BRILLIANT APPROACH FOR LOAD BALANCING IN CLOUD COMPUTING

M.Padmavathi
Department of Computer Science, Swarna Bharathi Institute of Science and Technology
macherlapadmavathi@gmail.com

Shaik. Mahaboob Basha
Principal, Alhabeeb Institute Of Science And Technology
principal@alhabeebcollege.ac.in

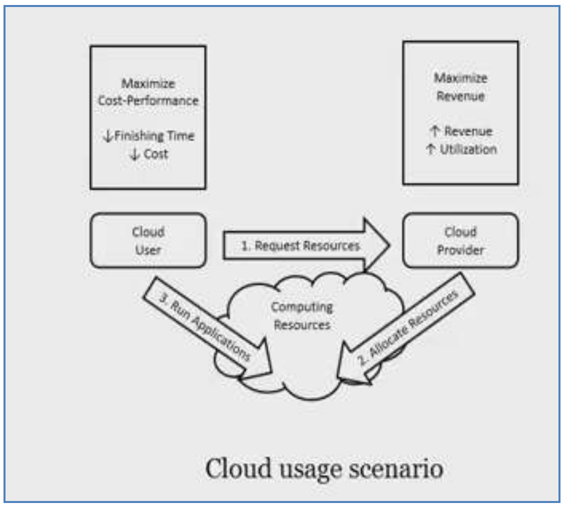
Abstract:

Cloud is a parallel and distributed computing system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing Resources based on Service-level Agreements (SLA) established through Negotiation between the Service Provider and Consumers. Virtualization is like something that is not real but provides all the facilities that are of real world. Virtualization is a part of Cloud Computing, because different services of cloud can be used by user .Many Cloud task schedulers had discovered to perform Scheduling task but most of them are static, not elastic, not economic and time consuming. In this paper a cloud computing scheduling policy has been proposed to minimize the imbalance between the work loads of different systems.

1. INTRODUCTION

1.1 Cloud Computing:

Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.



1.1.1 Cloud Computing Services:

A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold on demand, typically by the minute or the hour. It is elastic and a user can have as much or as little of a service as they want at any given time and The service is fully managed by the provider (the consumer needs nothing but a personal computer and Internet access).

. Although cloud computing has evolved over the time it has been majorly divided into three broad service categories: Infrastructure as a Service (IaaS), Platform as a Service:(PaaS) and Software as a Service (SaaS) which are broadly discussed below: Storage as a Service (SaaS) Communications as a Service (CaaS) Network as a Service (NaaS) Network as a Service (NaaS) Network as a

Service (NaaS) Monitoring as a Service (MaaS).

Software as a Service (SaaS)

Software as a Service(SaaS) is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet. Software as a service (SaaS)is a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted. It is sometimes referred to as "on-demand software", and was formerly referred to as "software plus services".

Platform as a Service (PaaS)

Platform as a Service (PaaS)delivers hardware and software tolls, usually those needed for application development, to its clients as a service. cloud user can execute programs without install in-house hardware and software. PaaS doesn't replace a business' entire infrastructure but instead, a business relies on PaaS providers for key services, such as Java development or application hosting. A PaaS provider, however, supports all the underlying computing and software; users only need to log in and start using the platform-usually through a Web browser interface.

Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) provides virtualized computing resources, a third party provider provides hosts hardware, Infrastructure, software and storage services. Other characteristics of IAAS environments include the automation of administrative tasks, dynamic scaling, desktop virtualization and policy-based services .IAAS providers also host users applications and handle tasks including system maintenance backup and resiliency planning. virtualization plays a dominant role in Iaas.

