Profiling: An Application Assignment Approach for Green Data Centers

Meera Vasudevan, Yu-Chu Tian, Maolin Tang
School of Electrical Engineering
and Computer Science
Queensland University of Technology
Brisbane Australia
{meera.vasudevan, y.tian, m.tang}@qut.edu.au

Erhan Kozan
School of Mathematical Sciences
Queensland University of Technology
Brisbane Australia
Email: e.kozan@qut.edu.au

***Abstract*—In the past few years, there has been a steady increase in the attention, importance and focus of green initiatives related to data centers. While various energy aware measures have been developed for data centers, the requirement of improving the performance efficiency of application assignment at the same time has yet to be fulfilled. For instance, many energy aware measures applied to data centers maintain a trade-off between energy consumption and Quality of Service (QoS). To address this problem, this paper presents a novel concept of profiling to facilitate offline optimization for a deterministic application assignment to virtual machines. Then, a profile-based model is established for obtaining near-optimal allocations of applications to virtual machines with consideration of three major objectives: energy cost, CPU utilization efficiency and application completion time. From this model, a profile-based and scalable matching algorithm is developed to solve the profile-based model. The assignment efficiency of our algorithm is then compared with that of the Hungarian algorithm, which does not scale well though giving the optimal solution.**

***Index Terms*—Energy efficiency, linear programming, heuristic algorithm, optimization, cost function**

I. Introduction

During the last few years, data center technology has evolved significantly. Conventionally, it has focused on providing more bandwidth and higher speed to support the ever increasing high-bandwidth applications. Nowadays, the number of connected file servers, database servers, storage servers, network components, and power and cooling systems within a data center is rapidly increasing. This is to satisfy the relentless need for more space, more storage facilities, more cooling and more bandwidth. Coupled with the everyday usage of data centers by the growing number of Internet users, the energy needed to power these systems is predictably escalating at an alarming pace. This demands efficient energy management for modern data centres.

The urgent importance of 'greening' data centers has made itself known in the form of carbon footprints and exorbitant energy costs [1]. According to Vaid [2], energy consumption accounts for more than 35% of the current data center operational expenses, and this estimate is expected to rise to 50% in a few years. Le *et al.* [3] have studied the energy and cost distribution of data centers. They conclude that deploying green initiatives at data centers reduces the carbon footprint by 35% at only a 3% cost increase. The Smart2020 analysis in [4] estimates that 14% of the Information and Communication Technology (ICT) carbon footprint is accounted for by data centers and these levels are projected to double by year 2020. Moreover, according to [5], data centers consume 1.1% to 1.5% of the world's total electricity.

Efforts have been made to address the 'greening' data center issue. For example, many organizations and researchers are developing energy aware measures that enable to perform more work with maximum performance efficiency and minimum energy consumption [6]. However, in most cases, deploying an energy optimized solution invariably degrades the performance efficiency in terms of inefficient CPU usage of physical nodes and delay in job completion times.

To tackle this challenging issue, this paper presents a novel concept of profiling to facilitate offline optimization for a deterministic application assignment to virtual machines. From this profiling concept, a profile-based linear programming model is established to describe the optimization problem. The model is solved through a simple profile-based matching algorithm. Case studies are also conducted to demonstrate our approach. The theory of utilizing application and virtual machine profiles in terms of energy-efficient application assignment is novel and as yet has remained unexplored.

The remainder of the paper is organized as follows. Section II reviews related work. Section III formally formulates the problem. A solution to the problem is presented in Section IV. Case studies are conducted in Section V to demonstrate our approach. Finally, Section VI concludes the paper.

II. Related Work

Application assignment involves assigning computing tasks or applications to data center resources such as CPU and memory. The factors considered in assignment schemes include application runtime, server workload, resource requirements or availability, energy consumption and performance efficiency. Thus, a key issue is how to formulate and solve the application assignment problem subject to various constraints.

Early attempts to discuss the classical assignment problem include [7], [8] and [9]. The most common approach used to solving these problems is to model the problem in linear

programming and then solve the problem with the Hungarian algorithm [10]. However, the Hungarian algorithm does not scale well and thus is not suitable for large-scale application assignment problems as those in modern data centers.

