International Journal of Grid and Distributed Computing
Vol. 10, No. 6 (2017), pp.1-10
http://dx.doi.org/10.14257/ijgdc.2017.10.6.01

Parallelism in CPU virtualization and Scheduling using Cuckoo Search Algorithm

Debabrata Sarddar[1], *Sougata Chakraborty[2] and Priyajit Sen[3]

[1]Assistant Professor, Department of Computer Science & Engineering, University of Kalyani, West Bengal, India
[2]Senior System Engineer at IBM India Private Limited, Kolkata, West Bengal, India
[3]Student of Master of Technology, Department of Computer Science & Engineering, University of Kalyani, West Bengal, India
[1]dsarddar1@gmail.com, [2]me.sougata.chakraborty@gmail.com, [3]priyajit91@gmail.com

Abstract

Network load is one of the major shortcomings of cloud computing environment. Sometimes, many light and heavy weighted processes hamper the speed of the network. To avoid the load of cloud network, we need to have a mechanism in which parallel execution of processes will occur so that the scheduling will be much more efficient and thus, throughput will increase. Network load can be minimized. For this, we are going to propose a new concept "Parallelism in CPU virtualization and scheduling using Cuckoo Search Algorithm" in which the processes will be allocated to the resources for their execution depending on the availability of efficiency of those resources. If any resource can accept many processes, then that resource can be allocated for the execution of those processes at the same time. Thus, resources can be utilized absolutely and this leads to parallelism in CPU virtualization and many processes can be served at any instance of time, throughput is maximized. At the same time, processes do not starve for a long time and resource utilization will be maximized.

Keywords: *CPU Virtualization, Parallelization, Cuckoo Search Algorithm, Bio inspired algorithm, Swarm scheduling algorithm, Evolutionary scheduling algorithm, fitness function*

1. Introduction

Bio inspired algorithms mainly deal with the behavior of different animals. They are classified into two types:

1.1. Swarm Scheduling Algorithm

Swarm scheduling algorithms deal with the behavior of ants, bees, cuckoos, bats etc. They are used to find their food, best nest, and they show other characteristics and those algorithms help us to solve different complex problems like process scheduling, optimization, resource management and fault tolerance etc.

1.2. Evolutionary Scheduling Algorithm

Evolutionary scheduling algorithm is based on the Darwin's theory of evolution. Example. Genetic algorithm, Genetic programming and Bee genetic algorithm. Genetic

ISSN: 2005-4262 IJGDC

algorithm and genetic programming have large scope in artificial intelligence, machine intelligence etc.

1.3. Cuckoo search Algorithm

Cuckoo search is a kind of bio inspired algorithm under the category of swarm scheduling algorithm. In this algorithm, scheduling is based on the behavior of cuckoo birds. Cuckoo birds lay their eggs in the nest of host bird very rapidly by choosing best suitable nest of any host bird and the host bird is not aware of the matter. If the host bird can identify cuckoo's cleverness, then it leaves the nest or throws cuckoo's eggs from the nest. Otherwise, if it is not understood by the host bird then cuckoo's eggs grow up and become birds. Thus, when they can fly with their own tiny wings they leave the host bird's nest and new cuckoo bird appears in host bird's nest. So, there is no need to create a nest by the cuckoo bird and therefore, both time and effort are minimized. The algorithm of cuckoo search is given below:

1.3.1. cuckoo_search (p, r)

```
    Step 1: Initialization
        Initialization of nests and random initial solution.

    Step 2: Evaluation
        Get the current best nest.

    Step 3: Loop construction
        While ( $f_{\min} > Max$ generation)
        Get the cuckoo value by random walk, if not replace it by Levy's flights.

    Step 4: Evaluation
        Evaluate the quality fitness $Fn_j$.
        Randomly choose nest among $n$, say $j$.

    Step 5: Condition
        If ( $Fn_i > Fn_j$ )
        Replace $j$ value by new solution.
        End

    Step 6: Solution construction
        Retain the best solution and nests.
        Rank the solution and nests to choose the best.
        Pass to next generation.
End while, else go to step 2.
```

1.4. CPU Virtualization

In CPU virtualization, many computers can run on a single physical machine.

1.5. Parallel Process Execution

In parallel process execution, more than one process is in execution in parallel.

