or more integrated computing resources by Service Level Agreements (SLAs) among a variety of Cloud Service Providers (CSP) and Cloud users [2]. In this platform, resources are grouped and dynamically configured and accumulated via virtualization and the cloud user necessities variety of requirements can potentially differ over time and improvement must be accommodated.

In this work, we recommend a multi-agent system has of a group of agents, their actions are justified and decisions are taken by their own intelligence. The successful communication between the agents requires the potential to collaborate, harmonize and confer with each other [2]. The Collaboration process involves multiple agents to extract knowledge from different sectors and targeted to their goal. In other hand harmonization is the process to accomplish the agreement between the agents. Negotiation is carried out through establishing communication between agents and external users [4]. The Swarm Intelligence has a constructive framework depend on social insect's nature and behaviors. Social insects mostly like ants and bees which following unique mechanism to carry out complex tasks. This cooperation exists without any supervised control. Each individual agent obeys set of guidelines by means of locally available data [5]. This unique behavior in-turn creates great accomplishments that no single agent acts by their own. Subsequently the swarm intelligent system also includes robustness towards their misbehavior or loss; they are dynamic in nature at any unstable environment and deploy distributed action.

II. RELATED WORKS

There are number of mature approaches has evolved for discovering cloud services in cloud market places. Sauli hamza et al deploys multiple agents for cloud resource discovery. The semantics have stored in database that leads to be time consuming process at different cloud environments and also the heuristics approach cause the larger complexity of public and hybrid clouds [4]. As our inspiration proposal is the Honey bee's life in natural world. The entire population is depend on the meta-heuristic enthused by the innate foraging activities of honey bees referred as Bees Algorithm [5]. It has initiated with a multiple number of scout bees are being positioned randomly in foot patch search space. The best sites identified based on fitness values. The chosen selected bees which will be foraged to their nearest sites to carry out a neighborhood search. Peter wright et al, have constraints based resource discovery model which has two phases at multi provider cloud environments. The constraints need to specifically pointed depend on variety of applications [12]. If the constraints are too large which involves several attributes that leads to greater complexity in resource discovery at hybrid clouds and this could be more time consuming process. Das et. al., has investigated a multi-agent model to provide economically finest performance in basic data centre. The authors defined their methodology which was purely depend on a hierarchical order by which the resource authority allocates resources to application manager module, in returns it becomes possible charge of administrating each physical server [7]. Subsequently Stelibis Sotiriadis et al, employ an automated self-organized agent for service discovery this involves multi agent system. This lagging in communication between the agents [3].

However Mauro Andreolini et. al. has delivered the optimized framework for VM migration through examine individually at each physical node and its associated virtually configured machines along with the major objective of reducing migrations immediately with crucial instance [8]. Taekgyeong Han et al, the resource discovery method has determined by publish/subscribe messaging algorithm, in this work there are three key layers has involved, i). Provider agent, ii). Subscriber agent and iii). Broker agent. This model defines event – based algorithm that need to satisfy a range of constraints even changing in attributes values at different scenarios[17]. Author Rajiv Gandhi et. al findings are revealed based on the maximum speed with the server which is not efficient for best performance. They have investigated load balancing strategy with a middle dispatcher called as central load balancing strategy for VM's. The load balancing resolution has taken by the centralized server based on global status information [26]. The cloud computing mechanism inherits some distinct features like resource sharing and resource polling. The collected resources are mutually pooled and shared with multiple cloud users and their applications. This also supports in multi enterprises and different platform users.

The Resource pooling & sharing has taken part in combining resources, Mapping and scheduling with shared resources through SLA between CSP and cloud users . Furthermore this supports in self-directed resource allocation and handling dynamically changing requests. Subsequently a cloud user has to decide the suitable resources to negotiate and accomplish them depend on the obtainability of resources. K.M.Sim et al have projected parallel negotiation strategy [13,14] that establish the negotiation actions between consumer with broker agents and broker with provider agents. Since many number of broker agent service obtain possible request from each individual consumer agent and most number of consumer agent transmit the service request to each one of the broker agent, the many to many negotiation model is most significant for modeling negotiation in parallel form between consumer agents and broker agents. Since a Cloud service may be efficiently balanced based on the multiple types of Cloud resources, negotiation deliverables in the resource markets between provider agent and broker agent may influence the consolidation in service based market between broker agents and consumer agents. Henceforth, a smart one too many negotiation models are deployed to the concurrent negotiation actions between broker agents with multiple provider agents.

In previous work agents have the capability to make own decisions based on users default recommendations and converse with each other by collaboration, negotiation and harmonization. The lifecycle of cloud service includes 1.Service requirements. 2. Service discovery, 3.Service negotiation, 4.Service composition, and 5.Service consumption. The service requirements segment, the cloud users has to define budgetary requirements based on technical and functional aspects [15]. Moreover author Jinhua hu et. al., has described the scheduling the fittest resource by means of genetic algorithm models depend on the historically stored data which is already available and current status information having enough knowledge about the central module [16]. The projected idea is not sufficient with the user requirement which is dynamic in nature. The possibility be the record keeping with obtainable VM and the cloudlet requirement would be prominent process for different resource allocations. However the service discovery segment comprises probing for best possible resource that match cloud user necessities. In the Service Negotiation segment establish SLA between variety of cloud users and External Coordinators (brokers). Subsequently author Barbagallo et. al., describe the bio-inspired algorithmic model based on the scout worker movement model, through which the scout agents are permitted to migrate from local physical node to remote node to considerably identify a appropriate destination for VM's which are transferred[18]. Yee et. al., has defined a modified Particle Swarm Optimization(PSO) is detailed as discrete model particle swarm optimization for multiple task allocation which considerably reduces the total execution time with data transfers among optimal resources[20]. The Service Composition segment, a broker merge a range of services from multiple providers, and distributes a distinct virtualized resource to a cloud user. Finally the service consumption segment, the service is conveying to the cloud user.