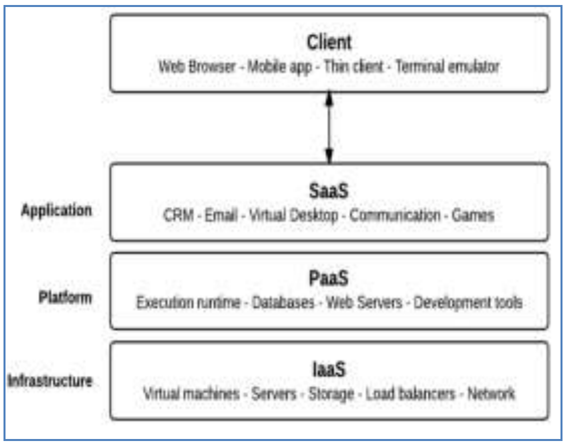

Client
Web Browser - Mobile app - Thin client - Terminal emulator
Application
SaaS
CRM - Email - Virtual Desktop - Communication - Games
Platform
PaaS
Execution runtime - Databases - Web Servers - Development tools
Infrastructure
IaaS
Virtual machines - Servers - Storage - Load balancers - Network

Fig 1.1: cloud computing basic services

1.1.1.2 Virtualization:

Virtualization is a broad concept and it creates virtual resources like hardware, software, platform, storage, desktop, operating system and network resources etc. In cloud computing technology, virtualization plays a very key role to share the infrastructure in the clouds. To overcome this problem basically virtualization technology was used. By using virtualization, all severs and the software applications which are required by other cloud providers are maintained by the third party people, and the cloud providers has to pay the money on monthly or annual basis. Virtualization can be categorized as follows and the brief taxonomy can be described by the following figure. Emulation means system pretend to be another system, where as virtual machine acts to be more than one machine.

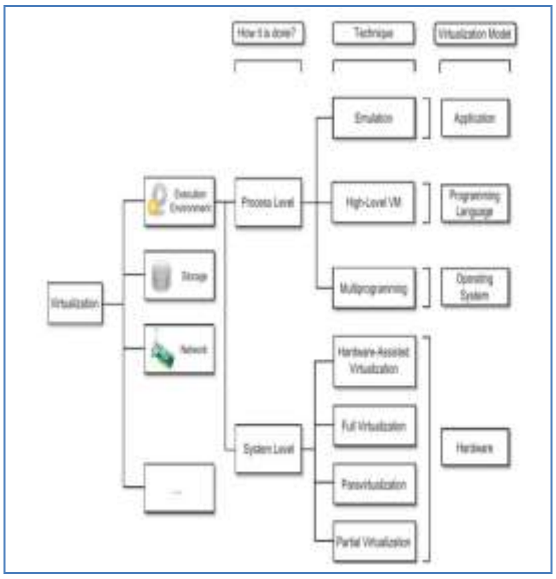


Fig 1.2: overview of virtualization

2. LOAD BALANCING:

A load balancer distribute work load in a manner that maximizes speed and capacity utilization and ensures that no one server is overworked or under loaded, which could degrade performance. If a single server goes down, the load balancer redirects traffic to the remaining online servers. When a new server is added or deleted from to the server group, the load balancer automatically starts to send requests to it.
Load balancing workflow can be summarized as following steps:

1. Cloud let is a connection of multi no of Systems and it stores the entire user Requests and it is Connected to the Datacenter Controller (DCC).
2. Different Cloud users from different locations may send request to the cloud let.
3. Primary Tasks assigned to Virtual machines after wards work load allotted to the Virtual machine with minimum work load.
4. Load Balancer can maintain a table, which stores Virtual Machine Id.

Various Parameters to be considered in Load Balancing: The Existing Load Balancing Algorithms has to consider the following issues.

Table 2.1: Various parameters to be considered in load balancing

Parameter Name	Description	Value to be
Overhead Associated	It is composed of overhead due to movement of tasks, inter-processor and inter-process communication.	minimized
Throughput	No. of Tasks whose Execution has been Completed.	maximized
Resource Utilization	Utilization of Resources.	maximized
Scalability	Ability of an Algorithm to perform load balancing for a System with any finite Number of Nodes.	Improved
Response Time	Amount of time taken to Respond by a particular Load Balancing Algorithm in a Distributed System.	minimized
Fault Tolerance	An algorithm to perform uniform load balancing in	maximized

	spite of arbitrary node or link failure	
Migration time	Time to Migrate the jobs or resources from one node to other	minimized

3. PARTICLE SWARM OPTIMIZATION (PSO):

PSO relies on the collective behavior of a colony of insects, such as ants, bees etc; a flock of birds; or a school of fish. The word particle represents, for example, a bee in a colony or a fish in a school. Each individual in a very swarm behaves mistreatment its own intelligence and the cluster intelligence of the swarm. As such, if one particle discovers a good path the remainder of the swarm also will be ready to follow the great path instantly even if their location is far away in the swarm.

