ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Metaheuristic Approach for Scheduling Real-Time Tasks to the Heterogeneous Processors

Mrs. G.Umarani Srikanth M.E. (Ph.D.)[1]
Ms. G.Sailish Fredisha B.E. (M.E.)[2]

[1]Associate Professor, Dept of PG Studies in engineering, S.A Engineering College, Chennai-77, Tamil Nadu.
gmurani@yahoo.com

[2]PG student, Dept of PG Studies in engineering, S.A Engineering College, Chennai-77, Tamil Nadu.
sailishf@gmail.com

Abstract **-** In this paper, we addressed the problem of partitioning periodic real-time tasks in a Heterogeneous multiprocessor platform by considering both timing constraints and energy consumption perspectives: our objective is to compute the feasible partitioning solution that results in reducing energy consumption on heterogeneous processors by using advance ant colony optimization. A local search heuristic and Max-Min Ant System can be used in Advance Ant Colony Optimization to improve the task assignment solution in term of both resource utilization and energy consumption. AACO can also provides task priority for an independent task .Extensive Java Agent Development Framework (JADE) multi agent simulator results clearly show that the proposed approach provides more approximate, effective and efficient way for reducing its energy consumption.

Keywords: **-** Task Scheduling, Heterogeneous Processors, Ant Colony Optimization, Local search Heuristic, Max-Min Ant System.

I. INTRODUCTION

A set of periodic tasks can be allocated to a set of heterogeneous processors without deadline violations is an major issues in Heterogeneous computing environment. There are several difficultly in Implementing real-time applications upon multiprocessors. A major difficulties are scheduling algorithms need to specify an execution order of the tasks and also determine which processor to be used. Using Bin Packing Problem [1], task can be allocated to an homogeneous multiprocessor platform in an efficient way. In homogeneous platform where all processors are identical in nature so assigning task is very easier compare to that of heterogeneous environment. The Task allocating problem becomes more difficult when the computing environment consists of heterogeneous multiprocessors. The major reasons are a piece of code may need different execution times upon different processors. A periodic tasks is allowed to have different resource utilization and energy consumption on different processors. Static allocation of task is performed well in heterogeneous processor because computational times and deadlines of the given tasks are known in priori and do not refer to the current system state [1, 6]. Goal of a scheduling algorithm is to minimize the total execution time of the task. Scheduling algorithm can also optimize the solution by lowering its energy consumption.

II. PROBLEM DESCRIPTION

Given a set T of periodic real-time tasks and a set M of m different configuration processors, find a task-to-processor Assignment and compute task-level speeds on each processor such that [2]:

a. The tasks assigned to each processor can be scheduled in a feasible manner, and
b. The total energy consumption of M is minimum (among all feasible task allocations).

In the Heterogeneous Multiprocessor Platform, Each processor is limited to operating only one instruction per clock cycle, and runs at variable speed according to the type of task it is performing. The energy consumption of a task on a Processor is calculated by

Energy Consumption = Processor Speed*clock cycle

III. DESIGN GOALS

Design of efficient scheduling for heterogeneous processor achieves the following goals:

a. To lower energy consumption of a heterogeneous processors.
b. Dynamic allocation of task to a heterogeneous processors.
c. To minimize execution time of the tasks.

IV. METAHEURISTIC INFORMATION

A. Local Search Algorithms:

The simplest implementation of a local search mechanism, known as iterative improvement, firstly creates an initial solution by some means (possibly just generating one at random). The algorithm then checks through some or all of the neighbours of the initial solution looking for an improved solution. If an improved solution is found then the current solution is replaced with it and the process repeats until no further improvement can be found [1].

