IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-8, Aug 2018 E-ISSN: 2347-2693

Energy Preserves Task Scheduling In Heterogeneous Virtual Machine Framework

RV. Deepa[1]*, E. Ramaraj[2]

[1*] Department of Computer Science, Alagappa University, Karaikudi, Tamil Nadu, India
[2]Department of Computer Science, Alagappa University, Karaikudi, Tamil Nadu, India

*Corresponding Author: rvdeepa89@yahoo.com.sg Tel:7373610169

Available online at: www.ijcseonline.org

Accepted: 15/Aug/2018, Published: 31/Aug/2018

***Abstract*:-**In Virtual machine, Energy conservation is the major problem and it provides benefits such as reducing costs, increased reliability of the system and it also provides protection to the environment. Energy–aware scheduling is used to achieve these benefits. Existing energy-aware scheduling algorithms are not real time task oriented and also it lacks in system schedulability. Vacation queuing model is used for real-time, a periodic, independent task to solve this problem. The system which is proposed here can achieve energy optimization by combining the virtual machine resources with current exploitation. The eminence of hardware and nodes are well-organized with virtual network topologies. Vacation system is implemented with sojourn time to guarantee the schedulability of real-time tasks, efficiently. Simultaneously, energy consumption via dynamic VMs consolidation is concentrated. There are two strategies i.e. scale up and scale down to achieve a suitable trade-off sandwiched between task's schedulability and energy preservation. Energy conservation is achieved by switching the active host to sleep mode when the system does not perform any action. The task should be completed within the deadline and each user must provide the deadline to avoid rejection. The deadline is analyzed and acknowledgement is provided to the scheduler for each task completion.

***Keywords*:-** Virtualmachine, Scheduling, Deadline, Resources.

I. INTRODUCTION

Virtual machine consists of a number of resources that are different with one other via some means and cost of performing tasks in virtual machine using resources of virtual machine is different so scheduling of tasks in virtual machine is different from the traditional methods of scheduling and so scheduling of tasks in virtual machine need better attention to be paid because services of virtual machine. Task scheduling plays a key role to improve flexibility and reliability of systems in virtual machine. In virtual machine, resources in any form the sequence and requirements of the task and subtask are always dynamically allocated i.e., firewall, Network. So, this leads task scheduling in virtual machine to be a dynamic problem means no earlier defined sequence may be useful during processing of task. The reason of the last scheduling to be dynamic, because uncertain of the flow task, execution paths are also uncertain and at the same time resources available are also uncertain because there is a number of tasks are present that are sharing them simultaneously at the same time. The scheduling of tasks in virtual machine means want the best suitable resource available for execution of tasks or to allocate processor machines to tasks in such a manner that the completion time is minimized as possible. Tasks are then chooses according to their priorities and assigned to available processors and computer machines which satisfy a predefined objective function.

II. LITERATURE SURVEY

2.1 Efficient Optimal Algorithm of Task Scheduling in Virtual machine Environment

Virtual machine comes in focus development of grid, virtualization and web technologies. Virtual machine is an internet based that delivers Infrastructure as a service (IaaS), platform as a service (PaaS), and software as services (SaaS). In SaaS, software application is made available by the virtual machine provider. In PaaS an application development platform is provided as a service to the developer to create a web based application. In IaaS infrastructure is provided as a service to the requester in the form of Virtual Machine (VM).These services are made available on a subscription basis using pay as you-use model to customers, regardless of their location. Virtual machine still under in its development stage and has many issues and challenges out of the various issues in virtual machine scheduling plays very important role in determining the effective execution. Scheduling refers to the set of policies to control the order of work to be performed by a computer system. There have been various types of scheduling algorithm existing in distributed system, and job scheduling is one of them. The main advantage of job scheduling algorithm is to achieve a high performance and the best system throughput. Scheduling manages availability of CPU memory and good scheduling policy gives maximum utilization of resource. We compared three algorithms Time Shared, Space shred and generalizes priority algorithm.

