Implementing Job Scheduling to Optimize Computational Tasks in Grid Computing using PSO

Surendra Kumar Patel
Dept.of Information Technology
Govt. N.P.G. College of Science, Raipur (C.G.), India

Anil Kumar Sharma
Dept.of Information Technology
Govt. Kaktiya P.G. College, Jagdalpur (C.G.), India

Anurag Seetha
Dept. of Computer Science and Engineering
Dr. C.V.Raman University, Bilaspur (C.G.), India

ABSTRACT
Grid computing is a recent advancement technology that enables resource sharing and dynamic allocation of computational resources, thus getting higher access to distributed data, promoting operational elasticity and collaboration. So, efficient resource management is one of the fundamental requirements in grid computing. Resource management is required in an environment where resources are quite limited and need to be utilised properly. Due to the complex and dynamic properties of grid environments, existing traditional model-based methods may result in poor scheduling performance. To overcome of such problem, we need to develop improved algorithm that reduces the computation time. This paper proposed PSO algorithm specifically focused on improving computational grid performance in terms of equal load balance for all jobs and total computation time, which enhance grid throughput, utilization, response time and more economic profits.

Keywords
Grid Computing, Resource Sharing, Job Scheduling, PSO algorithm

1. INTRODUCTION
The computational grid has provided a reliable infrastructure that helps researchers solving massive applications over geographically distributed nodes by sharing heterogeneous resources such as super computers, work stations, networks, software and even simple desktop workstation [1].

Resource sharing is the essence character of grid. There are a wide range of heterogeneous and geographically distributed resources in grid. For example, there are single processor, multiprocessors, shared memory machine, distributed memory machines, workstations, etc., and they are with different capabilities and configurations, and are managed in multiple administrative domains with different policies. In practical, the resource management and scheduling in such a complex environment are confronted with a great challenge [2][3]. Job scheduling is a decision process by which application components are assigned to available resources to optimize various performance metrics. The main goal of scheduling is to maximize the resource deployment and minimize processing time of the all jobs. Various research works has been done on job scheduling problem in grid, and different algorithms have their advantages and disadvantages but still further analysis and research needs to be done to improve the performance of scheduling algorithm in computational grid.

This proposed work mainly focuses on a PSO-based grid resource scheduling algorithm. The rest of the work is organized as follows: Section-2 gives a detailed literature review and the grid resource scheduling process and Section - 3 describes the Simulation tools and environment Section-4 Proposed algorithm Section-5 Simulation results and discussion and in Section-6 concludes the whole work.

2. RELATED WORK
Particle Swarm Optimization (PSO) is a population based search algorithm inspired by bird flocking and fish schooling originally designed and introduced by Kennedy and Eberhart [4] in 1995. In contrast to evolutionary computation paradigms such as genetic algorithm, a swarm is similar to a population, mean while a particle is similar to an individual. The particles fly through a multidimensional search space in which the position of each particle is adjusted according to its own experience and the experience of its neighbors. PSO system groups local search methods (through self experience) with global search methods (through neighboring experience), attempting to balance exploration and exploitation [5]. In 1997 the binary version of this algorithm was presented by Kennedy and Eberhart [6] for discrete optimization problems.

A combined optimization algorithm using PSO and SA was proposed for the task scheduling problem [7]. A particle consists of m segments and every segment has n distinct job numbers, representing the processing orders of n jobs on m nodes or machines. Thus, we have m machines and n jobs, which alter the continuous optimization problem to a discrete optimization problem. The real best possible values are rounded to the adjacent integers. In this algorithm, the PSO results are given to the SA algorithm to avoid being trapped in a local minimum.

The discrete PSO algorithm was proposed for task scheduling in grid systems. The scheduler aims to simultaneously minimize makespan and flow time [8,11,12]. This algorithm uses a matrix in which each column represents a job's resource allocations and each row represents jobs allocated to a resource. This inspired us to illustrate our proposed method with a 2D matrix.

3. SIMULATION TOOL AND ENVIRONMENT
Even though there are various tools available for simulating resource scheduling algorithms in Grid computing environments such as Bricks, OptorSim, SimGrid, GangSim, Arena, Alea, and GridSim, The simulation was carried out using the Optorsim simulator [9].