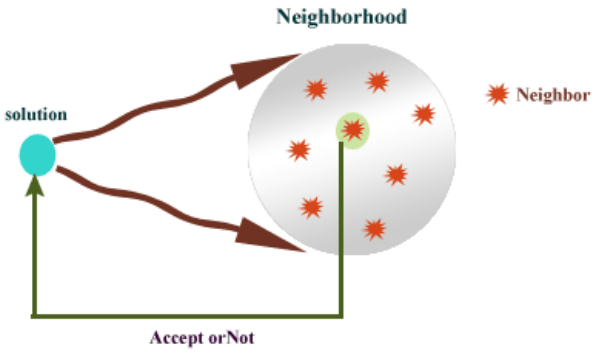

Figure 1: Local Search Algorithm

B. Ant Colony Optimization:

Ant Colony Optimization (ACO) [1,3], is inspired by the behavior of real ants, which are able to find the shortest path between food sources and their nest. Ant Colony Optimization employs simple agents called artificial ants to imitate the behavior of ant colonies.

Application of ACO:

a. Traveling salesman problem.
b. Quadratic assignment problem.
c. Job shop scheduling.
d. Sequential ordering
e. Network routing

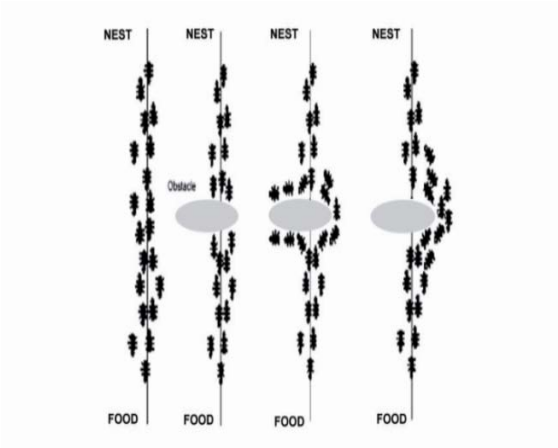

Figure 2: Natural Behaviors of Ant

C. Max-Min Ant System:

The MAX–MIN Ant System (MMAS) algorithm achieves a strong exploitation of the search history by allowing only the best solutions to add pheromone during the pheromone trail update. This mechanism which is used to for limiting the strengths of the pheromone trails effectively avoids premature convergence of the search [4, 5].

V. PROPOSED SYSTEM

Scheduling a set of implicit deadline sporadic tasks on a heterogeneous multiprocessor platform to meet all deadlines.A major advantage of my algorithm over other metaheuristics is that the dynamic allocation of task over the heterogeneous processor.

Figure 3: Dynamic allocation of task

It is new that our algorithm is designed to optimize assignment solutions in terms of both resource utilization and energy consumption. Our Advance Ant colony Optimization algorithm differs from the Ant Colony Optimization in that the artificial ants do not use any heuristic information during solution construction. Instead, our algorithm uses local search heuristic and Max-Min Ant System.. The reason is that with a constructive heuristic that is expensive to compute, ACO cannot scale up to large problem instances with competitive performance.

VI. ACHITECTURE

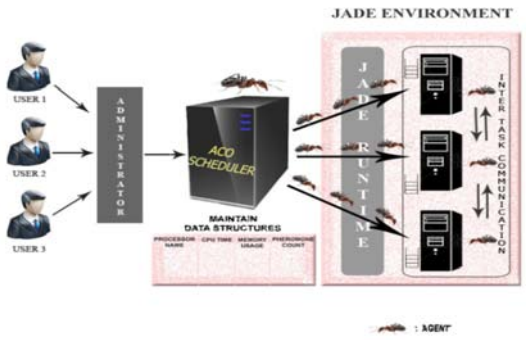

Figure 4: Architecture

VII. IMPLEMENTATION

a. Artificial Ants
b. Node Agents
c. Local Search Algorithm Implementation
d. Pheromone Update.
e. communication Layer and GUI

A. Art`ificial Ants:

The Artificial Ants is used to schedule the independent tasks to the nodes agent based on advance ant colony optimization three cases are there to assign task to an heterogeneous processors.

