Application of Greedy Algorithms to Virtual Machine Distribution across Data Centers

Arnab Kumar Paul*, Sourav Kanti Addya†, Bibhudatta Sahoo‡ and Ashok Kumar Turuk§
Department of Computer Science and Engineering
National Institute of Technology, Rourkela, India
Email: {*arnabkrpaul, †kanti.sourav, ‡bibhudatta.sahoo, §akturuk}@gmail.com

***Abstract*—Cloud computing allows users to access resources on demand. The size of data centers increase with the increasing demand for resources by users. Increase in the size of data centers is directly proportional to energy consumption. The total energy requirement has to be minimized by distributing virtual machine requests over data centers optimally, with the consideration of prices of distribution of virtual machines. These two parameters are taken into account to frame the objective function for the Virtual Machine Distribution across Data Centers. Here both servers and workloads are classified as IO bound and CPU bound. A greedy algorithm framework has been used to obtain sub-optimal solutions for virtual machine distribution problem. Simulation results obtained indicates in favor of *best fit allocation*.**

Keywords*- cloud computing, data center, greedy algorithms, virtual machine

I. Introduction

Cloud computing acts as a model to enable users to have on demand access to computing resources with minimal management effort. It has emerged as a popular computing model to support large scale processing of huge volumes of data. Several multinational organizations such as Google, Yahoo, Microsoft, Amazon and IBM have built cloud platforms for enterprises and users to access the cloud services [1], [2].

Data Centers have been used to provide powerful computing resources for critical areas, such as nuclear physics, scientific simulation and geothermal experiments. A Data Center (DC) usually deploys a large number of Physical Machines (PMs) packed densely to maximize space utilization [3].

Virtualization is one of the key concepts of data center management. The major advantage of virtualization is the possibility of running several operating system instances on a single PM thus utilizing the hardware capabilities more fully which allows administrators to save money on hardware and energy costs. In literature, these individual operating system instances are defined as Virtual Machines (VM). The computing resources of DC are made available to the users through VMs. The VM scheduling in a cloud computing environment is very crucial as the number of users continuously increase. The VM scheduling algorithm greatly affects the performance of the whole system and its throughput [4].

The placement of VMs to DCs is called Virtual Machine Distribution (VMD). The effectiveness of VMD is related to a Quality of Service (QoS) as per the services. The objective of VMD will be a minimum number of data centers with a much higher per-data center utilization. More flexibility and availability of DCs is achieved, while the hardware costs and operational expenses such as power, physical space etc. are reduced [5].

In this paper, a scenario has been considered where Cloud Service Provider (CSP) receives requests for VMs from the users who need their applications to run on DC. An optimization problem has been formulated for the distribution of VM requests over DCs, which keeps into account the minimization of energy and expenses incurred to a Cloud Service Provider (CSP). The VMD problem has been presented as an optimization problem using the problem model of classical bin packing problem. The solution to the virtual machine distribution is presented using greedy algorithms. An analysis is made by comparing the simple greedy algorithms (Best-Fit(BF), First-Fit(FF), Next-Fit(NF), Worst-Fit(WF) and Random Allocation(RA)) [6].

The remainder of the paper is organized as follows. Section II discusses related work. Section III gives the problem definition along with the statement of the problem and the constraints. Section IV shows the results of simulation done on the proposed cloud model. Section V concludes the paper.

II. Related Work

Virtual machine distribution problem has been addressed in various research topics which include load placement over shared resources, dynamic resource allotment and the classical bin packing problem.

Bin Packing is one of the oldest and most well studied NP-hard problem in computer science [7], [8]. The classical bin packing problem formulates the decision of putting a number of items in fixed space bins. The main idea behind this is to minimize the number of bins used [5].

