ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 2, April 2014

Second National Conference on Trends in Automotive Parts Systems and Applications (TAPSA-2014)

On 21st & 22nd March, Organized by

Sri Krishna College of Engineering & Technology, Kuniamuthur, Coimbatore-641008, Tamilnadu, India

Effectual Multiprocessor Scheduling Based on Stochastic Optimization Technique

A.Gowthaman[1] P.Nithiyanandham[2]

PG Student [VLSI] , Dept. of ECE, Sathyamabama University,Chennai , Tamil Nadu, India [1]

PG Student [VLSI] , Dept. of ECE, Sathyamabama University,Chennai , Tamil Nadu, India [2]

ABSTRACT: The problem of task assignment in heterogeneous computing systems has been studied for many years with many variations. PSO [Particle Swarm Optimization] is a recently developed population based heuristic optimization technique. The hybrid heuristic model involves Particle Swarm Optimization (PSO) algorithm and Simulated Annealing (SA) algorithm. The PSO/SA algorithm has been developed to dynamically schedule heterogeneous tasks on to a heterogeneous processor in a distributed setup. PSO with dynamically reducing inertia is implemented which yields better result than fixed inertia.

KEYWORDS: GA, HPSO, Inertia, PSO, TAP, Distributed System, Simulated Annealing

I. INTRODUCTION

HE problem of scheduling a set of dependent or independent task in a distributed computing system is a well – Tstudied area. In this paper, the dynamic task allocation methodology is examined in a heterogeneous computing environment.

- Task definition – the specification of the identity and the characteristic of a task force by the user, the compiler and based on monitoring of the task force during execution.
- Task assignment – the initial placement of tasks on processors.
- Task scheduling – local CPU scheduling of the individual tasks in the task force with the consideration of overall progress of the task force as a whole.
- Task mitigation – dynamic reassignment of tasks to processor in response to changing loads on the processors and communication network

Dynamic allocation technique can be applied to large sets of real – world application that are able to be formulated in a manner which allows for deterministic execution. Some advantages of the dynamic technique over static technique are that, static technique should always have a prior knowledge of all the tasks to be executed but dynamic techniques do not require that.

II . RELATED WORK

Several research works have been carried out in Task Assignment Problem [TAP]. The traditional methods such as branch and bound, divide and conquer, and dynamic programming gives the global optima, but it is often time consuming or do not apply for solving typical real – world problems.

The research [2, 5 ,12] have derived optimal task assignment to minimize the sum of task execution and communication cost with the branch and bound methods. Traditional methods used in optimizations are deterministic, fast and give exact answer but often get stuck on local optima. Consequently another approach is needed when traditional methods cannot be applied for modern heuristic are general purpose optimization algorithms. Multiprocessor scheduling methods can be divided into list heuristics, meta heuristics. In list heuristics, the tasks are maintained in a priority queue, in decreasing order of priority. The tasks are assigned to the few processors in the FIFO manner. A meta heuristic is a heuristic method for solving a very general class of computational problems by combining user-given black-box procedures, usually heuristics themselves, in a hopefully efficient way.

Available meta heuristics included SA algorithm [9], Genetic Algorithm [2,6], Hill climbing, Tabu Search, Neural networks, PSO and Ant Colony Algorithm. PSO yields faster convergence when compared to Genetic Algorithm, because of the balance between exploration and exploitation in the search space. In this paper a very fast and easily

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 2, April 2014

Second National Conference on Trends in Automotive Parts Systems and Applications (TAPSA-2014)

On 21st & 22nd March, Organized by

Sri Krishna College of Engineering & Technology, Kuniamuthur, Coimbatore-641008, Tamilnadu, India

implemented dynamic algorithm is presented based on Particle Swarm Optimization [PSO] and its variant. Here a scheduling strategy is presented which uses PSO to schedule heterogeneous tasks on to heterogeneous processors to minimize total execution cost. It operates dynamically, which allows new tasks, to arrive at any time interval. The remaining of the paper is organised as follows. Section 3 deals with problem definition, section 4 illustrates PSO algorithm. The proposed methodologies are explained Section 5.