The following figure depicts how PSO operation can be transferred to the technical aspect; first identify the similarities between the operations and find out the generic behavior of different PSO algorithms and transform to technical aspects.

The PSO firstly generates a random initial population of particles, every particle represents a possible resolution of system, and every particle is delineated by three indexes: position, velocity, fitness. Initially each particle is assign a random velocity, in flight, it dynamically adjusts the velocity and position of particles through their own flight experience (personal best position), as well as their neighbor's (global best position). Thus, the whole group will fly to the search region with higher fitness through continuous learning and updating. This process is repeated until reach the maximum iterations or the predetermined minimum fitness

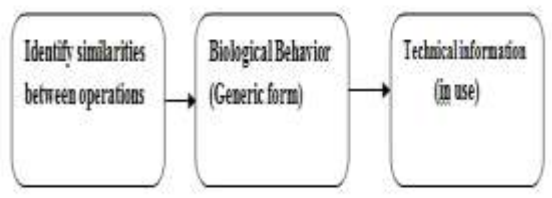

Identify similarities between operations
Biological Behavior (Generic form)
Technical information (in use)

Fig 3.1: General taxonomy of PSO Algorithm

The PSO firstly generates a random initial population of particles, every particle represents a possible resolution of system, and every particle is delineated by three indexes: position, velocity, fitness. Initially each particle is assign a random velocity, in flight, it dynamically adjusts the velocity and position of particles through their own flight experience (personal best position), as well as their neighbor's (global best position). Thus, the whole group will fly to the search region with higher fitness through continuous learning and updating. This process is repeated until reach the maximum iterations or the predetermined minimum fitness

Steps in PSO algorithm can be briefed as below:

1) Initialize the swarm by assigning a random position in the problem space to each particle

2) Evaluate the fitness function for each particle.

3) For each individual particle, compare the particle's fitness value with its pbest . If the current value is better than International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-2, May 2012 141 the pbest value, then set this value as the

pbest and the current particle‘s position, xi, as pi.
4) Identify the particle that has the best fitness value. The value of its fitness function is identified as guest and its position as pg.
5) Update the velocities and positions of all the particles using (1) and (2).
6) Repeat steps 2–5 until a stopping criterion is met (e.g., maximum number of iterations or a sufficiently good fitness value).

Table 3.1: Bio-inspired Algorithms with cloud computing Applications

S. No	Natural Tendency	Name of the Bio-inspired Algorithm	Cloud computing Application
1	Flash property	Firefly Algorithm(FA)	Tasks execution based on priority
2	Foraging(Searching for food)	AntColonyOptimization(ACO) Artificial Bee Colony(ABC)	Load balancing and Scheduling
3	Breeding(lays eggs in other birds nest)	Cuckoo search	Load balancing ,Scheduling and best node selection
4	Evolution based(generate offspring)	Genetic Algorithms	Work flow and VM Management

4. ANT COLONY OPTIMIZATION (ACO)

Each ant starts from a randomly selected city (vertex of the construction graph). Then, at each construction step it moves along the edges of the graph. Each ant keeps a memory of its path, and in subsequent steps it chooses among the edges that do not lead to vertices that it has already visited. An ant has constructed a solution once it has visited all the vertices of the graph. At each construction step, an ant probabilistically chooses the edge to follow among those that lead to yet unvisited vertices. The probabilistic rule is biased by pheromone values and heuristic information: the higher the pheromone and the heuristic value associated to an edge, the higher the probability an ant will choose that particular edge.

Once all the ants have completed their tour, the pheromone on the edges is updated. Each of the pheromone values is initially decreased by a certain percentage. Each edge then receives an amount of additional pheromone proportional to the quality of the solutions to which it belongs (there is one solution per ant).
This procedure is repeatedly applied until a termination criterion is satisfied.
The following are the two basic techniques in Ant System(AS)
(1) Ant generation. Check the cloud platform periodically, and generate ants if and only if there are existing overload nodes or underload nodes; and

(2) To find target nodes. According to searching rules, the ant is looking for the target nodes which meet the conditions of load balancing in its surrounding area. The target node is also called the candidate node for load balancing.