2.2 Enhancing Virtual machine Scheduling based on Queuing Models

The most important problem is how to build a model that can maximize server utilization and minimize waiting time in queuing models. Therefore, a mathematical model is proposed to deal with multiple tasks and resources based on the basis of maximizing the benefit of the virtual machine provider and decrease the response time of the system. The main objective of this paper is to improve the performance of virtual machine system using queuing models as a tool. Furthermore, proposed model verified experimentally in several models that achieve higher utilization and response times compared to other models. Finally, scheduling algorithm that compute the lower and upper waiting time for all jobs at the waiting queues is introduced. It consists of four modules: multiple waiting queues for incoming requests, global scheduler based on SA algorithm, local schedulers and waiting queues for each local scheduler. Submit requests (R1, R2,…, Rn) from different sites are submitted to the Global Scheduler (GS). Each request is then transported to the local controller via the communication network and then sent to the local queue. Simulation studies to examine the effectiveness of different models within this framework are used. Suppose a model in which many users submit a request for execution of the work by any of a large number of sites. At each site, two components are placed: a global scheduler (GS), who determine where to send the jobs sent to this site, a Local Scheduler (LS), responsible for determining the order in which jobs is performed in this particular site.

2.3 An Energy-Saving Task Scheduling Strategy Based on Vacation Queuing Theory in Virtual machine

In order to solve the problem, we take advantage of the vacation queuing model to analyze the energy consumption of a virtual machine system, and present a task scheduling algorithm based on similar tasks. The main contributions of this paper include: (1) We make the first attempt to apply an exhaustive service, vacation queuing theory to model a virtual machine system; furthermore, considering the different states of a compute node, the different energy consumption characteristics, and latency during state transition of the heterogeneous virtual machine system, we improve the vacation queuing theory by adding idle period—when there are no tasks arriving at a compute node, the node goes through a period of idle time instead of entering a vacation at once in order to avoid frequent switches between different states. (2) We analyze the expectations of task sojourn time and energy consumption of a virtual machine system based on the busy period and busy cycle under steady state.

III. EXISTING SYSTEM

TASK SCHEDULING

Scheduling is the process of allocating tasks to available resources on the basis of tasks' qualities and need. The main goal of scheduling is increased utilization of the resources without affecting the services provided by virtual machine. The Job management is the fundamental concept of virtual machine systems task scheduling problems are main which relates to the efficiency of the whole virtual machine system. Job scheduling is flexible and convenient. Jobs and job streams can be scheduled to run whenever required, based on business functions, needs, and priorities. Job streams and processes can set up daily, weekly, monthly, and yearly in advance, and run on-demand jobs without need for assistance from support staff.

Job scheduling is global centralized - As virtual machine. it is a model which provide the centralized resource by the mirror service to multiple distributed applications. Therefore, virtualized technology and mirroring services make the task scheduling of virtual machine achieve a global centralized scheduling.

Each node in the virtual machine is independent - In virtual machine , the internal scheduling of every virtual machine node is autonomous, and the schedulers in the virtual machine will not interfere with the scheduling policy of these nodes.

The scalability of job scheduling - The scale of resources supply from virtual machine provider may be limited in early stages. With the addition of a variety of resources, the size of the abstract virtual resources may become large, and the application demand continues increasing. In the virtual machine, task scheduling must meet the scalability features, so that the throughput of the task scheduling in the virtual machine may not be too low.

Job sc2heduling can be with dynamism self-adaptive. The virtual resources in virtual machine system may also expand or shrink at the same time.

The Task scheduling is divided two parts: The unified resource pool scheduling and the scheduling of applications and virtual machine are primarily responsible. For example, Map Reduce task scheduling. However, each scheduling consists of two two-way processes that are scheduler leases resource from virtual machine and scheduler callbacks the requested resources after use. The former process is scheduling strategy. It is the latter one is callback strategy. The set of task scheduling is the combination of the scheduling and callback resource strategy.

Scheduling Types

1) Static scheduling is a schedule task. It is about the information complete structure of tasks and before execution mapping of resources, execution/running time is estimate of task.

2) Dynamic scheduling must depend on not only the submitted tasks to virtual machine environment.

Disadvantage:

Task scheduling problem the essence of the resolution of the task scheduling problem is to set up a scheduling policy. Based on this, suitable mapping relationship can be established between application tasks and computing resources in order to achieve reasonable distribution and efficient execution of application tasks using the limited computing resources.

One task cannot be run on two virtual machines, each virtual machine can only handle one task at one time and each has different properties

- Implement in heterogeneous environment in difficult.
- Difficult to analysis dissimilar tasks.

