Journal of Theoretical and Applied Information Technology
20th January 2014. Vol. 59 No.2

ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

BASIP A VIRTUAL MACHINE PLACEMENT TECHNIQUE TO REDUCE ENERGY CONSUMPTION IN CLOUD DATA CENTRE

[1] AJITH SINGH. N AND [2] M. HEMALATHA
[1, 2] Department of Computer Science, Karpagam University, Coimbatore, Tamil Nadu, India
E-mail: [1]ajithex@gmail.com , [2]csresearchhema@gmail.com

ABSTRACT

Infrastructure as a Service (IaaS) in cloud computing provides Infrastructure as a service for the demand of user from small instance to large instance in pay per use model. The services include like computer resource, networking and data storage. An API (Application Programming Interface) is used to access the infrastructure and a dashboard to control the server and to create and manage different Virtual Machines in the cloud data centre. Multiple cloud users access the service simultaneously. Due to continuous access of the services, a deadlock can happen, and it may lead to a system crash, although cloud computing is designed to overcome such problem, a proper Virtual Machine Placement Technique is needed for implementation to rule out such incidences. Deadlock can occur in cloud computing as the system is inherited from distributed computing and virtualization. A Virtual Machine Placement Technique known as BASIP is proposed to overcome the issue of deadlock by using a banker algorithm with Stochastic Integer Programming. Further, the proposed algorithm is being simulated with hundreds of servers and thousands of virtual machines. The proposed algorithm was simulated with different overload detection and VM selection algorithm. The BASIP algorithm is experimented with 800 servers with 1024 Virtual Machines. From the experimental results, BASIP algorithm reduces energy significantly.

Keyword: *Virtual Machine, Cloud Computing, Live Migration, Bankers and Stochastic Integer Programming, Deadlock Avoidance, Resource Allocation, Energy, Cloud Data Centre, Minimum Utilization Rank, Polynomial Regression, BASIP*

1. INTRODUCTION

The challenges for the cloud provider is to handle all its resources in an energy efficient way while meeting the Service-Level Agreement (SLA), which is a contract between the customers and Cloud provider (CP) on the Quality of Service (QoS). Since the cloud computing came to the market, meaning of computing is totally changes cloud computing which was started a few years back has occupied the total IT market with its functionality. Research studies published on [1] highlights that cloud computing market revenue to approach $20 Billion by the end of 2016. Companies like Amazon Web Service (AWS), Google, Salesforce.com, IBM, Microsoft and Oracle have converted their traditional datacentre to cloud datacentre. As a starter Amazon is the one who started giving cloud services in 2006 with Elastic Compute Cloud (EC2) [2][3] Amazon's total revenues are $61 billion followed by Microsoft and Rackspace.com. Google has started with Google compute engine, which is much faster, than AWS. To make it fast it all depends how the resource are utilized in the datacentre especially in the virtual machine placement. Virtual machine placement is the process of mapping VMs to the most suitable Physical Machine (PM) based on the requirement of VMs characteristics to achieve the QoS without any violation of the SLA with user and cloud provider to achieve energy efficient. VMs provisioning and optimization to minimize the cost of computing by applying different policy and migration method in the cloud data centre.

As virtualization is a core technology of cloud computing, the problem of VMs placement has become a hot topic recently. This VMs placement is an important approach for improving power efficiency and resource utilization in cloud infrastructures. Virtual machine are of different configuration and cloud computing is a heterogeneous environment, allocating multiple VMs to PMs has to be done wisely so that a good load balancing will be achieved by taking care of power efficiency. VMs placement is an important approach for improving power efficiency and resource utilization in cloud infrastructures.

Reports from Google white paper that use of cloud computing significantly reduce power consumption than the traditional data centre. Based on their analysis which was published as “Google Apps: Energy Efficiency in the Cloud” [4], a typical company or organization that migrates to the cloud computing could save an estimated 68–87% in energy for its office computing and reduce similar amounts of carbon emissions. From the Google report, it is clear that migration to the cloud service can save energy from 68-87% for typical companies. Migration to cloud U.S. companies could achieve annual energy savings of $12.3 billion and carbon reductions of 85.7 million metric tonnes by 2020 equivalent to the annual emissions of over 16.8 million passenger vehicles [4].

To improve the performance of cloud service provider various load balancing method are applied to improve the performance for example like round robin method. Specializations form of distributed computing is cloud computing. However, the underlying concept is the same. The terms distributed systems, and grid and cloud computing actually refer to slightly different thing, this are based on delivering computing resources through a large and often global network of computers. Services access via the internet usually refers to cloud computing, cloud computing fully depend on internet. Services, which are accessible, can be anything business software, CRM, website, Social network etc. As the cloud computing is having the features of distributed computing and virtualization there is a possibility of occurrence of deadlock. Cloud provider wants to reduce the upfront cost by minimizing the server and hosting many virtual machines on a single host or multiple host compared to the number of jobs arrived for availing the Virtual Machines. Jobs will be competing to acquire the same VMs at the same time leading to a deadlock. Describe in Wikipedia “Deadlock is a common problem in multiprocessing systems, parallel computing and distributed systems, where software and hardware locks are used to handle shared resources and implement process synchronization” [5].

