VOL. 9, NO. 4, APRIL 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

POWER AWARE BASED WORKFLOW MODEL OPTIMIZATION USING FUZZY BEE COLONY METHOD

T. Sunil Kumar Reddy[1], P. V. Krishna[2] and P. Chenna Reddy[3]
[1]Department of CSE, SVTM, Madanapalle, India
[2]SCSE, VIT University, Vellore, India
[3]Department of CSE, JNTU University, Pulivendula, India
E-Mail: sunilre2013@gmail.com

ABSTRACT

Grid computing uses the concept of resource sharing which is used in virtual machines and cores, to enhance the capacity of a parallel CPU. The energy utilized and the time taken for executing a task is the most important criteria to be considered during the work-flow scheduling. This paper recommends the frame work of PERMA-G implementing the techniques of optimization, modeled after the fuzzy bee colony method as a solution to the issues of energy utilization and task completion time by the virtual machines. It replicates the honey bees' nature on fuzzy to dynamically calculate the time taken to run the computational task schedules to decrease the power utilization, cost and the task completion time. Performance evaluation is on the basis of service for Grid, power utilization and process.

Keywords: power estimation, power reduction, Grid load balancing, fuzziness.

1. INTRODUCTION

Distributed computing model is the basis for Grid computing to provide services to diverse applications through resource sharing using varied co-ordination [1]. The Grid computing system permits both heterogeneous and parallel computing for collating and resource sharing to the concepts such as virtualization and resource consumption for on demand situations in software services [2]. Power consumption is one amongst the significant issues to cater to the demands of diverse applications in Grid resource-scheduling. Decreasing the execution time is critical to performance factors with workflow-scheduling. Execution time can be decreased by increasing the clock frequencies but it results in increased power consumption and heat dissipation [3]. There should be a judicious balancing between the performance and power consumption. Energy-aware design need not always decrease the energy or power consumption. But, there is a possibility to reduce peak consumption of power with delay in the processor. The efficient technique to decrease power consumption by decreasing CPU voltage is done via Dynamic Voltage Scaling (DVS), where the CPU is slowed down but with minimal performance loss. DVS is the algorithm suggested for Source language level (High Level Language) which varies with different languages. DVS is implemented in PERMA framework [1] in areas having just one entry and one exit for control. Execution time and cost efficiency are the most critical QoS factors for heterogeneous and Grid computational systems. In [4], the algorithm of Optimizing Probabilistic Load balancing in the technology of Grid computing selects the top resources depending on the past status maintained by the minimum completion time. Response time is decreased by establishing the load balancing. In [5], we have a Task Load Balancing Strategy, a technique of hierarchical load balancing on the basis of neighborhood property. Local balancing gets priority privilege in this strategy followed by upper hierarchical balancing.

In Grid computing, Dynamic Load Balancing [6] depicts a model of task load balancing in the ambience of Grid using the system details to the earlier reference. This strategy has the following salient features: (i) It utilizes task-level balancing of load; (ii) It privileges transfer of local tasks in order to decrease cost of communication; (iii) It implements the strategy of distributing with decisions made locally. This system transfers the Grid to the tree structure, irrespective of the topological structure of the Grid.

The foraging of honey-bee [7][9], applied a natural phenomenon for the method of distributed biased random sampling in order to maintain the load of individual node through an evaluation of the closest global mean measure. Lastly, similar services connected through local rewiring are measured as a method of enhancing the load balancing through active system restructuring. Regarding the load balancing, as the need for web servers goes up or down, the dynamic services are assigned to adjust to the varying needs of the customer. The servers are arranged in a group for virtual machines (VMs), with separate virtual service queues for each VM. All servers that process a request out of their own queue compute a reward or a profit, which is equivalent to the quality shown in the waggle dance of the bees. In [8], a mathematical model of the framework of cloud computing with optimization technique of fuzzy bee colony is illustrated. The algorithm of Honey Bee Colony [10][11] is applied in scattered web servers for their web services.

