Volume 6, Issue 12, December 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Survey on Energy Aware Metaheuristic Technique in Cloud Environment

Mukesh Kumar
Research Scholar (M.Tech), Dept. of CSE,
Guru Kashi University, T.Sabo, Punjab, India

Harpal Singh
AS & HOD Deptt. of Compter Applications,
Guru Kashi University, T.Sabo, Punjab, India

Abstract— New technological breakthroughs and massive production provide cheap and easy-to-use products that are more accession to the average person, which leads to worldwide usage of emerging technologies. One of the main enablers of technological progress and modern civilization more generally is the energy that drives this machinery. However, due to its global usage, technological machinery creates an increasing demand for more energy. In this paper we will discuss the energy aware metaheuistic techniques in cloud environment.

Keywords— ACO, PSO, GA, Cloud Computing

I. INTRODUCTION

Cloud computing can be classified as a new paradigm for the dynamic provisioning of computing services supported by state-of-the-art data centers that usually employ Virtual Machine (VM) technologies for consolidation and environment isolation purposes [1]. In industry these services are referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) respectively. A recent Berkeley report [2] stated: ‘‘Cloud computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service’’. Therefore, Cloud service providers need to adopt measures to ensure that their profit margin is not dramatically reduced due to high energy costs. The rising energy cost is a highly potential threat as it increases the Total Cost of Ownership (TCO) and reduces the Return on Investment (ROI) of Cloud infrastructures.

Figure 1: Cloud Computing [14]

Nowadays energy consumption problem is a major issue for data centers. The energy consumption increases significantly along with its CPU frequency getting higher. With Dynamic Voltage and Frequency Scaling (DVFS) techniques, CPU could be set to a suitable working frequency during the running time according to the workload. On the other side, reducing frequency implies that more servers will be utilized to handle the given workload. It is a critical problem to make a trade-off between the number of servers and the frequency of each server for current workload.

In this paper, we investigate energy aware metaheuristic technique.

II. LITERATURE SURVEY

Various papers have been studies. Main papers are followings:

In Xin-She Yanget al [2013]: This paper provides a timely review of the bat algorithm and its new variants. A wide range of diverse applications and case studies are also reviewed and summarized briefly here. Further research topics are also discussed. Bat algorithm (BA) is a bio-inspired algorithm developed by Yang in 2010 and BA has been found to be very efficient. As a result, the literature has expanded significantly in the last 3 years.

In Qi Zhang et al [2012]: In this paper, they provide a control-theoretic solution to the dynamic capacity provisioning problem that minimizes the total energy cost while meeting the performance objective in terms of task scheduling delay. Specifically, they model this problem as a constrained discrete-time optimal control problem, and use Model Predictive Control (MPC) to find the optimal control policy.

In Dian PalupiRini et al [2011]: In this paper, they provide a Basic Information about Particle Swarm Optimization (PSO) is a biologically inspired computational search and optimization method developed in 1995 by Eberhart and Kennedybased on the social behaviors of birds flocking or fish schooling. Found by the PSO.

III. ENERGY EFFICIENCY

Energy efficiency refers to a reduction of energy used for a given service or level of activity, as defined by the World Energy Council. However, defining the energy efficiency of data center equipment is extremely difficult because it represents a complex system with a large number of components from various research areas such as computing, networking, management, and the like.

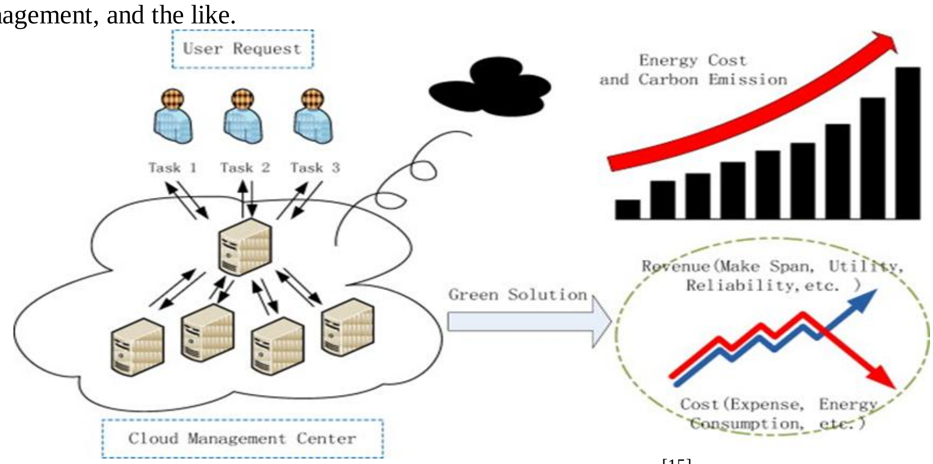

Figure 2: Energy Efficiency Model [15]

It is known that one of the most effective approach for reducing energy cost is to dynamically adjust the data center capacity by turning off unused machines, or to set them to a power-saving (e.g. “sleep”) state [3]. In a production data center where resource requests for tasks can arrive dynamically over time, deciding the number of machines to be switched off is not only affected by the efficiency of the scheduling algorithm, but also time-dependent characteristics of resource demand [4]. While over-provisioning the data center capacity can lead to sub-optimal energy savings, under-provisioning the data center capacity can cause significant performance penalty in terms of scheduling delay, which is the time a task has to wait before it is scheduled on a machine [5].