Finding solutions to bin-packing problems using heuristic algorithms has been addressed by several researchers. Some of the work falls into the category of greedy algorithms (e.g. Best-Fit(BF), First-Fit(FF), Next-Fit(NF), Worst-Fit(WF)) [6]. In [8] a survey has been done on the existing greedy heuristics for bin packing problem in one dimension. The study in [9]

978-1-4799-5364-6/14/$31.00 © 2014 IEEE

shows the approximability of vector bin packing(VBP) and the related multi-dimensional bin packing(MDBP) problem. Authors in [10] applied a First-Fit Decreasing(FFD) algorithm in a modified form. Here placement of the workload was modeled as an instance of 1-D bin packing problem.

In [11] a VM placement problem has been studied which maximizes the number of applications that can be hosted on a shared platform. There are many papers which have studies reflecting various resource management methods with varying principles in data center management [12]–[14]. In the earlier implementations of resource allocation, priority was given to users who had not received their share at the expense of those who had already surpassed their share [15], [16]. Lottery scheduling [17] and using an economic model [18] are two of the direct approaches which have been used where resources are allocated on the basis of lottery or capital. There has also been research on an online algorithm in [19] and [20] which caters to the changing resource requirements by allocating servers to application instances dynamically.

In [21] dynamic VM placement problem is addressed where an existing mapping is used as the initial point and then new placement solutions are generated to balance the load among hosts. In [22], Genetic Algorithm is used to simultaneously minimize total resource wastage, power consumption and thermal dissipation costs for VM placement. In [23], the VM placement problem is constructed as a constraint satisfaction problem with minimization of number of used servers and migration costs acts as the objective. Linear Programming is used for VM placement in [24] and [25]. In [26], constraint programming paradigm is used with a flexible and energy-aware framework in order to allocate VMs to data centers.

We have presented a cloud system model for distributing VM requests across DCs without violating the service level agreement (SLA). The greedy heuristics are applied for the distribution of VMs to optimize the energy and the expenses incurred to the CSP.

III. Distribution Model

The problem model that has been referred in this paper is depicted in Fig. 1. The proposed cloud architecture has been divided into three layers.

All the organizational and individual users reside in the topmost layer. The requests generated in this layer form a *VM Request Set* which is sent to the Cloud Service Provider Layer (CSPL). This layer acts as an intermediate layer which consists of two sub-layers. The upper sub-layer consists of the VM Request Set which comes from the topmost layer. The VM Requests consists of the CPU usage, memory, number of cores and the type of request (CPU based or IO based). The next sub-layer is the Data Center Control Layer (DCCL), which has information of the VM requests and the DCs. This layer is responsible for VM placement across DCs considering the different constraints namely, DC capacity constraint, user request constraint and the SLA between the user and CSP. Many heuristics are applied for the VM placement. VM request subsets placed into DCs are then passed to the next layer called the Data Center Set Layer (DCSL). This layer is responsible for PM allocation of the individual VM Request Subset into each DC. The assumption made in the paper states that each DC is composed of a large number of homogeneous PMs.

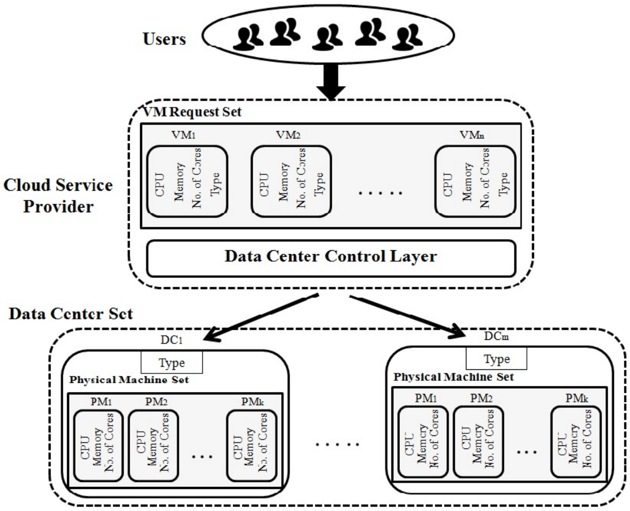

Fig. 1. Proposed Cloud Architecture

Virtualization technologies justify the assumption of using homogeneous resources in terms of capacity and computing capability [27] . The cloud architecture proposed in this paper is assumed to be placed at a particular geographic location. The maximum and minimum energy consumption differs in every DC denoted as *energymax* and *energyidle*. The distribution of VMs into different DCs also has a price associated with it, denoted as *dcprice*.