III. PROBLEM DEFINITION

This paper considers the TAP with the following scenario. The system consists of a set of heterogeneous processors (n) having different memory and processing resources, which implies that tasks (r) executed on different processors encounters different execution cost. The communication links are assumed to be identical however communication cost between tasks will be encountered when executed on different processors. A task will make use of the resources from its execution processor.

The objective is to minimize the total execution and communication cost encountered by task assignment subject to the resource constraints. To achieve minimum cost for the TAP, the function is formulated as

$$Min\ Q(x) = \sum_{i=1}^{r}\sum_{k=1}^{n} e_{ik}x_{ik} + \sum_{i=1}^{r-1}\sum_{j=i+1}^{r} c_{ij}\left(1 - \sum_{k=1}^{n} x_{ik}x_{jk}\right) \longrightarrow (1)$$

$$\sum_{k=1}^{n} x_{ik} = 1 \quad \forall\, i = 1,2,\ldots,r \longrightarrow (2)$$

$$\sum_{i=1}^{r} m_i x_{ik} \leq M_k \quad \forall\ k = 1,2,\ldots,n \longrightarrow (3)$$

$$\sum_{i=1}^{r} p_i x_{ik} \leq P_k \quad \forall\ k = 1,2,\ldots,n \longrightarrow (4)$$

$$x_{ik} \in \{0,1\} \quad \forall i,k \longrightarrow (5)$$

X_{ik} --> set to 1 if task 'i' is assigned to processor 'k'
n --> number of processor
r --> number of task
e_{ik} --> incurred execution cost if task 'i' is executed on processor 'k'
c_{ij} --> incurred communication cost if task 'i' & 'j' executed on different processor
m_i --> memory requirement of task 'i'
p_i --> processor requirement of task 'i'
M_k --> memory availability of processor 'k'
P_k --> Processor capability of processor 'k'
Q(x) --> objective function which combines the total execution and communication cost
(2) --> says that each task should be assigned exactly one processor
(3) & (4) --> are the constraints that to assure the resource demand should never exceed the resource capability.

IV . PARTICLE SWARM OPTIMIZATION

PSO is a stochastic optimization technique which operates on the principle of social behaviour like bird flocking or fish schooling. In a PSO system, a swarm of individuals (called particles) flow through the swarm space. Each particle represents the candidate solution to the optimization problem. The position of a particle is influenced by the best particle in its neighbourhood (ie) the experience of neighbouring particles. When the neighbourhood of the particle of the entire swarm, the best position in the neighbourhood is referred to as the global best particle and the resulting algorithm is referred to as 'gbest' PSO. When the smaller neighbourhood are used, the algorithm is generally referred to as the 'lbest' PSO. The performance of each particle is measured using a fitness function that varies depending on the optimization problem.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 2, April 2014

Second National Conference on Trends in Automotive Parts Systems and Applications (TAPSA-2014)

On 21st & 22nd March, Organized by

Sri Krishna College of Engineering & Technology, Kuniamuthur, Coimbatore-641008, Tamilnadu, India

Each particle in the swarm is represented by the following characteristics , x_{ij} - the current position of the particle, v_{ij} - the current velocity of the particle, $pbest_{ij}$ – personal best position of the particle. The personal best position of the particle 'i' is the best position visited by particle 'i' so far. There are two versions for keeping the neighbour best vector namely 'lbest' and 'gbest'. In the local version each particle keeps track of the best vector 'lbest' attained by its local topological neighbourhood of particles. For the global version, the best 'gbest' is determined by all particles in the entire swarm. Hence the 'gbest' model is a special case of the 'lbest' model.