IV. PROPOSED SYSTEM

SYSTEM ARCHITECTURE

Dynamic scheduling is the process in which arrived task is uncertain at the run time and resource allocation is difficult when simultaneously many tasks arrive. The task components/task arrived is not known until it reaches the dispatching center node. Thus the execution time of the arrived tasks may not be identified. Therefore, the allocation of tasks is carried-out on the fly when the application executes the project and focuses on the periodic scheduling and independency of real-time tasks. This approach dynamically creates virtual clusters dealing with the conflicts between the parallel and serial tasks in the system. The tasks are available dynamically for scheduling over time within the scheduler. The dynamic scheduling is more flexible to be able to determine the run time in advance. Consider a case in which four states are viewed, running state, idle state, sleep state and recover state. The scheduling is planned with the measure in order to identify the degree of efficient task computation relative to the completion time of the application. The degree of task computation is used as a utilization value to identify a level of virtue for executing task on processor. This in turn implies the energy consumption of that processor efficiently. In this proposed approach, the task load is attuned dynamically without the prediction of running time of all the tasks. TSAST scheduling strategy is implemented to provide best tradeoff in scheduling the tasks. Dynamic scheduling algorithm for this scheduling mechanism has been proposed to generate scheduling based on the shortest average execution time of tasks. In Vacation queue service system, the real-time independent tasks are scheduled in the virtual machine. It aims at saving the energy dynamically starting hosts, closing hosts, creating VMs, cancelling VMs and migrating VM according to the system's workload. The real time task must have deadline and it should be completed within the deadline to avoid task rejection. The idle node is switched to sleep mode to reduce the energy consumption by VM migration. The deadline is time and date which the user provides to complete their tasks within the time limit. The VMs scalability is dynamically adjusted to save energy. The scheduler analyses the deadline and also provides acknowledgement for the completed tasks.

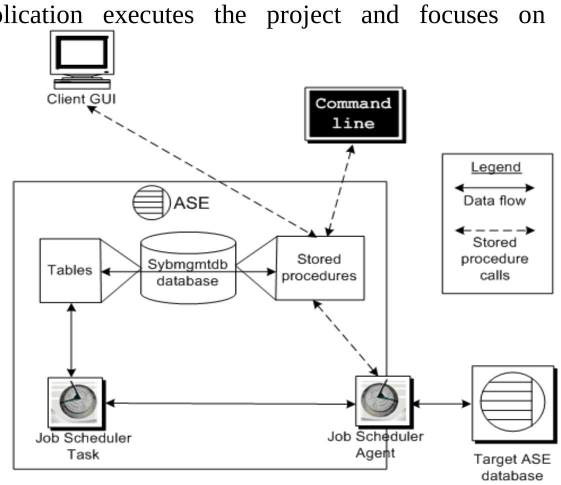


Figure 1.1 System Architecture in Task Scheduling

Advantages:

- Implement in heterogeneous environments
- Easy to analyze dissimilar tasks
- Energy and time are conserved in various busy states.

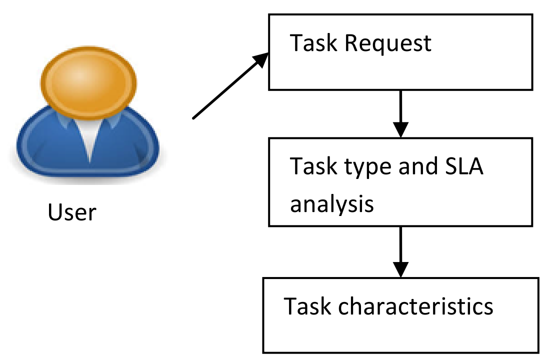


Figure1.2 Task Scheduling

V. VIRTUAL MACHINE FRAMEWORK CONSTRUCTION

The virtual machine service provider is responsible for maintaining an agreed-on level of service and provisions resources accordingly. A CSP, who has significant resources and expertise in building and managing distributed virtual machine storage servers, owns and operates live Virtual machine systems, it is the central entity of virtual machine. Virtual machine provider activities for utilizing and allocating scarce resources within the limit of virtual machine environment so as to meet the needs of the virtual machine application. It requires the type and each application to needed the amount of resources in order to complete a user job. The order and time of allocation of resources are also an input for an optimal resource allocation. Virtual machine user represents a person or organization that maintains a business relationship with, and uses the service from, a virtual machine provider. Users, who stores data in the virtual machine and rely on the virtual machine for data computation, Virtual machine consists of both individual consumers and organizations. Virtual machine consumers use Service-Level Agreements (SLAs) for specifying the technical performance requirements to be fulfilled by a virtual machine provider. In this module, scheduler gets the tasks from users and analyzes task types.