The VMD deals with n VM requests, each consisting of CPU speed, memory (RAM), number of cores and type of operations (CPU based or IO based) as requirements. All these requests constitute the VM request set $\{VM_1, VM_2, ..., VM_n\}$.

Each DC combines *k* homogeneous PMs where each PM has its own CPU, memory and number of cores as specifications. All the PMs constitute the Physical Machine Set (PMS) $\{PM_1, PM_2, ..., PM_k\}$ in every DC.

The DC Set consists of *m* DCs $\{DC_1, DC_2, ..., DC_m\}$. Here every DC has a type such that VM_i to be placed in DC_j should have the same type (CPU or IO). Every DC in the DC set is associated with a price to place the VM in the DC. This price is fixed in the SLA between the user and the CSP. A vector defined which consists of the prices of all the DC's in the DC set is as follows :

$$dcpricevector = \{dcprice_1, dcprice_2, ..., dcprice_m\}$$

Every DC is also associated with *energymax* and *energyidle* values which is used to calculate the energy consumed by the DC upon placing the VMs.

Both the energy vectors are defined as:

$$energymaxvector = \{dcemax_1, dcemax_2, ..., dcemax_m\}$$
$$energyidlevector = \{dceidle_1, dceidle_2, ..., dceidle_m\}$$

A. Constraints

The significant constraints for VM distribution across DCs, are as follows :

1) Assignment Constraint: This constraint sees to it that the DC into which the requested VM is to be placed provides all the requested resource demands in all the four dimensions considered here, namely CPU, memory, number of cores and type of request.

2) Capacity Constraint: This constraint checks the condition where the total resource requirements of the VM Request Set should be less than or equal to the total available resources of the DC Set in all the dimensions.

3) Placement Constraint: This defines the constraint to distribute a VM to only one data center from the DC Set provided all the resource requirements are satisfied in all dimensions.

B. Problem Statement

VMD problem is defined with consideration that there are *k* PMs in every DC. Suppose $CPM_r(i)$ defines the capacity of PM_i in r^{th} dimension.
Thus the capacity of a DC in r^{th} dimension can be written as:

$$\mathrm{CDC_r(i)} = \sum_{j=1}^{k} CPM_r(j) \tag{1}$$

The dimensions considered in this paper are CPU speed, memory and number of cores.

Here a homogeneous set of *m* data centers constitute the DC Set whose total capacity in the r^{th} dimension will thus be:

$$\mathrm{CDCSET_r} = \sum_{j=1}^{m} CDC_r(j) \tag{2}$$

Similar to equation 2, we can formulate the equation for the capacity of *n* VM requests which constitute the total capacity of the VM Request Set :

$$\mathrm{CVMRSET_r} = \sum_{j=1}^{n} CVM_r(j) \tag{3}$$

where $CVM_r(j)$ is the resource requirement of the j^{th} VM in the r^{th} dimension.

The utilization of DC *i* in the r^{th} dimension can be defined as the ratio of the total VM requirements of all the VMs placed in the i^{th} DC to the total resource of DC_i in the r^{th} dimension.

$$\mathrm{UDC_r(j)} = \frac{\sum_{\mathrm{i=1}}^{\mathrm{n}} CVM_r(i) \times V_{ij}}{CDC_r(j)} \ ; \ \forall j \in \{1, ..., m\} \tag{4}$$

where, $V_{ij} = 1$ if VM_i is placed in DC_j.