Deadlock can lead to unwanted performance and may violate SLA. Henceforth, in cloud computing a virtual machine placement technique is required to allocate the VMs and to balance the load to avoid deadlocks.

Another issue is that cloud provider never known the demand of resource from user, to determine the variation in demand and costs. To overcome the issue a hybrid algorithm known as BASIP is proposed in this research, which is based on Banker,’s Algorithm with Stochastic Integer Programming. BASIP after simulation proof to be an optimal solution to reduce energy in data centre, while solving the problem of resource allocation it avoid deadlock with payment plan.

2. RELATED WORKS

Large amounts of data and computational resource that can be used based on pay per use and released when finished is provided by cloud computing. User and cloud provider to achieve QoS in their services have maintained certain SLA. Violation of any SLA will lead to poor performance of the service. A VM placement solution should determine the optimal placement for each VM as soon as it arrives,

A study by using the game-theoretic method to solve the optimization problem of resource allocation in network systems from the viewpoint of cloud providers done by *Guiyi et al., 2010*. Cloud computing is based on QoS and cost which is considered by both the provider and user. *Guiyi et al* used game theory to solve the problem in which author first used binary integer programming method to obtain initial independent optimization and based on the result an evolutionary mechanism is designed to achieve the final optimal and fair solution [6].

Energy-aware data centre is the latest thing which all the scientist is concern about author (Mohsen Sharifi et al., 2011) approach that energy in data centre can be reduce and they proposed four models, namely the target system model, the application model, the energy model, and the migration model, to identify the performance interferences between processor and disk utilizations and the costs of migrating VMs. They consider fitness metric to evaluate the merit of consolidating a number of known VMs on a PM based on the processing and storage workloads of VMs. Based on their survey they proposed an energy-aware scheduling algorithm using a set of objective functions in terms of this consolidation fitness metric and presented power and migration models. The proposed scheduling algorithm assigns a set of VMs to a set of PMs in a way to minimize the total power consumption of PMs in the whole datacentre. Result shows that 24.9%

power saving and around 1.2% performance degradation when the proposed scheduling algorithm is used compared to when other scheduling algorithms are used [7]

S Chaisiri et al., 2009 look on the service provide by the cloud provider and proposed an algorithm, which could minimize the cost factor. The algorithm is based on stochastic integer programming that works in different stages possibly two stages. On one of it calculates the demand of VMs in reservation phase and another is to calculate the numbers of VMs allocated in both the utilization and on demand phases. Author tries to combine and try to put and design a new algorithm for VMs placement that is on Integer Linear Programming Problem (ILP), ILP could solve the NP-Hard problem [8].

Jiandun Li et al., 2011 proposed a hybrid energy-efficient scheduling algorithm using dynamic migration. Algorithm was implemented using Eucalyptus (v2.0.1) to setup the base environment for private clouds and powering up/down a node was implemented via powernap package. And dynamic migration was implemented through Libvirt. A VM workflow that conform to Gaussian distribution N(2440,1550) to simulate VM requests within a cycle was generated. The workflow consisted of 16 requests with the expected spectrum for left capacity set to [M-2, M+1]. Their proposed algorithm was experimented with & without migration, Round robin (RR) and Greedy approach respectively to schedule the workflow. The results of the experiments showed that it could not only reduce the response time, conserve more energy, but also achieve higher level of load balancing [9].

Considering the energy efficiency factors (such as energy cost, carbon emission rate, workload, and CPU power efficiency) author Saurabh Kumar Garg et al., 2011 proposed a optimal scheduling policies which run in multiple data centres for a cloud provider. This energy efficiency change across different data centres depending on their location, architectural design, and management system. The scheduling policies are sure to achieve a median as much as 25% of energy savings in comparison with profit based scheduling policies causing higher profit and fewer carbon emissions [10].

