A Hybrid Algorithm Using Genetic Algorithm – Hadoop MapReduce Optimization for Energy Efficiency in Cloud Computing Platform

Iza'in Nurfateha Ruzan[1], Suriayati Chuprat[2], Pegah Razmara[3]

[1]Faculty of Computing, Universiti Teknologi Malaysia,
Johor Bahru, Malaysia

[2]Advanced Informatics School (UTM AIS), Universiti Teknologi Malaysia,
Kuala Lumpur, Malaysia

[3]Faculty of Computing, Universiti Teknologi Malaysia,
Johor Bahru, Malaysia

Abstract: ***Cloud computing is an emerging model for distributed utility computing. It has become commercially attractive and continues to grow as it promises minimum maintenance and costs in comparison with traditional data centers. Clouds are normally composed by large and power-consuming data centers as it was designed to support the elasticity and scalability required by its customers. However, while cloud computing reduces the energy consumption at the customer site, it become an issue for the providers who have to deal with increasing demand and performance expectations. The study of this research paper is to formulate a hybrid algorithm using genetic algorithm and Hadoop MapReduce framework to further promotes the energy efficiency in cloud computing platform.***

Keywords: Cloud computing, energy efficient, genetic algorithm, mapreduce.

1. Introduction

The disclosure of cloud computing has been much predictable by a world with an ever increasing computing evolution of on-demand information technology service and product. The term "cloud" has been predicted by researchers to a large expansion based on virtualized resources, also an analogy for the internet, that is based on business model infrastructure, whereby the user can access anytime at anywhere, wherever and whenever they wanted to, because cloud computing offers service in real-time from the internet. The creation of cloud has been a help for both hosting companies and consumers can have access to the internet, whereas cloud users can also access, store and share any amount of information online.

The use of cloud computing is expanding and for the system to be effectively utilized, massive storage facilities are required and consequently enormous amount of energy consumption is consumed to operate and support the system. All contents that are stored and shared in the data centres, which includes videos, photographs and texts, must be accessible to the users in real-time. To address climate change, there is a necessity to move away from generating a large amount of electrical power using coal-fired power stations, especially which is needed to run cloud infrastructures such as data centers, backup power and cooling equipment. To move towards low carbon footprint, Information and Communication Technologies (ICT) equipment and cloud computing infrastructure should be powered by clean renewable energy.

Cloud computing is the new era or evolutions that conveys the use of information technology services and resources that are provided on a service basis. The computing power widely being used in internet is referring to the cloud itself and it is rapidly transforming into the largest virtual computer in the world. There are dozens of different definitions for Cloud Computing and there seems to be no consensus on what a Cloud is. The National Institute of Standards and Technology (NIST) [1] define of cloud computing:

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

NIST and also the other researcher [2][3][4][5]. Categorize cloud computing into 3 elements of service offerings, Figure 1 shows the cloud computing architecture.

- PaaS (Platform as a service)
- Iaas (Infrastructure as a service)
- SaaS (Software as a service)

Licensed Under Creative Commons Attribution CC BY

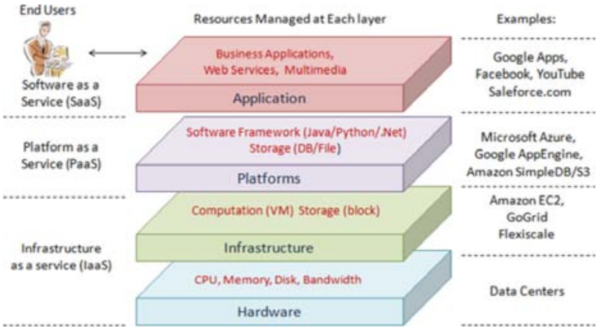
End Users
Resources Managed at Each layer
Examples:
Business Applications, Web Services, Multimedia
Application
Google Apps, Facebook, YouTube Saleforce.com
Software as a Service (SaaS)
Software Framework (Java/Python/.Net) Storage (DB/File)
Platforms
Platform as a Service (PaaS)
Microsoft Azure, Google AppEngine, Amazon SimpleDB/S3
Computation (VM) Storage (block)
Infrastructure
Amazon EC2, GoGrid Flexiscale
Infrastructure as a service (IaaS)
CPU, Memory, Disk, Bandwidth
Hardware
Data Centers

Figure 1: Cloud Computing Architecture [5]

The growth in the utilization of cloud computing has led to the production and increase in the carbon footprint, which consequently contributes to the world's climate change [6]. Although cloud computing has rapidly emerged as a widely accepted computing paradigm, research on cloud computing is still in its early stages, in which the related issues are as follows:

- Security: to make end users feel comfortable with "cloud" solution that holds their data, software and processes, there should exist assurances that service are highly reliable and available, as well as secure and safe, and that privacy is protected [7][8][9][10]. Software frameworks: handling varying and unpredictable loads and offering a highly available and reliable service in the face of hardware and software failures and evolution. These problems lead to the familiar challenges of constructing secure, reliable and efficient software [5][10][11].

- Quality of service (QoS): capability of acquiring and releasing resource on-demand. This is the objective to satisfy its service level objectives (SLOs) while minimizing the operational cost [10][12][13].
- Standardization: the failure of comprehensive cloud computing standards to gain traction and lack of security standards addressing issues such as data privacy and encryption [9][10][14][15].
- Power consumption: the use of Information and Communications Technologies [16] equipment is expected to rapidly increase the power consumed by ICT equipment and it is recognized that the power consumption of ICT equipment should be one of key issues, to stop the global growth emissions when it is now on top of priority issues [10][17][18][19].

We study cloud computing in the aspect of energy efficient techniques, approaches and methods in the scheduling performance. Many existing issues have not been fully addressed, while new challenge keeps emerging from industry applications. The remainder of this paper is organized as follows: Section 2 introduces background of the problem. Section 3 discusses related work. Hadoop MapReduce environment overview in section 4 followed by energy efficient task scheduling model in section 5. Genetic algorithm based on energy efficient task scheduling model in section 6. Sections 7 discuss the experimental evaluation and discussion. Finally, section 8 the conclusion of this study.

2. General Background of the Problem

The use of the Information and Communication Technologies (ICT) steadily leads to the increase in electricity rates. The amounts of communications are rapidly growing, especially with the use of smart devices, peripherals, computers and data centers [20]. Driven by the increasing demand for data processing and storage to large number of equipment, data centers can consume large amount of energy and emit large amount of carbon [21]. There are two principle sources of energy consumption on data processing center, also known as data center, one of which comprises the information systems that include hardware, servers, data storage systems and network interface. The second source is associated with the cooling systems, which is essential to guarantee the proper performance of information system; the more the data centers is utilized, the more heat is generated and consequently, increases the cooling requirement.

The drastic growth of on-demand upon cloud computing networks increases the energy consumption in the data center. This has become a critical issue and major concern for both industry and society [20][22][23][24]. The system performance improved with more energy consumption from the hardware and this has caused the energy consumption to increase and indirectly gives negative pressure to the cloud environment. Hence infrastructure providers are under enormous pressure to reduce energy consumption. It was reported that cloud computing data centers utilized approximately 1.5% of the total energy consumption in 2006 and it could be double in 2011 [25], the latest finding has shown that the data center consumed about 1.3% of the worldwide electricity supply in 2011 [6] and by 2020 it was predict that 25% of the energy is used in ICT systems, while data centers will consume 18% of the total, and the rest will be consumed by personal devices, office computers and servers [26].