III. THE HONEY BEE'S FRAMEWORK

In multitenant cloud platform the resources are widely scattered; The Honey Bee's framework is an effective and optimized model which involves three crucial components based on the developments of cooperative intellectual behavior of honey bee swarms [13]. 1. Food sources, 2. Employed forager's bees, and 3.Scout bees. The model also defines two primary operations: the discovering the new food source (nectar) and the desertion of a food source. At the initial stage the Scout bees have discovered the new food sources and have to inform to all other scout bees by waggle dance. The dance describes a mean of contact in the camp. This derives basic types of information (i). The Direction to the new food sources. (ii). Distance between hive and food sources. (iii). Quality of Nectar. This helps in other scout bees in the hive to travel on determined location. Once food sources are identified then scout bees are converted into recruit bees.

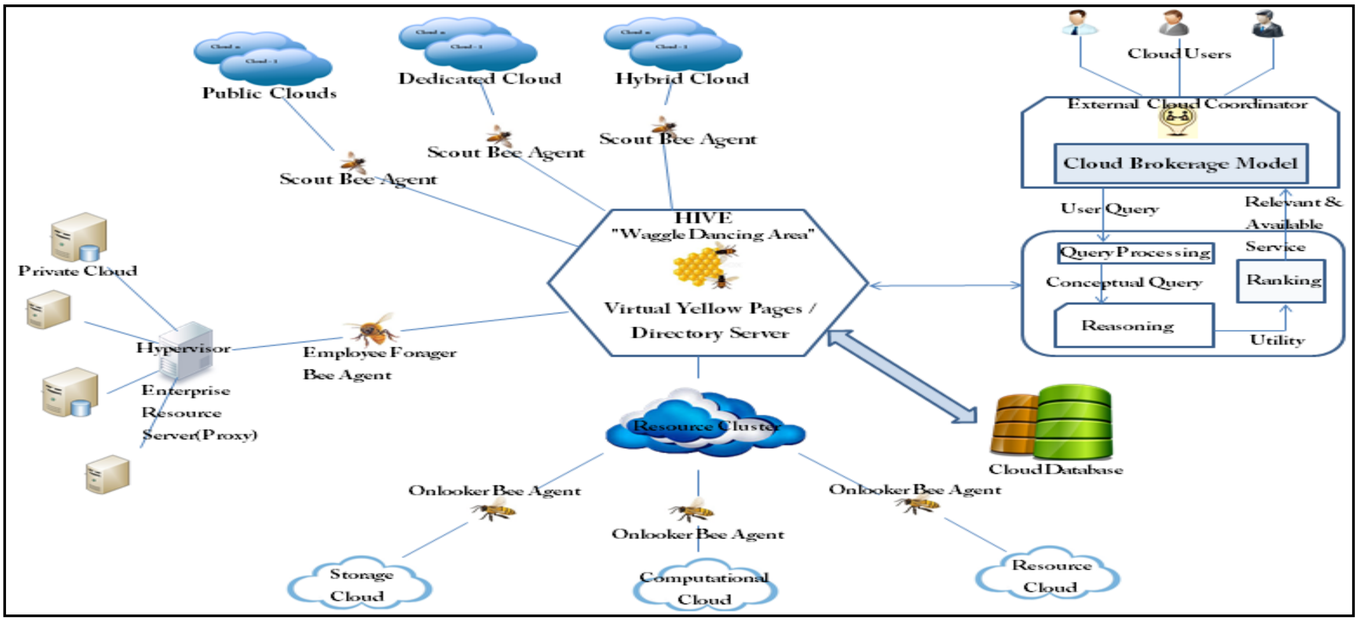

Fig. 1.A Honey Bee Agents based Cloud Discovery Model

Huge numbers of recruit bees will directed to most promising places to gather the nectar effectively and quickly. The Employed foraging bee obtains a freight of nectar from the known food patches and proceeds to the central hive and unpacks to a food store. After unpacking the foodstuff, the bee has the corresponding options; it is significant that all bees will initiate foraging simultaneously. This procedure confirmed that new bees will inaugurate new foraging with fair rate to the total quantity of bees [24]. The configuration towards self-organization may rely four basic properties as follows: i) The Positive feedback: It is the equilibrium between nectar quantity of food sources. ii) The Negative feedback: it is the message being exploited between the bees when there are no possible food sources. iii) Fluctuations: When the scouts bees travel randomly for discovering the new food sources. iv) The multiple interactions: Bees would share their information about new as well as the old food sources with the nest mates on the waggle dancing area[11].

IV. The Cloud Service Discovery Model

The Architecture comprises of five segments i). Cloud users in which they associated with Cloud Brokerage Negotiation Model. ii). the user query has been accelerated by Query Processing Unit. iii). Scout bee agents iv). Employee forager bee agents and v). Onlooker bee agents. There are various type users in the cloud environments. The cloud users have to define budgetary requirements based on technical and functional aspects.

External Cloud Coordinator (ECC): The external cloud coordinator provides feasibility for accessing the different set of cloud services, and provides interface to cloud users. They perform exactly like in real-world markets and intercede between variety of cloud user and CSP based on tendency of buying capacity[9]. An External Cloud Coordinator (ECC) can consent requests from many cloud users and follow through acquiescing their necessities. The SLA stipulates the specification on the type of service is being delivered, in the form of metrics granted by the cloud users, penalties for violation will be included. This module is instantiate by each data center and is accountable for undertaking subsequent activities: (i) Distribution of different cloud services, in terms of infrastructure oriented and platform oriented alliance; (ii) Tracking consignment on data center through commence intervention with other CSP in the form of dynamic scaling facilities across multiple data centers for managing the ultimate demands; and (iii) monitor the infrastructure based application processing and supervision on approved SLA's were delivered. The External Cloud Coordinator will acts as a market maker for collaborating CSP and cloud users. Moreover this aggregates the infrastructure based demands commencing the Cloud brokerage negotiation model and assesses them by obtainable supply currently being distributed.