1.6. Serial Process Execution

In serial process execution, one process is in execution at a time.

1.7. Process Scheduling

Process is a program in execution time. The processor is allocated to the processes by the different process scheduler for the execution. This process is known as process scheduling. The process scheduler is the part of the operating system that takes the responsibility to decide the process execution.

1.8. Discussion about Scheduling Algorithms

1.8.1. First Come First Serve Scheduling

In First Come first serve scheduling (FCFS), the process will appear first in the ready queue and that will be allocated for the execution first. Henceforth, the processes will be allocated as per the order of their coming into the ready queue.

1.8.2. Shortest Job First Scheduling

In Shortest Job First scheduling (SJF), the process with the lowest burst time will be allocated first to the processor and then as per the increasing order of burst time the processes will be allocated for their execution.

1.8.3. Round Robin Scheduling

It is a preemptive scheduling algorithm, where the CPU executes a process for a specific time called the time quantum and then leaves the process even if their complete execution is not over.

1.8.4. Throughput

It is the measure of amount of work completed or number of process executed per unit time.

1.8.5. Waiting Time

It is the measure of time for which a process waits in the ready queue for getting allocated to the CPU for its execution.

1.8.6. Response Time

It is the measure of time during the first response time for the execution of a process and completion time.

1.8.7. Turn Around Time:

It is the measure of time when the process arrives in the ready queue and leaves the ready queue.

1.8.8. Remaining Time:

It is the measure of time for which the burst time is remaining to complete the total execution of a process.

2. Related Work

In his paper “Bio inspired computing –A review of algorithms and scope of applications”, Arpan Kumar Kar has discussed about different existing bio inspired algorithms and their real-life applications.

In the paper “Computing with the collective intelligence of honeybees – A survey”, Anguluri Rajasekhar et al. have discussed a comprehensive survey of the entire horizon of research so far undertaken on the algorithms inspired by the honey bees.

In the paper “Insect sensory systems inspired computing and communications”, Zhanshan (Sam) Ma *et al.*, have proposed a mechanism to introduce the state-of-the art research in insect sensory systems from entomological perspectives, to propose potential new research problems inspired by insect sensory system focusing on unexplored fields and to justify how and why insect sensory systems may inspire novel computing and communications paradigms.

In the paper “A Novel Family Genetic Approach for Virtual Machine Allocation”, Christina Terese Joseph *et al.*, have proposed a novel technique to allocate virtual machines using the Family Gene approach to prove the reduction of energy consumption and the rate of migrations.

In the paper “A comprehensive review of cuckoo search: variants and hybrids”, Iztok Fister Jr. et al. have discussed a timely review of all the state-of-the-art developments in the last five years, including the discussions of theoretical background and research directions for future development of cuckoo search algorithm.

In the paper “Task Scheduling Using PSO Algorithm in Cloud Computing Environments”, Ali Al-maamari *et. al.*, have proposed a Dynamic Adaptive Particle Swarm Optimization algorithm (DAPSO) has been implemented to enhance the performance of the basic PSO algorithm to optimize the task runtime by minimizing the make span of a task set, and in the same time, maximizing resource utilization.

In the paper “Cluster Based Bee Algorithm for Virtual Machine Placement in Cloud Data Centre”, Ajith Singh *et al.*, have proposed a technique for solving Virtual machine placement problem is implemented with the Honey Bee algorithm with hierarchical clustering to minimize energy consumption in servers.

In the paper “Task Scheduling in the Cloud Computing Based on the Cuckoo Search Algorithm”, Nima Jafari *et al.*, have proposed a new evolutionary algorithm which implement Cuckoo Search Algorithm to schedule the tasks in Cloud computing.

In the paper “A Brief Review of Nature-Inspired Algorithms for Optimization”, Iztok Fister Jr. et al. have gone through a review study to present a relatively comprehensive list of all the nature inspired algorithms in the literature to inspire further research.

In the paper “Bio-Inspired Scheduling of High Performance Computing Applications in Cloud: A Review”, Delsy Jyothi *et al.*, have gone through a survey on the recent advancement in biologically inspired swarm optimization methods including ant colony optimization algorithm, artificial bee colony algorithm, cuckoo search and particle swarm optimization algorithm.