Case 1: Feasible Tour - ant stops with all tasks begin allocated
Case 2: Infeasible Tour - ant stops with an task not yet been assigned
Case 3: Partial Tour - ant don't met any stop condition

B. Node Agents:

Tasks are assigned to the Node Agents and the execution of the tasks take place in the node agent. After completing its execution its send back the information to the artificial ants. The information such as time taken to execute the tasks, CPU usage, memory usage, energy efficiency etc

C. C Local Search Algorithm Implementation:

Local search procedure is used for the artificial ants to improve their solutions. Our local search heuristic starts off with an initial assignment solution, and then tries to find better solutions using the following neighborhood operations [6,8]. Hence, the local search is designed to search for an improving solution using an neighbor operations

a. 1-opt
b. 2-opt

Data Structure for the Local Search Algorithm: We designed a data structure that allows the ants to quickly identify processor. The processor list is implemented. The processor list contains information regarding the execution time of the task in the processor, energy efficiency, Don't look bit, memory and CPU utilization. In order to implement an data structure in an efficient way don't-look bit are used [7]. All don't look bits are turned off at the beginning. We turn the bit for task is on whenever a search for an improving move fails, and turn it off whenever an improving move is performed between two exchanged tasks. The ants ignore all tasks whose don't-look bits are on. After implementing these ideas, search time is reduced to some extends.

D. Pheromone Update:

MAX–MIN Ant System (MMAS) is used to updating the pheromone trails. Best solution is selected in the following way: During the iterations that best solution is found, the pheromone trails are reinitialized to some values. if the best solution is not found then already present current best solution is selected as an best solution and its pheromone trails are gradually increasing [4,5].

E. Communication Layer and GUI:

JADE is used for the agent creation and communication. We need to send the tasks information to the nodes and get the information about how the tasks executed back in the node agent and send back the information to the Artificial Ants and represent that data in the Graphical User Interface.

VIII. TECHNICAL ARCHITECTURE OF JADE

JADE based AACO approach is presented in fig 5.It is built on a java-based agent platform(JADE).ACO Scheduler agent is responsible for the Initialization, Iteration Control, and Termination stages of the algorithm. Since the Iteration stage is most computational exhaustive, it will be delegated to multiple jade agents. The ants which are the jade agents residing in different container will perform the iterative construction of the scheduling. To update the iteration counter, the jade agents are coordinated by the ACO Scheduler agent. Exchange of pheromone information is also achieved by predefined agent messages.

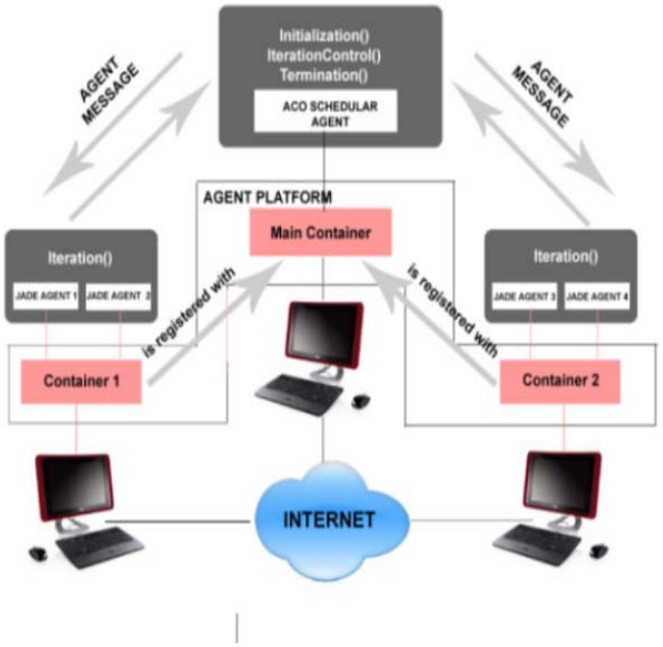

Figure 5: Technical Architecture

IX. SIMULATION RESULTS

For the purpose of demonstrating the efficiency of AACO, we have constructed a simulation and applied it to a multiprocessor scheduling problem. The simulation created for this paper is able to reduce the energy consumption of a heterogeneous processor. Fig 6 Shows the Main Scheduler Agent and GUI Along With Jade Application. Now Main Scheduler Are ready for assigning task to the Node Agent.