Cloud Brokerage Negotiation Model (CBNM): This model facilitates the following negotiation activities i) between cloud users and ECC ii). Between ECC and Cloud Service Provider (CSP). However this is envisioned as a many-to-one negotiation model mounted for negotiation between cloud users and external cloud coordinator. Since the dynamic composition of different cloud services using variety of cloud resources. The external cloud coordinator (ECC) could potentially negotiate with multiple cloud Service Providers (CSP's) for multiple resources. Negotiation process involves multiple rounds. When Cloud users generates the proposal through the possible requirements like most desired price to be afforded, the desired reservation price in case of demands between the cloud users[25]. Typically an ECC initially propose most favored deal initially. If no agreement is reached, negotiation proceeds to the next round. At every phase, an ECC determines amount of concession using the strategy. i) Negotiation between multiple cloud users ii) when the agreement can be processed. iii) The deadlines for conflict between the cloud users for selecting same resources.

Query Processing Unit (QPU): This module collects the possible details from different types of cloud users like type of resource is required, prices and schedule details. The QPU consults with cloud ontology which is maintained by the online datacenter [24]. It also processes the similarity reasoning between a cloud user's requirements (both functional and technical) and CSP specifications on variety of services. The Cloud ontology holds set of cloud based ideas and trade off between cloud concepts and the reasoning on cloud services.

Similarity Reasoning: The similarity reasoning need to fulfill three basic requirements: (i). Technical, (ii) Functional iii) Budgetary [1]. Indeed it is hard to discover services which are accurately match these requirements. Again it is upsurge the choice of finding significant alternatives. The QPU conclude the similarity between Cloud1 and Cloud2 by including their basic reachable nodes [17]. Let α(Cloud1)∩α(Cloud2) be the total number of obtainable nodes mutually shared by Cloud1 and Cloud2. α(Cloud1)∩α(Cloud2) is to calculate the general features betweenCloud1and Cloud2.

$$sim(Cloud1, Cloud2) = \left| \frac{\alpha(Cloud1) \cap \alpha(Cloud2)}{\alpha(Cloud1)} \right| -------------------1$$

$$sim(Cloud2, Cloud1) = \left| \frac{\alpha(Cloud1) \cap \alpha(Cloud2)}{\alpha(Cloud2)} \right| -------------------2$$

$$sim(Cloud1, Cloud2) = \rho \left| \frac{\alpha(Cloud1) \cap \alpha Cloud(2)}{\alpha(Cloud1)} \right| + (1-\rho) \left| \frac{\alpha(Cloud1) \cap \alpha Cloud(2)}{\alpha(Cloud2)} \right|$$

Compatibility Reasoning: (Write for 32 bit and 64 bit) the compatibility reasoning moderately achieve the high degree of similarity. Since if two siblings represents different version of software which has maximum similarity [19]. However they differ in terms of chronological ordering. To calculate the compatibility between two nodes(x & y) and described by chronological ordering as follows:

$$compatibility(Cloud1, Cloud2) = sim(Cloud1, Cloud2) + \frac{(\mu^{|Comp_{cloud1} - Comp_{cloud2}|})}{\theta} -------------3$$

Where $Comp_{cloud1}$ and $Comp_{cloud2}$ represented by sim(Cloud1, Cloud2) for the label values x and y. respectively, defined in equation (1). $Comp_{cloud1}$and $Comp_{cloud2}$ has two different version of software and represents by chronological orderings. The eq. 1, 2 is coarse-grain measurement since Cloud1 and Cloud2 has two different versions of software but having high degree of similarity. $(\mu| Comp_{cloud1} - Comp_{cloud2}|)/\theta$ is the fine-grain measurement since Cloud1 and Cloud2 has small variations.

Ranking: The service ranking is accomplished by two forms, Service matching and Service ranking. The Service matching is a process of ordering the queries on each node. The ontology database is tending to retrieve the relevant information based on the service. The attributes are also taken into consideration [19]. Service Ranking has described as the obtained services are ordered and their attribute values are concerned. These attributes may be qualitative and quantitative. Some of the attributes are measurable where as others are based on users experience.

The Honey Bee Agent Model: The Scout Bee agents travel randomly across multiple clouds. The gathered knowledge was memorized and communicated to all other scout bee agents in the hive through waggle dancing mechanism [6]. The waggle dancing area has considered as virtual yellow pages or directory server. The sign of communication helps to identify i). Route to reach the cloud sources. (ii). Type of service available in cloud network. This process supports in converting scout bee agents into onlooker bee agents which means they are known to the resources in different type of clouds. The entire system follows a unique structure. The scout bee agents and starts searching spontaneously for services based on some internal inspiration or possible external clue [23]. Once the scout bees are identified the new services availability then it transformed as recruit bee agent. Subsequently for the private and dedicated cloud service, the bee agent operates its own capability to remember the location and then instantaneously starts publishing it. Therefore, the bee agent will become an "Employee forager bee agent". After exploiting the knowledge from employee forager bee agent, the onlooker bee agents travel directly to corresponding location for collecting the resources [24]. Each onlooker bee travel independently communicates with one another. The onlooker bee agent gathers information about type of resource available in cloud network and stability factor of that resource, time and cost of each resource etc., once all the required information gathered by the onlooker bee agent, they assembled in originated place where all the relevant resources are clustered together.