3.1 Architecture

OptorSim is designed to fulfill the above requirements, with an architecture (Figure 1) based on that of the EDG data management components. In this model, computing and storage resources are denoted by Computing Elements (CEs) and Storage Elements (SEs) correspondingly, which are organized in Grid Sites. CEs run all jobs by processing data files, which are stored in the SEs. A Resource Broker (RB) controls the scheduling of jobs to Grid Sites. Each site handles its file content with a Replica Manager (RM), within which a Replica Optimizer (RO) contains the replication algorithm which drives automatic creation and deletion of replicas [10].

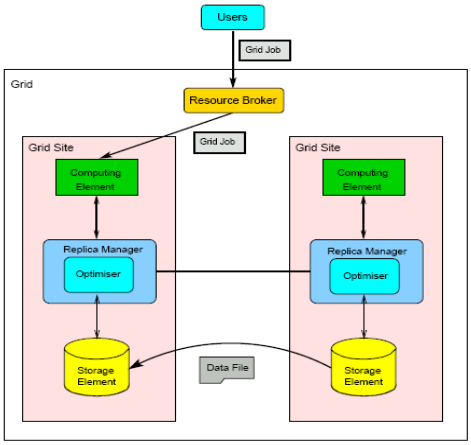

Figure-1: OptorSim Architecture [10].

3.2 Metrics

3.2.1 Mean Job Time

This is defined as the average time required executing a job starting from the time it is scheduled to the Computing Element until it complete processing all of the required files. It is fetched by accumulating the time taken by each job and divided by the number of jobs, as shown in the following formula:

$$MJET = \frac{\sum T_{Departure}\ T_{Arrive}}{n}$$

Where,

TArrive : start time of job execution,

TDeparture: completion time of job execution and

n: total number of processed jobs in the simulation.

3.2.2 Number of Replications

The replication promotes high data availability, low bandwidth utilization, increased fault tolerance, and enhanced scalability. In grid environment, replication is one of the key factors affecting performance of Data Grids. With this, it is suggested that well-defined replication strategies will smooth data access, and reduce job execution cost. However, replication is bounded by two factors: the size of storage available at different sites within the Data Grid and the bandwidth between these sites [15, 16].

3.2.3 Total Number of Local File Accesses

Table shows the number of local file accesses of each strategy and for different numbers of jobs. The fourth row in Table indicates the percentage gain of DPRSKP (Decentralized Periodic Replication Strategy based on Knapsack Problem) compared to other approaches. This percentage is computed as follows:

$$Gain\ in\ \% = \frac{Value_{DPRSKP} - Max(Value_{Best\ Client}, Value_{DR2})}{Max(Value_{Best\ Client}, Value_{DR2})}.100$$

3.2.4 Effective Network Usage

ENU is a measure of how well the replication strategy uses the network [11]. It is computed as:

$$ENU = \frac{N_{remote\ \ file\ access} + N_{replications}}{N_{remote\ \ file\ access} + N_{local\ \ file\ access}}$$

Where **Nremote file access** is the number of accesses that Computing Element reads a file from a remote site, **Nreplication** is the total number of file replication that occurs, and (Nremote file access + Nlocal file access) is the number of times that Computing Element reads a file from a remote site or reads a file locally. A lesser value indicates that the utilization of network bandwidth is more efficient [15].

4. PROPOSED ALGORITHM

4.1 Enhanced PSO based Job Scheduling

Particle Swarm Optimization (PSO) is one of the latest evolutionary optimization techniques inspired by nature. It is simulated the process of a swarm of birds preying. It has the higher capability of global searching and has been successfully applied to many areas such as neural network training, control system investigation and design, structural optimization and so on. It also has lesser algorithm parameters than both genetic algorithm and simulated algorithm. Furthermore, PSO algorithm works well on most global optimal problems. Conducting search uses a population of particles. Each particle corresponds to individual in evolutionary algorithm. A flock or swarm of particles is randomly generated initially, each particle's position representing a possible solution point in the problem space [16, 17].