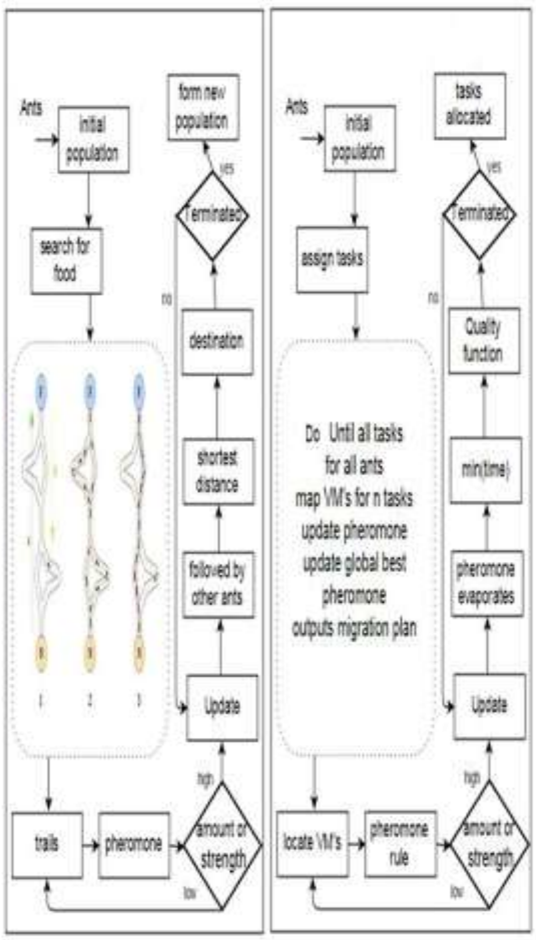


Fig 4.1:(a)ACO behavior inspired model deployed in (b) cloud process

ACO Basic Functionalities:

Pheromone tables

We replaced the routing tables in the network nodes by tables of probabilities, which we will call 'pheromone tables', as the pheromone strengths are represented by these probabilities. Every node has a pheromone table for every possible destination in the network, and each table has an entry for every neighbor.

For example, a node with four neighbors in a 30-node network has 29 pheromone tables with four entries each. One could say that an n-node network uses n different kinds of pheromones. The entries in the tables are the probabilities which influence the ants' selection of the next node on the way to their destination node.

Local search:

Local search starts from some initial solution and repeatedly tries to improve the current solution by local changes.

steps involved in local search

step1 : The definition of a neighborhood structure over the set of candidate solutions.

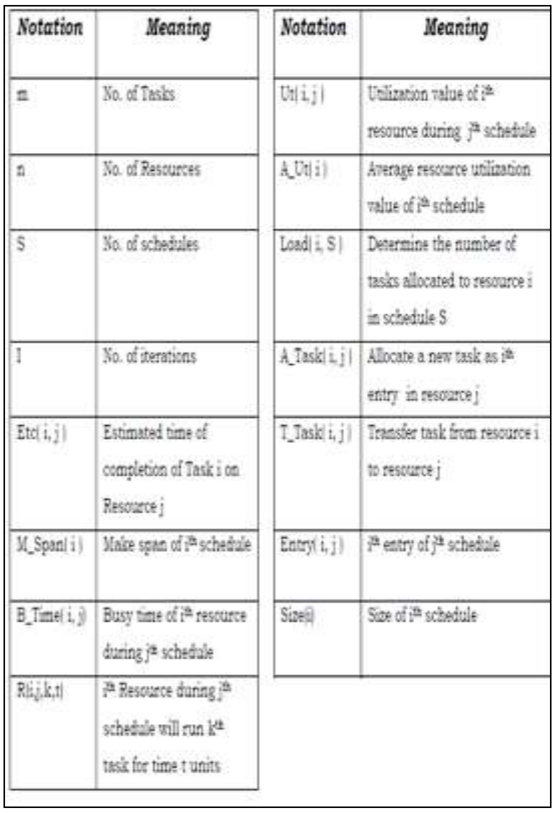
neighborhood structure :it defines for each current solution the set of possible solutions to which the local search algorithms can move.