PERMA-G framework employs the optimization technique of the fuzzy bee colony to solve the concerns of energy utilization and task execution time in virtual machines. It decreases the power consumption with the time for execution in VM through effective load balancing among the VMs. This paper depicts a model that identifies the amount of voltage needed to decrease the power utilization without impacting the time for execution and additionally extended to Grid through effective task

VOL. 9, NO. 4, APRIL 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

distribution based on load and the distance from the resource[12]. Organization of the paper is as follows: Section 2 explains the framework in the method proposed, the Optimized Fuzzy Bee PERMA-G (OFB-P). Section 3 presents the OFB-P model. Section 4 has a brief description of the OFB-P algorithm. Section 5 presents the discussion of the case and the results. Finally, section 6 offers a conclusion.

2. THE FRAMEWORK OF OPTIMIZED FUZZY BEE PERMA-G (OFB-P)

It is mostly the responsibility of the Grid to distribute an application for parallelizing among the machines, to handle applications executed among the machines and finally, to recover after detecting machine failures. We suggest the frame work of PERMA-G implementing the technique of fuzzy bee colony optimization, as presented in Fig 1.The PERMA-G presents the new model which calculates the energy utilization of tasks to host, scheduled in Grid systems running on multi-core. The suggested model describes resource computing in a Grid system through four parameters:

I) The machines' computational power or the count of operations that can be computed by its processor, i.e. μ_i.
II) The overall count of VMs integrating the processor i.e., N (VMs).
III) The energy utilization when the state of the processor is idle, i.e. E_i
IV) The energy utilization when a processor is loaded fully, i.e. E_l

The frame work of PERMA-G calculates the power μ_i, using energy consumption E for scheduling of VMs optimized by the technique of fuzzy bee colony for various resources. It assesses variation in the levels of discrete voltage among the devices, taking the overall execution time (ET) into consideration for the assigned tasks.

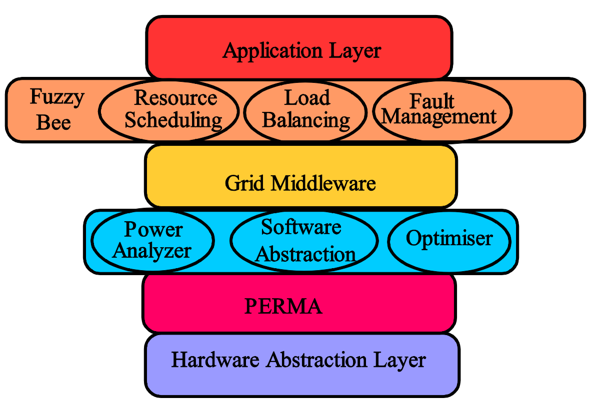

Figure-1. PERMA-G framework optimized by fuzzy bee optimization model.

Fig.1 depicts the PERMA framework's [3] deployment over the environment of Grid computing known as PERMA-G

3. OPTIMIZED FUZZY BEE PERMA-G MODEL

The nectar source, required by the scout bee R_n at n^{th} location is searched. The search of the scout bee starts from the hive and moves towards the location of the nectar from the initial location to the n^{th} location. Assume E_{ij} to be the time the bee takes to get to the required source of the nectar. In this case, the nectar is the resource which has to be assigned for the available Grid task as per the best resource fit. Once the nectar is found in time, the scout bee returns to the hive and dances so that the other scouts are influenced. When the resource is identified, it gives the request to allocate the available task from the Grid to provide web services. During the time of identification of the required source of nectar, the scout bee moves further to find some other nectar. Some more resources are identified based on the need within the time the resource completes the task. So, fuzziness can be used to get multiple degrees of nectar and also for storing that level of fuzzy nectar into a database of bee metaphor for future reference. This fuzziness is further applied to obtain multiple levels of resources based on the task execution time. It is then stored as a degree of fuzzy for future reference into the metaphor database.