Let’s identify two critical points where energy is not used in an efficient way but is instead lost or wasted. Both terms define inefficient use of energy from an agnostic point of view, where energy loss refers to energy brought to the system but not consumed for its main task (e.g., energy lost due to transport and conversion) [7]. This also includes energy used by supporting subsystems, such as cooling or lighting within a data center whose main task is the provision of cloud services. Energy waste refers to energy used by the system’s main task but without useful output (e.g., energy used while running in idle mode). Additionally, useless work by the system is also considered energy waste; for example, for a cooling subsystem, this would mean keeping the cooling at maximum during the night when temperatures are lower [5].

IV. CLASSIFICATION OF TASK SCHEDULING ALGORITHMS

Various algorithms are available for managing the energy in cloud environment. Figure 3 will show the various scheduling algorithms.

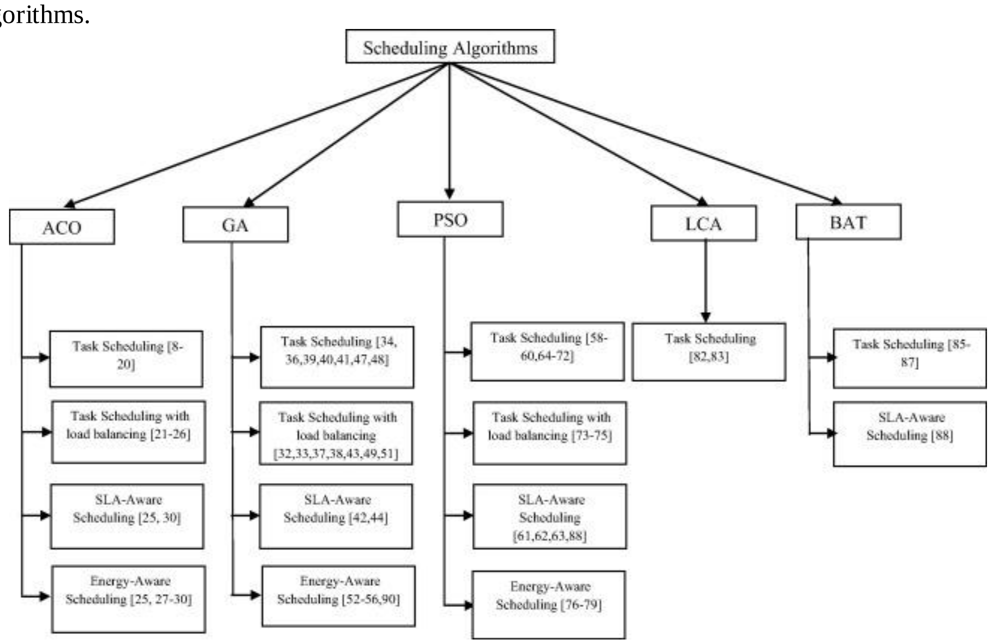

Figure 3: Various Scheduling Algorithms

Various Metaheurstics Algorithms:

Ant Colony Optimization ACO: Ant algorithms were first proposed by Dorigo and colleagues [7 9] as a multi-agent approach to difficult combinatorial optimization problems like the traveling salesman problem (TSP) and the quadratic assignment problem (QAP). There is currently a lot of on-going activity in the scientific community to extend/apply ant-based algorithms to many different discrete optimization problems [5, 11]. Recent applications cover problems like vehicle routing, sequential ordering, graph colouring, routing in communications networks, and so on.

Ant algorithms were inspired by the observation of real ant colonies. Ants are social insects, that is, insects that live in colonies and whose behaviour is directed more to the survival of the colony as a whole than to that of a single individual component of the colony. Social insects have captured the attention of many scientists because of the high structuration level their colonies can achieve, especially when compared to the relative simplicity of the colony's individuals. An important and interesting behaviour of ant colonies is their foraging behaviour, and, in particular, how ants can find shortest paths between food sources and their nest. While walking from food sources to the nest and vice versa, ants deposit on the ground a substance called pheromone, forming in this way a pheromone trail. Ants can smell pheromone and, when choosing their way, they tend to choose, in probability, paths marked by strong pheromone concentrations. The pheromone trail allows the ants to find their way back to the food source (or to the nest). Also, it can be used by other ants to find the location of the food sources found by their nest mates