Later, the authors of [11] discussed the assignment problem of nurses qualified to carry out jobs in different units of a hospital with the side constraints of seniority and job priority. They took into consideration of absentee staff and unexpected work overload. To deal with side constraints, an efficient approach was presented in [12] through an adaptive task allocation procedure. Basically, the approach uses system history to make non-greedy task assignments. More specifically, a host node has the freedom to choose not to host a particular task if it is deemed not suitable by certain standards pertaining to individual host nodes.

Considering the allocation problem as a mixed integer non-linear programming model, the authors of [13] have tackled task allocation, task scheduling and voltage assignment simultaneously for multi-core processor systems. A branch and bound algorithm is used to solve the problem. The constraints involved in the problem formulation specify that every task can only be executed on one processor and each task can only be carried out after the completion of its predecessor. Employing similar constraints, our model in this paper specifies that every application is only executed on a single virtual machine.

Recently, a trust model based task scheduling algorithm has been proposed [14]. It considers not only the transmission time of the data files but also the trust level of the data file host nodes. Such a consideration helps improve the Min-Min task scheduling algorithm to ensure the success of the task execution. This algorithm is specifically designed for data intensive tasks, which require multiple data files stored in various nodes. It overcomes the problem of high probability of data loss and error usually associated with data intensive tasks. However, in doing so, the algorithm compromises on the task execution time. In contrast, our model to be presented in this paper attempts successful execution of tasks in a timely manner with the help of data provided by the profiles.

An integer programming model has been presented in [15] to generate timetables on a daily basis to achieve an optimal allocation of teachers to courses. It allocates the best available teachers to their relevant courses in the optimal time slot. This is achieved by adopting a weighting system in which the priority of the courses chosen by teachers is used to determine the objective values. The problem addressed in [15] is similar to that of our work, which assigns the tasks to the most suitable virtual machines. While we also use a weighting system, the weightings are formed in an energy matrix, which is derived from analyzing the profiles of both tasks and virtual machines to satisfy the task allocation constraints.

The concept of profiles has been used in some existing approaches for resource allocation. The authors of [16] have investigated the relationship between resource demands and application performance metrics. They have presented an application profiling technique using a Canonical Correlation Analysis (CCA) method. The CCA analyzes the performance efficiency of the applications in term of their resource usage and builds application profiles. The resulting profiles are then used to build a performance prediction model. Similarly, a task migration algorithm has been proposed in [17] based on the profiles created from the dynamic behaviour of the static task allocation. These profiles are then used to determine the migration destination of the tasks. While profiling has been previously discussed in terms of performance and behaviour analysis of the applications after assignment to processors, the application of profiles in the decision making process of initial and continued task allocation has not been discussed. This will be implemented in our work in this paper.

In [18], an application placement framework, named EAPAC, is proposed to assign a certain number of mixed data-intensive applications to servers and to resolve resource conflicts. It is realized that the application processing time increases when there is a concurrency of resource requests. This can be avoided by ensuring that a mixture of applications with different resource requests is assigned to individual servers. The EAPAC consists of an application level load balancer and an application server manager. The load balancer assigns applications to server hosts while the server manager monitors the resource provisioning amongst servers. The EAPAC is claimed to be able to improve the task response time by 4 times as compared to Tang's method presented in [19] for dynamic application placement in data centers. However, the EAPAC is intended for deployment in non-virtualized environments. In contrast, the coupled application placement (CPA) framework proposed in [20] is deployed in virtualized data centers.

After careful study of the various energy aware measures, the following technological gaps have been identified, which motivate our research in this paper:

- On entering the data center, every application undergoes placement processing regardless of the frequency of its execution. To avoid the redundancy in application processing, we employ the profiling technique consisting of data related to applications such as their resource requirements and the duration that saves on application processing time.
- Application completion time is unknown prior to scheduling. In our research, the profiles enable us to be aware of the average completion times of the applications prior to assigning them to virtual machines.

In this paper, a novel profile-based application assignment algorithm will be presented for optimizing the energy of data centers while maintaining the constraints of CPU utilization efficiency and application completion time. Profiling applications and virtual machines allow the allocation manager to be aware of the resource requirements and availability, and application completion times without having to process applications anew.

III. Problem Formulation

This section formulates our research problem, briefly discusses the novel concept of Profiles and then presents a profile-based assignment model.

The research problem addressed in this paper is to optimize the energy of data centers through profile-based application assignment whilst maintaining efficient performance levels in terms of CPU utilization and job completion time. In essence, we attempt to minimize the CPU power for each of the physical nodes, which host the virtual machines for the successful execution of applications in a timely manner.

