GRID SCHEDULING USING ENHANCED PSO ALGORITHM

Mr. P.Mathiyalagan [1] U.R.Dhepthie [2] Dr. S.N.Sivanandam [3]
[1] Lecturer [2] Post Graduate Student [3] Professor and Head
Department of Computer Science and Engineering
PSG College of Technology

Abstract : Grid computing is a high performance computing environment to solve larger scale computational demands. Grid computing contains resource management, task scheduling, security problems, information management and so on. Task scheduling is a fundamental issue in achieving high performance in grid computing systems. A computational GRID is typically heterogeneous in the sense that it combines clusters of varying sizes, and different clusters typically contains processing elements with different level of performance.

In this, a heuristic approach based on particle swarm optimization algorithm is adopted for solving task scheduling problem in grid environment. Particle Swarm Optimization (PSO) is one of the latest evolutionary optimization techniques by nature. It has the better ability of global searching and has been successfully applied to many areas such as, neural network training etc. Due to the linear decreasing of inertia weight in PSO the convergence rate becomes faster, which leads to the minimal makespan time when used for scheduling. To make the convergence rate faster, the PSO algorithm is improved by modifying the inertia parameter, such that it produces better performance and gives an optimized result.

Keyword : Inertia, position updation, velocity, grid computing.

I. INTRODUCTION

Grid computing is the combination of computer resources from multiple administrative domains applied to achieve a goal, it is used to solve scientific, technical or business problem that requires a great number of processing cycles and needs large amounts of data.

One of the main strategies of grid computing is using software to divide and apportion pieces of a program among several computers, sometimes up to many thousands.

Some advantages of Grid Computing are,

1. Results can then be concatenated and analyzed upon job(s) completion.
2. Much more efficient use of idle resources. Jobs can be farmed out to idle servers.
3. Grid environments are much more modular and don't have single points of failure. If one of the servers/desktops within the grid fails there are plenty of other resources able to pick the load. Jobs can automatically restart if a failure occurs.
4. A client will reside on each server which sends information back to the master telling it what type of availability or resources it has to complete incoming jobs.
5. Jobs can be executed in parallel speeding performance. Grid environments are extremely well suited to run jobs that can be split into smaller chunks and run concurrently on many nodes. Using things like MPI will allow message passing to occur among computer resources.

Task scheduling is a key concept in computer multitasking and multiprocessing operating system design, and in real-time operating system design. In modern operating systems, there are typically many more processes running than there are CPUs available to run them. Scheduling refers to the way processes are assigned to run on the available CPUs. This assignment is carried out by software known as a scheduler.

The term heuristic is used for algorithms which find solutions among all possible ones. These algorithms, usually find a solution close to the best one and they find it fast and easily. Sometimes these algorithms can be accurate, that is they actually find the best solution, but the algorithm is still called heuristic until this best solution is proven to be the best.

Some of the heuristic techniques are, Simulated Annealing, Tabu Search, Swarm intelligence which consists of two successful approaches of Particle Swarm Optimization (PSO) and Ant Colony Optimization algorithm (ACO), Etc.

The PSO algorithm is used because, it belongs to the class of direct search methods used to find an optimal solution to an objective function in a search space. Direct search methods are usually derivative-free, meaning that they depend only on the evaluation of the objective function.

II. LITERATURE SURVEY

Particle swarm optimization (PSO) technique is employed in many optimization and search problems due to its simplicity and ability to tackle these problems successfully.

Kennedy and Eberhat developed a new evolutionary algorithm called particle swarm optimization (PSO).The PSO optimizes an objective function by iteratively improving a swarm of solution vectors, called particles, based on special management of memory. Each particle is modified by referring to the memory of individual swarm's best information. Due to the collective intelligence of these particles, the swarm is able to repeatedly improve its best observed solution and converges to an optimum.

