IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 5, Oct-Nov, 2013
ISSN: 2320 - 8791
www.ijreat.org

Implementation of Honey Bee Algorithm Through Cloud Computing

K.Ganeshan[1], S.Srinivasalou[2] and S.Kousick[3]

[1]**Computer Science and Engineering, Sri Aravindar Engineering College,India**

2**Computer Science and Engineering, Sri Aravindar Engineering College,India**

[3]**Computer Science and Engineering, Sri Aravindar Engineering College,India**

Abstract

This paper describes a population-based approach that uses a honey bees foraging model to solve job shop scheduling problems. The algorithm applies an efficient neighborhood structure to search for feasible solutions and iteratively improve on prior solutions. The initial solutions are generated using a set of priority dispatching rules. Experimental results comparing the proposed honey **bee colony**[1] approach witha survey of current research activities inspired by bee life. This work is intended to provide a broad and comprehensive view of the various principles and applications of these bio-inspired systems with existing approaches such as ant colony, tabu search and shifting bottleneck procedure on a set of job shop problems are presented. The results indicates the performance of thebee researches proposed in the paper are unified in the aim to contribute in their organization and to facilitate their application for solving other problems typically in the proposed approach is comparable to other efficientscheduling approaches.

Keywords—*Scheduling, honey bee colony, neighborhood search.*

1. INTRODUCTION

The cloud computing is a recent field in the computational intelligence techniques which aims at surmounting the computational complexity and provides dynamically services using very large scalable and virtualized resources over the Internet. It is defined as a distributed system containing a collection of computing and communication resources located in distributed datacenters which are shared by several endusers. There are two kinds of the cloud; the former is the public cloud in which services may be sold to anyone on the Internet. Here, Amazon Elastic Compute Cloud (EC2) , Google App Engine are large public cloud providers. The second type of the cloud is the private cloud. It is a proprietary network or a Dc datacenter that supplies hosted services to a limited number of clients (end-users). Cloud computing allows users to run applications remotely including information technology services called Software-as-a-Service (SaaS) . Besides, Infrastructure-as-a-Service (IaaS) refers to where the application is carried out via useful functions and services provided in the cloud. Between SaaS and IaaS, there are cloud platform services known as Platform-as-a-Service (PaaS) deliver operating system, programming language and solution stack. It aims to improve the applications deployment cost and reduce its complexity. It can be seen that the cloud resource use can be simplified and efficiency operated by the scheduled services use in an “optimal solution”[2]in the interest of the endusers.

To take up this challenge, we propose in this paper an efficient algorithm called Bees Life Algorithm (BLA) inspired by the bees’ colony life. It aims at an optimal job scheduling by assigning end-user tasks to the relevant datacenters in an optimal way.

2. RELATED WORK

2.1 Job scheduling in cloud computing:

Cloud computing is a latest new computing paradigm where applications, data and IT services are provided across dynamic and geographically dispersed organization. How to improve the global throughput and utilize Cloud computing resources proficiently and gain the maximum profits with job scheduling system is one of the Cloud computing service providers' ultimate objectives. The motivation of this paper is to establish a scheduling mechanism which follows the Lexi - search approach to find an optimal feasible assignment. Task scheduling has been treated as general assignment problem to find the minimal cost.

In the last years, the job scheduling problem in the cloud computing has been studied in few research activities because of the novelty of this research discipline however; we can mention the following approaches. As related work, we mention the work proposed in which developed a load Balancing framework for high-performance clustered storage systems. It provides a general method for reconfiguring a system facing dynamic workload changes.In, authors explored a novel approach calledRhizoma) similar to P2P systems in which the cloud application to be managed should expresses its resource requirements to the runtime as a constrained optimization problem. Then, Rhizoma fuses multiple real-time sources of resource availability data, from which it decides to acquire or release resources. Rhizoma results showed its performance. Nevertheless, this approach can be enhanced to run across multiple clusters and commercial utility computing providers. Authors of proposed to use statistical models to predict resource requirements for Cloud computing applications. Such a prediction proposal can help system design to reach scalability, job scheduling, resource allocation, and workload management. Also, they proposed an initial design of a workload generator in order to evaluate alternative configurations without the overhead of reproducing a real workload. An effective prediction model is a prerequisite for a good workload generator. The purpose of is to propose an approach for the storage and “analysis of very-large images on the cloud”[3][4][5]**.** This service has been implemented using multiple distributed and collaborative agents.