Jeffrey M. Galloway et al., 2011 presented a load balancing approach to IaaS cloud architectures that is power aware. They proposed a Power Aware Load Balancing algorithm (PALB) that applied to the cluster controller of a Eucalyptus private cloud that is power aware. This load balancer maintains the utilization of all compute nodes and distributes virtual machines in a way that is power efficient. The goal of this algorithm is to maintain availability to compute nodes while reducing the total power consumed by the cloud. The authors used Eucalyptus software for building the cloud architecture and proved that using PALB, organizations wanting to build local clouds using Eucalyptus would be able to save on energy costs. This is because Eucalyptus does not account for power consumption when applying its default load balancing technique. Depending on the job schedule distribution and virtual machine request size, organizations can save 70% - 97% of the energy consumed compared to using load balancing techniques that are not power aware which they showed through experimental results. As a part of their future work, they proposed the implementation of their local cloud “Fluffy” which is a standard Eucalyptus build with independent nodes for each component of the cloud [11].

Can Hankendi et al., 2011 proposed an efficient consolidation technique for multithreaded workloads through adaptive resource sharing on virtual environments. First, an experimental framework was presented to accurately evaluate energy/performance tradeoffs of co-scheduling multi-threaded applications on virtualized systems. Then the effect of application selection on energy efficiency was explored. It was shown that performance degradation due to resource contention can be minimized by setting memory and NUMA affinities for consolidated VMs. Based on all these analysis, the author proposed the adaptive VM reconfiguration algorithm based on power efficiency characteristics of multi-threaded workloads. All experiments are performed on an AMD 12-core Magny Cours (6172) server, virtualized by VMware vSphere 5.0 ESXi hypervisor. It was demonstrated that the proposed resource sharing technique outperforms the stateof-the-art co-scheduling techniques on a real-life multicore system. They presented a virtual machine reconfiguration algorithm that improves the overall throughput-per-watt of a real-life multicore system by up to 25% in comparison to existing consolidation methods [12].

To enhance Green computing within a scalable cloud-computing author Andrew et al., 2010 presented a new framework. The author discuss of using scheduling techniques related to power-aware for resource management enables live migration, and a minimal virtual machine design, by this the new framework which author describe will provide overall system efficiency will be vastly improved in a data centre based Cloud with minimal performance overhead. Author also explain the rising of computing use and concern of energy saving. For their framework, they demonstrate the potential of the proposed framework, the authors presented new energy efficient scheduling, VM system image, and image management components that explore new ways to conserve power. Author also describe that Future opportunities could explore a scheduling system that is both power-aware and thermal-aware to maximize energy savings both from physical servers and the cooling systems used. Such a scheduler would also drive the need for better data centre designs, both in server placements within racks and closed-loop cooling systems integrated into each rack. While a number of the Cloud techniques are discussed in this paper, there is a growing need for improvements in Cloud infrastructure, both in the academic and commercial sectors. It is believed that Green computing will be one of the fundamental components of the next generation of Cloud computing technologies [13].

Kusic et al., 2009 have stated the problem of continuous consolidation as a sequential optimization and addressed it using Limited Lookahead Control (LLC). The proposed model requires simulation-based learning, and the execution time reaches 30 minutes even 15 nodes. On the contrary, our approach is heuristic-based allowing a reasonable performance even for large-scale [14].

Based on the priorities in multi-application virtualized clusters researcher (Song et al., 2009) proposed an resource allocation technique, in their work they don't integrate migration of VMs to optimize the allocation table [15]. The author Gupta et al., 2012 points out that by using that utilizing the knowledge of the target application for a VM can lead more intelligent VMs placement decisions [16].

3. FORMALIZATION OF VIRTUAL MACHINE PROBLEM

In virtual machine placement problem, virtual machines are viewed as boxes, where various resource requests of each virtual machine considered as dimensions of box with non-negative values. Physical servers are considered as bins, where CPU, memory and bandwidth capacities are regarded as properties of box. The goal of virtual machine placement problem is to determine the minimum number of physical machines required by the set of virtual machines.

The problem of virtual machine placement in the datacentre is defined as: given a set of virtual machines $VM = \{vm_1, vm_2, \dots, vm_n\}$ and a set of physical machines $PM = \{pm_1, pm_2, \dots, pm_m\}$, where each vmi is a triplet $vm_i = (cpu_i, ram_i, bw_i)$, $1 \leq i \leq n$ denoted cpu, memory and bandwidth requirements of virtual machine respectively. Each pmj is also a triplet $pm_j = (cpu_j, ram_j, bw_j)$, $1 \leq j \leq m$ denoted resource capacity of physical machine. In addition, x_{ij}, $1 \leq i \leq m$, $1 \leq j \leq n$ and y_i, $1 \leq i \leq m$ are decision variables, $x_{ij} = 1$ if and only if vm_j is mapped onto pm_i, $y_i = 1$ if pm_i is used to host virtual machine. The objective is to minimize $\sum_{i=1}^{m} yi$ while finding all values of x_{ij}.