As discussed in a previous section, every DC is associated with three vectors, *dcpricevector*, *energymaxvector* and *energyidlevector*.
The *dcpricevector* continuously increases for the DCs in the DC Set from 1 to m. The three vectors are stated as follows :

$$dcpricevector = \{dcprice_1, dcprice_2, ..., dcprice_m\}$$
$$energymaxvector = \{dcemax_1, dcemax_2, ..., dcemax_m\}$$
$$energyidlevector = \{dceidle_1, dceidle_2, ..., dceidle_m\}$$

Taking the *energymaxvector* and *energyminvector* into account, the total energy consumption for the DC Set can be found. Energy consumption for DCj:

$$\mathrm{energyDC\,(j)} = ((\mathrm{dcemax_j} - \mathrm{dceidle_j}) \times \max_r\{UDC_r(j)\} + dceidle_j \quad ;\forall j \in \{1, ..., m\} \tag{5}$$

Here, the energy consumption of DC_j is calculated by adding the energy consumed by VMs placed into it and its idle energy consumption. The energy consumed by placed VMs is calculated by multiplying the maximum utilization of DC_j in all the three dimensions with the difference between the maximum and idle energy consumption of DC_j.

Thus the Total Energy Consumption (TEC) of the DCs in the DC Set can be calculated [27] by :

$$\mathrm{TEC} = \sum_{j=1}^{m} energyDC(j) \tag{6}$$

The two parameters associated with every data center are the price for distributing VMs across data center and the energy consumed by it. An objective function for DC_j (fnDC) can be formulated depending on the two parameters.

$$\mathrm{fnDC\,(j)} = \{\mathrm{w_{en}} \times energyDC(j)\} + \{w_{ex} \times (\sum_{i=1}^{n} dcprice_j \times V_{ij})\} \tag{7}$$

where w_{en} and w_{ex} are the weights associated with energy parameter and price parameter respectively, such that $w_{en} + w_{ex} = 1$ and $V_{ij} = 1$ if VM_i is placed in DC_j.

The overall function of the DC Set ($overallfnDC$) can be thus calculated by :

$$\mathrm{overallfnDC} = \sum_{j=1}^{m} fnDC(j) \tag{8}$$

$overallfnDC$ is calculated as the sum of the $fnDC(j)$ for all DCs in the DC Set. This value is used for analysis with respect to number of VM requests and is discussed in section IV. The above objective $overallfnDC$ can be achieved subjected to following constraints :

$$\sum_{i=1}^{n} CVM_r(i)\,.\,V_{ij} \le CDC_r(j) \ ; \ \forall j \in \{1, ..., m\}, \forall r \tag{9}$$

$$\sum_{i=1}^{n} CVM_r(i) \le CDCSET_r \tag{10}$$

$$\sum_{j=1}^{m} V_{ij} = 1 \ ; \forall i \in \{1, ..., n\} \tag{11}$$

$$V_{ij} \in \{0, 1\} \ ; \ \forall i \in \{1, ..., n\} \ , \ \forall j \in \{1, ..., m\} \tag{12}$$

Equation 9 satisfies the *Assignment Constraint* and equation 10 mathematically defines the *Capacity Constraint*. Equations 11 and 12 satisfy the *Placement Constraint*, where a VM can be placed in only one DC which is shown by the fact that V_{ij} can take only values 0 and 1.

The solution to VMD problem is NP-hard. Finding sub-optimal solution to NP-hard problems using algorithmic approach are roughly classified as *(i)* Exact Algorithms, *(ii)* Heuristics Algorithms and *(iii)* Approximation Algorithms [28]. This paper applies greedy heuristic algorithm of VM Request Set distribution across the DC Set on the basis of the *overallfnDC* and *TEC*. The DCCL uses all the information of the VM Request Set and the DC Set for VM distribution across DCs by applying certain greedy heuristics (random, next fit, first fit, best fit and worst fit) and some constraint conditions.

IV. Simulation Results and Discussion

The greedy heuristics applied to the proposed model in this paper are Random Allocation (RA), Next Fit Allocation (NFA), First Fit Allocation (FFA), Best Fit Allocation (BFA) and Worst Fit Allocation (WFA). We use an inhouse simulation using JAVA on a desktop computer with Intel (R) Core (TM) i7-3770 processor, 3.4 GHz and 4 GB memory, considering a fixed number of homogeneous DCs in the DC Set. The number of DCs considered in the simulation is fixed at 40. Every DC has 10 PMs each having 4 GB memory, 3.00 GHz cpu and 4 cores. VM Requests are generated in a uniformly distributed manner, where the values are generated randomly taking range for cpu fixed between 1 and 3 GHz, number of cores can be any integer value form 1 to 4 and memory can take values 1,2 or 4. For simulation purpose, the number of VM requests start from 25 and are taken upto 250 at intervals of 25 requests.