A heuristic approach proposed by Lei Zhang, Yuehui Chen, Bo Yang [1] based on particle swarm optimization is adapted to solving scheduling problem in the grid environment. Each particle is represented a possible solution. The approach aims to generate an optimal schedule so as to get the minimum makespan and maximum resource utilization while completing the tasks. The experimental results show that, the particle swarm optimization algorithm is more efficient and effective and provides better and optimized scheduling results than the Genetic algorithm. It is also found that the particle swarm optimization algorithm converges in the fast rate and it's suited for task scheduling. From the simulated experiment, the result of PSO algorithm is better than GA. Simulation results demonstrate that PSO algorithm can get better effect for a large scale optimization problem. Task scheduling based on PSO algorithm can be applied in the computational grid environment.

The hybrid particle swarm optimization algorithm was proposed by M. Fikret Ercan [2] for the application of PSO in scheduling hybrid flow-shops with multiprocessor tasks. In order to improve the performance of PSO, hybrid techniques were employed. The experimental results show that the PSO and hybrid methods are more efficient and effective in scheduling basis. For the fixed processor case, the scheduling problem becomes more difficult to solve and results are relatively higher. However, in fix processor case, again the PSO-SA hybrid produced the best performance and results. Hybrid methods demonstrated significant performance improvement though it was at the expense of increased computational complexity.

Particle swarm adaptation (James Kennedy) is an optimization paradigm that simulates the ability of human societies to process knowledge. The algorithm models the exploration of a problem space by a population of individuals; individuals' successes influence their searches and those of their peers. The algorithm is relevant to cognition, in particular the representation of schematic knowledge in neural networks.

Particle Swarm Algorithm for Tasks Scheduling in Distributed Heterogeneous System was proposed by Xiaohong Kong, Jun Sun and Wenbo Xu [3]. A distributed heterogeneous system consists of a suite of processors or machines with different processing capacities. It can be performance-to-cost efficient to meet the diverse computation requirements if properly deployed. Task scheduling is a crucial issue to improve the efficiency of this architecture. It has been incorporated an efficient population-based search technique, Particle Swarm Optimization (PSO), with list scheduling and propose a hybrid PSO algorithm for tasks scheduling. The experiment results show that the proposed algorithm outperforms other algorithms in these aspects of performance and scalability.

In practice, we reencode the particle variables to the scheduling in order to utilizing the search capability of PSO. The major motivation of using PSO is theoretically find out an optimal scheduling list rapidly, then adopt a task duplication technique in the task allocation procedure. As a result Hybrid PSO results a better performance evaluation.

The study of Brian Ivers ,Gary G.Yen [4] examines the optimization of the job shop scheduling (JSP) by a search space division scheme and use of the meta-heuristic method of particle swarm optimization (PSO) to solve it. The job shop scheduling problem (JSP) is a well known huge combinatorial problem from the field of deterministic scheduling. It is considered the one of the hardest in the class of NP hard problems. Particles are initialized in the search space of a particular problem by assigning them a position, which represents a solution to the objective function, and a velocity. The PSO algorithm is considered a very fast algorithm and is emerging as a widely studied used algorithm for optimization problems. Due to the memory characteristics of the PSO algorithm, it works better and there is no need to have the knowledge of other particles as, PSO takes care of it.

Improvement of Particle Swarm Optimization Based on Neighborhood Cognizance and Swarm Decision was proposed by ZHU Meijie, LIU Hanxing, SUN Weiwei, ZHU TongLin [5]. The original PSO usually converges prematurely, and falls into the local optimal solution. Aimed at the shortcoming of PSO, here they have put forward an Improved PSO based on Neighborhood Cognizance (NCPSO) and Improved NCPSO based on Swarm Decision (SDNCPSO). These two improved PSO can reduce the possibility of converging prematurely. The results of experiment prove that these two improved PSO can improve the performance of global convergence in PSO and make PSO converge to global optimal solution faster. The parameters used here is not optimized, the effect of these parameters and optimized parameter setting should be analyzed and taken care.