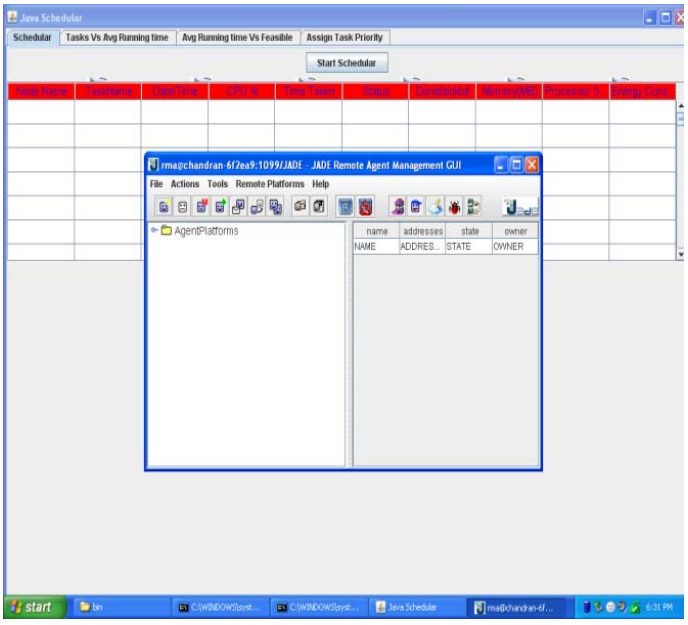

Figure 6: Main Scheduler Agent

The Fig 7 shows the priorities are given to the tasks which are dynamic in nature depend upon the necessary of the job.

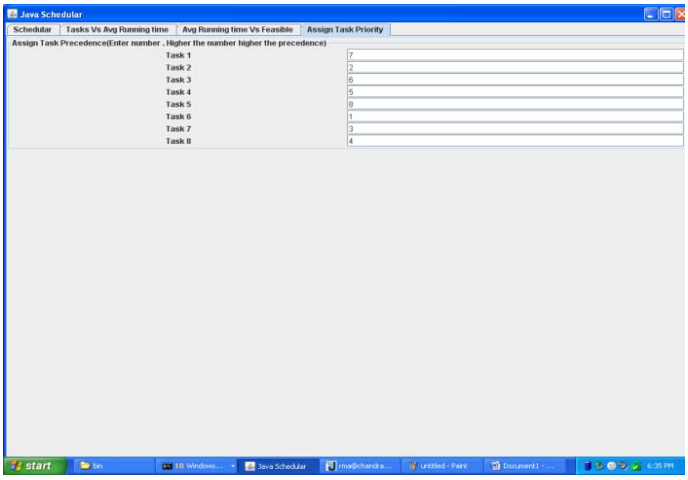

Figure 7: Task priority

Fig 8 shows an processor list which contain Node name,task name, cpu usage,memory usage,Don't look bit,energy consumption,etc.