3. Related Work

We have highlighted in the section 2 that rapid growth of the demand has lead large scale data center use in large amount of electricity. There are many approaches have been proposed to address these critical issues. DPM technique called Dynamic Voltage and Frequency Scaling (DVFS) combined reduction of the voltage supply and clock frequency, this is the direct relation power consumption in CMOS. The main point of these techniques is to lower down the CPU performance ratio, when is not fully apply [27]. In this case the run time and power consume were controlled by switching in between frequency and voltage. [28] Has suggested integrated approach of rate monotonic scheduling admission control test, the partitioning heuristics and speed assignment algorithm. They claimed to have the first works for energy aware on static priority periodic, which to solve real-time tasks on multiprocessors with partitioned approach where it divide into offline and online partitioned. Thus, the order has been given from the task allocation on scheduler. Hybrid metric is introduced which came out as the best method to maintain the energy consumption. Followed up by

Licensed Under Creative Commons Attribution CC BY

[29][30][31][32] where the research in on the timing constraints on DVS enabled settings for various tasks or system model and offline or online scheduling algorithm. [33], research works on parallel bi-objective hybrid metaheuristic for energy aware scheduling, where they use hybrid genetic algorithm to minimize makespan and energy consumption. Their study used Fast Fourier Transformation tasks graph approach to evaluated, the result of bi-objective metaheuristic improved in energy savings.

[34], works is on green energy efficient using DVFS techniques where it claimed that their works focus on the job scheduler where it schedules the task to execute the job in cloud data center. Evaluation results show that the works manage to reduce the power consumption on running the job tasks hence power up the resource utilization. This works is present to make the priority job come first and this technique can lower down the energy consumption of a server when it is in an idle mode or in a light workload. [35], works is to propose a new approach called DVFS Multi-Objective Discrete Particle Swarm Optimization (DVFS-MODPSO) for workflow scheduling in appropriated cloud computing infrastructures. The hybrid PSO algorithm is used to optimize the scheduling performance, these two techniques contribute to optimize the makespan, cost and energy in various space.

Another capable method to cut down the energy utilization through data centers is to implement the virtualization technology. [36][37], works using dynamic reallocation techniques on VMs where it obtained to have saving the energy when the idle server is in turn off/on position, it also very useful here and now regarding on cloud data centers. [38], reported with the provider perspective, GreenCloud architecture [39] the purposed is through supporting optimization on VM migration and VM placement to reduce virtualized data center energy consumption while guarantee the performance from users perspective. Comparable work is given by [40] who proposed Green Open Cloud (GOC). Advance reservation is design to support facilities through GOC for the next generation cloud data center. The function of GOC is to combine the workload by compromise with users so that idle servers can be switch on longer. [41], attempts that researcher mostly find solutions to minimize energy consumption in cloud environments through virtualization. Through this technology by using live migration to optimize the utilization of the available resources it has charter to overcome power consumption through multiple VMs on a single physical host. Similar in [42], they have proposed a cloud virtualization as a potential approach to reduce global warming and energy consumption. Their approach is to utilize less number of servers instead of using multiple servers to offer better service for multiple devices to improve in cloud energy efficiency.

For the use of large topology network might be have some of the use of thermal optimization, another approach which is power and thermal management (PTM). [43], works describes how much the cold air temperature should be used in the usage of large end user asking for on cloud. In contrast the focus on optimization over multiple system resource. [44][45], work is in the field of thermal management explores efficient methods of extracting heat from the data center. In [46] works they have find other approach focus on scheduling workloads in data center and the result came from the servers generate and minimize the energy expended by the cooling infrastructure, and it is leads to lower down cooling cost, decrease response times temporary, increase compaction and improve operational efficiencies and increase the hardware reliability. [47], in this research it presents the first closed form investigation of optimal solution for load distribution in a server with full of machine rack that minimizes the amount of computing and cooling energy. [48], works are focusing on the techniques such as voltage scaling and pinning which usually used for reducing energy consumption for thermal management over VMs migrations. In [49] explore a spatio-temporal thermal aware job scheduling as an extension to spatial thermal aware solutions same with [50][51]. These works is to enhance the temperature sharing within data center, and the outcome is to have the energy efficiency and the reliability of a data center operation.

On machine learning aspect which is in the neural network [52] have investigated and applied neural network predictor for saving the energy in cloud. [53], works were introducing a novel framework where it combining load demand prediction and stochastic state transition models. From this works they claimed that their model will lead to optimal cloud resource allocation through minimizing energy consumed while maintaining required performance levels. Throughout this research they have obtained both model neural network and linear predictor anticipate adequately future network load, whereas the linear predictor bring the most correct results. [54], works shows that the study on present workload fairly across all the nodes using back propagation learning algorithm to train feed forward Artificial Neural Network (ANN). These techniques can works accurately when active training set are used, where it can predict the demand where it allocates resources following that demand. From these works it can maintain the active servers in low energy consumption.

Task scheduling is an important part of cloud computing, which is a mechanism that maps users tasks to appropriate resources to execute, it is efficiency will directly affect the performance of the whole cloud computing environment [55]. [56], proposed green task scheduling, the intention of this works is to implement six innovative green task scheduling which it have two main aspect: assigning as many task as possible to a cloud server with the lowest energy, and setting the same optimal speed for all task assigned for all each cloud server. [57], proposed the task scheduling strategy for tree network of cloud computing environment. The process of this method is considered about the feature of tree network, this framework used master-slave model as a basic. In master-slave model, all the tasks are submitted on master node, then master node effectively and transparently partition and allocate tasks to each slave node. All tasks were completed through a number of parallel slaves. For example, MapReduce is one of the method represent the data processing frameworks based on master-slave modes, which is used in Hadoop [58][59][60]. [61], works stated that the power consumption by computing resources and storage in cloud can be optimized through energy aware resource

Licensed Under Creative Commons Attribution CC BY

allocation. They proposed energy efficient heuristic algorithm techniques and compared it with three energy-aware task consolidation heuristics through shifting number of tasks.

Load balancing can help in reducing energy consumption evenly through distributing the load and minimizing the resource consumption. Proper load balancing can help utilizing the available resources optimally; therefore minimize the resource consumption [23]. [62], works is on private cloud for finding the good solution of energy consumption. These works has proposed hybrid energy efficient scheduling approach to works on the experiment. The experiments results appear to save more time for users maintaining more energy and attain higher level of load balancing. [63], in their works they have proposed a scheduling algorithm which predicts electricity price peaks and controlling energy consumption through pausing mode of virtual machines. This works has been using open cloud manager OpenStack [64] approach to evaluate and show the reductions cost and energy consumption. [65], this research is aiming on a solution for online real time services using non-preemptive scheduling algorithm in order to minimize the process time of the migrated tasks. To overcome of increasing processing time of migrated tasks, a non-preemptive real time scheduling using check pointing algorithm is proposed which to minimize the migrated tasks also the penalty will be better with earlier completion of migrated tasks.