Resource Cluster: Cloud based services are diverse in nature that can be reasonably clubbing into clusters to expand their provision and use. The resource clustering mechanism is used to assemble multiple resource nodes that can functioned like single virtual cluster. This will augments the mutual load balancing and computing capability, and ease of use with grouped resources. Resource cluster mechanism falls under high speed dedicated network links between service instances for workload sharing, task aggregation, task scheduling [28].The cluster supervision working as dispersed middleware to overall cluster nodes that is characteristically accountable for these actions. The platform establishes a harmonization task which permits dispersed resources into single image. Then accomplishes resources within the single cluster. The resource clustering methodology has classified into four categories: i). Server based cluster: the physical and virtually distributed servers are equally clustered to increase the overall gain. The Hypervisors mounted at diverse substantial servers can be arranged to divide the virtual server execution state in order to initiate virtually clustered server. ii). Database Cluster is intended at civilizing data accessibility; this elevated availability has a harmonization characteristics that maintains the consistency of data being accumulated at special storage devices. iii).The Large Dataset Cluster is a data partitioning and sharing. The Intended data can be capably detached without any negotiating data security. iv). The Load balance cluster is the general resource cluster concentrated in distributing all the workloads between clustered nodes in order to augment capacity of resource through resource administration.

Hypervisor: The hypervisor is a method of virtualization arrangement that is mostly used to breed virtual server illustrations of a physical server. A hypervisor process is typically limited to one physical server. Hence only create virtual images of that server. Correspondingly a hypervisor can only allot virtual servers that makes to resource pools reside on the similar essential physical server [22]. Moreover the hypervisor process has partial virtual server supervision features, such as increasing the virtual server volumes. The Virtual Infrastructure Manager (VIM) delivers a range of features for governing numerous hypervisors across physical servers. The virtual servers are created via distinct hypervisor on separate physical servers.

Virtual Yellow Pages: The virtual yellow pages are scattered database of service clouds conserved and categorized by set of navigation honey bee agents. The navigation scheme should be mounted on each bee agents. The bee agents require either fractional model or fractional knowledge of both the task and the environment. This model deliver a novel scheme that gives system of virtual yellow pages to aid the bee agents decide where to go. The Directory server contain number of facilities and services as per the requirements by the cloud user. The bee agents select a set of services relevant for its task and articulate adaptive strategies to visit few resource clouds. The cloud resources are dynamic in nature[10]. Since the stability of virtual yellow pages has not static entities, by practicing adaptive learning methodology through waggle dance to keep the virtual yellow pages up to date. The virtual yellow pages have newly identified cloud sources that registered with one or more Scout bee agents to publicize their location. The Scout bee agents exhibit their service by the list of keywords. The recruit bee agent returns a list of corresponding services from its database for adaptive choice of best service, subsequently the scout bee agent looking few services to deliver feedback about the response time of sites and effectiveness of results. The overall strategy for identifying the ideal service we keep the average response of each service provider. Moreover this technique congregates the progress of dynamic system where resources appear and disappear by supplementing the first elite policy which inspires initial exploration of other scout bee agents.

V. THE PROPOSED ALGORITHM

In Honey bee discovery algorithm, each iteration consists of three steps: Dispatching the Employee forager bees onto their private and dedicated clouds and evaluating their resources for usage; After sharing the resource knowledge by waggle dance, the onlooker bee agents are distributed to the significant regions for evaluating the resources; on other hand the group of cloud sources is arbitrarily selected by the Scout bee agents and their resources are determined then transport them unsystematically onto tangible new cloud sources. The Greedy randomized adaptive search procedure is used to construct the first Scout bee agents. This is multi-iterative process [21], this includes two phases based on the heuristics function. A viable solution is produced in construction phase. In Local phase the best possible neighborhood identified using the outcome of construction phase. Finally the overall best solution kept as a finest result. The best possible result has generated by the construction phase through processing the series of iteration. At every round, the optimal choice of subsequent element will be incorporated. This will accomplished by ordering all Scout Bee agents in separate set referred as Set C with based on the greedy function g(s): C→R. The heuristic approach is formal since there imbursements linked with every individual element are updated dynamically. Consecutively the construction phase is tend reflect the variations fetched depend on the selection of previous section. The probabilistic element of a GRASP is exemplified by randomly deciding best agents in the given list. The unique list of best agents is referred as Limited Bees List (LBL). Let a α[0;1]. The pseudo code given below moderates a basic GRASP construction phase. Procedure construct (g(.),α, R)

Algorithm:

A. Construction of Initial Scout Bee Agents

1. *Initiate parameters (g(.),α, R)*
2. *Resource Count (R) = 0;*
3. *Initiate Scout bee agents set C;*
4. *While set C≠0 do*
   Initiate the Local Search (f(.),N(.),R, H_u)
   a. $H_u = \{(y \in N(x) | f(y)<f(x))\}$;
      i. *while* $|H_u| >0$ *proceed*
      ii. *Accumulate* $R \in H_u$;
      iii. $H_u = \{(y \in N(y) | f(y)<f(x))\}$;
   b. *end while;*
   c. *g(s) - Min Scout Bee Agents for Far Neighborhood cloud search*
   d. *g(s`) -Max Scout Bee Agents for Near Neighborhood cloud search*
   e. $LBL = \{(s \in C \mid g(s) \leq s` + \alpha(s` - s))\}$;
   f. *Randomly select s, from LBL;*
   g. $R = R \cup \{s\}$;
   h. *update the Scout bee agents in set C;*
5. *End While;*
6. *End Local;*
7. *End Construction;*

The above pseudo-code describes that α parameter is sum of greediness and arbitrariness in the given algorithm. The given value α = 0 exemplifies the greedy construction and α = 1 generates random construction. The heuristics construction procedure GRASP involves many feasible solutions .However this is not locally optimal based on the simple neighborhood solution. The local search optimization generated by iterative fashion through continuously substituting the current resolution with the finest solution in the respective neighborhood current solution. The Near neighborhood structure referred as N(x) and Far neighborhood structure N(y) that relates to the given solutions. The R defines the resource availability and H_u refers the Heuristics. The success factor is depending on the suitable option of a near and far neighborhood structure. The local search optimization mechanism requires exponential time from a random initial point; empirically their efficiency improves when the initial found solution improves. To cover the local optimization procedure through random start, the GRASP solutions requires equal period of time at every initial point. Furthermore the GRASP solution is significantly improved than the single solution acquired by a random starting point.