Improved proposed Algorithm:
Input: number of particles m,
Inertia weight w,
Acceleration constants C_1 and C_2,
Maximum velocity, V_{max}
Task vector T,
Maximum iterating times MaxIt,
Bandwidth matrix B,
Cost vector C,
Expect Fitness Function value f_e
Exist matrix E;
Output: task scheduling vector V
Process:

1. Set iterating times It=o, select a schedule vector V with the limit E as the initialize particle;
2. While It <= MaxIt

3. Get C_J by C, b_{ij} by B;
4. Initialize velocity and position vector of each particle randomly
5. Computing fitness $\boldsymbol{f = \sum_i C_j \frac{t_i}{b_{ij}}}$ and save f
6. If $f \leq f_e$ then break;
7. It=It+1;
8. Update V by $\boldsymbol{Ve^{t+1} = (w \times Ve^t) + \left(R_1 \times C_1 \times (P_{best} - Po)\right) + \left(R_2 \times C_2 \times (G_{best} - Po)\right)}$
9. Update P by $\boldsymbol{Po^{t+1} = Po^t + Ve^{t+1}}$
10. If V_{new} and P_{new} is not in E then selection a V and P with the limit E as the V_{new} and P_{new} ;
11. End While

5. SIMULATION RESULTS AND DISCUSSION

Enhanced PSO based Job Scheduling performs the task by using different metrics. Those metrics are compared with enhanced PSO based job scheduling are shown in charts which are given below:

Experiment summary table

Grid Summary Table	Random Allocation	Enhanced PSO based Job Scheduling
Mean Job Time	5637	292
Number of Replications	91	84
Total Number of Local File Accesses	571	1226
Effective Network Usage	0.5897436	0.06851549

Mean Job Time:

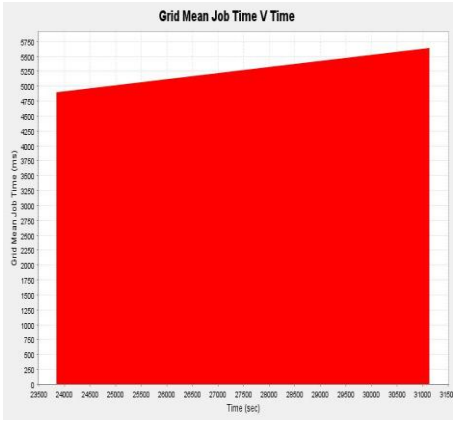

Figure-2: Random allocation

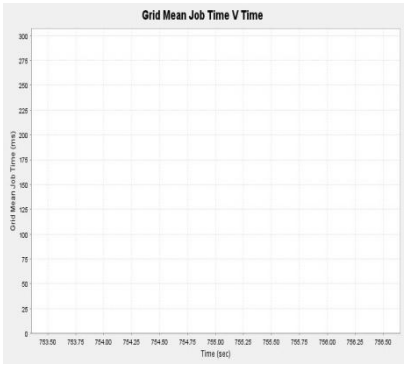

Figure-3: Enhanced PSO based Job Scheduling

Figure shows the Mean Job Time of all Jobs on Grid for random allocation and enhanced PSO based job scheduling. Thus the method of enhanced PSO based job scheduling has less mean job time than the random allocation.

Number of Replications

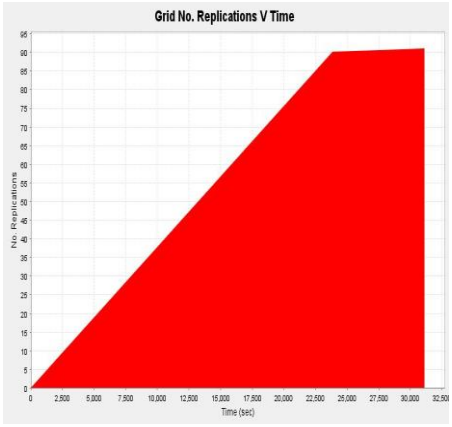

Figure-4: Random allocation

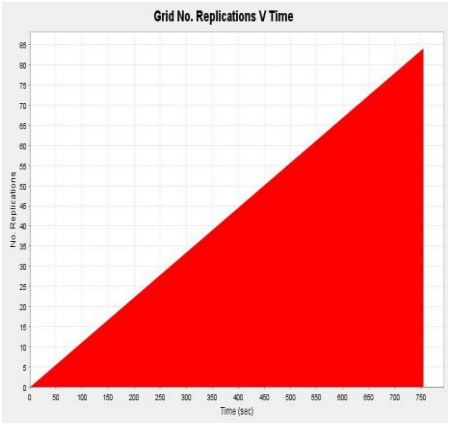

Figure-5: Enhanced PSO based Job Scheduling

Figure shows the total number of replication for random allocation and enhanced PSO based job scheduling. Thus the method of enhanced PSO based job scheduling has less replication rate than the random allocation.