This investigation [Shih-Tang Lo, Ruey-Maw Chen, Der-Fang Shiau and Chung-Lun Wu] introduced a particle swarm optimization (PSO) approach to solve the multi-processor resource-constrained scheduling problems [6]. There are two new rules are proposed and evaluated, named anti-inertia solution generation rule and bidirectional searching rule of PSO. The anti-inertia solution generation rule enables some jobs with anti-inertia velocity used to decide the start processing time, and escaping from local minimum. The bidirectional searching rule combines forward and backward scheduling to extend the search solution space. These two suggested rules applied in PSO scheme are capable of finding global minimum. The simulation results reveal that the proposed approach gives us the better and optimizes results. In a dynamic situation, there may be some emergency jobs arriving at a certain time or changing the resources available and requirement which also have to be considered.

An Improved PSO Algorithm was proposed (BU Yan-ping, ZHOU Wei, YU Jin-shou) against the optimal objective of to minimize the total completing time [7]. This presents an improved particle swarm optimization (PSO) algorithm with discrete coding rule for grid scheduling problem. The improved PSO algorithm can keep all the advantages of the standard PSO, such as implementation simplicity, low computational burden, and few control parameters, etc. A set of experiments show that the algorithm is stable and presents low variability. It has been tested the improved PSO algorithm against the MaxMin heuristic and found that improved PSO outperforms MaxMin by the total makespan and other performance. Fixed values for parameters are used and the algorithms are scripted in the Mat lab. But these optimization algorithms should be implemented for simulation part in the grid environment in java which can yield a better result and performance.

III. PSO ALGORITHM

Particle swarm optimization (PSO) is an algorithm modeled on swarm intelligence, that finds a solution to an optimization problem in a search space, or model and predict social behavior in the presence of objectives.

The PSO is a stochastic, population-based computer algorithm modeled on swarm intelligence. Swarm intelligence is based on social-psychological principles and provides insights into social behavior, as well as contributing to engineering applications. The particle swarm optimization algorithm was first described in 1995 by James Kennedy and Russell C. Eberhart.

The particle swarm simulates this kind of social optimization. A problem is given, and some way to evaluate a proposed solution to it exists in the form of a fitness function. A communication structure or social network is also defined, assigning neighbors for each individual to interact with. Then a population of individuals defined as random guesses at the problem solutions is initialized. These individuals are candidate solutions. They are also known as the particles, hence the name particle swarm. An iterative process to improve these candidate solutions is set in motion. The particles iteratively evaluate the fitness of the candidate solutions and remember the location where they had their best success. The individual's best solution is called the particle best or the local best. Each particle makes this information available to their neighbors.

They are also able to see where their neighbors have had success. Movements through the search space are guided by these successes, with the population usually converging, by the end of a trial, on a problem solution better than that of non-swarm approach using the same methods.

Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself i.e. its own experience and the position of the best particle in its neighborhood i.e. the experience of neighboring particles. When the neighborhood of a particle is the entire swarm, the best position in the neighborhood is referred to as the global best particle, and the resulting algorithm is referred to as the gbest PSO. When smaller neighborhoods 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.

Each Particle in the swarm is represented by the following characteristics:

- The current position of the particle
- The current velocity of the particle

The particle swarm optimization which is one of the latest evolutionary optimization techniques conducts searches uses a population of particles. Each particle corresponds to individual in evolutionary algorithm. Each particle has an updating position vector and updating velocity vector by moving through the problem space.

$V_i^{k+1} = wV_i^k + c_1\, rand_1(\ )\ x\ (pbest_i - s_i^k) + c_2\, rand_2(\ )\ x\ (gbest - s_i^k)$ -------- Eq (1)

$S_i^{k+1} = S_i^k + V_i^{k+1}$ -------- Eq (2)

Where,

v_i^k is the velocity of i at iteration k, s_i^k is the current position of i at iteration k. c_1 and c_2 are positive constants and rand1 and rand2 are uniformly distributed random number in [0,1].The velocity vector is range of [-Vmax,Vmax]. In Velocity updating eq (1), eq (3) terms that creates new velocity are,

- Inertia term, forces the particle to move in the same direction as before by adjusting the old velocity.
- Cognitive term (Personal best), forces the particle to go back to the previous best position.
- Social Learning term, forces the particle to move to the best previous position of its neighbors.