Since in this paper homogeneous PMs are considered and the number of PMs in a data center is constant, thus for all the simulations the values for FFA and BFA remain the same. Thus in the graphs shown below, only BFA is considered.

Fig. 2 shows the plot Number of Data Centers vs Number of VM Requests. It is observed that the number of DCs increases as the number of VMs is increased in NFA, BFA and RA. But it is seen that the number of DCs used remains 40 in WFA. This result arises due to the fact that WFA always allocates VM to the DC having the maximum amount of resources. Therefore, everytime a VM is to be placed a new data center is being looked into. Further observing the graph it is seen that BFA uses lesser number of DCs compared to NFA when there are large number of VM requests. This is because BFA considers the DCs from first, everytime a new VM is to be placed whereas NFA always considers distribution from the DC into the previous VM was distributed.

Fig. 3 displays the plot Total price vs Number of VM Requests. Since the number of DCs used increases on increasing the number of VM requests as seen in Fig. 2 and the fact considered in this paper that dcpricevector increases form DC_1 to DC_{40} explains the increase in the total price as the number of VM requests increase. Total price is calculated by Equation 13.

$$Totalprice = \sum_{j=1}^{m} \sum_{i=1}^{n} dcprice_j \times V_{ij} \tag{13}$$

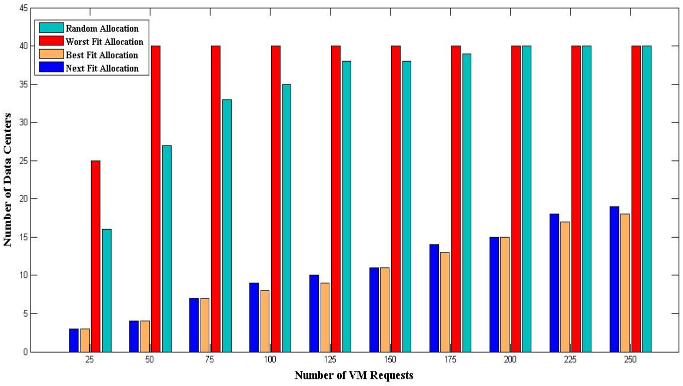

Fig. 2. Number of Data Centers vs Number of VM Requests

where, $V_{ij} = 1$ when VM_i is placed into DC_j.
It is also observed that as the number of VM requests increase BFA gives lesser price that NFA whose difference is negligible for small number of VM requests. RA and WFA is inconsistent as the price will depend on the VMs distributed to a particular DC.

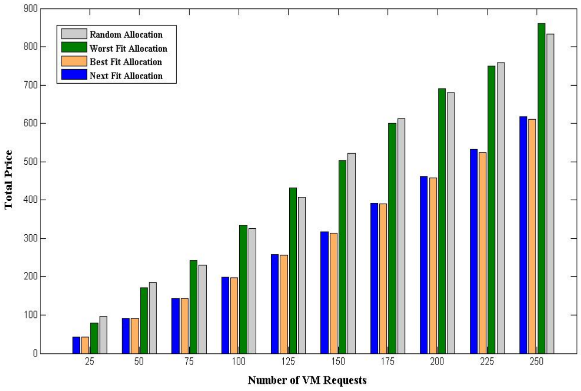

Fig. 3. Total price vs Number of VM Requests

Fig. 4 plots Energy Consumption vs Number of VM Requests. Energy consumption is calculated using equation 6. In contrast to figures 2 and 3, here the difference in energy consumption between BFA and WFA is not much. This is explained by the fact that energy consumption calculation depends on the utilization of a particular DC more than on the number of DCs used. Thus for WFA and RA, there are more number of DCs used each having lower utility than in the case for NFA and BFA where lesser number of DCs are used but each having more utility. Also total energy consumption increases continuously as the number of VM requests increase.