Total Number of Local File Accesses

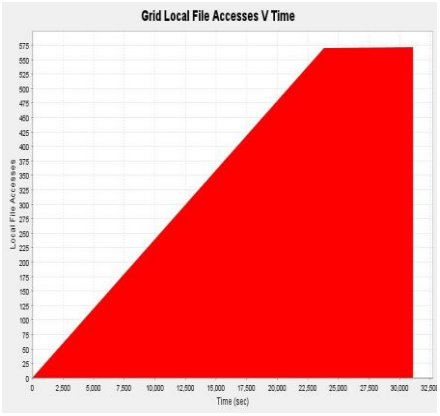

Figure-6: Random allocation

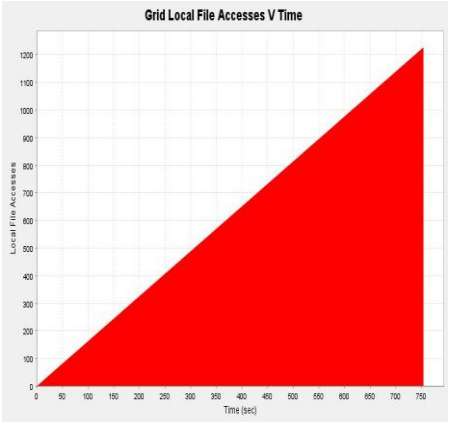

Figure-7: Enhanced PSO based Job Scheduling

The above figures of random allocation and enhanced PSO based job scheduling are shown for total number of local file accesses. Thus the enhanced PSO based job scheduling has high value when compare with random allocation.

Total Number of Remote File Accesses

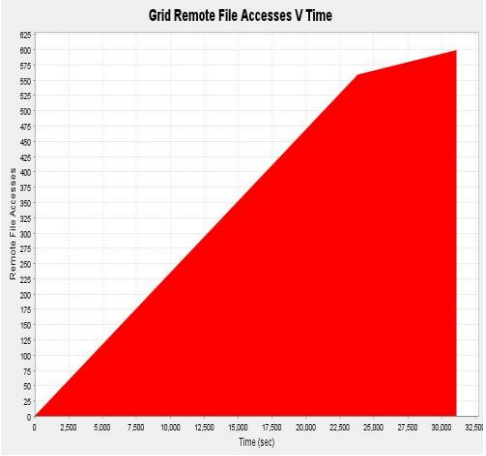

Figure-8: Random allocation

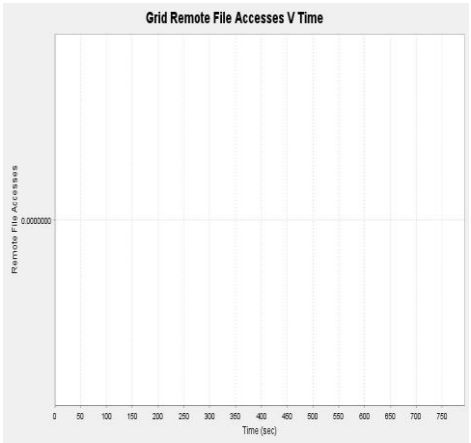

Figure-9: Enhanced PSO based Job Scheduling

The above figures show the total number of remote file accesses for random allocation and enhanced PSO based job scheduling. Thus the enhanced PSO based job scheduling has less value when compare with random allocation.