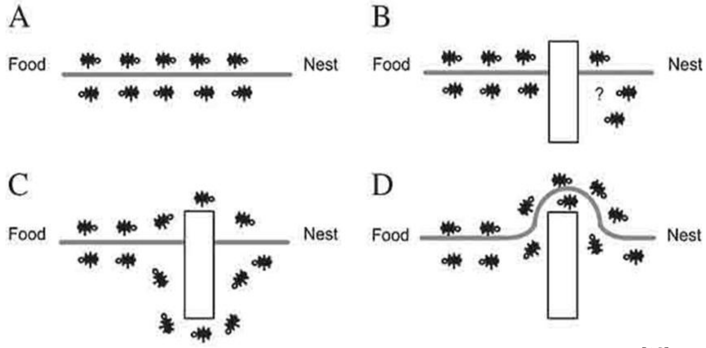

Figure 4: Finding by ants path from nest to food source [16]

Procedure ACO

1. **Initialization:**
   i. Initialize the pheromone value to a positive constant for each path between tasks and resources.
   ii. Optimal solution=null
   iii. Place the m ants on random resources

2. **Solution Construction of each ant:**
   Repeat for each ant
   i. Put the starting resource in tabu list of this ant (for the first task).
   ii. For all the remaining tasks
      a. Choose the next resources r_j for the nexttask t_i by applying following transition rule.

$$P_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum (\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}$$

if j c allowed, allowed means not in tabu list else 0

      b. Put the selected resource in previous step into tabu list of this ant
      End For
   Until each ant builds its solution
3. **Fitness**: Compute the fitness value of the solution of each ant
4. **Replacement**: Replace the Optimal solution with the ant's solution having best fitness value if its fitness value is better than Optimal solution.
5. **Pheromone Updation:**
   i. Update local pheromone for each edge
   ii. Update global pheromone
6. Empty tabu lists of all ants
7. Repeat steps 2 to 6 until stopping condition is met, Stopping condition may be the maximum number of iterations or no change in fitness value of ants" solution in consecutive iterations
8. **Output**: Print Optimal solution

End Procedure

Particle Swarm Optimization (PSO): Particle Swarm Optimization was first introduced by Dr. Russell C. Eberhart and Dr.James Kennedy in 1995. As described by Eberhart and Kennedy, the PSO algorithm is an adaptive algorithm based on a social-psychological metaphor; a population of individuals (referred to as particles) adapts by returning stochastically toward previously successful regions [1].

Particle Swarm has two primary operators: Velocity update and Position update. During each generation each particle is accelerated toward the particles previous best position and the global best position [12]. Each iteration a new velocity value for each particle is calculated based on its current velocity, the distance from its previous best position, and the distance from the global best position. The new velocity value is then used to calculate the next position of the particle in the search space. This process is then iterated a set number of times or until a minimum error is achieved.

Procedure PSO

1. **Initialization:** Initialize position vector and velocity vector of each particle.
2. **Conversion to discrete Vector:** Convert the continuous position vector to discrete vector
3. **Fitness:** Calculate the fitness value of each particle using fitness function.
4. **Calculating pbest:** Each particle's pbest is assigned its best position value till now. If particle's current fitness value is better than particle's pbest with current postion value.
5. **Calculating gbest:** Select the particle with best fitness value from all particles as gbest.
6. **Updation:** Update each particle's position vector and velocity vector using following equation:

$V_{i+1=}\omega V_i+C_1rand_1*(pbest-x_i)+C_2rand_2*(gbest-x_i)$

$X_{i-1}=X_i+V_{i+1}$

Where

ω=interia

c_1,c_2=acceleration coeffcients

$rand_1$, $rand_2$ =uniformly distributed random numbers and ε [0, 1]

pbest= best position of each particle

gbest= best position of entire particles in a population

i= iteration

7. Repeat steps 2 to 6 untill stoping condition is met; stopping condition may be the maximum number of iterations or no change in fitness value of particles for consecutive iterations.
8. **Output:** Print best particle as the final solution.

End procedure

Genetic based algorithms GA: GA was first introduced by Holland in 1975 and represents a population based optimization method based on a metaphor of the evolution process observed in nature. In GA, each chromosome (individual in the population) represents a possible solution to a problem and is composed of a string of genes. The initial population is taken randomly to serve as the starting point for the algorithm. A fitness function is defined to check the suitability of the chromosome for the environment. On the basis of fitness value, chromosomes are selected and crossover and mutation operations are performed on them to produce offsprings for the new population. The fitness function evaluates the quality of each offspring. The process is repeated until sufficient offspring are created [13]. GA are based on an analogy with the genetic structure and behaviour of chromosomes within a population of individuals using the following foundations:

I. Individuals in a population compete for resources and mates.
II. Those individuals most successful in each 'competition' will produce more offspring than those individuals that perform poorly.
III. Genes from `good' individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent.
IV. Thus each successive generation will become more suited to their environment.