A. Profiles

Profiling the applications and virtual machines allow the allocation manager to easily determine resource requirements and availability so that allocation decisions could be made promptly. With the help of the Profiles, the assignment algorithm will then identify the best possible virtual machine to host the application.

The initial step of building profiles involves the accumulation of a large amount of specific data such as energy, CPU, memory, application completion times, and application frequency. In this paper, our profiles primarily consider CPU utilization data and application completion times. Once the profiles have been created, they are updated regularly in order to maintain the performance efficiency.

Every incoming application to the data center and every virtual machine within the data center undergo profiling. Application profiles contain data related to the CPU requirements and average completion times of individual applications. Virtual machine profiles contain data related to the processing speed, availability and energy of the CPUs of the host nodes.

The amount of data needed to build profiles is large due to the extensive information required to carry out efficient sub-optimal allocations. However, once built, the profiles only need to be updated regularly with a considerably less processing time. Also, with the profiles, the allocation process speeds up and becomes easier to manage.

The profiles can aid in building a historical data of the applications and their execution frequency at the data center. This information can then be used to predict incoming applications to the data center and virtual machine availability at certain time periods. However, discussions of the prediction technique to the profile-based model is beyond the scope of this paper.

B. Mathematical Formulation

For N applications to be assigned to M virtual machines residing in L physical machines, denote

$$I \triangleq \{1, ..., N\}, \; J \triangleq \{1, ..., M\}, \; K \triangleq \{1, ..., L\}. \tag{1}$$

The assignment of an application a_i, $i \in I$, onto a virtual machine V_j, $j \in J$, is given by a binary decision variable $x_{ij}, i \in I, j \in J$, where:

$$x_{ij} = \begin{cases} 1 & \text{if } a_i \text{ is allocated to } V_j; \; i \in I, j \in J, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Our goal is to build a linear programming model to identify and carry out an optimal placement of applications on virtual machines based on CPU utilization efficiency, application completion time and energy cost. This is achieved through the use of the profiles of the applications and virtual machines.

1) CPU Utilization Efficiency: The profiles furnish our algorithm with the CPU requirement μ_i from application a_i, and CPU availability μ_j on virtual machine V_j, respectively, $i \in I, j \in J$. The CPU utilization efficiency μ_{ij}, is then defined as a function of both these values and expressed as a percentage as follows:

$$\mu_{ij} \triangleq \frac{\mu_i}{\mu_j} * 100; \; i \in I, j \in J. \tag{3}$$

2) Application Completion Time: The profiles include the average completion time θ_i of every application. To maintain the efficiency of allocation, the time T_{ij} taken by the individual virtual machine to complete an application must not be longer than the average application completion time θ_i, i.e.,

$$T_{ij} \leq \theta_i; \; i \in I, j \in J. \tag{4}$$

The concept behind this constraint is that an application may take 5 minutes to execute when assigned to machine V_1. However, the same application is capable of execution in 2 min when assigned to machine V_2 with a higher CPU speed.

3) Energy Cost: The energy cost of allocating application a_i to virtual machine V_j is denoted by $C_{ij}, i \in I, j \in J$. Algorithm 1 can be used to identify the minimum energy cost assignment from matrix $[C_{ij}]_{N \times M}$. The resulting energy cost of application allocation is directly proportional to the approximate power required to carry out the application in a virtual machine.

Algorithm 1: Derive minimum from energy cost matrix $[C_{ij}]_{N \times M}$

input : The cost matrix $[C_{ij}]_{N \times M}$
output: Allocation matrix $[x_{ij}]_{N \times M}$

```
Initialise [x_ij]_{N×M} as a null matrix ;
for i ← 1 to N do
    Set C_i1 as the minimum value ;
    Store the index data of C_i1 ;
    for j ← 1 to M do
        if C_ij is minimum then
            Update C_ij as the minimum value ;
            Update the index data ;
    Set x_ij to 1 as per the index data
```

Let w_k denote the total CPU utilization rate of a physical node P_k. It is expressed as:

$$w_k = \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} \mu_{ij} \tag{5}$$

Let P_k denote the energy consumption of the physical node. It is derived from the following equation [21]:

$$E(P_k) = \frac{(P_k^{max} - P_k^{idle}) * w_k}{100} + P_k^{idle}, \tag{6}$$

where P_k^{max} and P_k^{idle} are the power consumed at the maximum utilization and idle state, respectively. This equation

represents a linear relationship between the energy and CPU utilization [21], as graphically shown in Figure 1.