TABLE 1
REPRESENTATION OF PARTICLES

Particle Number	T1	T2	T3	T4	T5
Particle1	P3	P2	P1	P2	P2
Particle2	P1	P2	P3	P1	P1
Particle3	P1	P3	P2	P1	P2
Particle4	P2	P1	P2	P3	P1
Particle5	P2	P2	P1	P3	P1

The following equations are used for the velocity updation and the position updation

$$v_{ij} = w * v_{ij} + c_1 * rand1\left(pbest_{ij} - particle_{ij}\right) + c_2 * rand2\left(gbest_{ij} - particle_{ij}\right) \longrightarrow (6.1)$$

If we consider it for local best version then the above equation can be rewritten as

$$v_{ij} = w * v_{ij} + c_1 * rand1\left(pbest_{ij} - particle_{ij}\right) + c_2 * rand2\left(lbest_{ij} - particle_{ij}\right) \longrightarrow (6.2)$$

Equation (6.1) & (6.2) is followed by

$$particle_{ij} = particle_{ij} + v_{ij} \longrightarrow (7)$$

Here c1 & c2 are cognitive co-efficient and rand1 &rand2 are the two random variables drawn from U(0,1). Thus the particle flies through potential situation towards $pbest_i$ and gbest in a navigated way while still exploring new areas by the stochastic mechanism to escape from local optima. If c1=c2 each particle attracts to the average of pbest and gbest. Since c1 expresses how much particle trusts its own past experience it is called the cognitive parameter and c2 expenses how much it trusts the swarm called the social parameter. Most implementations use a setting with c1 roughly equal to c2. The inertia weight 'w' controls the momentum of the particle. The inertia weight can be dynamically varied by applying an annealing scheme for the w-setting of the PSO where 'w' decreases over the whole run. In general the inertia weight is set according to the equation. (8).

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} * iter \longrightarrow (8)$$

A significant performance improvement is seen by varying inertia.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 2, April 2014

Second National Conference on Trends in Automotive Parts Systems and Applications (TAPSA-2014)

On 21st & 22nd March, Organized by

Sri Krishna College of Engineering & Technology, Kuniamuthur, Coimbatore-641008, Tamilnadu, India

V . PROPOSED METHODOLOGY

This section discusses the proposed dynamic task scheduling using PSO. Table1 shows an illustrative example where each row represents the particle which corresponds to the task assignment that assigns five tasks to three processors. [Particle$_3$, T_4] = P1 means that in particle 3, the task 4 is assigned to processor 1. This section discusses simple PSO, hybrid PSO. In PSO, each particle corresponds to a candidate solution of the underlying problem. In the proposed method each particle represents a feasible solution for the task assignment using a vector of 'r' element and each element is an integer value between 1 to n. The below table shows an illustrative example where each row represents the particle which correspond to a task assignment that assigns the five task to three processors.

In a hybrid version hybridisation is done by performing simulated annealing at the end of an iteration of simple PSO. Generally TAP assigns 'n' tasks to 'm' processors. So that the load is shared and also balanced. The proposed system calculates the fitness value of each assignment and selects the optimal assignment from the set of solutions. The system compares the memory and processing capacity of the processor with the memory and processing requirements of the task assigned, else the penalty is added to the calculated fitness value.

The algorithm is used for dynamic task scheduling. The particles are generated based on the number of processors used, number of tasks that have arrived at a particular point of time and population size is specified. Initially particles are generated at random and the fitness is calculated which decides the goodness of the schedule. The pbest and gbest values are calculated. Then the velocity updation and position updation are done. The same procedure is repeated for maximum number of iterations specified. The global solution which is the optimal solution is obtained. When a new task arrives it is compared with the tasks that are in the waiting queue and a new schedule is obtained thus a sequence keeps on changing with time based on the arrival of new tasks.

Each particle corresponds to a candidate solution of the underlying problem. Thus each particle represents a decision for task assignment using a vector of 'r' elements and each element is an integer value between 1 to n. The algorithm terminates when the maximum number of iterations is reached. The near optimal solution is obtained by using Hybrid PSO.

5.1 Fitness Evaluation

The initial population is generated randomly and checked for the consistency. Then each particle must be assigned with the velocities obtained randomly and it lies in the interval [0,1]. Each solution vector in the solution space is evaluated by calculating the fitness value for each vector. The objective value of Q(x) in (1) can be used to measure the quality of each solution vector. In modern heuristics the feasible solutions are also considered. Since they may provide a valuable clue to targeting optimal solution. A penalty function is only related to constraints (3) & (4) and it is given by

$$Penalty(x) = \max(0, \sum_{i=1}^{r} m_i x_{ik} - M_k) + \max(0, \sum_{i=1}^{r} p_i x_{ik} - P_k) \longrightarrow (9)$$

The penalty is added to the objective function wherever the resource requirement exceeds the capacity. Hence the fitness function of the particle vector can finally be defined as in equation

$$Fitness(x) = (Q(x) + Penalty(x))^{-1} \longrightarrow (10)$$

Hence the fitness value increases the total cost is minimized which is the objective of the problem.