There are several implicit constraints in the above definition:

- Each virtual machine can only be hosted on one physical machine;
- For each type of resource, the amounts of resource requests of virtual machines sharing the same physical machine are smaller or equal to capacity of physical machine hosting them;
- The number of physical machines that host virtual machines are not more than me, $\sum_{j=1}^{m} yi \leq m$.[17]

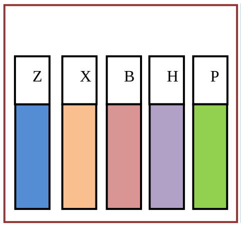

Figure 1

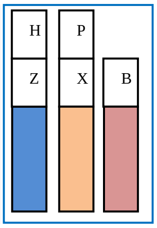

Figure 2

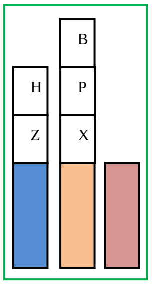

Figure 3

Figure 1: Traditional DC Without Virtualization

Figure 2: Cloud DC With Virtualization

Figure 3: Cloud DC With Virtualization

Example: Consider a datacentre (DC) with 5 VMs, Z, X, B, H, P with different requirements and services running on it. Figure 1 represents five different services running on different server, Figure 2 represents services running on a VM on the same server where it uses 3 servers instead of 5 and fig 3 use 2 servers to run 5 VMs. Based on the required minimum server will be used and other servers will be in sleep mode or low power mode, whenever required these offline servers can be put online. The choice of which VMs and how many VMs to place together on a server yields a range of different operating points varying in resource efficiency and performance. One may maximize efficiency by packing the VMs into the minimum number of servers required to satisfy the number of processor cores, memory and disk space requirements of each VM, but such packing hurts performance.

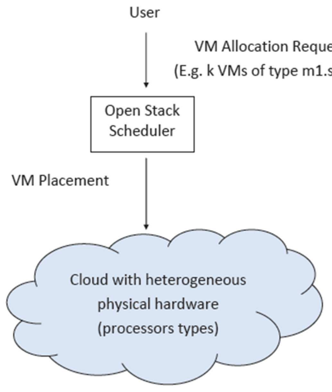

Figure 4: VM Placement In Open Stack [18]

3.1 Problem of Deadlock in Cloud Computing

Problem defined in [19] by using web server for e-commerce and database server on the Amazon EC2 server user has experienced a deadlock by using the instance to an m1.xlarge as well as in m1.large. User defined the problem that bad disk IO is often seen in EC2 server but in 2012 they faced the problem of deadlock especially on servers running the MySql database. Another user seeing the problem of deadlock in an EC2 server move the database to Amazon RDS (Relational Database Service) instance and declared that the problem of bad disk IO and deadlock was reduced in the RDS.

Reported by Matt Wilson on [20] found a Kernel deadlock in scheduler on multiple EC2 instance types, user report that kernel stop responding running on m2. 2xlarge EC2 instances. It shows full VCPUs are stuck waiting on spinlocks. User defined that it could be reason of scheduling code that leads to deadlock of the server which Xen hypervisor. However, it was fixed with new release of Linux-ec2 (2.6.32-346.51). User of Google App Engine got App deadline/deadlock/internal errors [21].

Mention in [22], a deadlock may occur during region initialization when index creation happens locally or through a remote request in vFabric GemFire. vFabric GemFire is a distributed data management platform providing dynamic scalability, high performance, and database-like persistence.

From the thread reported by many users on different platform in cloud computing it is clear that deadlock can happen in cloud computing especially when the instances is running database server. Since, cloud computing has inherited characteristic of distributed computing and virtualization there is a possibility of occurrence of deadlock.

4. PROPOSED METHOD

Our key idea is to consider the extent to which different VMs are affected, determine the VMs that degrade the least when placed together, and then consolidate these VMs to the extent that performance constraints are not violated.

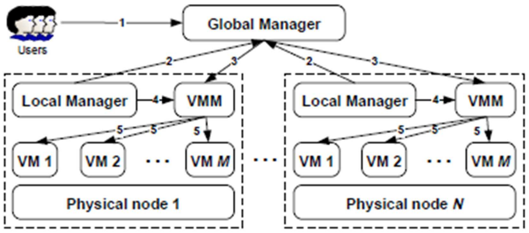

Figure 5: Physical Node With VM Based On Anton Et Al ., 2012 [23]

An algorithm for resource allocation fall into 4 categories [23]

1. Selection of sender hosts
2. Selection of VM
3. Selection of receiver hosts
4. Assignment of VM

The proposed algorithm BASIP (Banker Algorithm with Stochastic Integer Programming) use the method of the SIP (Stochastic Integer

Programming) and Banker's algorithm[24][25][26][27][28][29], to place the virtual machine in the data centre both the technique has its own merit and demerit. To determine the variation in demand and costs Stochastic Integer Programming (SIP) is used where by using SIP frequent re-allocation are not needed. If there is error in the estimation of then users might end up paying more. SIP is perfect for the determination of demand and costs but in lack in mapping of Virtual Machines to Physical Machines to overcome this problem this research work proposed a hybrid model of Banker algorithm and SIP [8]. Whereas Banker algorithm is a resource allocation and deadlock avoidance algorithm best for mapping the VMs to host. Banker algorithm the system is checked before deploying whether the system is ready to take any other process before allocation by giving a single in the safe and unsafe state. Whenever Banker algorithm allocate VMs to the host it check for safe and unsafe of the host and plus checking the SIP is true or not.