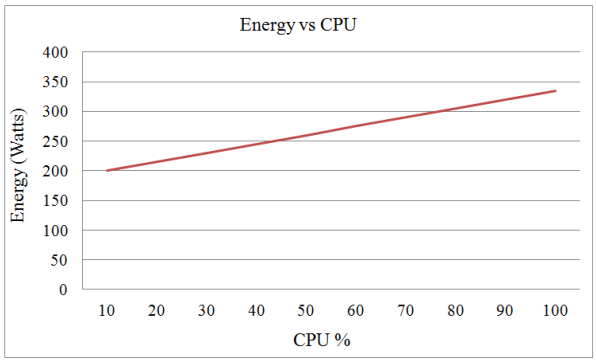

Fig. 1. Energy consumption versus CPU utilization [21].

C. Profile-based Assignment Model

The optimization problem featuring in our work is a semi-assignment problem. A set of N applications must be allocated to a set of M virtual machines until the applications are executed successfully.

The main objective of this problem is to minimize the total energy cost of the application assignment. A linear programming approach is employed to formulate this problem to make the best possible use of the available resources. It seeks to find optimal values from a set of feasible values for the decision variables. This will maximize or minimize the objective function and satisfy the given constraints.

The following are the constraints for our linear programming model:

- Constraint 1: CPU Utilization Efficiency. The major constraint ensures that the application assignment also takes into consideration the CPU utilization efficiency from Equation (3). A minimum CPU utilization efficiency value, α, is set for the assignment to take place: $\mu_{ij} \geq \alpha$. As a result, whilst most of the servers are working in acceptable capacity, the idle nodes can be switched off.
- Constraint 2: Application Completion Time as shown in Equation (4). The assignment needs to consider the discrete completion time that individual applications take in distinct virtual machines. These times are mainly dependent on the CPU speed and memory available to a virtual machine. Therefore, assignments that allow for a reasonable application completion time when compared to the average completion time should take place.
- Constraint 3: Each application a_i must be assigned to one virtual machine V_j only, in order to avoid redundancy in the form of multiple virtual machines attempting to execute the same application.
- Constraint 4: The maximum number of applications that can run on a virtual machine V_j at any given time is given by X_j^{max}. The value of X_j^{max} depends on virtual machine capacity and varies for different virtual machines. This constraint ensures that the virtual machine is not overloaded and continues to perform efficiently.
- Constraint 5: Binary Constraint as shown in Equation (2).

We define an objective function z to minimize the energy cost of application assignment. As a result, our Profile-based Assignment Model is as follows:

$$
\begin{aligned}
\min z \quad &= \quad \sum_{j=1}^{m}\sum_{i=1}^{n} C_{ij}x_{ij} \qquad (7)\\
\text{s.t.} \quad & \quad \mu_{ij} \geq \alpha;\ \forall i \in I, j \in J;\\
& \quad T_{ij} \leq \theta_i;\ \forall i \in I, j \in J;\\
& \quad \sum_{j=1}^{M} x_{ij} = 1 \ \ \forall i \in I;\\
& \quad \sum_{i=1}^{N} x_{ij} \leq X_j^{max};\ \forall j \in J;\\
& \quad x_{ij} = 0 \text{ or } 1;\ \forall i \in I, j \in J.
\end{aligned}
$$

Figure 2 displays a flowchart on the working of our profile-based assignment model. Solving the model, a Profile-based Matching Algorithm will ne developed in the next section.