VI. WORKLOAD HETEROGENEITY

The scheduler is responsible for assigning incoming tasks to active machines in the cluster. It also reports the average number of tasks in the queue during each control period to help the controller make informed decisions. The prediction module receives statistics about the usage of all resources (CPU and memory) in the cluster and predicts the future usage for all of them. Implement technique include the similar kind of workloads which is having fixed number of parameters such as length, number of CPU's required and buffer size of input and output files, etc. Workload consolidation in data centers is achieved with the help of virtualization, which helps in saving the energy as unused nodes can be put to sleep mode or by shut down of the machines. The proposed approach uses the varying workloads and simulation results show the increased number of completed jobs. This approach does not give any information about the energy consumption of the data center and SLA.

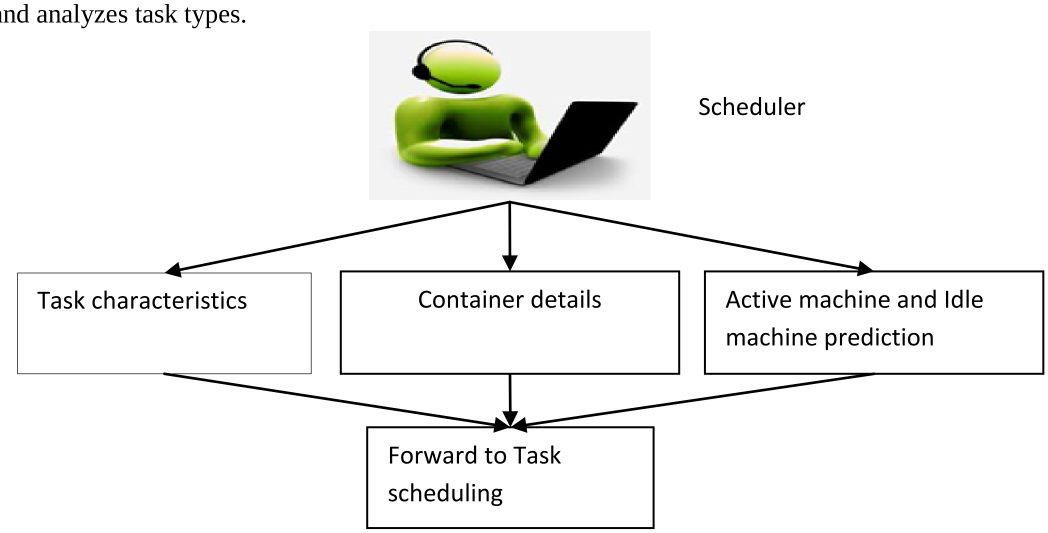


Figure 1.3 Task scheduling Steps

VII. VACATION M/G/1 QUEUE SYSTEM

In this module, we can allocate task based on vacation M/G/1 queue system and contains various states. We can implement finite capacity M/G/1/K queue with the variation that the server goes for vacations when it is idle. This service model is referred to as one provided that extensive service, as the server cannot go for a vacation until all the jobs currently in the system have been served. This is the service model being considered here as this leads to a simple methodical model. Note that it is also possible to have a gated service model where the server only serves individual customers that it finds in the system when it first starts service following its vacation. It then leaves for vacation again. The four states are described as follows:

(1) Running state: The state when a compute node is working;

(2) Idle state: If there are no tasks arriving at a compute node, the node goes through an idle period to avoid frequent switches from the deep sleep state. The threshold of idle period is T_i^{idl}.

(3) Sleep state: After the idle period of T_i^{idl}, if there are no incoming tasks, the compute node goes into sleep state.

(4) Recovering state: When a task arrives at the compute node under sleep state, the compute node needs to recover and then start to execute the task. The recovering state is a transition state—in this state, a compute node is woken up from the sleep state and transitioned to the running state.