As the inspiration idea is the bees’ life in nature, we can mention some works in this context. In a population-based metaheuristic inspired by the natural foraging behavior of honey bees called Bees Algorithm. It starts with a number of scout bees being placed randomly in the search space. The best sites visited in point of view fitness are chosen as selected bees which will be foraged their close sites to carry out a neighborhood search.

Another algorithm based on the reproduction behavior principle in the bee colony has been proposed in. It is called marriage in honey bees optimization algorithm. It starts with initializing the queen’s genotype at random and then a heuristic is applied to improve the queen’s genotype, therefore preserving the assumption that a queen is usually a good bee. Next, a set of mating–flights is undertaken relatively to the queen’s energy and speed. The queen then moves between different states (solutions) in the space and mates with the encountered drones according to probability criterion. After laying, the queen is replaced with the fittest brood and so on.

2.2 Problem statement and notations

Job Scheduling aims at assigning jobs to datacenters in the cloud so that the execution time (makespan) of the overall tasks of jobs is minimized. This problem can be formulated as follows.

Denoted by:

$$Jobs = \{j_1, j_2, j_3, \dots, j_n\}$$

a set of ‘n’ jobs to be scheduled. Moreover, we consider for each job ‘i’:

$j_iTasks = \{jTask_{i1}, jTask_{i2}, \dots, jTask_{in}\}$, as a set of ‘m’task partitions of Jobi disseminated among ‘m’ cloud datacenters (DCs) in order to be executed. Consequently, each cloud datacenter can carries out a disjoint subset of the decomposed jobs set. For its assigned jobs, DCj ensures the execution of their tasks as follows:

$$DC_jTasks = \{jTask_{aj}, jTask_{bj}, \dots, jTask_{rj}\}$$

The union of these overall disjoint subsets gives the whole set of jobs. For example, DCj carries out:

$DC_jTasks = \{jTask_{3j}, jTask_{6j}, \dots, jTask_{9j}\}$which are tasks of jobs J_3, j_6 …, and $j_9$respectively.

Therefore, the total execution time of all job tasks (‘r’).tasks) assigned to DCj would be:

$$Makespan(DC_jTasks) = Max(jTask_{kj}.StartTime + jTask_{kj}, ExeTime)$$

Where $jTask_{kj}.StartTime$ is the time when job task 'k' $jTask_{kj}$ starts executing on DCi and $jTask_{kj}, ExeTime$ is the execution time of $jTask_{kj}$ at DC_j.

Thus, the job scheduling problem in the cloud computing could be defined as searching of a set:

$$DCTasks \;\; \{DC_1Tasks, DC_2Tasks, \ldots, DC_mTasks\}$$

and:

$$DC_jTasks = \{jTask_{aj}, jTask_{bj}, \ldots, jTask_{rj}\}$$

with $0 < r \leq n$

which reduces: *Makespan(DCjTasks)*

In order to evaluate the quality of the requested solution (DCTasks), a fitness function is defined as follows (used to calculate the above makespan):

$$Fitness(DCTasks) = \sum (Fitness(jTask_{ij}, DC_j))$$

and

$$Fitness(jTask_{ij}, DC_j) = jTask_{ij}.TimetoExe$$

where, $jTask_{ij}.TimetoExe$ is the execution time of task of job 'i' needs to run in DCj.

3. BEES LIFE ALGORITHM

3.1 Nature of bees

A colony of honey bees can extend itself over long distances (up to 14 km) and in multiple directions simultaneously to exploit a large number of food sources. A colony prospers by deploying its foragers to good fields. In principle, flower patches with plentiful amounts of nectar or pollen that can be collected with less effort should be visited by more bees, whereas patches with less nectar or pollen should receive fewer bees.

The foraging process begins in a colony by scout bees being sent to search for promising flower patches. Scout bees move randomly from one patch to another. During the harvesting season, a colony continues its exploration, keeping a percentage of the population as scout bees.