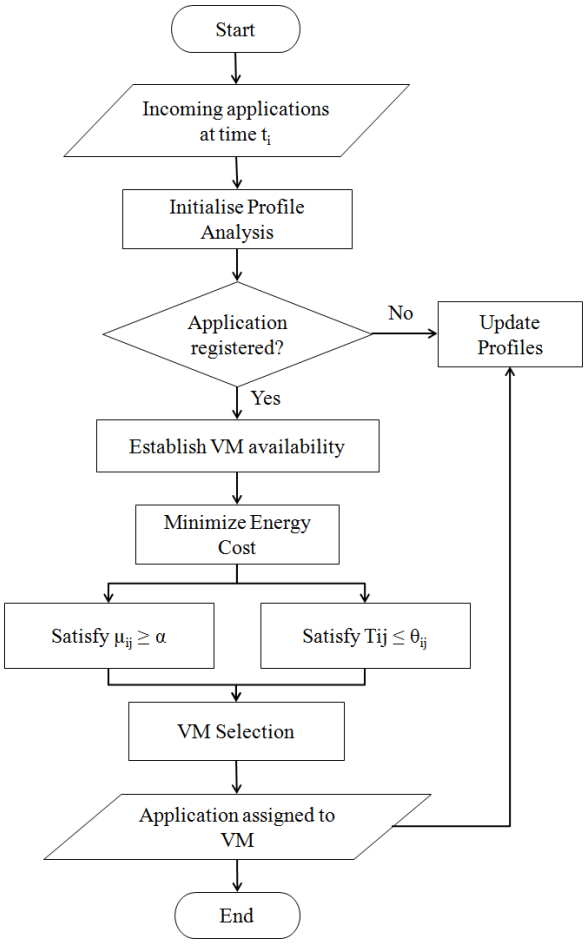

Fig. 2. Profile-based linear programming model.

IV. Profile-based Matching Algorithm

To solve the profile-based linear optimization problem formulated in Equation (7), this section presents a Profile-based Matching Algorithm. The algorithm makes use of some heuristics to simplify the problem-solving process. The motivation behind developing heuristics rather than employing conventional solution techniques is to allow for scalability and side constraints imposed on our assignment problem. The main objective of the Profile-based Matching Algorithm is to solve the energy optimization problem whilst satisfying constraints of CPU utilization efficiency and application completion time as discussed in the previous section. The algorithm also aims to obtain near-optimal allocation solutions which scale well in preference to the Hungarian Algorithm, which obtains optimal solutions but does not scale well [22].

Our profile-based matching algorithm is shown in Algorithm 2. The initial and most crucial element of the algorithm is the deciphering of the Profiles. Once the necessary data have been retrieved, the matrix $[C_{ij}]_{N\times M}$ is built depending solely on the minimum energy cost of assignment. In theory, the energy cost of allocation updates periodically so that real-time events could be taken into consideration in order to improve the efficiency of the allocation manager.

The algorithm then verifies the assignment by applying the constraints such the maximum CPU utilization efficiency and an acceptable application completion time are achieved. For the allocation to take place, the CPU availability of virtual machines, $\mu_j, j \in J$, must not be less than the CPU requirement of applications, $\mu_i, i \in I$.

If the conditions are satisfied, the algorithm moves on to the next assignment. In case of assignment unsuitability, the next best assignment is considered and the same process follows until a suitable assignment is achieved and the matrix $[C_{ij}]_{N\times M}$ is modified accordingly.

Algorithm 2: Profile-based matching

```
Read energy cost [C_ij]_{N×M} data from profiles;
Read utilisation efficiency and application completion
  time data from profiles;
Set scope to number of applications to be allocated;
while Within scope do
    Initialise [x_ij]_{N×M} and [Temp[i][j]]_{N×M} as null
      matrices;
    Copy matrix C_ij to a temporary matrix Temp[i][j];
    for i ← 1 to N do
        Set Temp[i][1] as the minimum value;
        for j ← 1 to M do
            if Temp[i][j] is minimum then
                Update Temp[i][j] as the minimum value
        Subtract minimum value from each value
    for each value in the matrix Temp do
        if Zero then
            Check utilisation efficiency constraint;
            Check application completion time constraint;
            if Constraints are satisfied then
                Confirm allocation as x_ij = 1;
                break;
            else
                Set value Temp[i][j] to a large number;
                goto step 7
    Output final allocation matrix x_ij
```

V. Case Studies

This section summarizes our case studies to demonstrate the profile-based linear programming modelling (Section III) and profile-based matching algorithm (Section IV).

The mathematical optimization of our profile-based prediction model involves identifying the best possible solution from a set of feasible solutions whilst considering the minimization of energy cost of assignment as the objective function. We use C++ to execute and test our profile-based matching algorithm.

The Hungarian Algorithm provides a high quality of solution in spite of the high solution time and poor scalability [23]. Therefore, we compare and observe the proximity of our algorithm efficiency to that of the optimal solution provided by the Hungarian Algorithm. The factors comprising the efficiency criteria include energy cost, CPU utilization and application completion time.