The set of task I ={1,2…,n}; The set of Onlooker agent J={1,2,…,m}; b_j = maximum resource volume of agent j; c_{ij} = cost of the task I assigned to each j; a_{ij} = resource needed for the task allocated to onlooker agent j; x_{ij} = decision variable (x_{ij} = 1 if the job assigned to onlooker agent j; 0, otherwise);

B. Initialization and evaluation of fitness values between Scout bee agent and Employee forager bee agent

1. *Initialize parameters*
2. *Construction of Initial Employee Forager Bee Agent using Greedy randomized Adaptive Search Procedure (GRASP)*
3. *Local search and Evaluate fitness of Initial Scout Bee Agent*
4. *I=0*
5. *Repeat*
6. *N=0*
7. *Repeat*
   a. *Near Neighborhood cloud search*
   b. *Far Neighborhood cloud search*
   c. *Calculate fitness value using their probabilities*
   d. *Assignment of Scout Bee Agent to Recruit Bee Agent according to mutual probabilities*
   e. *For all Scout Bee Agent*
      i. *Migration of Neighborhood cloud structure*
   f. *Find the best Recruit Bee Agent, replacement of respective Onlooker Bee Agent*
   If fit(Best Recruit Bee Agent)<fit(Onlooker Bee Agent)
   g. *Find the Feasible Recruit Bee agent , replace with Best Solution,*
   If fit (BestFeasible Onlooker Bee Agent)<fit(Best)
   h. *N=N+one;*
8. *Until (N= Employee Forager Bee Agent)*
9. *I=I+one;*
10. *Until(I=Maximum_Iteration)*

C. Evaluation of fitness values between Employee forager bee agent and Onlooker bee agent

0. *Parameter Initialization*
1. *n = Total number of Employee Forager bee Agents*
   a. *m = Total number of Onlooker bee Agents(m>n)*
   b. *I = Maximum Iteration Number*
   c. α_j *= Penalty control parameter for* j^{th} *agent*
   d. *Migration-Length : Length of Neighborhood cloud Structure*
2. *Initialize Employee Forager Bee Agent with GRASP Algorithm.*
   σ^i *:* i^{th}*Employee Forager Bee Agent in the entire population*
3. *Evaluate Employee Forager Bee Agents Fitness Function for Minimization*

$$\sum_{j=1}^{m}\sum_{i=1}^{n} c_{ij}x_{ij} + \alpha\sum_{j=1}^{m}\max\left\{0, \sum_{i=1}^{n} b_{ij}x_{ij} - a_j\right\}$$

4. *Repeat*
   Cycle = 1
   1. *Number of Scout Bee Agents = 0,1*n*
   2. *For each Employee Forager Bee Agents*
   a. *Apply Near Neighborhood cloud search*
      i. *If fit (NearNeighbour)<fit(Employee Forager Bee Agent) then*
      Employee Forager Bee Agents = Near Neighbour
   b. *Apply Far Neighborhood cloud search*
      i. *If fit(FarNeighbour)<fit(Employee Forager Bee Agent) Then*
      Employee Forager Bee Agents = Far Neighbour
   c. *Calculate their probabilities depend on the fitness.*
   d. $P_i = \dfrac{\sum (1/fit_i)^{-1}}{fit_i}$ *(For Minimization)*
   e. *Identify the possible bee agents (onlooker bee agents) that sent to food patches discovered by Employee forager Bee agents, According to earlier determined probabilities.*
   f. N_i *= Number Onlooker Bee Agents send to the* i^{th}*Cloud =* P_i**m*
   g. O_{ij} *:* j^{th} *Onlooker Bee Agent of* i^{th} *solution (j=1,..,*N_i*)*
      $\{O_{i1}, O_{i2}, \ldots O_{iNi}\}$ *= Migration of Neighbourhood clouds*
   h. *Determine the significance of fitness of each Onlooker Bee Agent*
   If overall fitness value and stability of bee agents(Onlooker bee agents) is improved than the suitability factor of Employee forager Bee agent solution then replace with Onlooker Bee Agents solution.
   If (min(fit(O_{ij}*))<fit(*σ^i*)then* $\sigma^i = O_{ij}$
5. *Best solution*
   If fit($Best_{cycle-1}$*)>Min(fit(*σ^i*))i=1,..,n then* $Best_{cycle} = \sigma^{i\ i0}$
   Else $Best_{cycle} = Best_{cycle}$*-1*
   Until (i=n)
6. *Scout Bee Agents*
   a. *Initialize the Scout Bee Agents using GRASP Algorithm*

b. *The total number of worst Employee Forager Bee Agents in the total population is compared with scout bee agent solutions.*

c. *If the current scout bee agent solution is improved than Employee forager agent solution then replace the recent solution with scout bee agent solution. Else consider Employee Forager Bee agent solution is reassigned to the next round without making any further change.*

7. *Cycle = Cycle +one*
8. *Until (Cycle = Total_Iteration)*

The initial Employee Forager Bee agent in the colony is built based on Greedy randomized Adaptive Search Procedure. The local search has multiple iteration and to find basic optimal solution. It is works with likelihood basis to the mutual probability function. The local search is optimized at each level until find the good solution.

D. Local Search optimization.

1. *Let S_j = Ø $\forall j = 1,...m$ (S_j is task given to the agent j)*
2. *Build a separate list of Onlooker Bee Agents for every task L_i., firstly L_i =(1,…,m) $\forall i$.*
3. *Consider random order of the following tasks, i=1;*
4. *While (no task have been allotted)repeat*
   a. *Choose randomly any Onlooker Bee agent j* from L_i using the mutual probability function that rely on the type of bee agent j and the required resource for the task i;*

$$P_{ij} = \frac{a / b_{ij}}{\sum_{l \in L_i} a_i / b_{il}}, \; j \in L_i$$

   The probability of minimal cost of The Onlooker Bee agent is being selected.