BASIP Algorithm (Hybrid Model of Banker's Algorithm and Stochastic Integer Programming**)**

H: set of Hosts

I: current mapping I: **H**→ $\{0,1,\ldots,N_{max}\}$

Where $N_{max} = \max\{N_T : H\epsilon \boldsymbol{H}\}$

I*: Proposed Mapping

M: set of Virtual Machines where method assigned to each **H** is associated with current mapping **I**

M*: set of Virtual Machines where methods assigned to each **H*** is associated with proposed Mapping **I***

```
Advancer (H, I, M)
$\boldsymbol{\hat{H}} = \{H \in \boldsymbol{H} : H_E \neq \emptyset\}$
While ($\boldsymbol{\hat{H}} \neq \emptyset$)
    If (Overload Detection (H )= =True)
        (H*,g*,M*)← Select($\boldsymbol{\hat{H}}, \boldsymbol{I}, \boldsymbol{M}$)
        If (! M*)
            Break
        I*← $\{H^*, I(H^*)\} + \{(H^*, I(H^*) + 1)\}$
        else
            M*← update (g*,M*,M)
            If (Banker (H, I*, M*) =SAFE && (SIP=True))
                I← $I*$
                M←M*
    $\boldsymbol{\hat{H}} \leftarrow \boldsymbol{\hat{H}} - \{H^*\}$

Banker (H, I*, M*)
$\hat{H} \leftarrow H$
While ($\hat{H} \neq \emptyset$)
    aMaps ← False
    for $T\epsilon\hat{T}$
        futureMapFit ← True
        $\hat{M} \leftarrow M^*$
        For ($g\epsilon A(H^{(I^*(H))})$)
            If (Overload Detection(H) = =True)
                M*← $Select(\{g\}, 0, \hat{M})$
                If (! M*)
                    futureMapFit ← False
                    break
                else
                    $\hat{M} \leftarrow update(g, M^*, \hat{M})$
        If (futureMapFit)
            M*← $release\left(H^{(I^*(H))}, M^*\right)$
            aMaps ← True
            $\hat{H} \leftarrow \hat{H} - \{H\}$
```

If (!aMaps)

Return UNSAFE

Return SAFE

SIP (Stochastic Integer Programming)

Decision variable $X_{ij}^{(r)}$ denotes the number of VMs in the class V_i, allocated to provider P_j. c_{ij} denotes the cost, in reservation phase, charged by provider P_j for hosting one VM from class V_i.

Stage 1: This defines the number of VMs to be provisioned in reservation phase

Stage 2: This defines the number of VMs allocated in utilization and on-demand phase.

Minimize: $SIP = \sum_{V_i \in \nu} \sum_{P_j \epsilon P} c_{ij} X_{ij}^{(r)} + \varepsilon_{\Omega}[\mathfrak{Q}(X_{ij}^{(r)}, \omega)]$

Subject to: $X_{ij}^{(r)} \epsilon \{0,1, \ldots\}, V_i \in \nu, P_j \epsilon P$

- $X_{ij}^{(r)}$ denotes number of VMs provisioned in first stage.
- $\varepsilon_{\Omega}[\mathfrak{Q}(X_{ij}^{(r)}, \omega)]$ denotes cost in second stage.
- Here $\omega \epsilon \Omega = D\ x\ \Pi \mathrm{p_j} \epsilon \mathrm{p}$ denotes realizations (set of demands and prices)

5. RESULT & DISCUSSION

The proposed method BASIP, a deadlock avoidance and payment plan for resource allocation to improved the VM placement, and reduce the energy in cloud data centre. The proposed algorithm check whether the host is safe or unsafe by using Banker's Algorithm then it check the what kind of payment plans based on the two condition resource is allocated to the VM. After the simulation with various overload detection algorithm and VM selection this research work obtain the minimum energy and the final output is below.