A. Case Study One - Feasibility and Scalability

Our test setup includes a data center with 100 physical nodes consisting of 4 virtual machines each. The applications and their corresponding profiles are randomly generated for test purposes.

Using our profile-based assignment model leads to the results shown in Figure 3. The average number of applications running in a single node amounts to 7. Groups 15, 75 and 80 have null values, signifying switched off servers. The resulting application assignment satisfies our objective of minimized energy cost and constraints of CPU utilization efficiency and application completion time, thereby supporting the feasibility of our Profile-based Assignment Model.

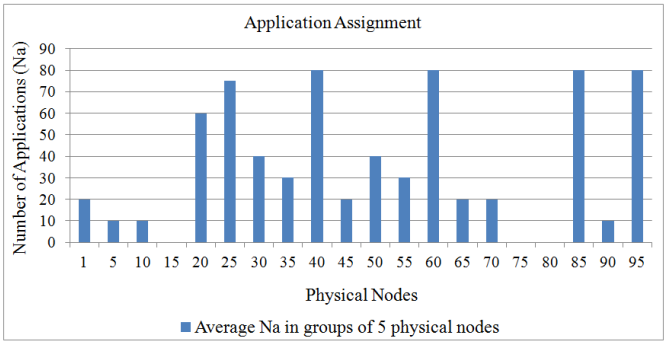

Fig. 3. Application Assignment

In order to test the scalability of our algorithm, we augment the number of virtual machines to 400, 800, 1,200 and 1,600, respectively. The number of applications ranges from 1,000 to 4,000. Our Profile-based Matching Algorithm (PMA) and the Hungarian Algorithm (HA) are each applied to the problem to obtain the execution time in seconds. The results are tabulated in Table I. In terms of the solution time criterion, it is seen from Table I that the PMA presented in this paper is capable of finding a near optimal solution in much less time as compared to the HA which consumes much more time in finding the optimal solution to the problem. Table I also shows that there is a high escalation in solution time of the HA when the numbers of nodes and applications $(M \times N)$ increase. In contrast, the solution time increase from the PMA is remarkably steady.

TABLE I
SOLUTION TIME FOR PROFILE-BASED MATCHING ALGORITHM AND HUNGARIAN ALGORITHM

Nodes	VMs	Applications	PMA Time (s)	HA Time (s)
25	100	500	3	5
100	400	1000	5	9
200	800	1500	11	21
300	1200	2000	17	83
400	1600	2500	33	108

B. Case Study Two - CPU Utilization Efficiency

The proposed Profile-based linear programming assignment model makes the best possible use of the available resources of the physical nodes. The scenarios considered in this case study are depicted in Table II. The physical nodes are classified into two groups: Node Groups 1 and 2. Node Group 2 comprises of the servers with higher processing speed and memory when compared to servers in Node Group 1. The CPU utilization of both groups are monitored for a period of 24 hours to obtain Figure 4. It is seen from Figure 4 that our assignment algorithm makes the maximum utilization of Group 2 throughout the day when compared to Group 1. This implies that the assignment of applications to high processing nodes are preferred in our Profile-based Matching Algorithm.

Table III compares the average CPU utilization efficiency of our profile-based matching algorithm and the Hungarian

TABLE II
CASE STUDY TWO SCENARIOS

Scenario	1	2	3	4	5
VMs	400	400	800	800	1000
Applications	500	1500	2000	2500	4000

TABLE III
AVERAGE CPU UTILIZATION EFFICIENCY (%).

Scenario	1	2	3	4	5
PMA	34.33	60.26	61.05	63.12	62.98
HA	41.05	73.42	70.89	68.71	66.59

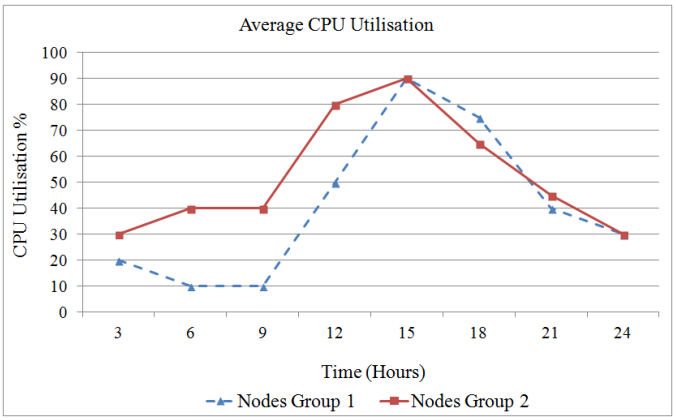

Fig. 4. Average CPU utilisation versus time.