   Allocate current task i with Onlooker Bee Agent j $j^* : Sj = Sj^* \cup \{i\}$.

   Let i=i+ one;

   if $\sum_{i \in S_j} b_{ij^*} > a_{j^*}$ discard j from given list.*

   Repeat step four
5. *Let $\sigma(i) = j$ if $i \in S_j$*

E. Neighborhood Cloud Structure

The migration of near neighborhood structure is relying on the resource required to complete the given task. The far neighborhood structure is a unique case that, if no optimal solution discovered in near structure. ∂- Migration of Near Neighborhood structure, ∂`- Migration of far Neighborhood structure

Migration (∂)

1. *Let S = {i | i∈{1,…,n}} , k=1 Migration to Neighborhood = ∂*
2. *If S = Ø else end; otherwise i_k is ejected from ∂_k . S=S-{i_k}*
3. *Let j* is onlooker agent j which minimizes*

$$c_{ikj} + \alpha_j \max\{0, (\sum_{i \in l, \partial(i)=j} a_{ij}) + a_{ikj} - b_j\}$$

   Among all agents $j \in J \setminus \{\partial(i_k)\}$
4. *Assign i_k to j*, Output ∂`, Calculate Fitness ($\partial$`)*
5. *If Fitness (∂`) < Fitness (Migration to Neighborhood) the Migration to Neighborhood = $\partial$`*
6. *k: k+1, return to step 2*
7. *Output Migration to Neighborhood.*

After, the optimal solution found the resultant resources information are stored and maintained by the central hive, which considered as virtual yellow pages or Directory server. The avail function defines the resultant amount of resources in the directory.

$$avail(i) = \begin{pmatrix} a_{i,\partial(i)} - P_{\partial(i)}(\partial) \\ a_{i,\partial(i)} \end{pmatrix} \text{if } a_{i,\partial(i)} > P_{\partial(i)}(\partial)$$

VI. EXPERIMENTAL SETTINGS

The Requests and advertisements from scout bee agents are produced randomly and comprise the information of the distinctive resources like CPU speed, network bandwidth, disk capacity, OS type, and network latency. The cloud user query where negotiated by query processing module and provide the service in available timeslots. The experiments and evaluation that commenced in order to enumerate the competence of Cloudsim using JXTA Protocol standard in exhibiting and simulating cloud computing environments. The investigation was accompanied on a Celeron machine with the configuration of 2.00GHz with 7MB of L2 cache and 4 GB of RAM running a standard Ubuntu - Linux version 8.04 and JDK 1.6.0.05. To evaluate the above standards a virtual cloud computing environment required which includes three types of data centers (private, public and Hybrid), generally there are three form of honey bee agents were involved in resource discovery in heterogeneous cloud environments.

The perspective of cloud user and variety of providers there are series of experiments were performed. The total number of hosts in each data center in each experiment was diverse from 100 to 100000. As the objective of these tests was to assess the time taken to discover the services (resources) located at different types cloud environments. Moreover no consideration was given to the user workload. The average setup time in deploying the simulation environment is the time variance between the subsequent events: (i) the overall time taken for Cloudsim to load the runtime environment (Java virtual machine). (ii) The illustration at which Cloudsim entities such as Bee agents and components are fully prepared and ready to progress events. (iii) The time taken to initiate the number of hosts in a data center increases, depending on the total host loaded subsequently the traffic load also increases[27]. The inner performances of the system can be assessed by virtualization policy in each data center of service provider, which consist number of virtual machines and maximizing the fragmentation of universal operation (the discovery operation).

VII. EVALUATION METRICS

The following metrics were used in the evaluation process. The total number of discovered services DiscvSer*- the scout bee agents. The number of relevant services to be clustered DiscvServClus*- onlooker bee agents. The number of discovered services in private clouds EmpDiscvServ*- Employee forager bee agent. The overall service arrangement is DiscvSerComp*. The number of cloud user queries and discovered services based on the requirements DCServ*(overall inputs, outputs, prerequisites and operational parameters of the discovered services). The cloud user query has evaluated by the query processing engine and reasoning module and finally rated on raking basis. This enriches that the cloud user query need to discovered or if exists that will directly utilized from the database which is directly connected with virtual yellow pages or directory server. The service model depicts the cloud service offered from different cloud environments, which semantically described as a WSML web service and the competence is mentioned in terms of its nature and the proficiencies.

Evaluation scenario 1: The physically well-defined cloud services these services are accessed by default Employee forager bee agents. Moreover it is semantically annotated by private clouds or dedicated clouds which consider in this scenario that entire services are meaningful. (Fig. 2)

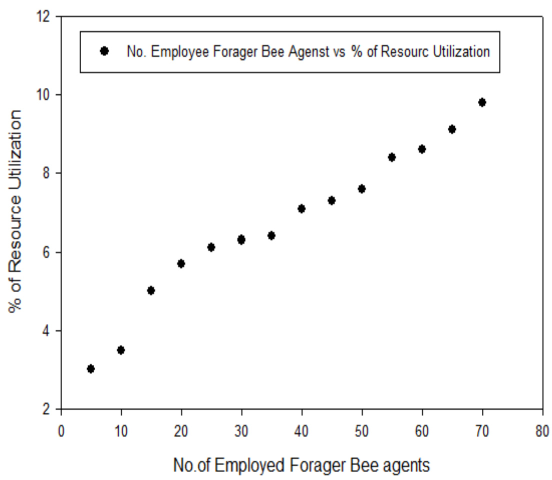

Fig.2. Percentage of utilized resources by Employee forager bee agents.