IV PROPOSED METHODOLOGY

The inertia weight *w* weighting function in Eq (1), controls the momentum of the particle. The inertia weight can be dynamically varied by applying a scheme for the setting of the PSO, where *w* decreases over the whole run. The decrease depends on the start and end value of the weight given. A significant performance improvement is seen by varying the inertia.

The Inertia term *w*, is provided with the below Eq (3) to make the convergence faster and easier.

$w = w_{end} + (w_{start} - w_{end}) * \beta$ ------ Eq (3)

Where ,

$\beta = (1 / 1 + (\alpha x / x_{\max})$

Parameters description,

w_{start},Startvalue of the inertia weight (1.5)

w_{end}, end value of the inertia weight (0.3)

x , current iteration number

$x_{\max}$, maximum iteration number

α , used to manipulate the gradient of the decreasing factor, set to 5.

The inertia term should linearly decrease in order to facilitate exploitation over exploration in later states of the search.

V PARTICLE SOLUTION REPRESENTATION FOR GRID SCHEDULING

In grid environment, one of the most important issues is how to represent a solution for task scheduling. The solution representation ties up with the PSO algorithm performance. It's defined one particle as a possible solution in the population. The dimension n corresponding to n tasks and each dimension represents a task. The position vector of each particle makes transformation about the continuous position. The smallest position value, namely, (SPV) rule is used first to find a permutation corresponding to the continuous position x_i^k.

For the n tasks and m resource problem, each particle represents a reasonable scheduling scheme. The position vector x_i^k has a continuous set of values. Where $\boldsymbol{x_i^k}$ is the position value of i particle with respect to the n dimension and s_i^k is the sequence of task of *i* particle in the processing order with respect to the n dimension.

Then the operation vector $\boldsymbol{r_i^k}$ is defined by following Equation, R= $\boldsymbol{s_i^k}$ mod m.

Table 1 illustrates the solution representation of particle $\boldsymbol{x_i^k}$ of PSO algorithm for 9 tasks and 3 processors. It's defined as that the start of sequence number is zero.

Dimension	x_i^k	s_i^k	r_i^k
0	**3.01**	**5**	**2**
1	**7.96**	**8**	**2**
2	**-0.91**	**0**	**0**
3	**0.78**	**2**	**2**
4	**-0.31**	**1**	**1**
5	**1.85**	**4**	**1**
6	**5.26**	**7**	**1**
7	**4.75**	**6**	**0**
8	**1.77**	**3**	**0**

Table 1 Solution Representation

The Initial population of particles is constructed randomly for PSO algorithm. The initialized continuous position values and continuous velocities are generated by the formula,

$\boldsymbol{X_k^0 = x_{min} + (x_{max} - x_{min}) * r}$ ----- Eq (4)

Where, $x_{min} = -0.4$ and $x_{max} = 4.0$ and r is the random number between 0 and 1.

$\boldsymbol{V_k^0 = V_{min} + (V_{max} - V_{min}) * r}$ ----- Eq (5)

Where, $V_{min} = -0.4$ and $V_{max} = 4.0$ and r is the random number between 0 and 1.

VII STEPS

STEP 1:

Initialization, set the contents for this PSO algorithm. Define the active resource and the list of tasks. The dimension of PSO algorithm is the number of tasks. Initialize position vector and velocity vector of each particle randomly by using the eq, (4) and eq (5) given below

$X_k^0 = x_{min} + (x_{max} - x_{min}) * r$

$V_k^0 = V_{min} + (V_{max} - V_{min}) * r$

Apply the SPV rule to find the permutation for the tasks.

Evaluate each particle in the swarm using an objective function. Find the best fitness value and set the global best value.

STEP 2:

Update iteration variable.

STEP 3:

Update inertia weight. Applying eq (3)

$w = w_{end} + (w_{start} - w_{end}) * \beta$

Where ,

$\beta = (1 / 1 + (\alpha x / x_{max})$

STEP 4:

Update velocity.

Apply the eq (1),

$V_i^{k+1} = wV_i^k + c_1 \, rand_1(\,) \times (pbest_i - s_i^k) + c_2 \, rand_2(\,) \times (gbest - s_i^k)$

and update velocity of each particle.