Algorithm (HA) according to the different Scenarios shown in Table II. It is observed from Table III that in terms of the average CPU utilization, the HA is more efficient than the PMA by approximately 6.64%. But this higher efficiency is achieved through significant sacrifice of scalability in obtaining the solution to the assignment problem. It can be observed from the behavior of our algorithm that the average CPU utilization efficiency of the our PMA algorithm is consistent across the virtual machines with the increase in the scale of the assignment problems.

C. Case Study Three - Energy

The average CPU utilization is deduced after the assignment at different time intervals for both the PMA and HA. Energy consumption is then derived from CPU utilization data from the following equation [21]:

$$E(P_k) = \frac{(P_k^{max} - P_k^{idle}) * w_k}{100} + P_k^{idle}, \tag{8}$$

where P_k^{max} and P_k^{idle} have been previously defined as the the power consumptions at the maximum utilization and idle state, respectively. In our case studies, it is assumed that:

$$P_k^{max} = 350 \text{ Watts}, \; P_k^{idle} = 200 \text{ Watts}. \tag{9}$$

Figure 5 shows the total energy consumption of the servers in a 24 hour period for both the PMA and HA. The average energy consumptions for PMA and HA are 260.33 Watts and 245.83 Watts, respectively. This suggests that our Profile-based Matching algorithm is 2.86% closer in efficiency to the Hungarian algorithm. Figure 5 also presents a slight anomaly recorded for the HA towards the end of the 24 hour period, where the energy consumption deviates from the expected behavior. This phenomenon will be investigated in the future.

D. Summary of Case Studies

To summarize all case studies conducted above, it has been shown that

- The PMA is feasible and scalable within the tested range of 100 to 1600 virtual machine nodes;

- The PMA has a steady solution time albeit compromising the CPU utilization efficiency for increasing problem sizes; and
- The energy conservation achieved is close to that of the optimal Hungarian algorithm solution.

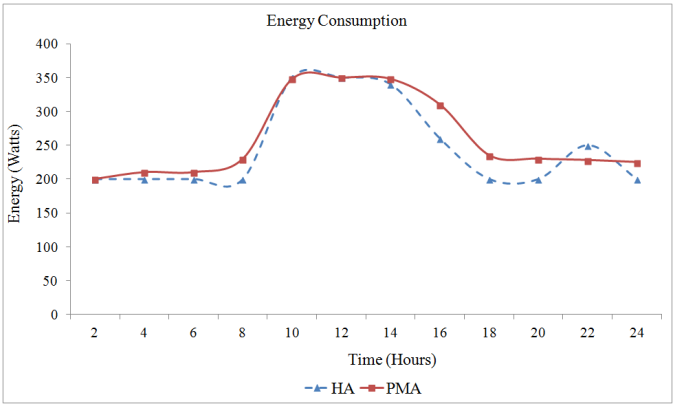

Fig. 5. Energy consumption.

VI. Conclusion

One of the critical issues that data centers are facing is how to minimize the energy consumption whilst maintaining high performance efficiency. This paper has presented an idea of utilizing the concept of Profiles for the assignment of applications to virtual machines. A profile-based assignment model has been established, and is solved using a profile-based matching algorithm. The proposed approach allows for the cheapest or a near-cheapest energy cost assignment whilst satisfying the constraints of CPU utilization efficiency and application completion time. Its feasibility and scalability have been demonstrated through preliminary case studies.

With the profile-based technique proposed in this paper, our future work will incorporate a profile-based predictive optimization approach, which will not only improve the efficiency of application allocation but will also allow deployment of profile-based modelling in real-world environments. Furthermore, larger workload, more realistic application arrival patterns, more comprehensive case studies, and comparisons with benchmarks and other assignment algorithms will also be considered in our future work.

References

[1] S. Rafiei and A. Bakhshai, "A review on energy efficiency optimization in smart grid," in *Proceedings of the IEEE 38th Annual Conference on Industrial Electronics Society*, Montreal, Quebec, Canada, 25-28 Oct 2012, pp. 5916–5919.