Evaluation scenario 2: The scout bee agents travel randomly and collect best possible cloud services. According the simulation work these services are automatically generated using a special tool "RandServGen". This tool produces a number of services, interpreted using random semantic concepts prescribed by ontologies. However these services are restricted to limited numbers. The Employee forager bee agents are dedicated to private clouds where they utilized permanent resources.

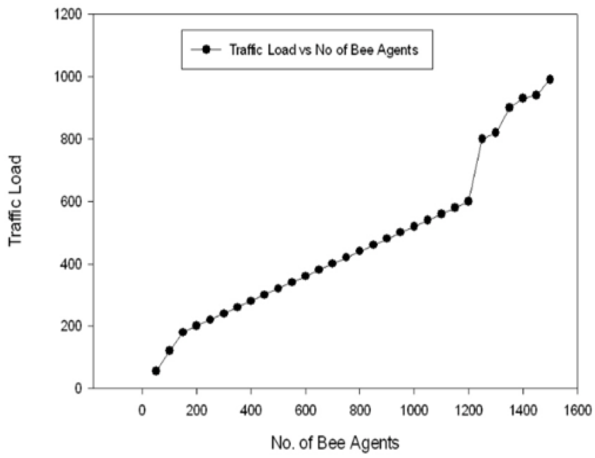

Fig.3. Traffic load based on number of Bee agents

Evaluation scenario 3: The total traffic load is directly proportional to the total number bee agents. When the scout bee agents and onlooker bee agents increases in turn this will create impact on traffic load. (Fig. 3) The traffic load gradually increases and if the number cloud user requests increases in turn this will affect network utilization.

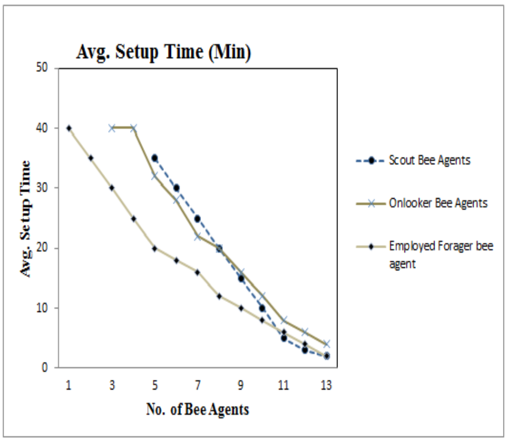

Fig.4. Average setup time for all three types of agents

Evaluation scenario 4: Each individual bee agent can travel independently and interconnect with other agents. Henceforth each bee agent can have different scheduling to connect with cloud resources. The knowledge revealed by the Scout bee agents and Employee forager bee agent to be delivered to onlooker bee agents through 'waggle dance'. The average setup time is evaluated based on above three conventions. (Fig. 4)

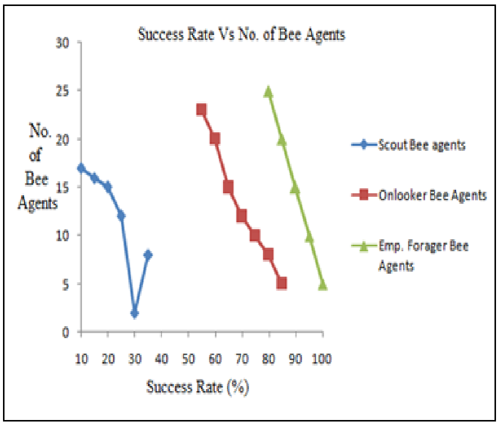

Fig.5. The Success rate on all three types of agents

Fig.6. Failure rate based on number Employee forager bee agents

Evaluation scenario 5: The effectiveness is calculated depend on the success rate of every individual bee agent. The success rate is accomplished by evaluating the cloud user quires and time taken to deliver the service. The failure rate is being calculated depend on the exploitation of different bee agents. Once Employee forager bee agents and utilization of onlooker bee agents increases obviously that reduce the failure rate. (Fig. 5-6)

VIII. CONCLUSION

The originality and implication of the above presented work are, inaugurating the idea of applying the agent mechanism in building honey bee stimulated discovery framework for cloud service discovery. This work enriched by two novel ways: by originating the arena of cloud based resource management by inventing numerous novel attitudes for enabling cloud service discovery, the negotiation of various cloud service and cloud service assignment. In multi-agent point of view, this work validates the solicitation of i) supportive problem solving paradigms using bee agents to mechanizing cloud service configuration, ii) composite and synchronized consultations to cloud SLA's and iii) Bee agents to construct a cloud service discovery engine. The influences of this proposed work as follows: 1. User query is handled by the query processing unit. Then the query with compare with preexistence services which is already available in the database. Depends on the maximum usage the corresponding service to be rated. The cloud negotiation module establishes the SLA's demand factor, availability and time duration etc., the uniqueness of query processing unit is a multi standards search engine module which gathers various requirements from variety of cloud users. The reasoning module enables the similarity among cloud services and regulates various levels of equivalent between the corresponding availability and plans between cloud user and a cloud service provider. A typical cloud brokerage model was invented to sustain cloud economical part. The negotiation mechanism already detailed in cloud markets. This knowledge is extracted from the previous work. This proposal has only equipped on Bee agent based problem solving methods and protocols mentioned in increase the usefulness for handling cloud resources discovery.

REFERENCES

[1]. K.M. Sim. "Agent-Based cloud computing". IEEE transactions on services computing. Vol.5,No. 4.November 2012,pp.564 – 577.

[2]. C. Jingliang, Z. Ming. "Research on web service discovery based on mobile agents". In proceeding of

thenternational conference on computer design and applications. Qinhuangdao. 2010, pp. 385-388.