Fig. 5 shows the plot value of $overallfnDC$ vs Number of VM Requests. *overallfnDC* is calculated using equation 8 after VM distribution is completed subject to the constraints discussed in Equations 9, 10, 11 and 12. *overallfnDC* takes into account two parameters; energy and price. The weights

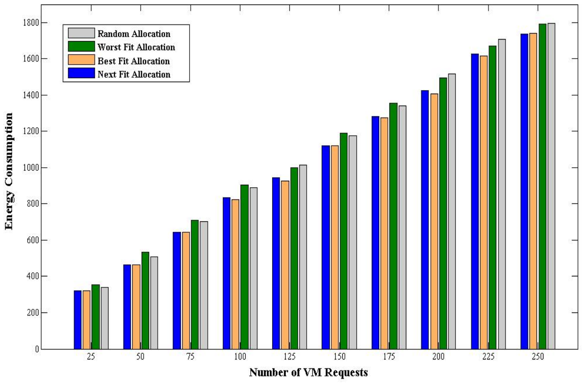

Fig. 4. Energy Consumption vs Number of VM Requests

associated with both the parameters are balanced by taking both values as 0.5. Thus it is seen that considering both number of DCs used and utility of DCs, the stark difference between BFA and WFA is reduced. NFA and FFA give similar values which is lesser than WFA and RA because of the difference in number of DCs used.

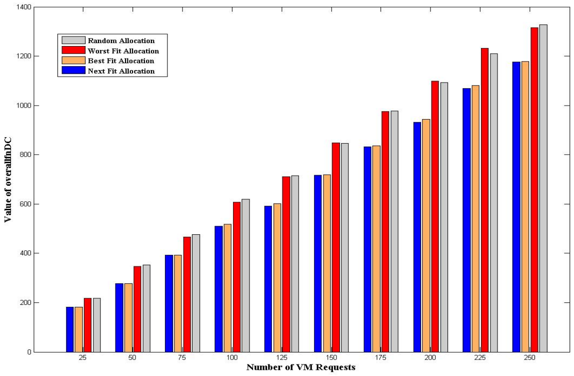

Fig. 5. Value of overallfnDC vs Number of VM Requests

All the above explained plots clearly show that BFA gives the best result for VM distribution across DCs for the proposed cloud model.

V. Conclusion

There has been a significant increase in the number of physical machines in each data center to deal with the increasing demands of resources which has resulted in increasing energy consumption and price for allocation. These two parameters have been studied in the cloud model proposed in this paper and analyzed by applying greedy algorithms. It is clearly seen that best fit allocation outperforms all the other greedy heuristics. In this work, three constraints have been considered, but the increasing availability of various applications on cloud leads to more number of constraints to be met in SLA. Performance of VM distribution on data center can be further analyzed using greedy heuristics for larger dimensions.

References

[1] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference*, 2009.

[2] B. F. Cooper, E. Baldeschwieler, R. Fonseca, J. J. Kistler, P. Narayan, C. Neerdaels, T. Negrin, R. Ramakrishnan, A. Silberstein, U. Srivastava *et al.*, "Building a cloud for yahoo!" *IEEE Data Eng. Bull.*, 2009.

[3] E. Mohammadi, M. Karimi, and S. R. Heikalabad, "A novel virtual machine placement in cloud computing," *Australian Journal of Basic and Applied Sciences*, 2011.

[4] M. T. Thiruvenkadam and V. Karthikeyani, "An approach to virtual machine placement problem in a datacenter environment based on overloaded resource," *International Journal of Computer Science and Mobile Computing*, 2014.

[5] J. Xu and J. A. Fortes, "Multi-objective virtual machine placement in virtualized data center environments," *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom), IEEE*, 2010.

[6] E. Feller, L. Rilling, and C. Morin, "Energy-aware ant colony based workload placement in clouds," in *Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing*. IEEE Computer Society, 2011, pp. 26–33.