In the paper “Task scheduling in Cloud Using Hybrid Cuckoo Algorithm”, Sumandeep Aujla et al. have proposed the optimization solution for the task scheduling using hybrid cuckoo algorithm.

In the paper “Multi-objective evolutionary algorithms: A survey of the state of the art”, Aimin Zhoua et al. have surveyed on the multi-objective evolutionary algorithms vividly.

In the paper “Swarm intelligence inspired classifiers for facial recognition”, Salima Nebti and Abdallah Boukerram have analyzed and designed the classifiers based on swarm intelligence.

3. Proposed Work

We are going to propose a novel approach to show the parallel execution of processes in virtual machines. Existing algorithms can give us the idea of executing different processes but not simultaneously. In the existing scheduling algorithms, once a process captures any resource for execution then at a time, only one process can be executed even in Round Robin scheduling also. But our idea is to apply cuckoo search algorithm for the execution of processes in parallel and scheduling can be done in such a way that a process can be fit within a resource. If the remaining space of the resource is sufficient to allocate another process, then the process will be allocated to the resource. Our proposal is about scheduling and here resource is the processor (CPU). The processor might be allocated to more than one process at once depending on its efficiency. It is also a non-preemptive process scheduling algorithm. We are going to illustrate our proposed method like the cuckoo lay eggs in crow's nest and the crow is not aware of the cuckoo's cleverness. When the eggs get broken and tiny cuckoo birds come out from the eggs, they fly away from crow's nest. Thus, the cuckoo bird does not need to create its own nest and the effort needed to make nests will not be required at all. Our thinking is in this bio-inspired approach, the host and the parasite are the two different machines. The host is a remote machine. The parasite can execute a task even if the required infrastructure is not available in parasite machine. The parasite machine will acquire the space and infrastructure if available after the allocation of the host's own task. Then parasite can execute its work without asking the permission from host machine. If it is caught by the crow, then instantly the eggs will be thrown from the nest and thus in our approach the connection can be broken or lost by the host. The algorithm, given below, will illustrate it more specifically.

3.1.1. Proposed Algorithm:

Step 1: Input n processes $p_1, p_2, p_3, p_4 \ldots, p_n$ along with their job size or burst time represented by $burst(p_i)$.

Step 2: Input m number of available resources $r_1, r_2, r_3, \ldots, r_m$ along with their efficiencies or working capability represented $eff(r_i)$ to be allocated for the execution of n processes.

Step 3: Initially the process (p_i) having lowest burst time among all the processes will be allocated to the available resource.

Step 4: $for\ i = 1\ to\ n$
    Do the following steps:

Step 5: We represent the remaining efficiency of resources by $rem(r_i)$ then $rem(r_i) = eff(r_i) - burst(p_i)$.

Step 6: Call $cuckoo\_search(p, r)$ and check the fitness of the host so that parasite can allocate its task in its remaining efficiency.

Fitness can be achieved by using following formula:

$f_i = f_{min} + \left(f_{max} \ ? \ f_{min} \right)\beta$, where $\beta \in [0,1]$- a random vector drawn from a uniform distribution.

```
If the current resource is full then go to step 5.
    Else
    $for\ j = 1\ to\ n\ and\ j\,! = i$
    $check\ if\ the\ eff\left(p_j\right) <= rem\left(r_j\right)$
    then
    $p_j\ will\ be\ allocated\ to\ the\ same\ resource\left(r_i\right).$
```

Step 7: End

Our proposed algorithm deals with the CPU virtualization and process scheduling. By the algorithm, there are *n* number of processes with their burst time or job size and there are limited numbers of resources. The challenging task is to allocate processes over the resources by checking the remaining efficiency of the resources. First, the host machine allocates its job to the currently available resources then if the resource is capable to accept more tasks then parasite machines will try to allocate their tasks by checking the burst time of their processes and remaining resource's efficiency. The fitness of the resources will be checked from the formula given below:

$f_i = f_{min} + \left(f_{max} \ ? \ f_{min} \right)\beta$, where $\beta \in [0,1]$- a random vector drawn from a uniform distribution. For the simplicity, we may assume $f_{min} = 0\ and\ f_{max} = 1$. β being a random vector can be considered as 1 which gives always the maximum fitness of the nest. By checking the value of f_i, other processes, having job size less than f_i, will be allocated to the current resource. This way, parallelism can be achieved and this leads to requirement of less time for the execution of all processes of the cloud network.