[3]. S. Sotiriadis, N. Bessis, Y. Huang, P. Sant."Towards decentralized grid agent models for continuous resource discovery of interoperable grid virtual organizations digital information management". In proceeding of the Fifth International Conference on Digital Information Management (ICDIM).Thunder Bay, ON. 2010,pp.530–535.

[4]. S. Hamza, K. Okba, B. Aicha-Nabila, A. Youssef ." A Cloud computing approach based on mobile agents for Web services discovery". In roceeding of the Second International Conference on Computing Technology(INTECH).Casablanca. 2012, pp. 297 – 304.

[5]. D.T. Pham, E. Kog, A. Ghanbarzadeh, S. Otri, S. Rahim, and M. Zaidi, "The Bees Algorithm – A Novel Tool for Complex Optimization Problems," In proceeding of second international virtual conference on Intelligent Production Machines and Systems IPROMS, Oxford, Elsevier, 2006.

[6]. A. Zhou, B-Y. Qu, H. Li, S. Z. Zhao, P.N. Suganthan, Q. Zhang. "Multi objective evolutionary algorithms: A survey of the state of the art". Swarm and evolutionary computation; Vol.1, No.1. March 2011,pp. 32-49.

[7]. R. Das, J. O. Kephart, C. Lefurgy, G. Tesauro, D.W. Levine, and H. Chan, "Autonomic multiagent management of power and performance in data centers," In proceeding of AAMAS'08. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp 107–114.

[8]. Mauro Andreolini, Sara Casolari, Michele Colajanni, Michele Messori, " Dynamic load management of virtual machines in a cloud architectures", Cloud Comp Munich, Germany, 2009, pp. 201-21.

[9]. Anton Beloglazova, Jemal Abawajyb, Rajkumar Buyyaa, " Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing", Future Generation Computer Systems, Vol. 28, Issue 5, May 2012, pp 755–768

[10]. J. Wei, M. Jun, Z. Xiu, X. Huan." Research on cloud manufacturing resource integrating service modeling based on cloud-Agent". In proceeding of the IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS). Beijing 2012, pp.395 -398.

[11]. D. Barbagallo, E. Di Nitto, D. J. Dubois, and R. Mirandola, "A Bio-inspired self- organizing data center," in proceeding of SOAR'09. Berlin, Heidelberg: Springer-Verlag, 2010, pp127–151.

[12]. W. Peter, L.Yin et al., "Constraints-based resource discovery model for multi-provider cloud environments". Journal of Cloud Comput Adv Syst.Vol.1, No.6. June 2012, pp.2-14.

[13]. K. M. Sim, "Agent-based cloud commerce," In proceeding of IEEE International conference Ind. Eng. Eng. Manage., Hong Kong, 2009, pp. 717–721.

[14]. K. M. Sim and B. Shi, "Concurrent negotiation and coordination for grid resource coallocation," IEEE Trans.Syst., Man, Cybern. B, Cybern.vol. 40, no. 3, pp. 753–766, May 2010.

[15]. V. SashiKanth, N. Agireddi, M. Shakti."An ontology based cloud service generic search engine". In proceeding of the 8th international conference on computer science &education (ICCSE'13).Colombo2013,pp. 335-340.

[16]. Jinhua Hu, Jianhua Gu, Guofei Sun, Tianhai Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", In proceeding of third International Symposium on Parallel Architectures, Algorithms and Programming, 2010 pp89-96.

[17]. H. Taekgyeonget al, "An ontology-enhanced cloud service discovery system". In Proceeding of international multi conference of engineers and computer scientists. Hong Kong 2010, pp. 17-19.

[18]. D. Barbagallo, E. Di Nitto, D. J. Dubois, and R. Mirandola, "A bio-inspired algorithm for energy optimization in a self-organizing data center," In proceeding of SOAR'09. Berlin, Heidelberg: Springer-Verlag, 2010, pp127–151.

[19]. L. Xiaoyong, D. Junping et al., "Adaptive and attribute-based trust model for service level agreement guarantee in cloud computing". IET Inf. Secur.Vol. 7. No. 1 March2013, pp. 139–50.

[20]. Yee Ming Chen, Shin-Ying Tsai, "Optimal Provisioning of Resource in a Cloud Service", International Journal of Computer Science Issues, Vol. 7, No 6, 2012, pp95-99.

[21]. S. Muhammad et al.,"Swarm intelligence based routing protocol for wireless sensor networks: survey and future directions Information science". J. Information Sci. Vol. 181. No. 20, October 2011,pp. 4597-4624.

[22]. Oracle Enterprise Transformation solutions series Cloud reference architecture an oracle white paper, Nov.2012.

[23]. A. Reza, H. Ramin, Z. Koorush, H. Bahareh. "A multi-objective artificial bee colony algorithm". Swarm and Evolutionary Computation. Vol. 2. No.1 February 2012,pp. 39–52.

[24]. K. Dervis, A. Bahriye."A comparative study of Artificial Bee Colony algorithm". Appl Math Comput. Vol. 214, No.1August2009, pp. 108–132.

[25]. G. Yongqiang, G. Haibing,Q. Zhengwei, H. Yang, L. Liang. "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing". J Comput Syst Sci. Vol. 79, No. 8 December2013, pp. 1230–1242.

[26]. K. Rajiv Gandhi et al, "A Hybrid Meta Heuristic Algorithm for Discovering Classification Rule in Data Mining". Inter. J. Comput. Sci. Net. Secur. Vol. 12, No. 4 March 2012, pp. 117-122.

[27]. L. Guanghui et al., "On the effectiveness of incorporating randomness and memory into a multi-start meta heuristic with application to the Set Covering Problem". J. Comput. Industrial Eng. Vol. 51. No. 3.November 2006, pp.362–374.

[28]. J. Li. "Grid resource discovery based on semantically linked virtual organizations". Future Gener Comput Syst. Vol. 26, No. 3 March 2010, pp.361-373.