5.2 Simple PSO

As we have already seen in Section 2 Classical PSO is very simple. There are two versions for keeping the neighbours best vector namely 'lbest' and 'gbest'. The global neighbourhood 'gbest' is the most inituitive neighbourhood. In the local neighbourhood 'lbest' a particle is just connected to a fragmentary number of processes. The best particle is obtained from, the best particle in each fragment.

5.2.1 gbest PSO

In the global version, every particle has access to fitness and best value soo far of all particles in the swarm. Each particle compares with fitness value with all other particles. It has exploitation of solution spaces but exploration is weak. The implementation can be depicted As a flow chart as shown in Fig. 1

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 2, April 2014

Second National Conference on Trends in Automotive Parts Systems and Applications (TAPSA-2014)

On 21st & 22nd March, Organized by

Sri Krishna College of Engineering & Technology, Kuniamuthur, Coimbatore-641008, Tamilnadu, India

5.2.2 lbest PSO

In the local version, each particle keeps tracts of best vector lbest attained by its local topological neighbourhood of particles. Each particle compares with its neighbours decided based on the size of the neighbourhood. The groups exchange information about local optima. Here the exploitation of the solution space is weakened and exploration becomes stronger. This implementation can be depicted as a flow chart as shown in Fig. 2.

5.3 Hybrid PSO

Modern meta-heuristics manage to combine exploration and exploitation search. The exploration seeks for new regions, and once it finds a good region, the exploitation search kicks in.

However, since the two strategies are inter-wound, the search may be conducted to other regions before it reaches the local optima. As, a result many researchers suggest employing a hybrid strategy, which embeds a local optimizer in between the iterations of the Meta – heuristics.

VI . DYNAMIC TASK SCHEDULING USING HYBRID PSO

The procedure for Hybrid PSO is as follows,

1. Generate the initial swarm.
2. Initialize the personal best of each particle and the global best of the entire swarm.
3. Evaluate the initial swam using the fitness function.
4. Select the personal best and global best of the swarm
5. Update the velocity and the position of each particle using the equations
6. Obtain the optimal solution in the initial stage.
7. Apply simulated annealing algorithm to further refine the solution.
8. Repeat step 3- step 7 until the maximum number of iterations specified.
9. Get the Optimized Result
10. Simulate the Process
11. Generate the initial swarm.
12. Initialize the personal best of each particle and the global best of the entire swarm.
13. Evaluate the initial swam using the fitness function.
14. Select the personal best and global best of the swarm
15. Update the velocity and the position of each particle using the equations
16. Perform the Hybrid PSO Process
17. Find the optimized result
18. Start back the simulation process.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 2, April 2014

Second National Conference on Trends in Automotive Parts Systems and Applications (TAPSA-2014)

On 21st & 22nd March, Organized by

Sri Krishna College of Engineering & Technology, Kuniamuthur, Coimbatore-641008, Tamilnadu, India