Ex: iterative improvement, hill climbing, gradient-descent for maximization and minimization problems, respectively,

Step 2:The local search algorithm searches for an improved solution within the neighborhood of the current solution.

Step 3: If an improving solution is found, it replace the current solution and the local search is continued.

These steps are repeated until no improving solution is found in the neighborhood and the algorithm terminates in a local optimum.



Notation	Meaning	Notation	Meaning
m	No. of Tasks	Ut(i, j)	Utilization value of i^{th} resource during j^{th} schedule
n	No. of Resources	A_Ut(i)	Average resource utilization value of i^{th} schedule
S	No. of schedules	Load(i, S)	Determine the number of tasks allocated to resource i in schedule S
I	No. of iterations	A_Task(i, j)	Allocate a new task as i^{th} entry in resource j
Etc(i, j)	Estimated time of completion of Task i on Resource j	T_Task(i, j)	Transfer task from resource i to resource j
M_Span(i)	Make span of i^{th} schedule	Entry(i, j)	i^{th} entry of j^{th} schedule
B_Time(i, j)	Busy time of i^{th} resource during j^{th} schedule	Size(i)	Size of i^{th} schedule
R(i,j,k,t)	i^{th} Resource during j^{th} schedule will run k^{th} task for time t units		

Fig: Notations in ACO Load Balancing Algorithm

The ACO meta heuristic is:

```
Set parameters, initialize pheromone
trails
SCHEDULE_ACTIVITIES
  ConstructAntSolutions
  DaemonActions    {optional}
  UpdatePheromones
END_SCHEDULE_ACTIVITIES
```

5. CONCLUSION AND FUTURE SCOPE

In this paper basics issues of cloud computing and load balancing are discussed .In addition to that, the load balancing techniques issues based on Swarm intelligence has been discussed. Ant Colon Optimization ((ACO) issues and functionalities are discusses. We have discussed how the ACO functionalities can be applicable for load balancing. The research work can be proceeded to implement the modified ACO functionalities to load balancing algorithm. Objective for this paper is to develop an effective load balancing algorithm using Ant colony optimization technique to maximize elasticity and dynamic nature to provide services to the clients and to minimize different performance parameters like CPU load, Memory capacity, network load for the clouds of different sizes.

In this paper, phenomenon updating, tabu search and local search concepts based on ant colony optimization has been discussed and hence it is proved that ACO algorithm can be used to develop load balancing and scheduling algorithms. As part of the future work an efficient load balancing algorithm can be developed.

REFERENCES:

[1] DanchoDanchev,"Building and Implementing a successful Information Security Policy" Windowsecurity.com- Windows Security Resources for IT admins.

[2] David Escalante and Andrew J. Korty, Cloud Services: Policy and Assessment, EDUCAUSE Review,vol. 46, no. 4 (July/August 2011)

[3] Richard N. Katz, "Looking at Clouds from All Sides Now", EDUCAUSE Review, vol. 45, no. 3(May/June 2010): 32-45

[4] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A Practical Approach, TATA McGRAW-HILL Edition 2010.

[5] Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.

[6] Mladen A. Vouk, Cloud Computing Issues, Research and Implementations, Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces, 2008, June 23-26.

[7]Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNInternational Journal of Computer Science and Network Security,VOL.10 No.6, June 2010.

[8]Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A PracticalApproach, TATA McGRAW-HILL Edition 2010.

[9] P. L. McEntire, J. G. O'Reilly, and R. E. Larson, Distributed Computing: Concepts and Implementations. New York: IEEE Press, 1984.

[10] L. Rudolph, M. Slivkin-Allalouf, E. Upfal. A Simple Load Balancing Scheme for Task Allocation in Parallel Machines. In Proceedings of the 3rd ACM Symposium on Parallel Algorithms and Architectures, pp. 237-245, July 1991.

[11] William Leinberger, George Karypis, Vipin Kumar, "Load Balancing Across Near-Homogeneous Multi-Resource Servers", 0-7695-0556- 2/00, 2000 IEEE.