Assume $[E_1, E_2]$ to be the interval graph to the hive, for a specific fuzzy nectar degree being searched by the bee. We then get MinT: $[E_1, E_2] \rightarrow I$. Here I represents the unit interval [0,1]. According to the mathematical model, as soon as the required nectar is located, then scheduling of web service is done to the specific resource. The algorithm is suggested to search for a particular service in a Grid that is designed based on considerations of power saving to meet the OoS and the constraints of energy. E_{xy} can be used to denote the finishing time of a specific task T by ECT and execution time of a specific task T on a Virtual Machine VMs. During the load balancing on the basis of execution time, the tasks are relocated among the VMs to reduce the time of response. The task execution time changes with different VMs depending on the capacity of the VM. Balancing execution optimally between the VMs gives effective task completion time. The dynamic technique of OFB-P balances the load by factoring the task priorities depending on power consumption and execution time.

The tasks separated from the over-loaded VMs behave as Honey Bees. When submitted to an under-loaded VM, it updates the number of different priority tasks, depending on the execution as well as the load of tasks allocated to that VM. This is maintained duly in the datacenter accessed by the other tasks for allocating VMs. As the sorting of all VMs is done in an ascending order, the removed tasks are submitted to the VMs that are under-loaded. Using the information from the datacenter, the present work load of every available VM can be computed. The algorithm is a blend of the fuzziness

VOL. 9, NO. 4, APRIL 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

concept based on honey bee behaviour and the technique of dynamic task resource balancing.

$T_{xy} = 0$, when the bee does not shift out of the hive and dies due to starvation. The task expires without execution as it fails to move towards any resource.

$$T_{xy} = \frac{[a_{xy}(t)]^{\alpha} - [E_{xy}]^{\beta}}{\sum [a_{xy}(t)]^{\alpha} - [\frac{1}{E_{xy}}]^{\beta}}$$ When the bee moves towards the permitted nectars from the hive, where a_{xy} (t) depicts the arc fitness that measures the time taken to shift to the nectar y, from the hive arc, in a time t. When T_{xy} = 1, then the bee moves to the required nectar from the hive i.e., the task is assigned to the required resources. The binary variable α turns the arc's fitness off or on and the parameter β controls the considerable level of heuristic distance.

The functional objective being

$$MinT = \sum_{(x,y)=0}^{n} E_{xy} T_{xy} \qquad (1)$$

It must meet the following criteria

$E_{xy} >= 0$
$E_{xy} + E_{yz} >= E_{yz}$

The VM (C_{VM}) capacity is measured through P_i × Mips in which the processor is P_i. The overall VMs capacity (OC_{VM}) is given by the mean of C_{VM}. The VM with the loaded task is assessed by dividing the total count of tasks by the service rate (T_{VM}). The mean of OT_{VM} evaluates the overall tasks loaded in every VM (OT_{VM}) in the datacenter. Execution time needed for the VM (E_{VM}) is assessed by dividing T_{VM} with C_{VM} and the same needed for all VMs in the datacenter OE_{VM} is assessed by dividing OT_{VM} with OC_{VM}.

$$MinL = \sum_{(x,y)=0}^{n} (E_{VM} - OE_{VM})^2 \qquad (2)$$

The following criteria must be satisfied:
$E_{VM} >= 0$
$E_{VM} + E_{VM} >= E_{VM}$
The bee represents the task and the nectar represents the resource. Various levels of fuzziness are estimated by utilizing equations 1 and 2. If the MinL is below fuzziness and between [0-1], the system is considered balanced [13], otherwise, the system is unbalanced.