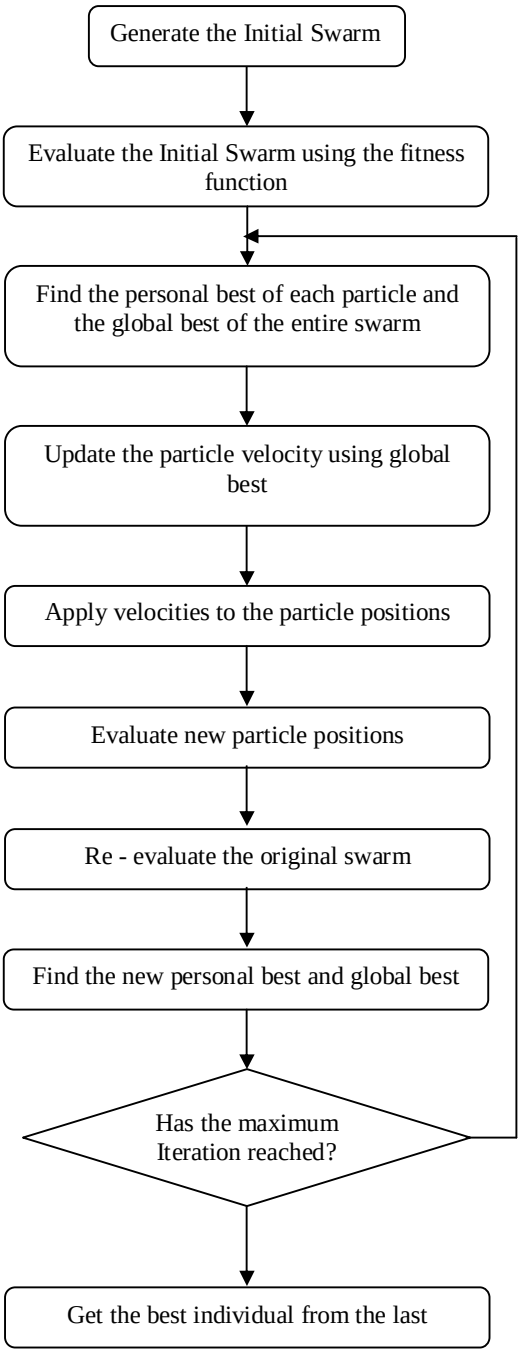

Fig 1: Global Best

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 2, April 2014

Second National Conference on Trends in Automotive Parts Systems and Applications (TAPSA-2014)

On 21st & 22nd March, Organized by

Sri Krishna College of Engineering & Technology, Kuniamuthur, Coimbatore-641008, Tamilnadu, India

VII. CONCLUSION

In many problem domains, the assignment of the tasks of an application to a set of distributed processors such that the incurred cost is minimized and the system throughput is maximized. Several versions of the task assignment problem (TAP) have been formally defined but, unfortunately, most of them are NP complete. In this paper, we have proposed a particle swarm optimization/simulated annealing (PSO/SA) algorithm which finds a near-optimal task assignment with reasonable time. The Hybrid PSO performs better than the local PSO and the Global PSO.

REFERENCES

[1] Abdelmageed Elsadek.A, Earl Wells.B, A Heuristic model for task allocation in heterogeneous distributed computing systems, The International Journal of Computers and Their Applications, Vol. 6, No. 1, 1999.
[2] Annie S. Wu, Shiyun Jin, Kuo-Chi Lin and Guy Schiavone, Incremental Genetic Algorithm Approach to Multiprocessor Scheduling, IEEE Transactions on Parallel and Distributed Systems, 2004.
[3] Batainah.S and AI-Ibrahim.M, Load management in loosely coupled multiprocessor systems, Journal of Dynamics and Control, Vol.8, No.1, pp. 107-116, 1998.
[4] Chen Ai-ling, YANG Gen-ke, Wu Zhi-ming, Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem, Journal of Zhejiang University, Vol.7, No.4, pp.607-614, 2006
[5] Dar-Tzen Peng, Kang G. Shin, Tarek F. Abdelzaher, Assignment and Scheduling Communicating Periodic Tasks in Distributed Real-Time Systems, IEEE Transactions on Software Engineering, Vol. 23, No. 12, 1997.
[6] Edwin S . H . Hou, Ninvan Ansari, and Hong Ren, A genetic algorithm for multiprocessor scheduling, IEEE Transactions On Parallel And Distributed Systems, Vol. 5, No. 2, 1994.
[7] Multiprocessor Scheduling using HPSO with Dynamically varying Inertia, International Journal of Computer Science and Applications, Vol 4 Issue 3, pp 95 – 106, 2007
[8] Multiprocessor Scheduling using Particle Swarm Optimization, International Journal of the computer, The Internet and Management, vol-17 No. 3, pp 11-24, 2009