Genetic algorithm (GA) is an optimization and search technique based on the principles of genetics and natural selection. A GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the fitness (i.e., minimizes the cost function). Often technique commonly used in the scheduling strategy is genetic algorithm. This method perform scheduling problem in the process of deciding which jobs to start execute and in which resources; by given a set of jobs and requirements, resources, and the system status [66]. [67], they come out the Improve Genetic Algorithm (IGA) whereas the goal is to have good consideration utilization. [68], this works is based on improving the performance of cloud computing with the deadline constraint, a task scheduling model is established for reducing the power consumption system of cloud computing also improving the profit of service providers. In this research they proposed a model solving method called multi-objective genetic algorithm (MO-GA), where it including the encoding rules, crossover, selection operators also a method of sorting Pareto solutions. [69], from this works it shows that they have using the cloud booster algorithm to measure the virtual machines allocation where it based on nodes weight which it indicates a value capacity of each nodes. They proposed the virtual machines allocation which considering the node weight and future prediction. The outcome of this research they produced adaptive genetic algorithm based on virtual machines resource migration strategy which focusing on load balancing systems and have resulted to have overload avoidance and energy saving for systems with multiple resource constraints. [70], this works is based on improving the energy efficiency of servers in data center through suitable task scheduling strategies. In this research they have proposed a new energy efficiency task scheduling model to solve with a practical encoding and decoding specifically designed with genetic algorithm operator techniques. From these techniques they improve the searching capability a local search operator is introduced. The outcome of this studied is shows that the proposed algorithm is effective and efficient.

4. Hadoop MapReduce Environment Overview

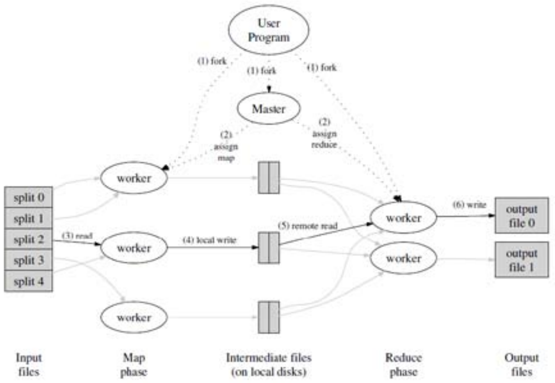

Figure 2: Execution Overview [58]

To deal with a very large amount of data which it needs to be done very quickly and fairly process in a day, Google's introduced new infrastructure where it use a very large amount a thousand more of machine in the data centre to have a good process between the end users. Because of this demand Google came in with the Google distributed files systems (GFS) and MapReduce computational model framework which its include the implementation of Hadoop from Apache whereas it deal with those aspect. Hadoop MapReduce popular with the works of script application where it focusing with large amount of data to be process, and because of this it is called as a software framework in parallel on enormous clusters and produce stable hardware and fault-tolerant aspect. A MapReduce usually works with partitioning input data into a set of Map (*M*) task splits in a parallel aspect. Then the framework allocates the output of map task, which it will become the input of Reduce (*R*) task. The number of partitions (*R*) and the partitioning function are specified by the user. Commonly both input and output jobs are stacked into file systems. The framework is responsible to manage the scheduling, observing and re-executes if the task failed in the systems.

When the user program calls the MapReduce function, the following sequence of actions occurs as shows in Figure 2. Fork is responsible to splits the input files into Map (*M*) pieces of typically 16 to 64MB per piece. It then starts up many copies of the program on a cluster machines. The special copies of program called as master which it assign the work into two and namely the Map (*M*) and Reduce (*R*) tasks. The scheduler works to schedule the map task close to physical storage locations on set of input data. Once the program is starting to split the worker who is assigned a map task reads the content of the corresponding input splits. Then it analyze key/value pairs out the input data and pass each pair to the user defines map function. The intermediate

Licensed Under Creative Commons Attribution CC BY

key/value pairs produced by the map function are buffered in memory.

From the intermediate files on local disk, the task will partition into Reduce (*R*) regions. When reduce worker is notified by the master about these locations, it use remote procedure to calls and read the buffered data from the local disks of map workers. After reduce workers has read all intermediate data for its partition, it will sort all occurrences of the same key are grouped together. The reduce worker iterates over the sorted intermediate data and for each unique intermediate key encountered, it analyze the key and the corresponding set of tasks value to the user reduce function. From this process the output of the program will have at the reduce function as the final output of the program. When all of the process has been completed, the master will wake up the user program, to returns back the user code [58].

5. Energy Efficient Task Scheduling Model

5.1 Mathematical Description

Firstly we give general mathematical description of the energy efficient task scheduling based on MapReduce. Using notation similar by [70] we have: *N* servers in a data center, denoted by $M = \{m_1, m_2, \ldots, m_N\}$. CS_k is CPU utilization. CO_k is optimal point. *D* represent as input data, *m* represent as splits with each size of 64M , denoted by $D = \{d_1, d_2, \ldots, d_m\}$.

Initialization step from above notation we randomly store these: *m* splits on N servers, use $m\times3$ matrix *P* to represent the storage location of each split element p_{ij} indicates the storage location of split d_i, where integer $p_{ij} \in \{1, 2, \ldots, N\}$, in which $_i = 1, 2, \ldots, m$ and $_j = 1, 2, 3$. Input data *D* will be processed by *m* map tasks and *r* reduce tasks, denoted by $T = \{t_1, t_2, \ldots, t_m, \ldots, t_{m+r}\}$. CPU required for every map task is *CM*, and for every reduce task is *CR*. From the initialization above it then become a problem of how to allow these $m+r$ tasks into the N servers, so that it will have the all servers reaches the highest of energy efficient.

5.2 Optimization Model

Using similar notation from [70], we have: Vector $S = \{s_1, s_2, \ldots, s_{m+r}\}$ represent thefinal task scheduling scheme, the i^{th} element indicates that task t_i is assigned on server m_{si}, where $1 \le s_i \le N$ and $i = 1, 2, \ldots, m + r$. Following with server m_k, denoted as M_k and R_k respectively, where $k = 1, 2, \ldots, N$. Then it will become $NM_k= |M_k|$ and $NR_k= |R_k|$. From all the above notation and implementation, single objective of optimization in energy efficiency task scheduling with MapReduce model as shown in this equation (1) bellow:

$$\sum_{k=1}^{N}\Big(COk - (CSk + NMk * CM + NRk * CR)\Big)\hat{}2 \qquad (1)$$

Subject to:

1) $$S \begin{cases} s_i \in \{p_{i1},p_{i2},p_{i3}\} & \text{if } i=1,\ldots,m \\ s_j \in \{1,2,\ldots,N\} & \text{if } j=m+1,\ldots,m+r \end{cases}$$

2) $NM_k = |\{s_i \mid s_i = k, \quad i =1,2,\ldots,m\}|, \qquad$ where $k = 1,2,\ldots,N$

3) $NR_k = |\{s_i \mid s_i = k, \ i =m+1,\ldots,m+r\}|, \qquad$ where $k = 1,2,\ldots,N$

4) $CS_k + NM_k * CM + NR_k * CR \le 1, \qquad$ where $k = 1,2,\ldots,N$

Equation (1) explain single objective optimization model for the energy efficient task scheduling problem based on MapReduce for cloud computing, the objective function indicates the minimum sum of the difference between servers optimal points with its CPU utilization after scheduling. Constraints (1) tell us that if a map task t_i (where $i = 1, 2, \ldots, m$) is assigned on server m_{si} , then server m_{si} must have stored its corresponding input data. This is because the MapReduce prefers moving the executive program to the node which stores the data when it deal with large amount of data to process, rather than moving the data as in traditional distributed computing. This scheduling scheme based on data location can avoid a large scale data movement, which not only reduces the network overhead, but also makes the map tasks locally read and process the data. Constrains (2) and (3) compute the number of map tasks NM_k and reduce tasks NR_k which are assigned to server m_k. Constrains (4) indicates that the CPU utilization of any server should not exceed 100% before and after the task scheduling where it to improve the energy efficiency of servers cannot be solved as easy load balancing among them to make all servers CPU utilization reach 100%. Considering the inter-relationship between energy consumption, resource utilization and performance of workload we know that task has been changed by the CPU utilization of servers [70][71].