4. Experimental Results

Table 1. FIFO Scheduling

Process	**Arrival Time (*ms*)**	**Burst Time (*ms*)**	**Waiting Time (*ms*)**	**Average Waiting Time (*ms*)**
P1	0	8	0	35/4=8.75
P2	1	4	7	
P3	2	9	10	
P4	3	5	18	

Table 2. SJF Scheduling

Process	**Arrival Time (*ms*)**	**Burst Time (*ms*)**	**Waiting Time (*ms*)**	**Average Waiting Time (*ms*)**
P1	0	8	0	31/4=7.75
P2	1	4	7	
P3	2	9	15	
P4	3	5	9	

Table 3. Round Robin Scheduling

Process	Arrival Time (*ms*)	Burst Time (*ms*)	Waiting Time (*ms*)	Average Waiting Time(*ms*)
P1	0	8	15	54/4=13.5 Considering, Time Quantum (Δ_t)=3 *ms*
P2	1	4	11	
P3	2	9	15	
P4	3	5	13	

Table 4. Proposed Scheduling Algorithm (Parallelism using Cuckoo Search)

Host Process Arrival Time (*ms*)	Host Process Burst Time (*ms*)	Parasite Process Arrival Time (*ms*)	Parasite process Burst Time (*ms*)	No. of Resources	Efficiency of resources
0	8	2	6	R1	10
1	4				
2	9	4	8	R2	8
3	5				

Average Waiting time for Host Processes= 5.25 *ms*.
Average Waiting time for Parasite Processes= 4.35 *ms*.

It is parallel execution of processes so the maximum waiting time will be the actual waiting time because all the processes are executed in parallel. Here we can see that the average waiting time of host processes is 5.25 *ms* and the average waiting time of parasite processes is 4.35 *ms*. But, as the processes are executed simultaneously then this may happen that when any host process is waiting for its allocation, the parasite process may execute its job. Thus, average waiting time is 5.25 *ms*.

The comparison of average waiting time is shown in the following figure:

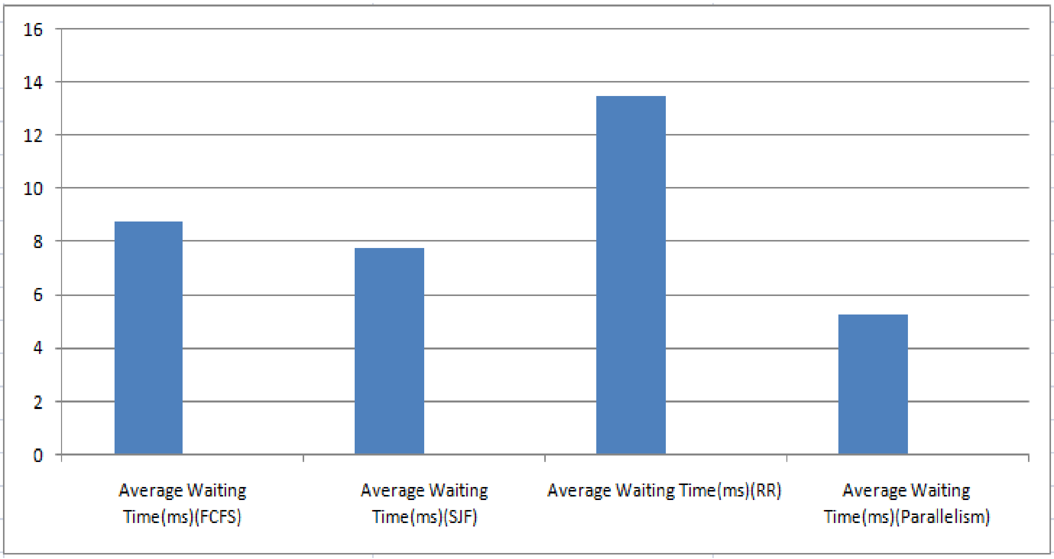

Figure 1. Comparison of Average Waiting Time

5. Conclusion

From the above result and analysis, we have come to know that due to parallelism in process scheduling, many processes can be executed in a resource. So, resource utilization is improved and at the same time average waiting time for processes will be minimized than other scheduling algorithms. Our proposed approach will be fruitful to serve in daily business days when many jobs or processes usually come to the server and due to the limited number of resources, processes need to wait for a long time. This may also happen that resources are not completely utilized. At that time, though some resources are not full at all, still no process can be allocated in resource. Our proposed approach can eliminate these problems.