VIII. ENERGY CONSUMPTION EXPECTION

In this module compute sojourn time that consists of waiting time in the local queue of a compute node. And service time of the node performing tasks. The sojourn time probability distribution function is first derived for a Markovian queue; with both batch arrivals and batch departures that admits a product-form in networks of such queues. The reversed process is then considered and it is confirmed that the unconditional reversed sojourn time has the same distribution as the forwards sojourn time. Using conditional forward and reversed sojourn times, a result is obtained for the sojourn time distribution on overtake-free paths in product-form networks of such batched queues. Then calculate the busy cycle and busy period, calculate mean power. Finally set the reasonable threshold idle time.

$$P\{\text{server is idle}\} = p_0 = \frac{r_0\overline{V}}{D} + \frac{(q_0)}{\lambda D}$$

IX. SIMILAR TASK SCHEDULING

In this module we can implement Task Scheduling Algorithm based on Similar Tasks (TSAST). TSAST follows the partitioned or non-hierarchical clustering approach. It involves partitioning the given data set into specific number groups called Clusters. every cluster is related with a centre point called centroid. Every point is declared to a cluster with the closest centroid. Proposed dynamic VM allocation algorithm using clustering is as:

Input: Virtual Machine`s of List V with their locality about the globe List D of data centers

Algorithm:

1. Select K points according to the number of datacenters in D
2. Choose datacenter from D
3. Form K clusters of VM`s from V by assigning closest centroid
4. Recomputed the centroid of each cluster
5. All the requested VM`s arranged in cluster form
6. Assign the VM`s to the available Host
7. If all the VM`s are allocated
8. Allocate the VM`s cluster to the selected datacenter
9. End if
10. Repeat [2] until D is empty
11. If all the VM`s are created in the datacenters
12. Send the virtual machine lets to the created VM`s
13. End if

TECHNIQUES

Vacation queue system

In this system, consider a multiple vacation model. Here a server, on returning from a vacation, goes for another vacation if it finds the method still null. In this case, the servers resume normal service if it finds one or extra jobs waiting when it returns from a vacation. Note that many vacations, one after the other, will be possible in this model.

$$P\{\text{server is on vacation}\} = \frac{r_0\overline{V}}{D}$$

$$P\{\text{server is busy serving a job}\} = \frac{(1-r_0)\overline{X}}{D} = \rho_c$$

$$P\{\text{server is not on vacation but is idle}\} = \frac{(q_0)}{\lambda D}$$

Similar task clustering

In this algorithm present a Task Scheduling Algorithm based on Similar Tasks (TSAST). The smaller the variance of service time of a compute node, the less energy that is consumed by the node. At the same time, the task scheduling algorithm needs to provide good service to users, such as a shorter task sojourn time. Usually, the longer the task queues, the longer the task waiting time and the task sojourn time. Considering the factors, for soft real-time tasks and non-real-time tasks, we schedule them on compute nodes with similar service times and shorter task waiting times; in this case, we call these tasks with similar service times on a node as similar tasks. For hard real-time tasks with strict deadlines, if the deadline cannot be satisfied when a task is scheduled to the compute node with a similar service time, it is scheduled to the compute node with the earliest finish time.

The algorithm initialization set KU=0

Where KU=$\left\{\frac{1}{\mu_i}\middle|1 \le i \le k\right\}$; //KU is mean service time set of all compute nodes

While (there are incoming tasks) do

Accept a task $task_j$ using FCFS strategy;

Record task parameters$task_j = \{com_j, t_j^{arr}, t_j^{dl}\}$;

For each node $node_i$

Calculate task execution time t_{ji} of $task_j$ in it:

$t_{ji} = \frac{com_j}{\sqrt[3]{pow_i^{run}/c_i}}, i \in k, //c_i \text{ is as constnat}$

End for

Choose a node set U=$\{node_i | t_{ji} + t_i^{wait} < t_j^{dl} - t_j^{arr}\}$;

// U is a set of the compute nodes which can finish

//the task before deadline

If $U \neq \emptyset$

Choose the compute node with minimum

$\left|\frac{1}{\mu_i} - t_{ji}\right| + \eta_{ji}.t_i^{wait}$ to $U; \frac{1}{\mu_i} \in KU$

Else

Choose the compute node with the minimum$\{t_{ji} + t_i^{wait}\}$;

End if

Update KU

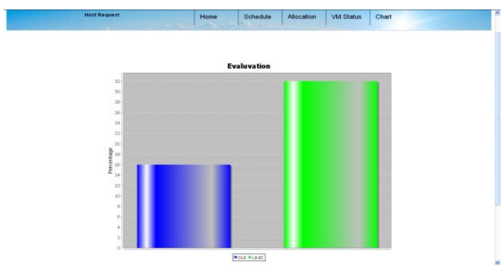
//update the mean service time of the selected node

End while

X. RESULT AND DISCUSSION

We can assess the execution of the framework utilizing the parameters, for example, (i) Dynamic load adjusts assets distribution (ii) similar task scheduling algorithm, different between usage assessment frameworks.