4. OPTIMIZED FUZZY BEE PERMA-G ALGORITHM (OFB-P) ALGORITHM

Input: T task
1: Estimate available task T, capacity and load to VMs to initialize the stable condition
2: If $T_{xy} = 1$ and MinL $\leq T_{xy}$
3: system is in stable state
4: Exit
5: Decision for handling instable state
6: If $T_{xy} = 0$ and L > OT_{VM}
7: Task load balancing and VMs allocation is not possible
8: else
9: Trigger task load balancing and VMs allocation
10: supply UVM
11: $UVM = OC_{VM} - \frac{OT_{VM}}{C_{VM}}$
12: supply OVM
13: $OVM = \frac{OT_{VM}}{C_{VM}} - OC_{VM}$
14: Sort UVM in ascending order and OVM in descending order
15: While LVM and OVM ≠ NuLL
16: Sort task in VMs by execution time
17: For each task T in VMs find LVM
18: if T is non preemptive
19: T_{he}= VM|MinT|MinL ∈ VM and $T_{VM} \leq C_{VM}$
20: T_{me}= VM|MinL ∈ VM and $T_{VM} \leq C_{VM}$
21: T_{le}= VM|MinT ∈ VM and $T_{VM} \leq C_{VM}$
22: if T is preemptive
19: T_{he}= VM|MinT|MinL ∈ VM
20: T_{me}= VM|MinL ∈ VM
21: T_{le}= VM|MinT ∈ VM
22: update number of task assigned to VM
23: update number of task based on execution time assigned to VM
24: update load on VMs
25: update sets UVM and LVM
26: end while

5. RESULT AND DISCUSSIONS

A cloud computing simulator has been used to evaluate the performance of our algorithm depending on reduction in execution time. The classes on the simulator have been extended to match the algorithm implementation. The makespan and cost of popular algorithms for scheduling such as Round Robin (RR), Non power aware (NPA), First Come First Serve (FCFS) Dynamic Voltage and Frequency Scaling (DVFS) are assessed in terms of MIPS, VMs, utilization of power and the host type as reflected in Fig(2). The suggested algorithm addresses the issues of least completion time with cost efficiency for task scheduling to give optimum results.

Table-1. The simulation setup.

Simulation time	80100
Host	10-50
VMs	50-300
RAM	512-24576
Bandwidth	1000-1000000
Scheduling	FCFS, DVFS, RR

VOL. 9, NO. 4, APRIL 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Table-2. Virtual machines setup.

Resource/ VM	VM Type1	VM Type2	VM Type3	VM Type4
MiPs	750	1000	1500	2000
Cores	1	1	1	1
RAM	512	512	1024	1024
Bandwidth	1000	1000	1000	1000
Storage	25000	25000	25000	25000

Table-3. Host setup.

Resource/ Host	Host type0	Host type1	Host type2	Host type3	Host type4
MiPs	1500	2000	2500	4200	6000
Cores	1	1	2	4	4
RAM	24576	24576	24576	24576	24576
Bandwidth	100000	100000	100000	100000	100000
Storage	100000000	100000000	1000000000	100000000	100000000

The simulation setup for the performance evaluation of OFB-P is depicted in table 3. The average power utilization and the evaluated VMs display better output in comparison to other approaches as illustrated in Fig (2) and (3).

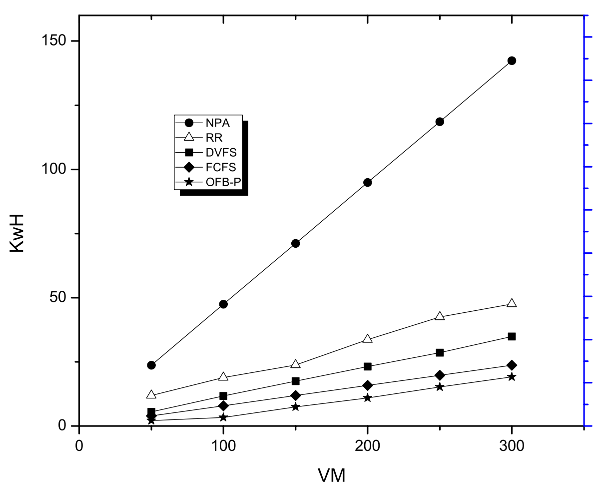

Figure-2. Power utilization based on number of VMs.