6. Genetic Algorithm based on Energy Efficient Task Scheduling Model

6.1 Encoding and decoding operation

Encoding operator is one of the compelling aspects in the genetic algorithm. Encoding principles is to focusing on gene and chromosome. Based on the nature of this energy efficient multi-job scheduling problem, we use the integer coding. We use vector $S = \{s_1, s_2, \ldots, s_{m+r}\}$ as an individual to produce a scheduling pattern, were to implement $m + r$ tasks that need to be processed, where the i^{th} component indicates task t_i is assigned on server m_{si} . In here the advantage of this approach is that we can simply apply it into simple multi point crossover operator for the evolution of individuals. To load the population, we give the following steps for encoding each individual.

```
for $i = 1, 2, \bullet\ , m;$ select random integer $j \in \{1, 2, 3\};$
        let $s_i = p_{ij}.$
for $i = m + 1, m + 2, \bullet\ , m + r;$ select a random integer
$k \in \{1,\ 2, \bullet., N\};$
set $s_i = k$
```

After encoding individuals we need to perform decoding operator, this method is to compute the individual fitness value in the encoding previously. We need to compose the integer value first. The composition of method is as follows:

```
for $k = 1, 2, \bullet , N;$ empty sets $M_k$ and $R_k.$
  for $i = 1, 2, \bullet , m,;$ put $i$ into set $M_{si}$, in which $si$ is the $i^{th}$
```

Licensed Under Creative Commons Attribution CC BY

```
    element of individual $S$;
  for $i= m + 1, m + 2, • , m + r$, put $i$ into set $R_{si}$.
  Let $NM_k= |M_k|$ and $NR_k= |R_k|$
```

6.2 Population operation

The population is called when a group of chromosome starts from genetic algorithm. Population contains with chromosome represent as N*pop* and N*bits* matric denoted N*pop* × N*bits* randomly generate using pop = round(rand(N*pop*, N*bits*)); where the function is to generates a N*pop* × N*bits* matrix of random order numbers between zero and one. Pop matrix in every row is a chromosome, which it is tally with various values of longitude and latitude. The pseudo code of population as follows: Initialize population name, *A*; population value, *var* number, number of servers and generation cycle value, *t*.

```
A = round ((rand (pop,var) * N);
gen = 0;
while
gen <=t;
t, represent the iteration in GA
```

6.3 Selection and mating operation

The selection part it is like match making the selected of two chromosome from the mating pool of N*keep* chromosome to generate two new offspring. In this part we were taking pairing into priority whereas it place in the mating population denoted N*pop* – N*keep* new offspring are create to change the dispose chromosomes. Selection have two method to represent with first pairing from top to bottom secondly random pairing, in this experiments we choose random pairing of selection. This method uniform random number to generate selected chromosomes represent as ma = ceil (N*keep* + (N*pop* – N*keep*) . * rand (1, N*keep*)) and pa = ceil (N*keep* + (N*pop* – N*keep*) . * rand (1, N*keep*)), ceil represent to cycle or rounds the numbers to the next highest integer. From this selection it will obtain in a dissimilar set of parents. Such as, the combining of the next generation is not the same for each of selection pattern. The selection pattern which is standard in genetic algorithm called as roulette wheel and tournament selection. The selection pressure ratio probability that fit chromosome is choosing to the probability that the average chromosome is chosen. The selected parents of the population will create more than one of new offspring which called as mating. Mating require two parents which generate two offspring, therefore the offspring of both parents hold binary codes. Thus, the parents need to generate a total of N*pop* – N*keep* offspring, so that the new chromosome population is back to N*pop* in the process of pairing method. The following show selection pseudo code:

```
ma = ceil (Nkeep + (Npop – Nkeep) . * rand (1, Nkeep));
pa = ceil (Nkeep + (Npop – Nkeep) . * rand (1, Nkeep));
```

6.4 Crossover operation

In crossover section we choose a random two point crossover operator for the evolution of individual where it is chosen for the parents. Thus, the parents swapped with the two point crossover of bits. Also to matches good parents sub-solutions to construct better offspring, to make individual meet the constraints of the model. The random selection of chromosome parts decides which group the bits are swapping. Initialization step of crossover we start to defines of population number, *last*; selection rate, *sel*; and number of chromosome kept, *M2*. After defining the parameters, the crossover function start to form defining the fitness value, *FV* from the single optimization algorithm energy efficient and the individual number of fitness value. The number values continue with the crossover chromosome, *ic*. Where the pop, *A* will do the process according the random number, *N* respectively. Thus, the new individual will obtain such as from A_3 and A_4 from individual A_1 and A_2. The following show crossover operation pseudo code:

```
last = 0;
sel = 0;
M2 = 2 * ceil (sel * population/2);
for ib=1:last
 [FV, ind] = sort ( FV);
 A = A (ind (1:M2) , : );
 [ib FV [1]]

for ic = 1:2:M2
 A (ceil (M2 * rand) , 1:cross) = A (ic , 1:cross);
 A (ceil (M2 * rand) , cross + 1:N) = A (ic + 1 ,
     cross +  1:N);
 A (ceil (M2 * rand) , 1:cross) = A (ic + 1 , 1:cross);
 A (ceil (M2 * rand) , cross + 1:N) = A (ic , cross + 1:N);
```

6.5 Mutation operation

Mutation process is about each gene independently works with mutation probability, p_m represent mutation rate. In this process the single-point mutation operator is use, to meet its individual constraints of this purpose approach, whereas it is come out with the new generation iteration. In here mutation number is randomly selected from total value in the population matrix, N*pop* × N*bits*. Mutation designed as *elite* solutions intend to generate without changes. Mutation process start with defining the parameter accordingly, next the value number of mutation represent as *nmuts*, obtain from the mutation rate and population also the selection from previous formula. The following show mutation operation pseudo code: Initialize mutation probability, *pc*; number of mutation, *nmuts*.

```
pc = 0;
nmuts = pc * N * (M -1);

for ic = 1:nmuts
 ix = ceil (M * rand);
 iy = ceil (N * rand);
pop (ix, iy) = 1-pop (ix, iy);
```

7. Experimental Evaluation and Discussion

7.1 Parameter values

Initializing the parameter value using the pseudo code aforementioned: population size, X=100; crossover probability (pc), pc=0.6; mutation probability (pm), pm=0.2; elitist number, k=5 and stop generation, t=2000. Assume the server is N=200 as in a data center, input data is 1250G and it splits into 20000, splits which means m=20000. Input data will splits and randomly the three servers set earlier will stored the data. Reduce tasks are required to implement the algorithm for completing the job tasks, we take 450 jobs as r=450. Based on observation 3 years servers in data center, different server may have different optimal point of performance of energy point for monitoring how long has it been used. In this three series of experiments we assumed the 1/3 first server used 0.9 optimal point, 1/3 middle server used 0.7 optimal point, while 1/3 end server used 0.5 optimal point. The analysis continues where we set the initial states of server as follows:

CS [1:5] = 0.5	CS[46:75] = 0.5	CS[116:145]= 0.5
CS[6:25] = 0.7	CS[76:95] =0.7	CS[146:165]=0.7
CS[26:45]= 0.9	CS[96:115]=0.9	CS[166:200]=0.9

7.2 Experiment 1

We implement this experiment under normal conditions situation to observe the performance of these three series algorithms. Consider the desired CPU for each map task and reduce task are CM = 0.0052 and CR = 0.002 designated. The experimental of algorithm under normal state were represent in Figure 3, (a) under normal state hybrid GA-MR (b) normal condition results hybrid GA-MR after scheduling (c) normal state task scheduling existing work and (d) Hadoop MapReduce scheduling result. Based on our observation in our proposed algorithm shows differences between existing work result when the selected server number 5, 75 and 145 with optimal point of 0.5, 0.7 and 0.9, the hybrid GA-MR algorithm show task is assign in server number 145 contrast with the previous work does not assign task on that point of server. For server number 25, 95 and 165 server with the same initial CPU utilization is 165 in proposed algorithm contrast with previous work; task is assign only on server 25. Continues with the server 45, 115 and 195 server 45 in hybrid GA-MR algorithm is assign task contrast with the previous work does not assign a task because it uses the same optimal point, 0.9 where we use the different of optimal point in the normal state condition. The assigning task or not situation happen because if initial CPU utilization have same number value with the optimal point of energy, certain point of server does not assign task on that point of server which it will reduce the energy consumption on that server. In different aspect we were comparing with the result show in Figure 3(a) and (d) where the flow of the task in this algorithm does not consider any of the optimal point of servers. Which it will make the situation of the task in Hadoop MapReduce algorithm will not consistent. The higher the optimal performance energy point , the more task it needs to deal with which in hybrid GA-MR we use the pure genetic algorithm rather than being modified as we can see in Figure 3(b) compare with the Figure 3(c). Furthermore, in term of finding optimal result in energy efficiency performance of server we have obtain 2.5107, while Hadoop MapReduce algorithm 6.4796 and existing work is 0.2404. Somewhat, the existing result is better performance of the total IT equipment power in PUE, which it could be consider that the existing proposed algorithm using modified genetic algorithm much more better performance energy contrast with our hybrid GA-MR algorithm. In this research we have taken the total sum value of Hadoop MapReduce as our benchmarks of the study, in which when we were compared it together somehow our hybrid GA-MR algorithm were optimal than the Hadoop MapReduce algorithm, hence it could improve the energy efficient performance of servers.

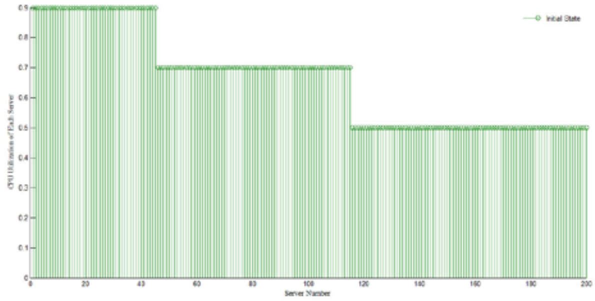

Figure 3: (a) Normal state Hybrid GA-MR algorithm

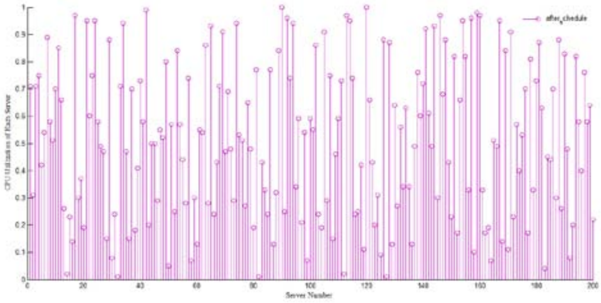

Figure 3: (b) Normal state Hybrid GA-MR algorithm after scheduling

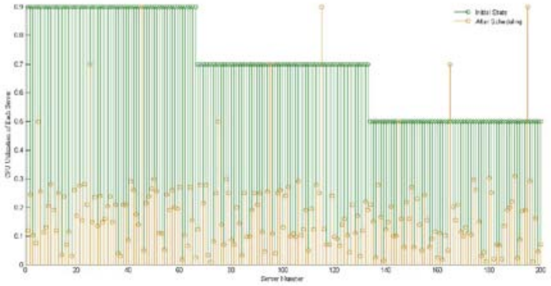

Figure 3: (c) Normal state Genetic Algorithm

Licensed Under Creative Commons Attribution CC BY

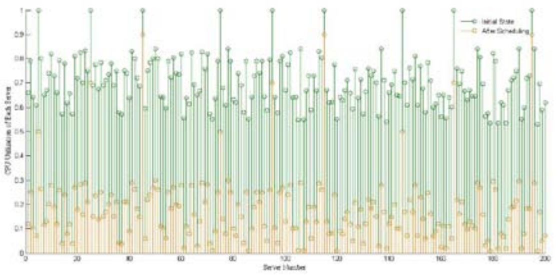

Figure 3: (d) Normal state Hadoop MapReduce scheduling

7.3 Experiment 2

We were handling the second experiment under low load condition of the worst case that is when systems run low load state to perform either the proposed algorithm works similar with the Hadoop MapReduce scheduling pattern. We were presuming the input data to process is small. Consider the desired CPU for each map task and reduce task are CM = 0.0025 and CR = 0.001 designated. The experimental of algorithm under low load state were representing in Figure 4, (a) low load condition of hybrid GA-MR after scheduling (b) low load state previous work and (c) Hadoop MapReduce scheduling. We begin the analysis with small value parameter task in Figure 4 (b) and (c), this Figure 4 (b) show that CPU utilization of energy generally distributes about 0.55 and 0.45 after scheduling, while Hadoop MapReduce which the optimal point were not taken consideration in this algorithm to perform a better performance. Figure 4 (c) has shown that the CPU utilization were distributes about 0.35 after scheduling. While our hybrid GA-MR algorithm perform well in this condition of low load, Figure 4 (a) shows that even if we have set the 1/3 first server is 0.9, 1/3 middle server 0.7, and 1/3 end server 0.5 as the optimal point of the certain point the initial CPU utilization will have a massive task running if it is not reach the optimal point of servers. Generally CPU utilization of energy point is distributes about 0.2. To the other aspect in term of the optimization value in energy efficiency point of server, we have obtained an optimal result compare with the others algorithm. Taken form the Hadoop MapReduce algorithm total sum value as a benchmarks in this research we have found that our hybrid GA-MR algorithm performs well in this low load condition which we have obtain the IT equipment power in PUE, 1.1342, while Hadoop MapReduce 22.3556 and 18.5475 by existing work of genetic algorithm.

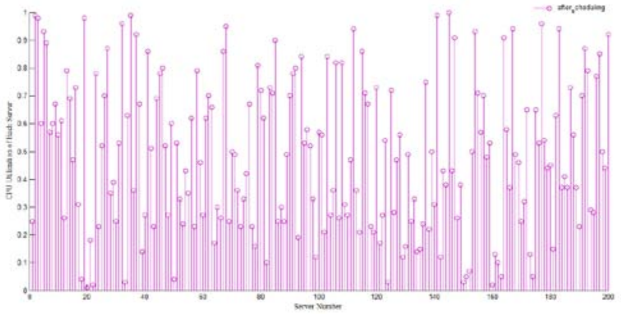

Figure 4: (a) Low load condition Hybrid GA-MR after scheduling

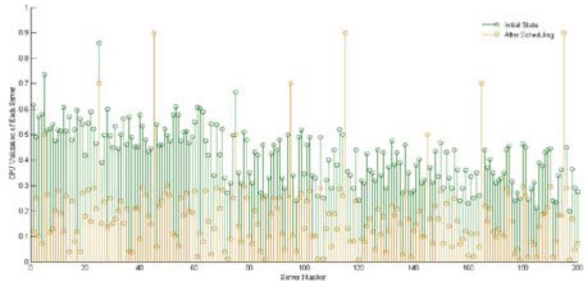

Figure 4: (b) Low load condition Genetic Algorithm

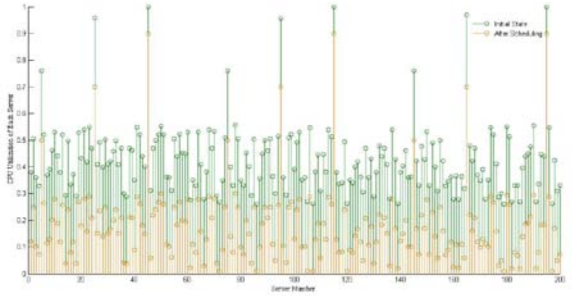

Figure 4: (c) Low load condition Hadoop MapReduce scheduling

7.4 Experiment 3

We implement the third experiment under heavy load condition of the worst case that is when systems run under heavy load state to perform either the proposed algorithm works similar with the Hadoop MapReduce scheduling pattern. We were presuming the input data to process is large. Consider the desired CPU for each map task and reduce task are CM = 0.007 and CR = 0.006 designated. The experimental of algorithm under heavy load state were representing in Figure 5, (a) heavy load condition of hybrid GA-MR after scheduling (b) heavy load state of existing work and (c) Hadoop MapReduce scheduling. Figure 5 (a) shows hybrid GA-MR scheduling algorithm under heavy load condition which we compare it with Figure 5 (b) and (c) under the same condition situation, based on our observation in the Figure 5 (a) we have found that the higher the optimal point of energy the more task it needs to deal with. Comparing with Figure 5 (b) in which it generally uses modification of optimal point value number to perform the experiment, while our hybrid GA-MR were basically used the same 1/3 first server 0.9, 1/3 middle server 0.7 and 1/3 end server 0.5. Contrast with the Hadoop MapReduce scheduling which they were not consider any value of the optimal point of server same in the experiment 1 and 2 previously. As we can see in the Figure 5 (a) the CPU utilization of performance energy of task in server distributes between 0.4 for the 1/3 first servers, 0.55 for the 1/3 middle server and 0.6 for the 1/3 end servers, while in Figure 5 (b) CPU utilization of performance energy of task in server distributes generally 1/3 first server is 1.0, 1/3 middle server is 0.95 and 1/3 end servers is 0.8. Contrast with the Hadoop MapReduce scheduling in Figure 5 (c) CPU utilization mostly distributes 0.8 and 1.0. We could see that the heavy load condition will hit the 1.0 of maximum CPU utilization. From other aspect the total sum for optimal point of energy in servers where we

Licensed Under Creative Commons Attribution CC BY

have measured for the hybrid GA-MR algorithm we obtain 13.1120, while 8.7965 by existing work genetic algorithm and 12.931 by Hadoop MapReduce scheduling. Somewhat, the existing result is better performance of the total IT equipment power in PUE, which it could be consider that the existing proposed algorithm using modified genetic algorithm much more better performance energy efficiency contrast with our hybrid GA-MR algorithm and Hadoop MapReduce scheduling algorithm.

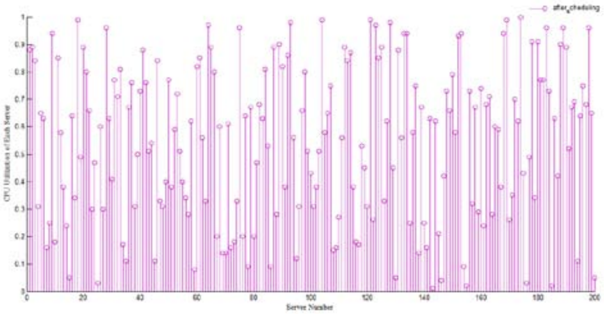

Figure 5: (a) Heavy load condition Hybrid GA-MR algorithm after scheduling

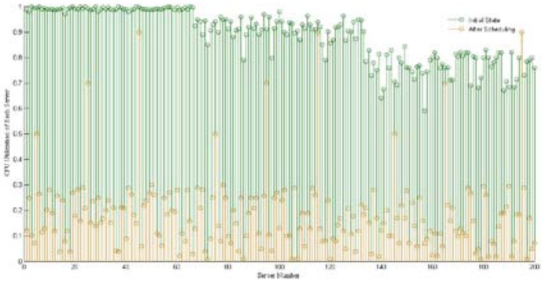

Figure 5: (b) Heavy load condition Genetic Algorithm

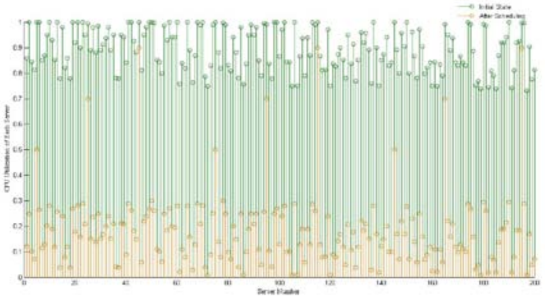

Figure 5: (c) Heavy load condition Hadoop MapReduce scheduling

8. Conclusion

After series of experiment under normal condition, low load condition and heavy load condition we have found that the performances of the energy act the utmost priority to perform energy efficient for assigning task. Which it is indirectly gives an impact to the equipment of the server if the need of power consumption is using to give the end users satisfactory using the internet. In this research we were focusing on how to have the minimum optimization value and indirectly can improve a bit the energy efficiency of servers through potential techniques. Finally, the optimization of energy efficiency is measure for having the value total sum of IT equipment power when the task is running on servers. We were using the Hadoop MapReduce as the benchmarks value for our proposed algorithm in which our algorithm is perform well in the low load condition than the existing work and Hadoop MapReduce, somehow it could improve a bit the energy efficiency on servers.

References

[1] M. Peter and T. Grance, "The NIST definition of cloud computing."*National Institute of Standards and Technology* 53.6, pp. 50, 2010.

[2] Y. Lamia, M. Butrico, and D. Da Silva, "Toward a unified ontology of cloud computing." *Grid Computing Environments Workshop, IEEE, GCE'08*, pp. 1-10, 2008.

[3] B. Rajkumar, A. Beloglazov, and J. Abawajy. "Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges." *arXiv preprint arXiv:1006.0308* (2010).

[4] Z. L. Mingyi, K. Li, and Y. Q. Zhang. "Green task scheduling algorithms with speeds optimization on heterogeneous cloud servers," *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*. IEEE Computer Society, 2010.

[5] Zhang, Q. L. Cheng, and R. Boutaba. "Cloud computing: state-of-the-art and research challenges." *Journal of internet services and applications* 1.1 (2010): 7-18.

[6] *Green, Make IT. "Cloud Computing and its Contribution to Climate Change."* Greenpeace International. 2010.

[7] C. Ivica and M. P. H. Larsson, *Building reliable component-based software systems*. 1st ed. Artech House, 2002.

[8] A. V. Mladen, "Cloud computing–issues, research and implementations," *CIT. Journal of Computing and Information Technology* 16.4 pp. 235-246, 2008.

[9] R. Bhadauria, R. Chaki, N. Chaki, and S. Sanyal, "Security Issues In Cloud Computing." *Acta Technica Corvininesis-Bulletin of Engineering* 7, no. 4, 2014.

[10] S. Y. Jing, S. Ali, K. She and Y. Zhong, "State-of-the-art research study for green cloud computing." *The Journal of Supercomputing* 65, no. 1 pp. 445-468, 2013.

[11] S. Bykov, A. Geller, G. Kliot, J. R. Larus, R. Pandya and J. Thelin, *Orleans: A framework for cloud computing*. Technical Report MSR-TR-2010-159, Microsoft Research, 2010.

[12] S. Ferretti, V. Ghini, F. Panzieri, M. Pellegrini and E. Turrni, "QoS-Aware Clouds," *IEEE 3rd International Conference on Cloud Computing*. pp.321-328, 2010.

[13] X. Fan, W. D. Weber and L. A. Barroso, "Power Provisioning for a Warehouse-sized Computer," *In Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA).* ACM New York. pp.13-23, 2007.

[14] G. Somani and S. Chaudhary, "Application performance isolation in virtualization," In *Cloud Computing, 2009. CLOUD'09. IEEE International Conference on*, pp. 41-48, September 2009.

Licensed Under Creative Commons Attribution CC BY

[15] S. Labes, J. Repschlager, R. Zarnekow, A. Stanik and O. Kao, "Standardization approaches within cloud computing: evaluation of infrastructure as a service architecture," *In Computer Science and Information Systems (FedCSIS), 2012 Federated Conference on*, pp. 923-930, September 2012.

[16] T. Cordeiro, D. Damalio, N. Pereira, P. Endo, A. Palhares, G. Gonçalves, D. Sadok et al., "Open source cloud computing platforms," *In Grid and Cooperative Computing (GCC), 2010 9th International Conference on*, pp. 366-371, November 2010.

[17] L. Benini, A. Bogliolo and G. D. Micheli, "A survey of design techniques for system-level dynamic power management," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 8, no. 3, pp. 299-316, 2000.

[18] R. Jejurikar, C. Pereira and R. Gupta, "Leakage aware dynamic voltage scaling for real-time embedded systems," In *Proceedings of the 41st annual Design Automation Conference*, pp. 275-280, June 2004.

[19] S. Kuribayashi, "Reducing total power consumption method in cloud computing environments," *International Journal of Computer Networks & Communications (IJCNC)* vol.4, no.2, March 2012.

[20] I. S. Moreno and J. Xu, "Energy-efficiency in cloud computing environments: towards energy savings without Performance degradation." *International Journal of Cloud Applications and Computing (IJCAC)* vol.1, no. 1, pp. 17-33, January 2011.

[21] S. K. Garg and R. Buyya, "Green cloud computing and environmental sustainability," *Harnessing Green IT: Principles and Practices,* 1st ed. John Wiley & Sons, Ltd. 2012, pp. 315-340.

[22] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee et al. "A view of cloud computing." *Communications of the ACM* vol. 53, no. 4, pp. 50-58, April 2010.

[23] N. J. Kansal and I. Chana. "Cloud load balancing techniques: A step towards green computing." *IJCSI International Journal of Computer Science* vol. 9, no. 1, pp. 238-246, January 2012.

[24] 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, March 2012.

[25] R. Brown, "Report to congress on server and data center energy efficiency: Public Law. 109-431," Lawrence Berkeley National Laboratory, pp. 1-137, June 2008.

[26] E. Gelenbe and R. Lent. "Optimising server energy consumption and response time," *Theoretical and Applied Informatics*, vol. 24, no. 4, pp. 257-270, 2012.

[27] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," *In Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, IEEE Computer Society, pp. 826-831, May 2010.

[28] T. A. AlEnawy and H. Aydin, "Energy-aware task allocation for rate monotonic scheduling," *In Real Time and Embedded Technology and Applications Symposium*, no. 11, pp. 213-223, 2005.

[29] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems." *In ACM SIGOPS Operating Systems Review*, vol. 35, no. 5, pp. 89-102, December 2001.

[30] H. Aydin and Q. Yang, "Energy-aware partitioning for multiprocessor real-time systems." In *Parallel and Distributed Processing Symposium, 2003.* Proceedings. International, pp. 9-pp. IEEE, 2003.

[31] Y. Liu and A. K. Mok, "An integrated approach for applying dynamic voltage scaling to hard real-time systems." *In Real-Time and Embedded Technology and Applications Symposium, 2003*. Proceedings. The 9th IEEE, pp. 116-123. IEEE, 2003.

[32] H. Aydin, R. Melhem, D. Mossé, and P. M. Alvarez, "Power-aware scheduling for periodic real-time tasks," *Computers, IEEE Transactions on* 53, no. 5, pp. 584-600, 2004.

[33] M. Mezmaz, N. Melab, Y. Kessaci, Y. C. Lee, E-G. Talbi, A. Y. Zomaya, and D. Tuyttens, "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems."*Journal of Parallel and Distributed Computing*, vol. 71, no. 11, pp. 1497-1508, November 2011.

[34] C. M. Wu, R. S. Chang and H. Y. Chan, "A green energy-efficient scheduling algorithm using DVFS technique for cloud datacenters," *Future Generation Computer Systems*, pp. 1-21, 2013. Available: http://dx.doi.org/10.1016/j.future.2013.06.009

[35] S. Yassa, R. Chelouah, H. Kadima and B. Granado, "Multi-objective approach for energy-aware workflow scheduling in cloud computing environments," *The Scientific World Journal*, vol. 2013, pp. 1-13, September 2013.

[36] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, IEEE Computer Society*, pp. 826-831, May 2010.

[37] Y. Jin, Y. Wen and Q. Chen, "Energy efficiency and server virtualization in data centers: An empirical investigation," In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pp. 133-138, IEEE, March 2012.

[38] S. K. Garg, C. S. Yeo and R. Buyya, "Green cloud framework for improving carbon efficiency of clouds." In *Euro-Par 2011 Parallel Processing*, pp. 491-502. Springer Berlin Heidelberg, 2011.

[39] L. Liu, H. Wang, X. Liu, X. Jin, W. B. He, Q. B. Wang and Y. Chen, "GreenCloud: a new architecture for green data center." In *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session*, pp. 29-38. ACM, June 2009.

[40] L. Lefèvre and A. C. Orgerie, "Designing and evaluating an energy efficient cloud." *The Journal of Supercomputing*, vol. 51, no. 3, pp. 352-373, 2010.

[41] A. Uchechukwu, K. Li and Y. Shen, "Improving cloud computing energy efficiency." In *Cloud Computing*

Licensed Under Creative Commons Attribution CC BY

Congress (APCloudCC), 2012 IEEE Asia Pacific, pp. 53-58. IEEE, November 2012.

[42] B. Yamini and D. V. Selvi, "Cloud virtualization: A potential way to reduce global warming." In *Recent Advances in Space Technology Services and Climate Change (RSTSCC), 2010*, pp. 55-57. IEEE, November 2010.

[43] E. Pakbaznia, M. Ghasemazar and M. Pedram, "Temperature-aware dynamic resource provisioning in a power-optimized datacenter." In *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 124-129. European Design and Automation Association, March 2010.

[44] C. D. Patel, R. Sharma, C. E. Bash and A. Beitelmal, "Thermal considerations in cooling large scale high compute density data centers." In *Thermal and Thermomechanical Phenomena in Electronic Systems, 2002. ITHERM 2002. The Eighth Intersociety Conference on*, pp. 767-776. IEEE, 2002.

[45] R. K. Sharma, C. E. Bash, C. D. Patel, R. J. Friedrich and J. S. Chase, "Balance of power: Dynamic thermal management for internet data centers." *Internet Computing, IEEE*, vol. 9, no. 1, pp. 42-49, 2005.

[46] J. D. Moore, J. S. Chase, P. Ranganathan and R. K. Sharma, "Making Scheduling" Cool: Temperature-Aware Workload Placement in Data Centers." In *USENIX annual technical conference, General Track*, pp. 61-75, April 2005.

[47] J. Li, Z. Li, K. Ren and X. Liu, "Towards optimal electric demand management for internet data centers," *Smart Grid, IEEE Transactions on*, vol. 3, no. 1, pp. 183-192, February 2012.

[48] I. Rodero, H. Viswanathan, E. K. Lee, M. Gamell, D. Pompili and M. Parashar, "Energy-efficient thermal-aware autonomic management of virtualized hpc cloud infrastructure." *Journal of Grid Computing*, vol. 10, no. 3, pp. 447-473, July 2012.

[49] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. K. S. Gupta and S. Rungta, "Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers," *Computer Networks*, vol.53, no. 17, pp. 2888-2904, December 2009.

[50] T. Heath, A. P. Centeno, P. George, L. Ramos, Y. Jaluria and R. Bianchini, "Mercury and freon: temperature emulation and management for server systems," In *ACM SIGARCH Computer Architecture News*, vol. 34, no. 5, pp. 106-116. ACM, October 2006.

[51] Q. Tang, S. K. S. Gupta and G. Varsamopoulos, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," *Parallel and Distributed Systems, IEEE Transactions*, vol. 19, no. 11, pp. 1458-1472, October 2008.

[52] T. V. T. Duy, Y. Sato and Y. Inoguchi, "Performance evaluation of a green scheduling algorithm for energy savings in cloud computing," In *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, pp. 1-8. IEEE, April 2010.

[53] J. Prevost, K. Nagothu, B. Kelley and M. Jamshidi, "Prediction of cloud data center networks loads using stochastic and neural models," *in System of Systems Engineering (SoSE), 2011 6th International Conference on*, pp. 276-281, June 2011.

[54] N. M. A. Sallami, A. A. Daoud and S. A. A. Alousi, "Load Balancing with Neural Network." *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol.4, no. 10, pp. 138-145, 2013.

[55] Y. Fang, F. Wang and J. Ge, "A task scheduling algorithm based on load balancing in cloud computing," In *Web Information Systems and Mining*, pp. 271-277. Springer Berlin Heidelberg, 2010.

[56] L. M. Zhang, K. Li and Y. Q. Zhang, "Green task scheduling algorithms with speeds optimization on heterogeneous cloud servers," In*Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, pp. 76-80. IEEE Computer Society, December 2010.

[57] W. Li, G. Xu, J. Zhao, X. Fu, Y. Dong, Y. Zhai and W. Dan, "Task Scheduling Strategy Based on Tree Network in Cloud Computing Environment," In *International Conference on Mechanical Engineering and Technology (ICMET-London 2011)*. ASME Press, 2011.

[58] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107-113, 2008.

[59] N. Maheshwari, R. Nanduri and V. Varma, "Dynamic energy efficient data placement and cluster reconfiguration algorithm for MapReduce framework," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 119-127, January 2012.

[60] T. White, *Hadoop: The definitive guide*, 2nd ed.; O'Reilly Media, Inc., 2012.

[61] D. Kumar and B. Sahoo, "Energy Efficient Heuristic Resource Allocation for Cloud Computing," *International Journal of Artificial Intelligent Systems and Machine Learning*, vol.6, no. 1, pp.32-38, January 2014.

[62] J. Li, J. Peng and W. Zhang, "A scheduling algorithm for private clouds," *Journal of Convergence Information Technology*, vol. 6, no. 7, pp. 1-9, July 2011.

[63] D. Lucanin and Ivona Brandic, "Take a break: cloud scheduling optimized for real-time electricity pricing," In *Cloud and Green Computing (CGC), 2013 Third International Conference on*, pp. 113-120. IEEE, September 2013.

[64] K. Pepple, *Deploying OpenStack*, 1st ed.; O'Reilly Media, Inc., 2011.

[65] R. Santhosh and T. Ravichandran, "Pre-emptive scheduling of on-line real time services with task migration for cloud computing," In *Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on*, pp. 271-276. IEEE, February 2013.

[66] B. Furht, *Handbook of social network technologies and applications*. Vol. 1. New York: Springer, 2010.

[67] C. Xiong, L. Feng and L. Chen, "A New Task Scheduling Algorithm Based on Improved Genetic

Licensed Under Creative Commons Attribution CC BY

Algorithm in Cloud Computing Environment," *AISS: Advances in Information Sciences and Service Sciences*, vol. 5, no. 3, pp. 32-38, February 2013.

[68] J. Liu, X. Luo, X. Zhang, F. Zhang and B. Li, "Job scheduling model for cloud computing based on multi-objective genetic algorithm," *IJCSI International Journal of Computer Science*, vol. 10, no. 1, pp. 134-139, January 2013.

[69] S. R. Suraj and R. Natchadalingam, "Adaptive Genetic Algorithm for Efficient Resource Management in Cloud Computing," *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, no. 2, pp. 350-356, February 2014.

[70] X. Wang, Y. Wang and H. Zhu, "Energy-efficient task scheduling model based on MapReduce for cloud computing using genetic algorithm," *Journal of Computers*, vol. 7, no. 12, pp. 2962-2970, December 2012.

[71] S. Srikantaiah, A. Kansal and F. Zhao, "Energy aware consolidation for cloud computing," In *Proceedings HotPower'08 Proceedings of the 2008 conference on Power aware computing and systems*, vol. 10, pp. 1-10. 2008.

Author Profile

Iza'in Nurfateha Ruzan received her Bachelor Science degree in Computer Science (Software Engineering) from Universiti Teknologi Malaysia in 2010. At present she is pursuing the Master of Science (Computer Science) fulltime research at Universiti Teknologi Malaysia. Her research interests include Cloud Computing, Scheduling Theory and Soft Computing.

Suriayati Chuprat received her Doctor of Philosophy (Mathematics) from Universiti Teknologi Malaysia in 2009. In 2013-2014, she attached to University of York, United Kingdom for a Post-Doctoral program. At present she is working as Senior Lecturer at Advanced Informatics School Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia. Her research interests include Real-time Systems, Scheduling Theory, Parallel Computing and Security Cloud Computing.

Pegah Razmara received her Master of Science from Universiti Teknologi Malaysia in 2010. At present she is pursuing the Doctor of Philosophy (Computer Science) from Universiti Teknologi Malaysia. Her research interests include Real-time Systems, Scheduling, Resource Allocation, Parallel and Distributed Computing.

Licensed Under Creative Commons Attribution CC BY