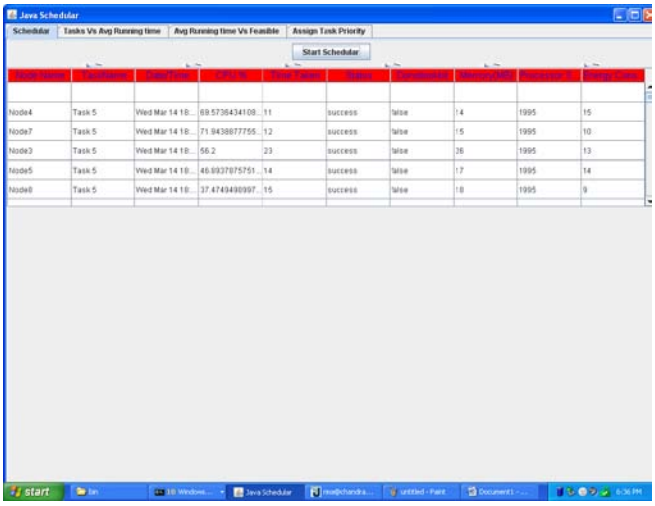

Figure 8: processor list

Fig 9 shows the Graph for the reduction of energy consumption. As the number of iteration increases then the energy consumption of the task in a processor has been reduced.

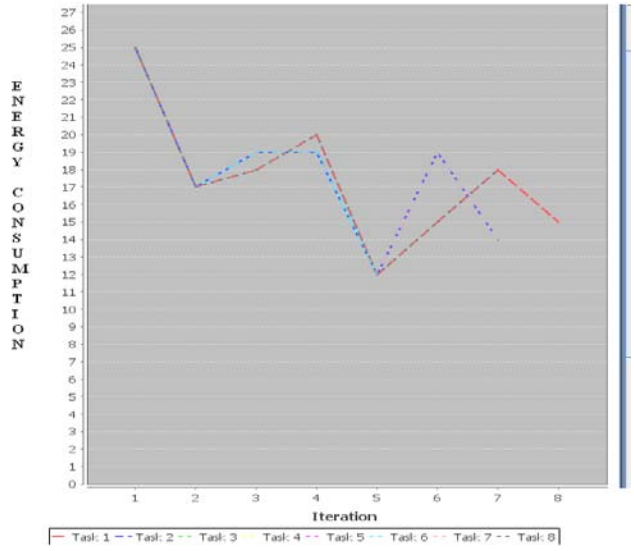

Figure 9: Graph

X. SIMULATION SOFTWARE

This paper uses the simulation technique java agent development framework. **(JADE)** using java, which is one of recent simulator used in efficient way to produce accurate result.

XI. CONCLUSION

In this Paper, we presented a new objective for reducing Energy consumption of an heterogeneous processors using an Metaheuristic Scheduling algorithm called as an Advance Ant Colony optimization (AACO). It is new that our AACO is designed to optimize assignment solutions in terms of both resource utilization and energy consumption. AACO is an effective scheduling algorithm for optimize the solution by lowing the energy consumption for an independent task with task priority. As a future work, we plan to extend our ideas on dependent task. Also the effect of fault tolerance can be considered in future.

XII. REFERENCES

[1]. G. Ritchie, Static multi-processor scheduling with ant Colony optimization & local search, Master Thesis, School of Informatics, University of Edinburgh, UK, 2003.

[2]. H. Aydin, Q. Yang, Energy-aware partitioning for multiprocessor real-time systems, in: Proc. Intl. Parallel and Distributed Processing Symposium, 2003.

[3]. M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press,2004.

[4]. T. Stützle, H.Hoos,Max–min ant system,Future Generation Computer Systems 16 (8) (1999) 889–914.

[5]. T. Stutzle, H. Hoos, Improvements on the ant-system: introducing the max–min ant system.

[6]. J. Carpenter, S. Funk, P. Holman, A. Srinivansan, J. Anderson,S. Baruah, A categorization of real-time Multiprocessor Scheduling problems and algorithms, in: Handbook of scheduling: Algorithms, Models, and Performance Analysis, Chapman Hall/CRC Press, 2003.

[7]. H.Jin, H.Wang, H. Wang, G. Dai, An ACO-based approach for task assignment and scheduling of multiprocessor control, in: Lecture Notes in Computer Science, vol. 3959, 2006.

[8]. J. Levine, F. Ducatelle, Ant colony optimisation and local search for bin packing and cutting stock problems, Journal of the Operational Research Society 55 (7) (2004) 705–716