- **ENERGY:** BASIP * LR * MUR = 15.90kWh
- **SLA :** BASIP * IQR * RS = 0.00028%
- **MIGRATION :** BASIP * MAD * RS = 691 VM migrates

Table 1 shows the simulation result of BASIP with Maximum Correlation(MC) as VM selection with various overload detection

Table 2 shows the simulation result of BASIP with Minimum Migration Time (MMT) as VM selection with various overload detection

Table 3 shows the simulation result of BASIP with Minimum Utilization (MU) as VM selection with various overload detection

Table 4 shows the simulation result of BASIP with Random Selection (RS) as VM selection with various overload detection

Table 5 shows the simulation result of BASIP with Minimum Utilization Rank (MUR) as VM selection with various overload detection. MUR is a the proposal method of the Ajith Singh et al., 2013

Figure 6 shows the energy consumption chart of BASIP
Figure 7 shows the SLA chart of BASIP
Figure 8 shows the migration chart of BASIP

Overload Detection/VM Selection	ENERGY	SLA	MIGRATION
IQR-MC	16.28	0.00035	748
LR-MC	16.21	0.00036	718
LRR-MC	17.03	0.00035	795
MAD-MC	16.40	0.00037	785
PR-MC	17.21	0.00034	765
THR-MC	16.53	0.00035	739

Table 1: BASIP - Maximum Correlation (MC) With Overload Detection

Overload Detection/VM Selection	ENERGY	SLA	MIGRATION
IQR-MMT	17.04	0.00034	820
LR-MMT	18.07	0.00029	714
LRR-MMT	16.55	0.00036	779
MAD-MMT	17.03	0.00033	778
PR-MMT	16.83	0.00032	776
THR-MMT	16.85	0.00033	792

Table 2: BASIP - Minimum Migration Time (MMT) With Overload Detection

Overload Detection/VM Selection	ENERGY	SLA	MIGRATION
IQR-MU	16.52	0.00036	744
LR-MU	16.65	0.00032	720
LRR-MU	17.27	0.00032	738
MAD-MU	17.35	0.00032	761
PR-MU	16.88	0.00031	773
THR-MU	16.61	0.00034	746

Table 3: BASIP - Minimum Utilization (MU) With Overload Detection

Overload Detection/VM Selection	ENERGY	SLA	MIGRATION
IQR-RS	17.80	0.00028	721
LR-RS	16.03	0.00035	772
LRR-RS	16.94	0.00035	730
MAD-RS	16.82	0.00030	691
PR-RS	17.08	0.00035	765
THR-RS	17.26	0.00032	735

Table 4: BASIP - Random Selection (RS) With Overload Detection

Overload Detection/VM Selection	ENERGY	SLA	MIGRATION
IQR-MUR	17.19	0.00031	699
LR-MUR	15.90	0.00039	831
LRR-MUR	16.19	0.00037	762
MAD-MUR	17.09	0.00037	793
PR-MUR	16.82	0.00031	774
THR-MUR	17.13	0.00034	784

Table 5: BASIP - Minimum Utilization Rank (MUR) With Overload Detection

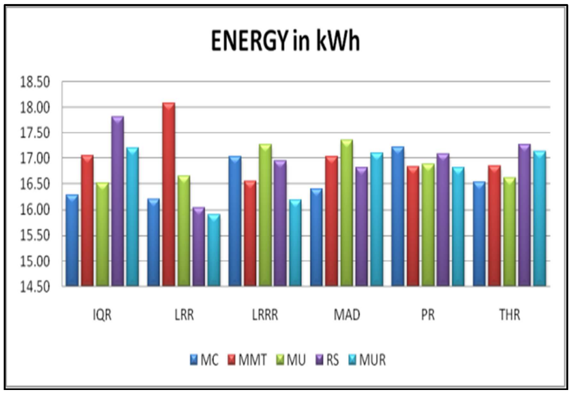

Figure 6: Energy in kWh using BASIP

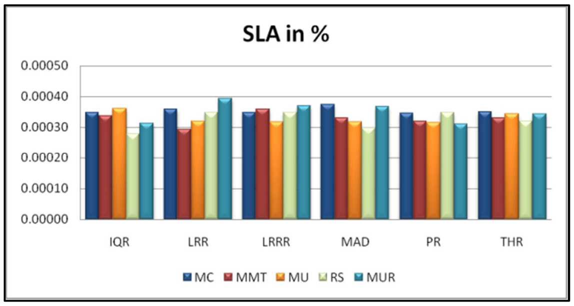

Figure 7: SLA - BASIP

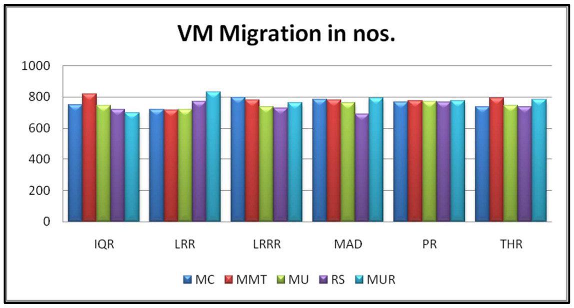

Figure 8: Migration - BASIP

In this work, researcher not only simulated with BASIP the proposed algorithm but to see and compare the result it is simulated with Banker's algorithm as well as SIP also. Below is the result after simulation.