When they return to the hive, those scout bees that found a patch which is rated above a certain quality threshold (measured as a combination of some constituents, such as sugar content) deposit their nectar or pollen and go to the "dance floor" to perform a dance known as the waggle dance.

3.2 BLA-Job Scheduling

Similarly in job scheduling the bee algorithm is explained to reduce the complexity of the job examined. The honey bees' effective foraging strategy can be applied to job shop scheduling problems and for solving "continuous problems"[3].

A feasible solution in a job shop scheduling problem is a complete schedule of operations specified in the problem. Each solution can be thought of as a path from the hive to the food source. The figure on the right illustrates such an analogy

The makespan of the solution is analogous to the profitability of the food source in terms of distance and sweetness of the nectar. Hence, the shorter the makespan, the higher the profitability of the solution path.

We can thus maintain a colony of bees, where each bee will traverse a potential solution path. Once a feasible solution is found, each bee will return to the hive to perform **waggle dance**. The waggle dance will be represented by a list of elite solutions, from which other bees can choose to follow another bee's path. Bees with a better makespan will have a higher probability of adding its path to the list of elite solutions, promoting a convergence to an optimal solution.

Using the above scheme, the natural honey bee'sself organizing foraging strategy can be applied to the job shop scheduling problem.

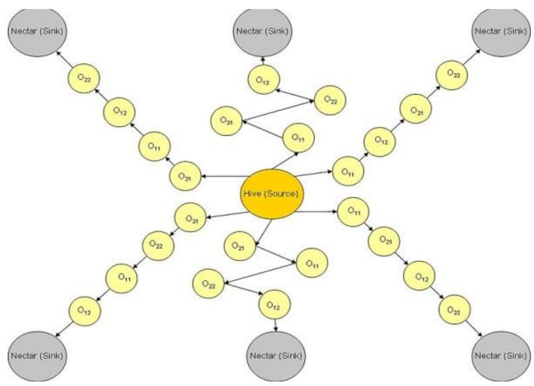

Fig - I ,Bee model enhancing job scheduling

3.3 Pseudo code for BLA

1. Initialise population with random solutions.
2. Evaluate fitness of the population
3. While (stopping criterion not met) //Forming new population.
4. Select sites for neighbourhood search.
5. Recruit bees for selected sites (more bees for best e sites) and evaluate fitnesses.
6. Select the fittest bee from each patch.
7. Assign remaining bees to search randomly and evaluate their fitnesses.
8. End While.

3.4 Flow Chart for BLA

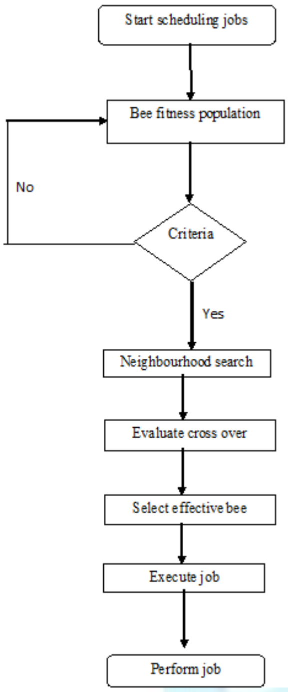

Fig 2, Flowchart representing honey bee algorithm

4. Optimization operators of BLA

To ensure the diversity in the generation of populations, two genetic operators are applied in the reproduction phase: crossover and mutation. Nevertheless, in the foraging phase, we define a greedy approach to ensure a neighborhood search in order to optimize the found individual among its neighbors. In the following sections, these operators are illustrated.

4.1 Crossover

It is a binary operator in which the queen mates with one drone randomly chosen. This process is repeated until reaching 'N' individuals with crossover probability of 'Pc'. In this paper, we applied two-point crossover as shown in figure 3. Parents are splitting at these two points chosen randomly to create children after exchanging the extremities with correction if children do not belongs to the search space.