References

[1] Arpan Kumar Kar, “Bio inspired computing –A review of algorithms and scope of applications”, Volume 59, 15 October 2016, pp.20–32.

[2] Anguluri Rajasekhar, NandarLynn, SwagatamDas, P.N.Suganthan, “Computing with the collective intelligence of honeybees – A survey”, Swarm and Evolutionary Computation, Volume 32, February 2017, pp.25–48.

[3] Zhanshan (Sam) Ma, Axel W. Krings, “Insect sensory systems inspired computing and communications”, Ad Hoc Networks, Volume 7, Issue 4, June 2009, pp.742–755.

[4] Christina Terese Joseph, Chandrasekaran K, Robin Cyriac, “A Novel Family Genetic Approach for Virtual Machine Allocation”, Procedia Computer Science, Volume 46, 2015, pp.558-565.

[5] Iztok Fister Jr., Dušan Fister and Iztok Fister, “A comprehensive review of cuckoo search: variants and hybrids”, Int. J. Mathematical Modelling and Numerical Optimisation, Vol. 4, No. 4, 2013.

[6] Ali Al-maamari and Fatma A. Omara, “Task Scheduling Using PSO Algorithm in Cloud Computing Environments”, International Journal of Grid and Distribution Computing Vol. 8, No.5, (2015), pp.245-256.

[7] Ajith Singh. N and M. Hemalatha, “Cluster Based Bee Algorithm for Virtual Machine Placement in Cloud Data Centre”, Journal of Theoretical and Applied Information Technology 30th November 2013. Vol. 57 No.3.

[8] Nima Jafari Navimipour and Farnaz Sharifi Milani, “Task Scheduling in the Cloud Computing Based on the Cuckoo Search Algorithm”, International Journal of Modeling and Optimization, Vol. 5, No. 1, February 2015.

[9] Iztok Fister Jr., Xin-She Yang, Iztok Fister, Janez Brest, Duˇsan Fister, “A Brief Review of Nature-Inspired Algorithms for Optimization”, ELEKTROTEHNIˇSKI VESTNIK 80(3): 1–7, 2013, ENGLISH EDITION.

[10] Delsy Jyothi, Anoop S, “Bio-Inspired Scheduling of High Performance Computing Applications in Cloud: A Review”, International Journal of Computer Science and Information Technologies, Vol. 6 (1), 2015, pp.485-487.

[11] Sumandeep Aujla, Amandeep Ummat, “Task scheduling in Cloud Using Hybrid Cuckoo Algorithm”, International Journal of Computer Networks and Applications (IJCNA) Volume 2, Issue 3, May – June (2015).

[12] Aimin Zhoua, Bo-Yang Qub, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, Qingfu Zhangd, “Multiobjective evolutionary algorithms: A survey of the state of the art”, Swarm and Evolutionary Computation 1 (2011) pp.32–49.

[13] Salima Nebti, Abdallah Boukerram, “Swarm intelligence inspired classifiers for facial recognition”, Swarm and Evolutionary Computation 32(2017) pp.150–166.

Authors

Debabrata Sarddar, Assistant Professor in the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, INDIA. He has done Ph.D. at Jadavpur University. He completed his M. Tech in Computer Science & Engineering from DAVV, Indore in 2006, and his B.E in Computer Science & Engineering from NIT, Durgapur in 2001. He has published around 200 research papers in different journals and conferences. His research interest includes wireless and mobile system, Cloud Computing and WSN.

Sougata Chakraborty is a Senior System Engineer at IBM India Private Limited in Kolkata. He completed M. Tech in Computer Science & Engineering from Jadavpur University in 2011. He had also completed his B. Tech in Information Technology from Murshidabad College of Engineering & Technology under West Bengal University of Technology in 2008. His research interests include Cloud Computing and Mobile Computing.

Priyajit Sen is presently pursuing M. Tech in Computer Science and Engineering at the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, India. He has completed his MCA from Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, India in 2015. His research interest includes Mobile Computing, Wireless Sensor Network and Cloud Computing.