These estimations are merged as limit of work stack adjusts and assets portion chart. The future utilizing subterranean insect state calculation actualize in utilizing 90% work stack radiators.



XI. CONCLUSION

The efficient dynamic scheduling algorithm is implemented in virtual machine for scheduling the task. There are two efficient approaches Vacation queue and TSAST algorithm. Using these algorithms, the analysis of any incoming task supplied by the user is done and then the scheduler allocates the task to the real-time controller. Before placing the task in a real time controller, the scheduler first checks whether the task can be completed by the real time controller or not. i.e., the real time machine is checked whether it has enough resources to complete the task or not. If there is insufficient resources in real time controller, then the scheduler allocates that task to the VM controller and the VM controller use the VM machines to complete the task. Otherwise, the real time controller allocates the task to the Physical Machine (PM) for task completion. The efficiency of the system performance is increased by using this algorithm. The energy-aware scheduling is analyzed for independent, periodic real-time tasks in virtual virtual machines. The scheduling objective is to improve the system's schedulability for real-time tasks and to save energy. Here, the virtualization technique is deployed and TSAST is used to save energy.

XII. FUTURE ENHANCEMENT

Scrutinize and build the vacation queue scheduling architecture and task oriented energy consumption. A novel energy-aware scheduling algorithm is implemented based on this system. The TSAST enhances the system's schedulability. The task's real-time requirements should be satisfied and the energy should also be saved to adjust the scalability of the active hosts'. Extensive simulation experiments have been conducted to compare it with other algorithms to evaluate the effectiveness of our TSAST. We can extend our work to implement various scheduling algorithms with deadline constraints.

REFERENCES

[1] G. Lovasz, F. Niedermeier , and H. De-Meer, Performance tradeoffs of energy-aware virtual machine consolidation, Journal of Networks Software Tools and Applications, vol. 16, pp. 37–38, 2013.

[2] J. X. Chen, Energy efficient design of virtual machine data center, Virtual machine IDC, vol. 57, pp. 481–496, 2011.

[3] O. Philippe and L. Jorge, Deep network and service management for virtual machine and data centers: A report on CNSM 2012, Journal of Network and Systems Management, vol. 21, pp. 707–712, 2013.

[4] Y. S. Jing, A. Shahzad, and S. Kun, State-of-the-art research study for green virtual machine , Journal of Super, vol. 65, pp. 445–468, 2013.

[5] T. L. Chen and H. L. Lachlan, Simple and effective dynamic provisioning for power-proportional data centers, IEEE Transactions on Parallel and Distributed Systems, vol. 24, pp. 1161–1171, 2013.

[6] C. Y. Lee and A. Zomaya, Energy conscious scheduling for distributed systems under different operating conditions, IEEE Transactions on Parallel and Distributed Systems, vol. 22, pp. 1374–1381, 2011.

[7] J. Guo, F. Liu, D. Zeng, J. C. S. Liu, and H. Jin, A cooperative game based allocation for sharing data center networks, in Proceedings IEEE Infocom, 2013, pp. 2139–2147.

[8] Z. Zhou, F. Liu, H. Jin, B. Li, B. Li, and H. Jiang, On arbitrating the power-performance tradeoff in SaaS virtual machines, in Proceedings IEEE Infocom, 2013, pp. 872–880.

[9] W. Deng, F. Liu, H. Jin, B. Li, and D. Li, Harnessing renewable energy in virtual machine datacenters: Opportunities and challenges, IEEE Network Magazine, vol. 28, pp. 48–55, 2014.

[10] F. Xu, F. Liu, L. Liu, H. Jin, B. Li, and B. Li, iAware: Making live migration of virtual machines interference aware in the virtual machine, IEEE Transactions on Computers, vol. 63, pp. 3012–3025, 2014.

Authors Profile

Mrs RV. Deepa received the UG degree in Saradha Niketan college,Amaravathiputhur from Karaikudi Alagappa University in 2010 and the PG degree in Alagappa Govt Arts college Karaikudi from Alagappa University in 2012.Currently working toward the Mphil degree in Alagappa university, Karaikudi.

Dr. E. Ramaraj is working as the Professor and Head of the Department of Computer Science, Alagappa Network Security, Remote Sensing and Big Data & Analytics. He has published more than 100 international journals.