Before crossover: Parents

$$P_i = \{< jTask_{ai,}jTask_{bi,}jTask_{ci,} \dots, jTask_{ri} >\}$$

$$P_j = \{< jTask_{Aj,}jTask_{Bj,}jTask_{Cj,} \dots, jTask_{Rj} >\}$$

After crossover: Children

$$P_i = \{< jTask_{kai,}jTask_{kbi}, \dots, jTask_{kri} >\}$$

$$P_j = \{< jTask_{kAj,}jTask_{kBj,} \dots, jTask_{kRj} >\}$$

4.2 Mutation

It is a unitary operator in which the child has the chance to be mutated according to the mutation probability of 'Pm'. We propose the mutation operator in which a random selected task will be chosen and replaced by another random selected task in the same datacenter as shown in figure 4 and so on for the other datacenters. If the new individual do not belongs to the search space, a correction process is carried out.

Child before mutation:

$$C_i = \{< jTask_{ai,}jTask_{bi,}jTask_{ci,} \dots, jTask_{kri} >\}$$

Child after mutation:

$$C_i = \{< jTask_{kai,}jTask_{kbi,} \dots, jTask_{kri} >\}$$

4.3 BLA complexity analysis

To analyze the computational resources running time, the performed complexity of BLA is calculated as follows. First of all, double O(N) time units are used to choose and to evaluate N bees which represent the population individuals. It means that the initialization performs O(N) + O(N) ≈ O(N) time units.

The total time complexity of each iteration can be calculated as follows:

$$O(N) + O\big((N + P_c) \times P_m\big) + O(FBest \times B) + O(FOthers \times (W - B))$$

We mention the BLA performs:

In the best case:

$$O(NS \times ((N + P_c) \times P_m) + (FBest \times B) + (FOthers \times (W - B)))$$

In the worst case:

$$O(ItMax \times ((N + P_c) \times P_m) + (FBest \times B) + (FOthers \times (W - B)))$$

It isconcluded that BLA performs a linear complexity which is the least possible.

5. CONCLUSION

In this paper, job scheduling problem in the cloud computing has been studied and solved with a novel population-based metaheuristic called Bees Life Algorithm which is inspired by nature life of bees. The makespan is the major objective aimed at such problem in this scalable computing area. In order to prove the reliability and the efficiency of this proposal, an implementation and a set of tests of BLA have been carried out and compared against GA asconventional algorithm in this context. The obtained results of BLA prove the efficiency and the performance of the proposed algorithm in terms of the execution time.

As perspective, the dynamic job scheduling in the cloud computing can be studied using BLA in which number of datacenters can be varied. In addition, another aspect can be treated such as the execution of jobs in real time.

References

[1] Chong C, Low MYH, Sivakumar AI and Gay KL (2006) "A Bee Colony Optimization Algorithm to Job Shop Scheduling" Simulation Conference, WSC 06. Monterey,

[2] CAPham D.T., Ghanbarzadeh A., Koç E., Otri S., Rahim S., and M.Zaidi "*The Bees Algorithm – A Novel Tool for Complex Optimisation Problems*"", Proceedings of IPROMS 2006 Conference, pp.454–461.

[3] Tovey, "The Honey Bee Algorithm: A Biological Inspired Approach to Internet Server Optimization", Engineering Enterprise, the Alumni Magazine for ISyE at Georgia Institute of Technology,

[4] 2004Yang X.S., "Engineering Optimizations Via Nature-Inspired Virtual Bee Algorithms". Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, LECTURE NOTES IN COMPUTER SCIENCE 3562:2005, pp. 317-323 , Springer Berlin / Heidelberg.

[5] Olague, G.; Puente, C. Honeybees as an Intelligent based Approach for 3D Reconstruction. ICPR 2006. 18th International Conference on Pattern Recognition. 2006, Volume 1, Issue , 0-0 0 Page(s):1116–1119.doi:10.1109/ICPR.2006.632

[6] Olague, G. Puente, C.. The Honeybee Search Algorithm for Three-Dimensional Reconstruction. 8 th European Workshop on EvolutionaryComputation in Image Analysis and Signal Processing. Lecture Notes in Computer Science 3907. Springer-Verlag. pp. 427–437. EvoIASP2006.BestPaper Award for EvoIASP 2006. doi:10.1007/11732242, ISBN 978-3-540-33237

[7] Tovey, "The Honey Bee Algorithm: A Biological Inspired Approach to Internet Server Optimization", Engineering Enterprise, the Alumni Magazine for ISyE at Georgia Institute of Technology, 2004.

[8] Karaboga, "An idea based on honey bee swarm for numerical optimization", Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.