Procedure GA

1. **Initialization:** Generate initial population P consisting of chromosomes.
2. **Fitness:** Calculate the fitness value of each chromosome.
3. **Selection:** Select the chromosomes for producing next generation using selection operator.
4. **Crossover:** Perform the mutation operation on the chromosomes.
5. **Mutation:** Perform the mutation operation on the chromosomes.
6. **Fitness:** Calculate the fitness value of these newly generated chromosomes known as offsprings.
7. **Replacement:** Update the population P by replacing bad solution with better chromosomes from offsprings.
8. Repeat steps 3 to 7 until stopping condition is met, Stopping condition may be the maximum number of offsprings.
9. **Output** best chromosomes as the final solution.

V. CONCLUSIONS

Tasks scheduling plays a core part in the cloud computing environment. Without a proper scheduling strategy, timely execution of the tasks and fault tolerance cannot be achieved successfully. As the requirements of the users are dynamic in nature, an efficient schedule should be established. In this paper, a hybrid technique that combines the particle swarm optimization and ant colony optimization algorithm has been proposed which effectively reduces the cost and minimizes the last dag finish time of the workflow. To prove its effectiveness, comparison through simulation environment has done between the proposed algorithm with PSO and ACOalgorithm.

REFERENCES

[1] Anton Beloglazov, JemalAbawajy, RajkumarBuyya, 'Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing' April 2011

[2] A. Gandhi, M. Harchol-Balter, R. Das, C. Lefurgy, Optimal power allocation in server farms, in: Proceedings of the 11th International Joint Conference on Measurement and Modeling of Computer Systems, ACM, New York, NY, USA, 2009, pp. 157–168.

[3] ALBCOM, LSI, UniversitatPolitècnica de Catalunya, JordiGirona 1-3: Ant colony optimization: Introduction and recent trends Campus Nord, 08034 Barcelona, Spain Communicated by L. Perlovsky Accepted 11 October 2005, Physics of Life Reviews 2 2005.

[4] A. Verma, P. Ahuja, A. Neogi, pMapper: power and migration cost aware application placement in virtualized systems, in: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, Springer, 2008, pp. 243–264.

[5] E. Dodonov, R. de Mell, A novel approach for distributed application scheduling based on prediction of communication events, Future Generation Computer Systems 26 (5) (2010) 740–752.

[6] E. Elnozahy, M. Kistler, R. Rajamony, Energy-efficient server clusters, Power Aware Computer Systems (2003) 179–197.

[7] E. Pinheiro, R. Bianchini, E.V. Carrera, T. Heath, Load balancing and unbalancing for power and performancee in cluster-based systems, in: Proceedings of the Workshop on Compilers and Operating Systems for Low Power, 2001, pp.182–195.

[8] Dian Palupi Rini, Siti Mariyam Shamsuddin, Siti Sophiyati Yuhaniz : Particle Swarm Optimization: Technique, System and Challenges, Faculty of Computer Science & Information System, University Teknologi Malaysia, Proceedings of the International Journal of Computer Applications (0975 –8887) Volume 14–No's, January 2011

[9] E. Dodonov, R. de Mell, A novel approach for distributed application scheduling based on prediction of communication events, Future Generation Computer Systems 26 (5) (2010) 740–752.

[10] E. Elnozahy, M. Kistler, R. Rajamony, Energy-efficient server clusters, Power Aware Computer Systems (2003) 179–197.

[11] E. Pinheiro, R. Bianchini, E.V. Carrera, T. Heath, Load balancing and unbalancing for power and performancee in cluster-based systems, in: Proceedings of the Workshop on Compilers and Operating Systems for Low Power, 2001, pp. 182–195.

[12] J.S. Chase, D.C. Anderson, P.N. Thakar, A.M. Vahdat, R.P. Doyle, Managing energy and server resources in hosting centers, in: Proceedings of the 18th ACM Symposium on Operating Systems Principles, ACM, New York, NY, USA, 2001, pp. 103–116.

[13] L. Chiaraviglio, I. Matta, GreenCoop: cooperative green routing with energy efficient servers, in: Proceedings of the 1st ACM International Conference on Energy-Efficient Computing and Networking, e-Energy 2010, Passau,Germany,2010,pp.191–194.

[14] http://www.ijser.org/paper/Application-of-ant-colony-optimization-for-Multicasting-in-MANET/Image_001.jpg.

[15] http://blog.smallbusinessadvocate.com/wp-content/uploads/2014/03/cloud-computing-from-any-device.jpg.

[16] http://ars.els-cdn.com/content/image/1-s2.0-S1568494614000568-gr1.jpg.