	ENERGY kWh	SLA %	MIGRATION
BASIP	LR-MUR	IQR-RS	MAD-RS
	15.90	0.00028	691
SIP	LR-MC	PR-MU	PR-MU
	21.13	0.00019	776
BANKER	LRR-MU	IQR-MU	IQR-MU
	24.15	0.00029	779

Table 6: Comparison Table of BASIP, SIP and BANKER

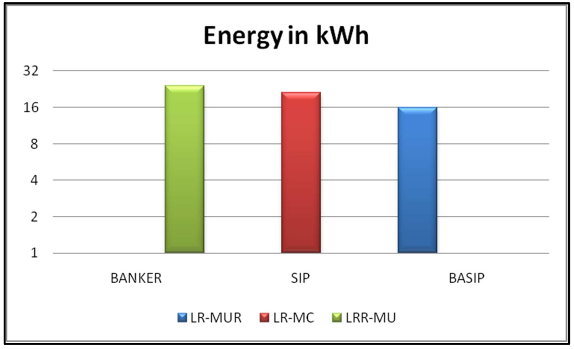

Figure 8: Energy Comparison of BANKER-SIP-BASIP

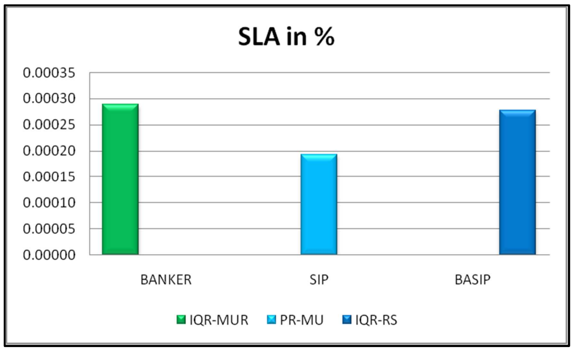

Figure 9: SLA Comparison of BANKER-SIP-BASIP

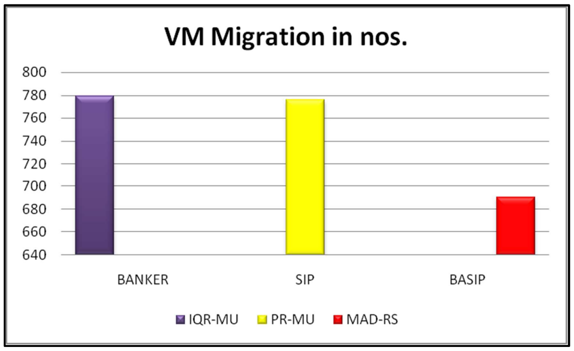

Figure 10: Migration Comparison of BANKER-SIP-BASIP

Overload Detection: DVFS-Dynamic Voltage Frequency Scaling, IQR-Interquartile Range, LR-Local Regression, LRR-Robust Local Regression, MAD-Median Absolute Deviation, THR- CPU utilization threshold, PR - Polynomial Regression
VM Selection Policy: MC-Maximum Correlation, MMT-Minimum Migration Time, RS-Random Selection, MUR - Minimum utilization Rank

6. CONCLUSION

As the increase in size of the data centres and with the cloud computing and virtualization technology, automation process of virtual machine placement has become an important issue. Virtual machine presents a great opportunity for cloud.

Cloud provider has to consider minimizing the cost and the factor related with that is processor, storage, memory, network and power. Virtualization technology benefits the computer and IT industries by enabling users to share expensive hardware resources by multiplexing VMs on the same set of hardware hosts. In this paper, an approach for solving the Virtual machine problem is dealt by BASIP (Banker Algorithm Stochastic Integer Programming). The study deal with processor, memory and power based, algorithm try to minimize the power usage by migration of VM from one host to another host that is known as VM placement. The host was shut down which are in idle mode or put in sleep which ultimately reduce the power consumption and reduce the cost factor, paper also consider the SLA are not violated while considering the reduction of power. The proposed algorithm BASIP take care that while placing VM in the host it check whether it is safe or unsafe to be allocated that in future such deadlock or resource shortage should not occur while considering the cost. BASIP combine with overload detection of LR and VM selection MUR reduces the energy upto **15.90**, which is the lowest, using of this placement technique in cloud data centre will significantly reduce the energy consumption and cloud provider and user will save lots of money on IaaS.

