Energy Efficient Heterogeneous Multicore Cloud Server for Independent Real Time Tasks in Cloud Environment

A.S. Radhamani and E. Baburaj

ABSTRACT

Multicore processors and cloud computers are two emerging classes of complex hardware that have the capability of complex hardware that have the capability to provide unique capacity to the normal user. In order for the user to efficiently bind all of this computational energy and resource utilization, a new hardware platform is needed. Moreover, the current cloud systems drive much intricacy onto the user, requiring the user to handle individual Virtual Machines (VM) and deal with many system level concerns. In this work we present the mechanisms and implementation of heterogeneous multicore system as cloud server, both for effective resource utilization and energy consumption, by Energy Efficient Dual Phase Scheduling (EEDPS) of workloads using Genetic Algorithm. Resource utilization and energy consumption are the two major issues of cloud computing providers, as a result to manage resources have to be efficiently utilized so that the required energy consumption can be achieved.

INTRODUCTION

The current multicore revolution guarantees extreme changes in fundamental architecture as the member of schedulable processing elements significantly increasing. Along with the multicore uprising, cloud computing & IaaS have been in advance attractiveness. This emerging computing model has an enormous potential to change the computing industry and programming models (Arumbrust, et al., 2009). In multicore processors, as the number of cores being added, the cloud computing providers have been growing at an infinite rate, as per the user demand. To achieve optimal efficiency in terms of energy, power and performance, when running parallel workloads of cloud users in a multicore cloud server provides a unique challenging in the computing era. In contrast with the traditional high performance computing systems which are designed with a slew of homogeneous processing elements, our programming model consists of a heterogeneous multicore processors, where each core itself consists of many cores, and individual core will act as a datacenter or distribution nodes in a computing cloud. The computing model is similar to Intel’s core i7(Named Nehalem) is entirely different from previous X86 multicore processors.

The computing model aims at streaming the real time demand with different Service Level Agreement (SLAs) in terms of reliability, scalability to performance. With the advance of heterogeneous multicore processors as cloud servers are widely applied in modern supercomputing datacenters. These cloud data centers are essentially require high energy to maintain operation (Bianchini, et al., 2004). Normally a typical datacenter with 1000 racks requires a power of 10mW to operate (Rivoreire, et al., 2007). As the high energy consumption results in high energy cost, which become a crucial issue, which will reduce the profit margin of cloud providers. Therefore there is an essential need for energy efficient solutions which can be achieved by dual phase scheduling that combines Data Center Selection (DCS) and Work Load Selection (WLS) scheduling algorithm to utilize cores effectively thereby minimizing the completion time of the task on the node thereby minimizing the overall energy consumption.

The scheduling of independent workloads or tasks in a set of heterogeneous computing cores has been studied to be an NP complete. Since the heterogeneous core have different processing capabilities. Many static and dynamic scheduling algorithms have been proposed to schedule task among different cores, to achieve the required performance. The most important performance parameters which can be studied in the heterogeneous environment are as follows, availability of the datacenters, service reliability, throughput of the datacenters, overall completion time of the workload, workload execution time, and workload waiting time and so on. As
one of the important performance metric in heterogeneous computing cloud, the completion of the tasks needs to be studied. Completion time of a specific task is important, because of it’s ability to describe the performance of the system. As a result minimizing, the completion time of a particular task can be considered as a goal of the proposed scheduling algorithm, due to its significance role. As the environment considered is heterogeneous, it is also necessary, to consider the energy consumption of the core. Since both completion time and energy consumption are highly dependent on each other and should not be optimized separately Because of the heterogeneity, finding the available core and scheduling the independent tasks are very important to achieve efficient resource utilization and high throughput. In this paper the goal of scheduling is to minimize the overall completion time of the independent tasks and energy consumption of the data centers. We propose an Energy Efficient Dual Phase Scheduling(EEDPS) scheme for heterogeneous computing cloud, which makes a good trade off between availability and performance the proposed method was parallel genetic Algorithm approach along with to achieve exact completion time, the concept of Directed Acyclic Graph (DAG) is assumed.

To meet the rising computational requirements parallel processing approach poses a number of problems not encountered in traditional sequential processing (Ali Allahaerdi, et al., 2006; Topcuoglu and Hariri, 2002), the most important of which is the multicore scheduling, three unique features distinguish our approach from a traditional GA algorithm are as follows First, in general, it is almost important to represent the precedence constraints of the workloads, since the result of other workloads are required before a particular workload can be executed. Hence to minimize the overall completion time, the scheduling algorithms typically schedule all the workloads on a given number of available data center without isolating precedence constraints, and hence a bond element is assigned to each workload to show the relation among the workload. Second a high quality initial solution is generated by scheduling the highly coupled task onto the same datacenter which can largely decrease the communication cost in the heterogeneous cores. Third, our approach, can fine tune the system, so that the datacenter that operate on high speed have maximum resource utilization, results in predictable energy consumption within the specific completion time.

Related Work:
Several research work have been carried out in the past decades, in the heuristic algorithm for job scheduling and generally, since scheduling problems in heterogenic multicore cloud systems is NP lard, as the problem size increasing the time required to complete the problem also increases exponentially, therefore it is extremely significant and necessary to develop algorithms to find solution to these problems. In this section we present some of the previous works in scheduling in a heterogeneous environment without energy constraints and with energy constraints.

Scheduling Policies Without Energy Constraints:

The problem of parallel system task scheduling, Performance Effective Genetic Algorithm (PEGA), for heterogeneous parallel multicore is presented in (Jastic singh and Guarvinder Singh, 2012) in which authors have considered only the finish time of the tasks to hybrid the throughout of the system. In (Yun Wen, 2011) a novel hybrid metaheuristic based approach for solving task scheduling in the heterogeneous multiprocessor system has been proposed by means of incorporating GA with Unable Network Search (UNS) approach. However this approach can not be applicable for real distributed computing environments. Based on GA and chromosome background tree, without using any classical algorithms a novel intelligent solution for minimizing the entire run time is explained by Mohammed Hassan Shenassa, and Mahdi Mahmooodi in (Mohammad Hassan Sherassa and Mahdi Mahnooodi, 2006). Hee-Jun Park and Byung Kook Kim presented as optimum scheduling, to minimize the computation period of interactive executive, by using spatial searching and enhanced branch and bound technique in (On Hee Tun park, 2001). But it has the draw back that it can be used for general network achievements and heterogeneous processors. Reza Entegari-Maleki and Ali Movaghar presented a task scheduling method to minimizing the overall response time of the tasks in grid computing environments. If address this goal, they have used a Discrete Time Market Chain (DTMC) (Roza Entezari Maleki and Ali Movaghar, 2012) with the grid environment. However, the other QoS parameter like reliability of task executive availability of the resources, cost of scheduling are not considered. Also in (Xianzhen Kong, et al., 2011) and (Sasmitha Kumari Nayak, et al., 2012) authors have described about efficient dynamic task scheduling in virtualized data centers with fuzzy techniques and in grid environment using GA respectively. A decentralized dynamic scheduling approach named the Community Aware Scheduling Algorithm (CASA), addressed a two phase solve with an collection of sub algorithms are explained in (Ye Huang, et al., 2011). To adapt GA to non-identical parallel machine scheduling problem, a new cross over operator and new optimality criterion were designed and implemented (Savas Balin, 2011). In (Ali Allahaerdi, et al., 2006), author classifies scheduling problems into those with batching and non-batching considerations, and with sequence-independent and sequence-dependent setup times.
Scheduling Policies with Energy Constraints:

To address the issue of energy efficient strategies (Xinomin Zhu, et al., 2012), authors have proposed in novel scheduling strategy-Adaptive Energy Efficient Scheduling AEES, per aperiodic and independent real time tasks by adjusting the voltage according to the workload conditions of the cluster. A two phase energy scheduling strategy called EETDS, to reduce the communication cost when allocating parallel dependent tasks in (Zerg, Z., et al., 2007). The most typical application model used in scientific and engineering field, five parallel applications are precedence constrained. For such applications, a parallel biobjective hybrid genetic algorithm were proposed by M. Mezmaz et al. (2011) considering two objectives namely makerpan and energy consumption which is based on DVS (Dynamic Voltage Scaling). In (Khan and Ahmad, 2009), the authors presented the problem for assigning a set of tasks onto the machines with the characteristics of DVS. Also in this approach authors have not considered the precedence constraints of the tasks. In (Topcuoglu, H., 2002) authors presented two novel scheduling algorithms for a bounded number of heterogeneous processors with an objective to simultaneously meet high performance and fast scheduling time, which are called the Heterogeneous Earliest-Finish-Time (HEFT) algorithm and the Critical-Path-on-a-Processor (CPOP) algorithm.

In some of the previous research studies, on scheduling, authors have considered, the system as Homogeneous for energy consumption problems (Kim, K.H., et al., 2007; Ge, R., et al., 2005). Also in (Ge, R., et al., 2005) authors proposed a dynamic priority for periodic and aperiodic tasks using a single processor system. In (Kim, K.H., et al., 2007), for the purpose of quality control, authors have concentrated their study based on deadline constrains of the tasks. Saurabh et al (2011), addressed an issue, that exploit heterogeneity among different data centers for a cloud provides by using near optimal scheduling policies, considers various energy parameters such as energy cost, carbon emission rate, workload and CPU power efficiency.

Unlike the various above methods, the EEDDS involves energy aware scheduling of independent tasks in heterogeneous multicore cloud systems have not yet received much attention. In this paper we investigate scheduling of independent tasks to minimize the total completion time of the date centers which satisfying minimum energy constraints in a heterogeneous multicore cloud system. As the completion time and energy consumption and resource utilization in our model one highly dependent on each other, should not be optimized separately.

Problem Formulation:

In this section, we address the system, energy and scheduling models in our proposed method.

Cloud Computing Model:

A cloud computing system is a set of distributed resources to be allocated services to be remote users through a network (Germain Renaud, 2009). In our paradigm, the cloud is assumed to be hosted in a data center which in built consists of heterogeneous cores. Each data center is also assumed as a heterogeneous multicore (or) many core processor which provides services to the end users. The communication among the datacenters can be represented by a task graph. An edge between two workloads represents inter service communication between two datacenters, in this graph.

The heterogeneous multicore cloud system used in our implementation consists of a set of C of c cores/datacenters. Each core c \( c \in C \) can operate with different speeds, as the system assumed is heterogeneous. The communication cost between the cores / datacenters are assumed as same.

Parallel Program Model:

This paper considers the parallel program model, with the following scenario. Scheduling in parallel heterogeneous multicore cloud system is classified into many classes based on the characteristics of the datacenters, worked to be scheduled and the availability of the datacenters. The heterogeneous multicore cloud system in, consists of a set of cores/datacenters, that have different speed (V) or processing capabilities, which implies that tasks executed on different datacenters encounter different completion time. The method used in the completion of the tasks on heterogeneous multicore system is to partition the huge task into set of tasks of appropriate size and an abstract model of the partitioned tasks can be represented by Directed Acyclic Graph (DAG). The heterogeneous multicore cloud, task scheduling problem includes the problem of assigning the data center with required precedence relationship to suitable cores, and the probability of ordering task executives on each the applications is represented by DAG, \( G=(U,E) \) where U is the set of ‘u’ task nodes, and E is the set of ‘e’ directed common edges between the tasks. Each edge \( e_{ij} \) represents the precedence constraint that \( u_i \) cannot be scheduled until \( u_i \) has been completed, and hence \( u_i \) is a predecessor of \( u_i+1 \) and \( u_i+1 \) is the successor of \( u_i \). As a result the task with no predecessor is called the initial task while the task with no successor is called the final task.

The scheduling problem in heterogeneous multicore cloud system is a dual phase scheduling to identify the datacenter for the given workload and to partition the workload between different cores of datacenter by accomplishing minimum completion time and energy consumption simultaneously. We assumed that each data center has ‘M’ different cores, \( M = \{ x_i; i = 1, 2, 3...m \} \) and T different tasks \( T = \{ t_i; i = 1, 2...m \} \) are
considered in a heterogeneous environment, every core works on different speeds \( V = \{ v_i; i = 1, 2, 3 \ldots n \} \) and processing capabilities. It is also assumed that the core \( c_1 \) much faster than \( c_2, c_3 \) and so on for any data center. 

If the processing speed (or) frequency is \( v(i,j) \) the completion time can be defined based on task size and is given by,

\[
\text{Completion time} = \frac{\text{Task size}}{\text{Speed of the core}}
\]

Maximum completion time \( = \left( \max \sum_{i=1}^{n} \frac{T(i,j)}{V(i,j)} \right) \) \( \text{(1)} \)

Before presenting the objective function it is necessary to state the communication cost between the cores for each data center. The communication cost among the cores for transferring the tasks from one core to authors is set such that, if the tasks are scheduled on the same core, it becomes same (or) negligible, on the other hand, if the tasks are scheduled on different cores small predefined value is assumed. 

The objective function is to minimize the total completion time and is a measure for distinguishing good solutions from bad solutions. Therefore the devised objective function is,

\[
\text{Min } C_{\text{max}} = \left[ \max \sum_{i=1}^{n} \frac{T(i,j)}{V(i,j)} \right] \quad \text{(2)}
\]

**Energy Model:**

Power consumption model in complementary metal-oxide semiconductor (CMUS) logic circuits is the basic for devising the energy model for the proposed method. While running a task with an average power pang, at the processor speed of \( f \) for \( T \) time units, the energy consumption of the core is mathematically given by the following.

\[
E = P_{\text{avg}} \times T 
\]

Where

\[
P_{\text{avg}} = \gamma \times V^{2}dd \times f 
\]

\( \gamma = \) switching capacitance

\( V_{dd} = \) supply voltage

\( f = \) core frequency

From the aforementioned equations, it can be observed that the supply voltage is the dominant factor, and in equation (4) \( P_{\text{avg}} \) is an increasing function of \( f \) but it is an inverse relation with energy in equation (3), and hence the energy can be minimized.

**Scheduling Model:**

To address the aforementioned objectives, a EEDPS scheduling model, based on previously proposed Community Aware Scheduling algorithm (Ge, R., et al., 2005) is presented, and EEDPS involves two phases namely Data Center Selection (DCS) and workload selection (WLS), which work together in an optimized scheduling process to achieve the required goal.

**Genetic Algorithms:**

The Genetic Algorithm (GA) have been recognized as an optimization technique for combinatorial problem and have the significant role in the area of parallel computing. The recent and well known application includes reliability design, scheduling and sequencing, transpiration, vehicle routing and scheduling and so on.

The general form of GA was described by Goldberg. Based on the mechanism of natural selection and natural genetics, GAs are considered as stochastic search techniques. GAs start with an initial set of random solutions called population as a result it is different from conventional search techniques. Hence the first step of GA starts by creating an initial population and each individual is called a chromosome, responding a solution to the problem. Once the chromosome has been coded, it is possible to evaluate the performance (or) fitness of individuals in a population. During each generation, genetic operation such as crossover, mutation and selection are applied in order to search better, solutions.
Genetic Scheduling Policies:
Based on heterogeneous dual phase scheduling an algorithm is created which can adapt the varying resource environments utilizing multiheuristic GA. The activation of data phase scheduling first to schedule an if workload to a data center. Second schedule the worked within the selected datacenter from step (1). Depending upon the objective to minimize the completion time of the workloads with the datacenter, we have designed various mapping policies.

Input: set of independent workloads and datacenters.
Input: set of independent workloads within datacenters
for each heuristic do
  generate schedule 1;
for each heuristic do
  generate schedule 2
end
while population not finish do
  copy and repeat heuristic schedules;
end
repeat
  cycle cross over;
  random mutations;
  rebalance;
  selection (based on roulette wheel),
  find best schedule;
  update mutation rate;
until stopping criteria met;
return schedule with minimum completion time.

Algorithm 1: Pseudo Code for Genetic Scheduling Policies:
The set of cores of the cloud system is heterogeneous. The resource utilization between cores in the datacenter can vary over time. The availability of each datacenter can also vary over time. Tasks/workload are independent and dynamic can be processed by any of the cores in the data center of the distributed cloud system.

In our approach described a chromosome (solution) is compassed of a sequence of \( N \) genes. The \( i^{th} \) gene of a solution is \( g \) is denoted \( g_i \). Each gene is defined by a task, core and frequency. They are denoted as \( t(g_i), c(g_i) \) and \( f(g_i) \). This mean that the task \( t(g_i) \) is assigned to the core \( c(g_i) \) with the frequency \( f(g_i) \).

Unlike the First Come First Served (FCFS) and Heterogeneous Earliest Finish Time (HEFT), the GA builds task parts \( t(g_1), t(g_2) \ldots t(g_n) \) of a solution \( g \). Therefore mutation and crossover operation of the GA affect only the task part of the genes for each solution; thus the role of GA is to give better task scheduling.

The role of fitness (evaluation) operator is to calculate the completion time and energy consumption of each solution. If the task, core, and frequency parts of each gene are known, the fitness can be determined. The Table1 describes the experimental parameters used in our implementation.

Table 1: Experimental Parameters.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Tasks</td>
<td>20–40–60–80–100–120–140–160–180</td>
</tr>
<tr>
<td>No of Cores</td>
<td>02 04 06 08 16 32 64 100 128</td>
</tr>
<tr>
<td>Core Frequency</td>
<td>2 3 4 5 6 7 8 9 10</td>
</tr>
<tr>
<td>Population Size</td>
<td>100</td>
</tr>
<tr>
<td>Number of Generations</td>
<td>50</td>
</tr>
<tr>
<td>Cross over rate</td>
<td>1</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Other Parallel Scheduling Approaches:
This section with a brief overview two existing on parallel scheduling approaches, namely FCFS and HEFT. Then our genetic algorithm based scheduling is presented.

FCFS (First Come First Served) Scheduling:
FCFS is the simplest scheduling algorithm for heterogeneous cloud applications. Tasks or work loads are dispatched according to their arrival time from the queue. Being its non preemptive characteristics, after selecting the data center it completes its run and the next task / work load and so on. Each parallel active data center is identified also from which the total computing power can be calculated, by the number of processing data center times the processor power, from which energy computation can be computed.
HEFT (Heterogeneous Earliest Finish Time):
We choose HEFT, since it is simple, popular and as competitive approach HEFT algorithm (Frederic Suter, 2004) has two phases, the first is the prioritizing phase, which assigns value to each task called upward rank, which is based on computation and mean communication costs. Then by sorting the tasks in decreasing order of rank, the task list is generated. The second phase processor selection phase, the task which has the highest upward rank is selected and assigned to the processor that minimizes its action time. The time when the execution of a task can actually start on a processor is termed as Earliest start time (EST), and an execution start when the processor become available. Earliest Finish Time (EFT) of a task can be computed by adding execution cost.

Performance Evaluation and Results:
In this section we evaluate the performance of the proposed energy efficient dual phase scheduling strategy using MATLAB simulations.

To illustrate the performance improvements gained by EEDPS, we compared it with existing FCFS and HEFT conventional approaches. The GA scheduler uses different number of generations based on whether stopping criterions are met. In our implementation 50 generations was chosen found to be large enough figure for the random mutations.

The performance metrics by which we evaluate the system include

1. Completion Time = \frac{\text{Task}}{\text{Frequency}}

2. Total energy consumption = Average power consumption x Total running time of task

3. Resource utilization = Time to execute tasks x Total running time of task in a datacenter.

In our simulation we assumed the datacenter which varies from 2 to 256. Fig1 (a) shows the performance of FCFS, HEFT and EEDPS in terms of completion time.

Fig. 1(a): No.of Data Centers Vs Completion Time.
Fig. 1(b): Workload Vs Completion Time.

Fig. 2(a): No. of Data Centers Vs Energy consumption per datacenter.

Fig. 2(b): No. of Data Centers Vs Improvement in Energy %

Fig. (3): No. of Data Centers Vs Datacenter Utilization%.
Also to it is observed that EEDPS has higher completion time at the beginning than that of the other schemes, because it provides optimum completion time with varying data centers as the proposed scheduling approach is of evolutionary GA. Though the conventional approaches provide better results like our approach, the completion time fluctuates, as the data center size increases. Also in FCFS as the load increases, newly arrived workloads and workload with higher priority, conditions have to wait for long time. Also from fig1 (b) HEFT is unable to offer high stability when load is high.

Fig (2)a show that energy consumption per datacenter decreases initially in both the conventional approaches and as the number of data center increases the energy consumption is also increases. This is because when the number of data center increases, the system of load is distributed to all the curves of data center in which allows the tasks to conserve more energy as the voltage increases when the number of data center is less than 32, EEDPS consumes almost equal energy than that of our conventional approach, but when it comes to a larger size, energy consumption of EEDPS in improved. Fig 2(b) shows the improvement in energy for different data centers. From the fig it is clearly stated that as the number of data center increases EEDPS provides better improvement in energy on heterogenic multicore cloud utilizing high scheduling. Fig (3) shows, the EEDPS exhibits its superior performance interiors of resource utilize FCFS her the par resource utilization, than the other two schemes, the results also high light data center resources in the cloud while making a good trade off between energy efficiency and scheduling.

Conclusion: We investigated in this paper an Energy Efficient Dual Phase Scheduling strategy, precedence-constrained through DAG model, for heterogeneous multicore cloud server. The parallel goal of our scheduling approach is to minimize the completion time, with the effective energy consumption and resource utilization. Our simulation result shows that, the average energy computation is reduced by 45% and the completion time by 22% than the conventional methods which are in use. The proposed method uses EEDPS to model cores and the workloads within the datacenter, and then applied GA to achieve optimum completion time thereby we achieved energy consumption and resource utilization than the existing methods. In future the other QoS measure parameters like reliability of task execution, cost of scheduling may be combined with new scheduling model, so that better result can be assessed.

REFERENCES