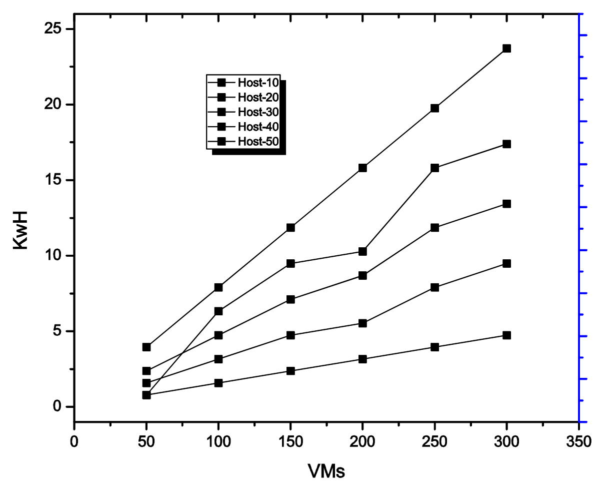

Figure-3. Power utilization based on Host and VMs.

The host's static power is efficiently made use of at an optimum level via VMs for a larger count of the execution of MIPS on type 4 hosts with VMs as depicted in Fig (3).

VOL. 9, NO. 4, APRIL 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

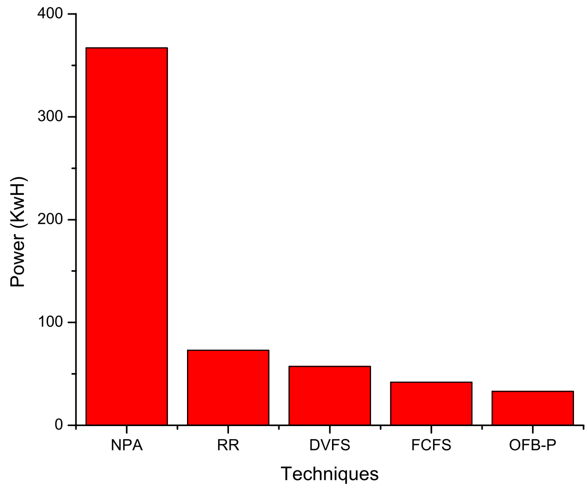

Figure-4. Power utilization performance.

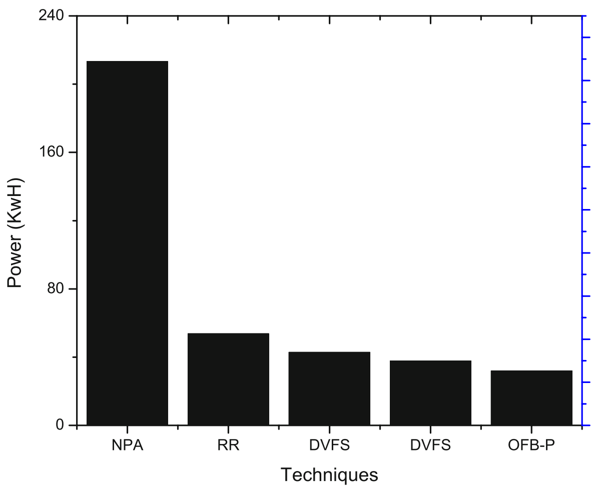

Figure-5. Power utilization performance for the 150 Host with 800 VMs.

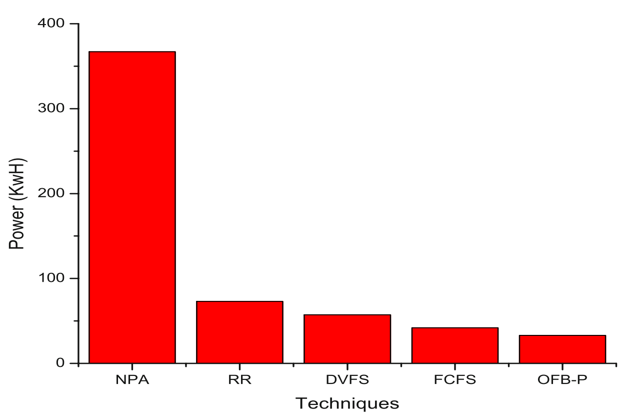

Figure-6. Power utilization performance for the 350 Host with 800 VMs.