[2] K. Vaid, "Invited talk: Datacenter power efficiency: Separating fact from fiction," in *Workshop on Power Aware Computing and Systems*, Vancouver, British Columbia, Canada, 3 Oct 2010.

[3] K. Le, R. Bianchini, M. Martonosi, and T. Nguyen, "Cost- and energy-aware load distribution across data centers," in *Proceedings of HotPower*, Montana, USA, 10 Oct 2009, pp. 1–5.

[4] M. Webb, "Smart 2020: Enabling the low carbon economy in the information age," The Climate Group and the Global e-Sustainability Initiative, London, UK, Tech. Rep., 2008.

[5] J. Koomey, "Growth in data center electricity use 2005 to 2010," Analytics Press, Oakland, California, USA, Tech. Rep., 1 Aug 2011.

[6] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 68–73, Dec 2008.

[7] D. Votaw and A. Orden, "The personnel assignment problem," in *Proceedings of the Linear Inequalities and Programming Symposium*, Washington, DC, USA, 14-16 Jun 1952, pp. 155–163.

[8] R. E. Machol, "An application of the assignment problem," *Operations Research*, vol. 18, no. 4, pp. 745–746, Jul-Aug 1970.

[9] T. A. Ewashko and R. C. Dudding, "Application of kuhn's hungarian assignment algorithm to posting servicemen," *Operations Research*, vol. 19, no. 4, p. 991, Jul-Aug 1971.

[10] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, March 1955.

[11] G. Caron, P. Hansen, and B. Jaumard, "The assignment problem with seniority and job priority constraints," *Operations Research*, vol. 47, no. 3, pp. 449–453, May-Jun 1999.

[12] A. Campbell, A. S. Wu, and R. Shumaker, "Multi-agent task allocation: learning when to say no," in *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, Atlanta, Georgia, USA, 12-16 Jul 2008, pp. 201–208.

[13] L.-F. Leung, C.-Y. Tsui, and W.-H. Ki, "Simultaneous task allocation, scheduling and voltage assignment for multiple-processors-core systems using mixed integer nonlinear programming," in *Proceedings of the International Symposium on Circuits and Systems*, vol. 5, Bangkok, Thailand, 25-28 May 2003, pp. 309–312.

[14] Y. Xu and W. Qu, "A trust model-based task scheduling algorithm for data-intensive application," in *Proceedings of the 6th Annual ChinaGrid Conference*, Liaoning, China, 22-23 Aug 2011, pp. 227–233.

[15] A. Sheikh and S. Khan, "Integer programming approach for optimal resource allocation in workflow automation design," in *Proceedings of the IEEE 9th International Multitopic Conference*, Karachi, Pakistan, 24-25 Dec 2005, pp. 1–5.

[16] A. V. Do, J. Chen, C. Wang, Y. C. Lee, A. Zomaya, and B. B. Zhou, "Profiling applications for virtual machine placement in clouds," in *Proceedings of the IEEE 4th International Conference on Cloud Computing*, Washington, DC, USA, 4-9 Jul 2011, pp. 660–667.

[17] J. Baxter and J. Patel, "Profiling based task migration," in *Proceedings of the 6th International Parallel Processing Symposium*, California, USA, March 1992, pp. 192–195.

[18] X. Shi, H. Jiang, L. He, H. Jin, C. Wang, B. Yu, and F. Wang, "Eapac: An enhanced application placement framework for data centers," in *Proceedings of the IEEE 14th International Conference on Computational Science and Engineering*, Dalian, China, 24-26 Aug 2011, pp. 34–43.

[19] 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*, Alberta, Canada, 8-12 May 2007, pp. 331–340.

[20] M. Korupolu, A. Singh, and B. Bamba, "Coupled placement in modern data centers," in *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing*, Rome, Italy, 25-29 May 2009, pp. 1–12.

[21] M. Blackburn, *Five ways to reduce data center power consumption (white paper)*. The Green Grid, 2 Apr 2008.

[22] J. A. Winter and D. H. Albonesi, "The scalability of scheduling algorithms for unpredictably heterogeneous CMP architectures," in *Proceedings of the 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, Anchorage, Alaska, USA, 24-27 Jun 2008, pp. 42–51.

[23] Y. Chaobo and Z. Qianchuan, "Advances in assignment problem and comparison of algorithms," in *Proceedings of the 27th Chinese Control Conference*, Kunming, Yunnan, China, 16-18 Jul 2008, pp. 607–611.