STEP 5:

Update Position, by applying the eq 2,

$S_i^{k+1} = S_i^k + V_i^{k+1}$

and update position of each particle.

STEP 6:

Apply the SPV rule to find the permutation.

STEP 7:

Update personal best, by evaluating the particle.

STEP 8:

Update global best.

STEP 9:

Stopping criterion. If the number of iteration exceeds the maximum number of iteration, then stop, otherwise go to Step 2.

IX EXPERIMENTAL RESULTS

The above algorithm is simulated in ALEA Grid simulation tool kit. The experimental results show that the modified particle swarm optimization algorithm is able to get the better schedule

Table 2 Comparisons between Existing PSO method and proposed PSO method

than the normal existing PSO algorithm in computational grid within an optimized time as shown in the above table 2.

ITERATION	***Existing method*** $e^{-(\alpha x / x_{max})}$	**Proposed Method** $(1 / 1 + (\alpha x / x_{max})$
1	1.4992	1.4992
2	1.4988	1.4985
3	1.4982	1.4977
4	1.4976	1.4970
5	1.4970	1.4962

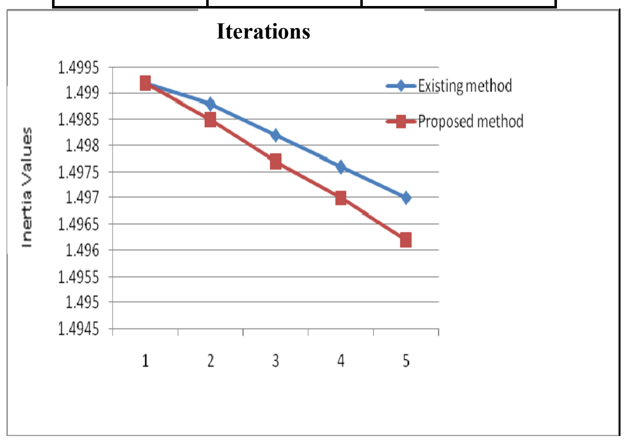

Figure1 Comparison Graph between iterations and inertia value.

The above figure 1 shows comparisons modified PSO and existing PSO method based on iterations and inertia values. The curve obviously shows the improvement.

X CONCLUSION

In this paper, a scheduling algorithm based on PSO is proposed for task scheduling problem on computational grids. Task scheduling algorithms based on PSO algorithm can be applied in computational grid environment. Each particle represents a feasible solution. This project aims at generating an optimal schedule so as to complete the tasks in a minimum time as well as utilizing the resources in an efficient way. The performance of the proposed approach is compared with the existing approach. The future work may include other hybridization techniques to further minimize the execution time.

XI REFERENCES

1. Lei Zhang, Yuehui Chen, Bo Yang ”Task Scheduling Based on PSO Algorithm in Computational Grid”, 2006 Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications.

2. M. Fikret Ercan ”A hybrid particle swarm optimization approach for scheduling flow-shops with multiprocessor tasks”, 2008 International Conference on Information Science and Security.
3. Xiaohong Kong, Jun Sun and Wenbo Xu ” Particle Swarm Algorithm for Tasks Scheduling in Distributed Heterogeneous System”, 2006 Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications.
4. Brian Ivers ,Gary G.Yen “Job Shop Optimization Through Multiple Independent Particle Swarms” 2007 IEEE Congress on Evolutionary Computation.
5. ZHU Meijie, LIU Hanxing, SUN Weiwei, ZHU TongLin “Improvement of Particle Swarm Optimization Based on Neighborhood Cognizance and Swarm Decision” 2007 IEEE.
6. Shih-Tang Lo, Ruey-Maw Chen, Der-Fang Shiau and Chung-Lun Wu “Using Particle Swarm Optimization to Solve Resource-constrained Scheduling Problems” 2008 IEEE Conference on Soft Computing in Industrial Applications.
7. BU Yan-ping, ZHOU Wei, YU Jin-shou “An Improved PSO Algorithm and Its Application to Grid Scheduling Problem ” 2008 International Symposium on Computer Science and Computational Technology.