REFERENCE:

[1] http://www.heraldonline.com/2013/08/20/5130129/cloud-computing-market-revenue.html
[2] http://en.wikipedia.org/wiki/Amazon.com
[3] http://aws.amazon.com
[4] http://static.googleusercontent.com/external_content/untrusted_dlcp/www.google.com/en//green/pdf/google-apps.pdf
[5] Wikipedia deadlock
[6] GuiyiWei, Athanasios V. Vasilakos,Yao Zheng &Naixue Xiong, "A game-theoretic method of fair resource allocation for cloud computing services". *J Supercomput*, 54(1), 2009, pp252–269.
[7] Mohsen Sharifi, Hadi Salimi,Mahsa Najafzadeh. "Power-efficient distributed scheduling of virtual machines using workload-aware consolidation techniques." *The Journal of Supercomputing*, 61(1), 2011, pp 46-66.
[8] Sivadon Chaisiri, Bu-Sung Lee, and Dusit Niyato, "Optimal virtual machine placement across multiple cloud providers". In Markus Kirchberg, Patrick C. K. Hung, Barbara Carminati, Chi-Hung Chi, Rajaraman Kanagasabai, Emanuele Della Valle, Kun-Chan Lan, and LingJyh Chen, editors, APSCC, 2009, pp103–110. IEEE
[9] Jiandun Li,Junjie Peng, Wu Zhang, "A Scheduling Algorithm for Private Clouds". Journal of Convergence Information Technology, 2011(7).
[10] Saurabh Kumar Garg, Chee Shin Yeo, Arun Anandasivam, and Rajkumar Buyya, 2011.

"Environment-Conscious Scheduling of HPC Applications on Distributed Cloud-oriented Data Centers. Journal of Parallel and Distributed Computing (JPDC), 71(6): 732-749

[11] Jeffrey M. Galloway, Karl L. Smith, Susan S. Vrbsky, "Power Aware Load Balancing for Cloud Computing". Proceedings of the World Congress on Engineering and Computer Science 2011, San Francisco, USA. 1(1)

[12] Can Hankendi and Ayse K. Coskun "Adaptive Energy-Efficient Resource Sharing for Multi-threaded Workloads in Virtualized Systemss". IRNet IEEE International Conference on Cluster, Cloud and Grid Computing, 2011

[13] Andrew J. Younge, Gregor von Laszewski, Lizhe Wang, Sonia Lopez-Alarcon, Warren Carithers, "Efficient resource management for Cloud computing environments". Green Computing Conference 1(1), 2010, pp357-364

[14] D. Kusic et al., 2009. Power and performance management of virtualized computing environments via lookahead control. Cluster Computing, 12(1):1-15

[15] Y. Song et al., "Multi-Tiered On-Demand resource scheduling for VM-Based data center." In Proc. of the 2009 9th IEEE/ACM Intl. Symp. on Cluster Computing, 2009, pp 148-155

[16] Gupta, A., D. Milojicic, and L.V. Kalé. Optimizing VM Placement for HPC in the Cloud, in Proceedings of the 2012 workshop on Cloud services, federation, and the 8th open cirrus summit 2012: ACM

[17] Y. Yu and Y. Gao, "Constraint Programming-Based Virtual Machines Placement Algorithm in Datacentre". International Federation for Information Processing, 2012, 295–304

[18] http://openstack-neat.org

[19] http://stackoverflow.com/questions/10717767/mysql-magento-performance-issues-on-amazon-ec2

[20] https://bugs.launchpad.net/ubuntu/+source/linux-ec2/+bug/929941

[21] https://groups.google.com/forum/#!topic/google-appengine/SyWkwNzjei4

[22] http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2041544

[23] Anton Beloglazov and Rajkumar Buyya, " Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers". Software: Practice and Experience, 41(1), 2011, pp23–50

[24] W. Dijkstra, Edsger,. "Selected Writings on Computing: A Personal Perspective". Springer Verlag, 1982

[25] LANG, S.-D, "An extended banker's algorithm for deadlock avoidance". IEEE Trans. Software Eng 1999, 25, 3

[26] S. A. Reveliotis, "Variations on Banker',s algorithm for resource allocation systems". Proc. Japan-USA Symp. Flexible Automat., 1998, pp.1185 -1192

[27] Sheau-DongLang, "An Extended Banker's Algorithm for Deadlock Avoidance". IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 25(3), 1999

[28] M. Martin, "Deadlock Avoidance in Distributed Service Oriented Architectures". Norman, OK: MS Thesis, University of Oklahoma, 2010

[29] Martin M., Grounds N.G., Antonio J.K., Crawford K. and Madden J, "Banker's Deadlock Avoidance Algorithm for Distributed Service-Oriented Architectures", PDPTACSREA Press, 1(1), 2010, pp43-50.

[30] Calheiros, R. N. , et al. "Cloudsim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services". Technical Report, GRIDS-TR-2009-1, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia