

Minimizing Energy Consumption by Task Consolidation in Cloud Centers with Optimized Resource Utilization

Mahendra Kumar Gourisaria¹, S. S. Patra², P. M. Khilar³

¹School of Computer Engineering, KIIT University, Bhubaneswar, India

²School of Computer Application, KIIT University, Bhubaneswar, India

³Department of Computer Science and Engineering, NIT, Rourkela, India

Article Info

Article history:

Received Jun 21, 2016

Revised Sep 18, 2016

Accepted Oct 02, 2016

Keyword:

CPU utilization

Energy consumption

Idle VM

Task consolidation

Virtualization

VMs

ABSTRACT

Cloud computing is an emerging field of computation. As the data centers consume large amount of power, it increases the system overheads as well as the carbon dioxide emission increases drastically. The main aim is to maximize the resource utilization by minimizing the power consumption. However, the greatest usages of resources does not mean that there has been a right use of energy. Various resources which are idle, also consumes a significant amount of energy. So we have to keep minimum resources idle. Current studies have shown that the power consumption due to unused computing resources is nearly 1 to 20%. So, the unused resources have been assigned with some of the tasks to utilize the unused period. In the present paper, it has been suggested that the energy saving with task consolidation which has been saved the energy by minimizing the number of idle resources in a cloud computing environment. It has been achieved far-reaching experiments to quantify the performance of the proposed algorithm. The same has also been compared with the FCFSMaxUtil and Energy aware Task Consolidation (ETC) algorithm. The outcomes have shown that the suggested algorithm surpass the FCFSMaxUtil and ETC algorithm in terms of the CPU utilization and energy consumption.

Copyright © 2016 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

S.S. Patra,

School of Computer Application,

KIIT University,

Bhubaneswar, Odisha, India.

Email: sudhanshupatra@gmail.com

1. INTRODUCTION

Data communications plays a crucial element of our daily lives and also in the computation. Most of our practices depends on gathering information through the client-server and distributed paradigm [1]. Now a days, client demands have tremendously increased in terms of the number of requests. To cater to the consistent amount of requests, computational capacities and facilities must be constantly reviewed and improved. To remain competitive, the proportional nonnegligible amount of the required energy has been often left behind.

Recent advocacy and tightly coupled with energy consumption called green or sustainable computing has received considerable attention in the field of computation as there is a huge need of minimizing the energy consumption. The scope of sustainable computing goes beyond the main computing

components, expanding into a much larger range of resources associated with auxiliary equipment, such as the water used for cooling and the physical/floor space occupied by the resources [14].

In Cloud computing, energy consumption and resource utilization are strongly coupled. Specifically, resources with a low utilization rate still consume an unacceptable amount of energy compared to the energy consumption of a fully utilized or sufficiently loaded Cloud computing [10]. According to recent studies [2-5] average resource utilization in most data centers can be as low as 20%, and the average energy consumption of idle resources can be as high as 60% (or peak power) [3]. To increase resource utilization, task consolidation is an effective technique, greatly enabled by virtualization technologies, which facilitate the concurrent execution of several tasks and, in turn, reduce energy consumption [4].

Power management has been broadly classified into static and dynamic management. Static power management has been dealing with fixed power and dynamic power managements with dynamic behaviors for additional degree of capability in virtualized data centers [6]. Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS) and Database-as-a-Service (DaaS) are the four levels of access in which clouds are deployed for the clients. The task has originated by the different type of customers according to their requirements. There are several heuristic algorithms proposed for local cloud for the centralized controller which has been power aware. Based on the system structure and the characteristics of the cloud infrastructures, a function between the resources of cloud and the combinatorial allocation task has been proposed, as an economics based optimization model [13].

The virtualization concept encapsulates the numerous services that have met the user needs in cloud computing scenario [7]. VMs have been designed to run on various servers which provide the multiple Operating system environments for different applications. Particularly, executing an application which requires resources has been made available for resource provisioning and VM provisioning. Resource provisioning is scheduling the requests for the physical resources where-as VM provisioning creates the instance of VMs as required by the different applications [8].

Server or workload consolidation is the main aim of the task consolidation problem. It allows the servers on a single physical server for minimization of energy consumed by a cloud data center. In the present paper, the task consolidation problem has been addressed to assign n tasks to a set of different resources and the utilization of nodes and distributed VMs are maintained by energy efficiency and load management. The availability of computer nodes during the power consumed by the cloud is the prime concern of the developed algorithm [9-10].

In this paper, the greedy heuristic algorithm has been evaluated and implemented for three basic task consolidations which assign tasks to the physical servers for minimizing the total energy consumed. The proposed heuristic is to minimize the number of idle VMs and minimize the number of idle VMs to as minimum as possible. It has also been shown that the performance improvement is based on different tasks. Section 2 defines the general model of cloud computing environment, energy consumption and task model of the system. We have firmly defined the problem of energy minimization based on the system model. Section 3 deals with the used heuristic algorithm and the algorithms are illustrated by means of an example in Section 4. Section 5 illustrates the set up for the simulation and analyzes the results generated by simulation. Finally, the conclusions have been described in Section 6.

2. CLOUD SYSTEM MODEL

The current section depicts the cloud and its function with the energy models. It also defines the job consolidation problem. The high level architecture of the cloud system is shown in Figure 1 [11]. Virtualization allows the cloud providers to create a set of VMs on a single physical machine that improves the Return on Investment (ROI). The energy consumption may be reduced by switching off the idle nodes, which eliminates the idle power consumption of the given system [12].

In the present work, the target system has been used which consists of a set N of r resources which can be interconnected in the sense that a common route exists between whichever two individual resources as shown in Figure 2. It assumes that the resources are identical in terms of their potential of computing. The virtualization technologies justifies this. The present study has however not considered the federated cloud environment in which the data centers can be placed at different physical locations and the client requests may be processed at various geographical locations.

Figure 3 shows the energy consumption has risen by 56% by data centers from 2005 to 2010 worldwide. Furthermore, CO₂ emissions of the ICT industries are currently approximated to be 2% of the global emissions. It has been observed that global emissions are equivalent to the emissions of the aviation industries. The energy model is conceptualized on the basis that energy consumption has a linear relationship

with processor utilization [11].

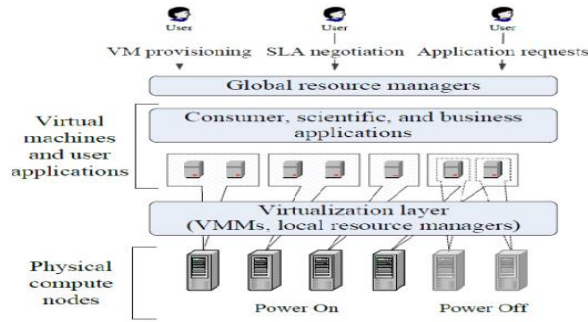


Figure. 1. Cloud System Architecture [11]

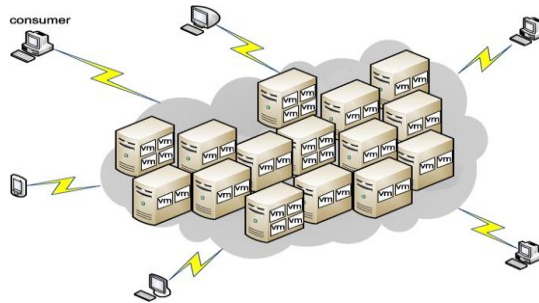


Figure. 2. The cloud Model [8]

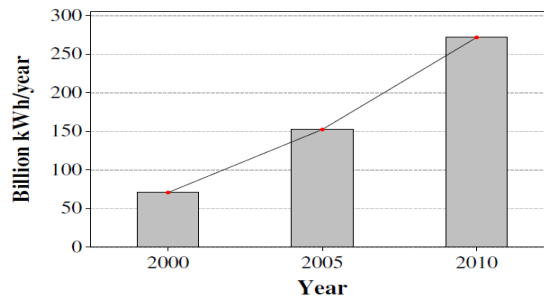


Figure 3. The Worldwide Data Centre Energy Consumption (2000-2010)[11]

This means that for a particular task, the processing time of a task and the processor utilization are the required parameters to determine the energy consumption for that task. The utilization Z_i is defined as for a resource r_i at any given time,

$$Z_i = \sum_{j=1}^t z_{i,j} \tag{1}$$

In (1) it has been observed that t is assigned as the number of tasks running at the current time where $z_{i,j}$ stands for the resource usage of the task $t_{i,j}$. The energy consumption E_i of a resource r_i at a given time is derived as

$$E_i = (\delta_{\max} - \delta_{\min}) \times Z_i + \delta_{\min} \tag{2}$$

In (2) it has assigned as δ_{\max} is meaning for the energy consumption at 100% utilization or peak load and δ_{\min} is assigning the minimum energy consumption as low as 1% consumption or in dynamic mode. In this paper, it has assumed that the resources in the objective arrangement are comprised of an efficient power saving method in favor of an inactive time slot. Particularly, the energy use of an inactive source at any specified time has been set to 10% of δ_{\min} . According to the energy consumption, the VMs can be broadly classified into six levels, the idle state and other five levels of CPU utilizations which has shown in Figure 4. In the present study, the task consolidation algorithm assigns a set M of m tasks to a set N of n cloud resources without violating the time constraints to minimize energy consumption and to maximize resource utilization.

$$E_i(V_i) = \begin{cases} \alpha \text{ watts/s, if idle} \\ \beta + \alpha \text{ watts/s, if } 0\% < \text{CPU utilization} \leq 50\% \\ 2\beta + \alpha \text{ watts/s, if } 50\% < \text{CPU utilization} \leq 70\% \\ 3\beta + \alpha \text{ watts/s, if } 70\% < \text{CPU utilization} \leq 80\% \\ 4\beta + \alpha \text{ watts/s, if } 80\% < \text{CPU utilization} \leq 90\% \\ 5\beta + \alpha \text{ watts/s, if } 90\% < \text{CPU utilization} \leq 100\% \end{cases}$$

Figure 4. Five Levels of CPU Utilization [9]

3. TASK CONSOLIDATION HEURISTIC ALGORITHM

Task allocation is a NP-Hard problem in the cloud. Heuristic and meta-heuristic algorithms are the two useful and efficient technologies for scheduling in cloud due to the ability to distribute and deliver the optimized solutions. In this section, we present the proposed task consolidation algorithm and the performance of the proposed algorithm has been compared with the existing ECTC and FCFSMaxUtil energy conscious task consolidation algorithms.

3.1 FCFSMaxUtil with Minimization of Idle VM

As the idle resources also consume power, the proposed algorithm always minimizes the number of idle resources by allocating a task at an instance to a VM which is currently idle. If no machine is idle it implements FCFSMaxUtil algorithm. The pseudo codes of all the algorithms are as follows:

3.2 ECTC with Minimization of Idle VM

As the idle resources also consume power, the proposed algorithm always minimizes the number of idle resources by allocating a task at an instance to a VM which is currently idle. If no machine is idle it implements ECTC algorithm. The pseudo codes of all the algorithms are as follows:

FCFSMaxUtil Algorithm

```

Input : Task Matrix (TaskId,Arrival Time,
                    Utilization, Execution Time)
Output : Allocation Table (Task Id, Machine Id, Start Time,End Time,Utilization)
[minArrivalTime maxArrivalTime]=
FindMinimumArrivalTimeMaximumArrivalTime(mat)
time= minArrivalTime
while (time <= maxArrivalTime)
do
    Tasklist = GetTasksatArrivalTime(mat ,
    time)
    sort the tasklist in ascending order of arrival
    for each task in tasklist
    do
        find the VM which has currently
        highest CPU Utilization
        Assign the task to the VM and
  
```

```

        update the Allocation table
        time= time +1
    End Algorithm

```

ECTC Algorithm

```

Input : Task Matrix (TaskId,Arrival Time,
                    Utilization, Execution Time)
Output : Allocation Table (Task Id, Machine Id, Start Time,End Time,Utilization)
for each task in tasklist do
    for each vm in vmlist do
        max=-1
        E=
        EnergyConsumedIncludingTheTask(task,vm)
        //Allocate the task into the VM to the
        maximum energy efficient
        if E > max
            max=E
            allocatedVm =Vm
        end if
    end for
    if allocatedVm !=NULL
        allocate task to allocated VM
        Update the Alloc table
    end if
end for
End Algorithm

```

FCFSMaxUtil with Minimization of Idle VM

```

Input : Task Matrix (TaskId,Arrival Time,
                    Utilization, Execution Time)
Output : Allocation Table (Task Id, Machine Id, Start
                        Time,End Time,Utilization)
[minArrivalTime maxArrivalTime]=
FindMinimumArrivalTimeMaximumArrivalTime(mat)
time= minArrivalTime
while (time <= maxArrivalTime)
do
    Tasklist = GetTasksatArrivalTime(mat, time)
    sort the tasklist in ascending order of arrival
    for each task in tasklist do
    for each Vm in Vmlist do
        if CPUUtilization(Vm)==0
            allocatedVm=Vm
            return
        else
            find the VM which has
            currently highest CPU
            Utilization
            Assign the task to the
            VM and update the
            Allocation table
        endif
    end for
    time= time +1
end while
End Algorithm

```

ECTC with Minimization of Idle VM

```

Input : Task Matrix (TaskId,Arrival Time,
                    Utilization, Execution Time)
Output : Allocation Table (Task Id, Machine Id, Start

```

```

        Time,End Time,Utilization)
for each task in tasklist do
  for each Vm in Vmlist do
    if CPUUtilization(Vm)==0
      allocatedVm=Vm
      return
    else
      max=-1
      E= EnergyConsumedIncludingTheTask(task,Vm)
      //Allocate the task into the VM to
      which it will be maximum energy
      efficient
      if E > max
        max=E
        allocatedVm =Vm
      end if
    end for
  end for
  if allocatedVm !=NULL
    allocate task to allocatedVM
    Update the Alloc table
  end if
end for

```

4. AN ILLUSTRATION

Consider a set of 10 VMs $V=\{V1,V2,...,V10\}$ and a set of 20 independent tasks $T=\{T1,T2,T3,..T20\}$ in which each task T_i has 4 tuples {TaskId, Arrival Time, Processing Time, CPU Utilization}. We have considered the threshold value of CPU utilization as 100%. The task table has been shown in Table 1. The task allocation table for various algorithms such as MaxMaxUtil, ECTC and our proposed algorithm are shown in Tables 2, 3,4 and 5 respectively.

Table 1. Example of Task Table

Task Id	Arrival Time	Processing Time	Utilization
1	1	12	54
2	1	5	62
3	1	7	31
4	1	12	51
5	1	9	67
6	2	8	59
7	2	11	57
8	2	8	31
9	2	10	54
10	2	10	66
11	2	17	61
12	3	17	45
13	3	13	43
14	3	9	59
15	3	7	13
16	3	12	40
17	4	12	63
18	4	11	22
19	4	6	18
20	4	14	33

Table 2. Task Allocation Table using FCFSMaxutil Algorithm

Task Id	Machine Id	Start Time	End Time	Utilization
1	1	1	12	54
2	2	1	5	62
3	3	1	7	31
4	4	1	12	51
5	5	1	9	67
6	6	2	9	59
7	7	2	12	57
8	3	2	9	31
9	8	2	11	54
10	9	2	11	66
11	10	2	18	61
12	2	6	22	45
13	3	10	22	43
14	5	10	18	59
15	7	3	9	13
16	6	10	21	40
17	8	12	23	63
18	2	6	16	22
19	4	4	9	18
20	9	12	25	33

Table 3. Task Allocation Table using ECTC Algorithm

Task Id	Machine Id	Start Time	End Time	Utilization
1	1	1	12	54
2	2	1	5	62
3	1	1	7	31
4	3	1	12	51
5	4	1	9	67
6	5	2	9	59
7	6	2	12	57
8	3	2	9	31
9	7	2	11	54
10	8	2	11	66
11	9	2	18	61
12	7	3	19	45
13	6	3	15	43
14	10	3	11	59
15	1	3	9	13
16	10	3	14	40
17	2	6	17	63
18	9	4	14	22
19	3	4	9	18
20	8	4	17	33

Table 4. Task Allocation Table using the Proposed Minimization of Idle VM Algorithm using ECTC

Task Id	Machine Id	Start Time	End Time	Utilization
1	1	1	12	54
2	2	1	5	62
3	3	1	7	31
4	4	1	12	51
5	5	1	9	67
6	6	2	9	59
7	7	2	12	57
8	8	2	9	31
9	9	2	11	54
10	10	2	11	66
11	8	2	18	61
12	1	3	19	45
13	4	3	15	43
14	3	3	11	59
15	5	3	9	13
16	7	3	14	40
17	2	6	17	63
18	9	4	14	22
19	5	4	9	18
20	10	4	17	33

Table 5. Task Allocation Table using the Proposed Minimization of Idle VM Algorithm using MAXUTIL

Task Id	Machine Id	Start Time	End Time	Utilization
1	1	1	12	54
2	2	1	5	62
3	3	1	7	31
4	4	1	12	51
5	5	1	9	67
6	6	2	9	59
7	7	2	12	57
8	8	2	9	31
9	9	2	11	54
10	10	2	11	66
11	2	6	22	61
12	3	8	24	45
13	5	10	22	43
14	6	10	18	59
15	7	3	9	13
16	8	10	21	40
17	9	12	23	63
18	3	8	18	22
19	4	4	9	18
20	10	12	25	33

5. SIMULATION RESULTS

The behavior of three task consolidation heuristic with 1000 tasks has been simulated here. The tasks are bought out for different groups of VMs with the use of incompatible ETC algorithm [6]. Matlab 2012 software has been used for simulation for 1000 tasks. The tasks arrive at the central server queue with a rate of λ having unlimited buffer size. It has taken the arrival interval between the tasks as 1 and the task arrival rate to be 30 in the present studies. The task consolidation algorithms behaviors are demonstrated for 10 and 15 VMs in Figures 5, 6 respectively. The consumption of energy on 15 VMs by varying the task size from 500 to 1500 has been shown in Figure 7.

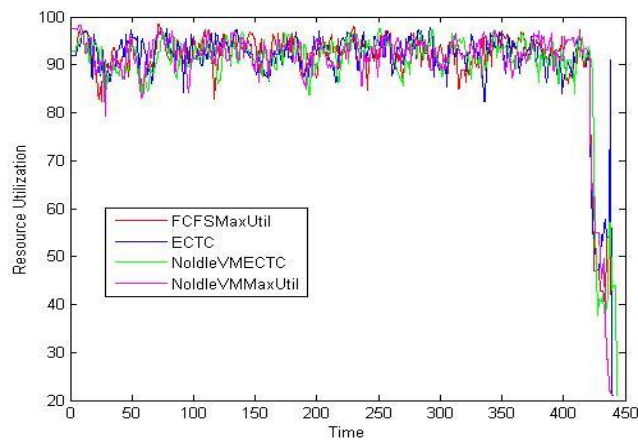


Fig. 5. CPU Utilization Comparison for 1000 Tasks on 10 VMs

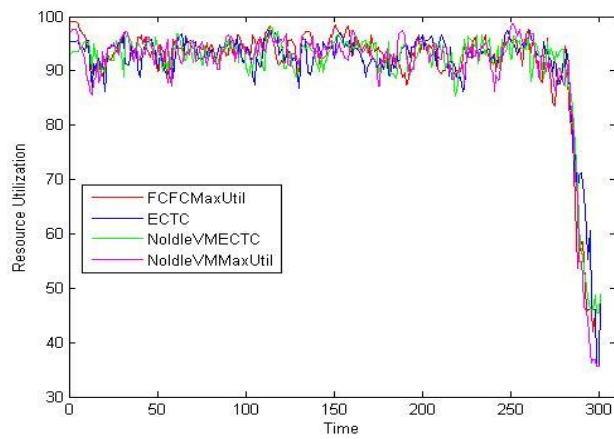


Figure 6. CPU Utilization Comparison for 1000 Taks on 10 VMs

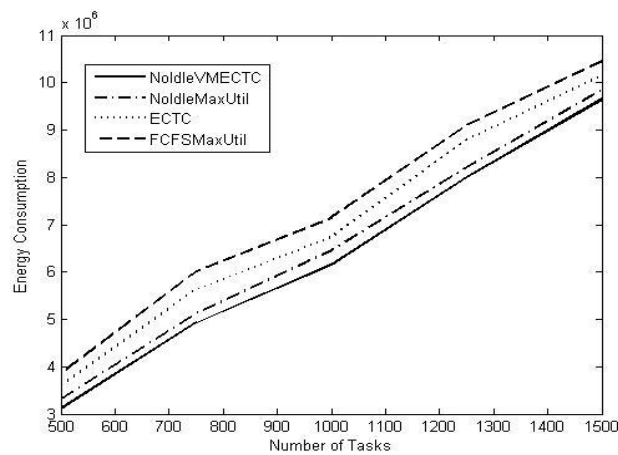


Figure 7. Energy Consumption for Number of Tasks on 15 VMs.

6. CONCLUSION

The simulation experiments have been successfully carried out which examines the behavior of heuristic task consolidation algorithms. It has also been optimized for energy consumption in a cloud

environment. The performance analysis has been demonstrated for the various task consolidation algorithms for the ETC matrix. The results drawn shows that the proposed algorithm has saved the energy as compared to the existing algorithms.

REFERENCES

- [1] D. Grigoras, "Advanced environments, tools, and applications for cluster computing," *NATO Advanced Research Workshop, IWCC 2001*, Mangalia, Romania, September 2001, *Lecture Notes in Computer Science*, 2326, Springer, 2002.
- [2] C. Lefurgy, X. Wang, and M. Ware, "Server-level power control," in *Proc 4th IEEE International Conference on Autonomic Computing (ICAC '07)*, Jacksonville, FL, USA, June 2007, pp. 4.
- [3] L. Barroso and U. Holzle, "The case for energy-proportional computing," *IEEE Computer*, 40(12), 2007, pp. 33-37.
- [4] P. Bohrer, E. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, and R. Rajamony, "The case for power management in web servers," *Power Aware Computing*, Kluwer Academic Publishers, 2002, pp. 261-289.
- [5] X. Fan, X.-D. Weber, and L.A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proc 34th annual International Symposium on Computer Architecture (ISCA '07)*, 2007, pp 13-23.
- [6] Wen, Gaojin, Jue Hong, Chengzhong Xu, Pavan Balaji, Shengzhong Feng, and Pingchuang Jiang. "Energy-aware hierarchical scheduling of applications in large scale data centers." *International Conference on Cloud and Service Computing (CSC) 2011*, pp. 158-165. IEEE, 2011.
- [7] Kim, Kyong Hoon, Rajkumar Buyya, and Jong Kim. "Power Aware Scheduling of Bag-of-Tasks Applications with Deadline Constraints on DVS-enabled Clusters." In *CCGRID*, vol. 7, pp. 541-548. 2007.
- [8] Lee, Young Choon, and Albert Y. Zomaya. "Energy efficient utilization of resources in cloud computing systems." *The Journal of Supercomputing*, vol. 2, pp. 268-280, 2012.
- [9] Hsu, Ching-Hsien, Shih-Chang Chen, Chih-Chun Lee, Hsi-Ya Chang, Kuan-Chou Lai, Kuan-Ching Li, and Chunming Rong. "Energy-aware task consolidation technique for cloud computing." In *Cloud Computing Technology and Science (CloudCom)*, 2011 IEEE Third International Conference on, pp. 115-121, 2011.
- [10] Srikantaiah, Shekhar, Aman Kansal, and Feng Zhao. "Energy aware consolidation for cloud computing." In *Proceedings of the 2008 conference on Power aware computing and systems*, vol. 10, pp. 1-5. 2008.
- [11] Koomey, Jonathan. "Growth in data center electricity use 2005 to 2010." *A report by Analytical Press, The New York Times*, 2011.
- [12] Hsu, Ching-Hsien, Kenn D. Slagter, Shih-Chang Chen, and Yeh-Ching Chung. "Optimizing energy consumption with task consolidation in clouds." *Information Sciences*, vol. 258, pp. 452-462, 2014.
- [13] Awada Uchechukwu, Keqiu Li, Yanming Shen. "Energy consumption in Cloud Computing Data Centers." *International Journal of Cloud Computing and Services Sciences (IJ-CLOSER)*, vol. 3, No.3, pp. 145-162, June 2014.
- [14] Angli Liu, "Theoretical Analysis on Scale-down Aware Service Allocation in Cloud Storage Systems." *International Journal of Electrical and Computer Engineering (IJECE)*, vol.3, No.1, pp. 21-29, February 2013.

BIOGRAPHIES OF AUTHORS



Mahendra Kumar Gourisaria is presently working as an Assistant Professor in the School of Computer Engineering at KIIT University, Bhubaneswar, Odisha. He has received his Master degree in Computer Application from Indira Gandhi National Open University, New Delhi and M.Tech in Computer Science and Engineering from Biju Patnaik University of Technology - Rourkela. He is pursuing his Ph.D. from KIIT University. He has published around 10 research papers in different international journals and conferences of repute. His area of research includes Cloud Computing, Data Mining, Soft Computing and Internet and Web Technology. He is a member of IAENG, UACEE and life member of ISTE, CSI and ISCA.



Sudhansu Shekhat Patra is currently an Associate Professor in the School of Computer Application, KIIT University, Bhubaneswar, India. He received his Master degree in Computer Application from Motilal Nehru National Institute of Technology, Allahabad, India, M.Tech(Computer Science & Engg) from Utkal University, Bhubaneswar, India and Ph.D. in Computer Science from KIIT University, Bhubaneswar, India. His research interests include grid computing, Cloud Computing, Algorithms. He is a life member of Indian Society for Technical Education.



Pabitra Mohan Khilar is currently an Assistant Professor in the Department of Computer Engineering, NIT, Rourkela, India. He received his Master degree in Computer Science and Engineering from NIT Rourkela and Ph.D. in Computer Science and Engineering from IIT, Kharagpur, India. His research interests include Parellel and Distributed Processing, Fault-Tolerance Computing, Cluster and grid computing, Wireless Adhoc Networks, Performance Evaluation of Distributed Networks and Distributed Embedded System. He is a member of MIEEE, MIE, MCSI, MISTE, MOEC, MOITS, MIETE, MISCA.