[7] E. G. Coffman Jr, M. R. Garey, and D. S. Johnson, "Approximation algorithms for bin packing: A survey," in *Approximation algorithms for NP-hard problems*, , Ed., 1996.

[8] S. S. Seiden, "On the online bin packing problem," *Journal of the ACM (JACM),ACM*, 2002.

[9] C. Chekuri and S. Khanna, "On multi-dimensional packing problems." in *SODA*, vol. 99. Citeseer, 1999, pp. 185–194.

[10] A. Verma, P. Ahuja, and A. Neogi, "pmapper: power and migration cost aware application placement in virtualized systems," in *Middleware 2008*. Springer, 2008, pp. 243–264.

[11] B. Urgaonkar, A. L. Rosenberg, and P. Shenoy, "Application placement on a cluster of servers," *International Journal of Foundations of Computer Science*, 2007.

[12] S. Borst, O. Boxma, J. F. Groote, and S. Mauw, "Task allocation in a multi-server system," *Journal of Scheduling*, 2003.

[13] R. J. Al-Ali, K. Amin, G. Von Laszewski, O. F. Rana, D. W. Walker, M. Hategan, and N. Zaluzec, "Analysis and provision of qos for distributed grid applications," *Journal of Grid Computing*, 2004.

[14] A. K. Amoura, E. Bampis, C. Kenyon, and Y. Manoussakis, "Scheduling independent multiprocessor tasks," in *AlgorithmsESA'97*, , Ed., 1997.

[15] G. J. Henry, "The unix system: the fair share scheduler," *AT&T Bell Laboratories Technical Journal*, 1984.

[16] J. Kay and P. Lauder, "A fair share scheduler," *Communications of the ACM*, 1988.

[17] C. A. Waldspurger and W. E. Weihl, "Lottery scheduling: Flexible proportional-share resource management," in *Proceedings of the 1st USENIX conference on Operating Systems Design and Implementation*, , Ed., 1994.

[18] I. Stoica, H. Abdel-Wahab, and A. Pothen, "A microeconomic scheduler for parallel computers," in *Job Scheduling Strategies for Parallel Processing*, , Ed., 1995.

[19] A. Karve, T. Kimbrel, G. Pacifici, M. Spreitzer, M. Steinder, M. Sviridenko, and A. Tantawi, "Dynamic placement for clustered web applications," in *Proceedings of the 15th international conference on World Wide Web*. ACM, 2006, pp. 595–604.

[20] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, "A scalable application placement controller for enterprise data centers," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 331–340.

[21] C. Hyser, B. Mckee, R. Gardner, and B. J. Watson, "Autonomic virtual machine placement in the data center," *Hewlett Packard Laboratories, Tech. Rep. HPL-2007-189*, pp. 2007–189, 2007.

[22] J. Xu and J. A. Fortes, "Multi-objective virtual machine placement in virtualized data center environments," in *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*. IEEE, 2010, pp. 179–188.

[23] F. Hermenier, X. Lorca, J.-M. Menaud, G. Muller, and J. Lawall, "Entropy: a consolidation manager for clusters," in *Proceedings of*

the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments. ACM, 2009, pp. 41–50.

[24] B. Kantarci, L. Foschini, A. Corradi, and H. T. Mouftah, "Inter-and-intra data center vm-placement for energy-efficient large-scale cloud systems," in *Globecom Workshops (GC Wkshps), 2012 IEEE*. IEEE, 2012, pp. 708–713.

[25] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimal virtual machine placement across multiple cloud providers," in *Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific*. IEEE, 2009, pp. 103–110.

[26] C. Dupont, G. Giuliani, F. Hermenier, T. Schulze, and A. Somov, "An energy aware framework for virtual machine placement in cloud federated data centres," in *Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012 Third International Conference on*. IEEE, 2012, pp. 1–10.

[27] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," *The Journal of Supercomputing*, vol. 60, no. 2, pp. 268–280, 2012.

[28] D. S. Hochbaum, *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 1996.