Utilization of power with varied number of hosts and with a fixed count of VMs is evaluated for diverse methods as presented in Fig (4) through Fig (6).

6. CONCLUSIONS

The proposed model is simple and efficient in its approach for power estimation and advances a formula to decrease energy utilization by means of execution time for Grid. This paper puts forward optimized PERMA-G frame work, implementing the optimization technique of fuzzy bee colony. It provides comprehensive solutions to all the issues pertaining to power consumption and task execution time in virtual machines. This framework utilizes fuzzy bee technique to balance the virtual machines' load depending on the task execution time and resource allocation time for the task. This technology enhances the framework of PERMA-G with optimization by a gross reduction both in power consumption and task execution time over the Grid.

REFERENCES

[1] Buyya R. and Pandey S. and Vecchiola C. 2009. Cloudbus toolkit for market-oriented cloud computing. Proceeding of the 1st International Conference on Cloud Computing (CloudCom).

[2] Suraj Pandey, Dileban Karunamoorthy and Rajkumar Buyya. 2011. Workflow Engine for Clouds, Cloud Computing: Principles and Paradigms. R. Buyya, J. Broberg, A. Goscinski (eds), ISBN-13: 978-0470887998, Wiley Press, New York, USA, February. pp. 321-344.

[3] M. Rajasekhara Babu, P. Venkata Krishna and M. Khalid. 2013. A framework for power estimation and reduction in multi-core architectures using basic block approach. International Journal of Communication Networks and Distributed Systems. 10(1): 40-51.

[4] L.D. Dhinesh Babu and P. Venkata Krishna. Versatile time–cost algorithm (VTCA) for scheduling non–pre-emptive tasks of time critical workflows in cloud computing systems. International Journal of Communication Networks and Distributed Systems. 11(4): 390-411.

[5] M. Moradi, M.A. Dezfuli, M.H. Safavi. A New Time Optimizing Probabilistic Load Balancing Algorithm in Grid Computing. Department of Computer and IT, Engineering, Amirkabir University of Technology, Tehran, Iran. IEEE 978-1-4244- 6349-7/10/©2010.

[6] B. Yagoubi and Y. Slimani. 2007. Task load balancing strategy for grid computing. Journal of Computer Science. 3(3): 186-194.

ARPN Journal of Engineering and Applied Sciences

[7] B. Yagoubi and M. Medebber. 2007. A load balancing model for grid environment, computer and information science. iscis 2007. In: 22nd International Symposium on, 7-9 November. pp. 1-7.

[8] M. Randles, A. Taleb-Bendiab and D. Lamb. 2009. Scalable self governance using service communities as ambients. In: Proceedings of the IEEE Workshop on Software and Services Maintenance and Management (SSMM 2009) within the 4th IEEE Congress on Services. IEEE SERVICES-I 2009, July 6-10, Los Angeles, CA (to appear).

[9] K. Mukherjee and G. Sahoo. 2009. Mathematical model of cloud computing framework using fuzzy bee colony optimization technique. In: Proceedings of the 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, December 28-29. pp. 664-668.

[10] Dhinesh Babu L.D. and P. Venkata Krishna. 2013. Honey bee behavior inspired load balancing of tasks in cloud computing environments. Applied Soft Computing. 13: 2292-2303.

[11] J.S. Chase, D.E. Irwin, L.E. Grit, J.D. Moore, S.E. Sprenkle, Dynamic virtual clusters in a Grid site manager, in: Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing, Washington, DC, USA, 2003, pp. 90–98.

[12] A. Sodan, Service control with the preemptive parallel job scheduler Scojo- PECT, Journal of Cluster Computing (2010) 1–18.

[13] X. Ren, S. Lee, R. Eigenmann, S. Bagchi, Prediction of resource availability in fine-grained cycle sharing systems empirical evaluation, Journal of Grid Computing 5 (2) (2007) 173–195.