Effective Network Usage

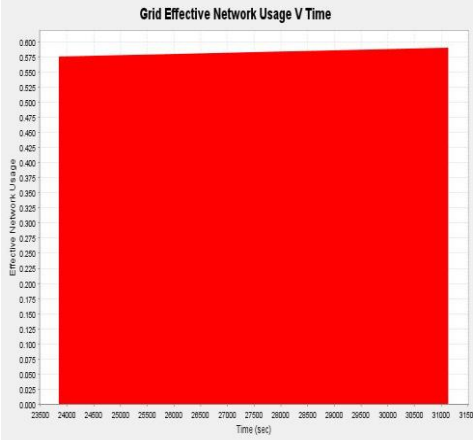

Figure-10: Random allocation

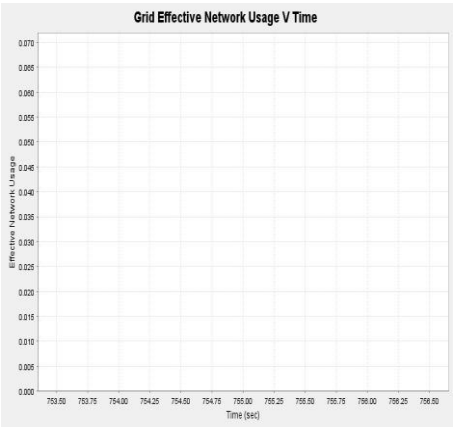

Figure-11: Enhanced PSO based Job Scheduling

Figure shows the effective network usage for random allocation and enhanced PSO based job scheduling. The random allocation has high value when compare with enhanced PSO based job scheduling method.

6. CONCLUSION

Grid computing can solve more complex tasks in less time and utilizes the resources efficiently. In this paper an algorithm is designed and implemented for an efficient job scheduling to maximize the resource utilization and minimize processing time of the jobs, that is simulation results shows that proposed enhanced PSO algorithm has optimal performance in terms of Mean Job Time, Number of Replications, Total Number of Local File Accesses and Effective Network Usage to other job scheduling approaches

7. REFERENCES

[1] Foster I., Kesselman C., 2004, "The Grid 2: Blueprint for a New Computing Infrastructure", Second Edition, Elsevier and Morgan Kaufmann Press.

[2] Foster, and C. Kesselman. 2003,"The Grid 2: Blueprint for a New Computing Infrastructure", Morgan Kaufmann, USA.

[3] [2] R. Buyya, D. Abramson, and S. Venugopal. 2005, "The Grid Economy". Proceedings of the IEEE, pp. 698-714.

[4] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks (1995) 1942–1948.

[5] A. Salman, I. Ahmad, S. Al-Madani, Particle swarm optimization for task assignment problem, Microprocessors and Microsystems 26 (2002) 363–371.

[6] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, IEEE international conference on Systems, Man, and Cybernetics (1997) 4104 – 4108.

[7] Weijun X, Zhiming W, Wei ZH, Genke Y (2004) A new hybrid optimization algorithm for the job-shop scheduling problem. In: Proceeding of the 2004 American control conference, vol 6, Boston, pp 5552–5557.

[8] Izakian H, Tork Ladani B, Zamanifar K, Abraham A 2009) A novel particle swarm optimization approach for grid job scheduling. Commun Comput Inf Sci 31:100–109.

[9] R. Buyya, "A grid simulation toolkit for resource modelling and application scheduling for parallel and distributed computing", www.buyya.com/gridsim/, accessed on January '2014.

[10] EU DataGrid Project. The DataGrid Architecture. Technical Report DataGrid-12-D12.4-333671-3-0, CERN, Geneva, Switzerland, 2001.

[11] Cameron, D. G., A. P. Millar, C. Nicholson, R. Carvajal-Schiaffino, F. Zini, and K. Stockinger 2004. Optorsim: a simulation tool for scheduling and replica optimisation in data grids. In Computing in High Energy and Nuclear Physics.

[12] Cameron, D. G., A. P. Millar, C. Nicholson, R. Carvajal-Schiaffino, F. Zini, and K. Stockinger 2004. Optorsim: a simulation tool for scheduling and replica optimisation in data grids. In Computing in High Energy and Nuclear Physics.

[13] Blythe, James, Sonal Jain, EwaDeelman, Yolanda Gil, Karan Vahi, AnirbanMandal, and Ken Kennedy. "Task scheduling strategies for workflow-based applications in grids." In Cluster Computing and the Grid, 2005. CCGrid 2005. IEEE International Symposium on, vol. 2, pp. 759-767. IEEE, 2005.

[14] Somasundaram, K. "Dynamic resource allocation in grid computing." (2014).

[15] Higashino, Wilson A., Miriam AM Capretz, and Maria Beatriz Felgar De Toledo. "Evaluation of Particle Swarm Optimization Applied to Grid Scheduling." InWETICE Conference (WETICE), 2014 IEEE 23rd International, pp. 173-178. IEEE, 2014.

[16] Izakian, Hesam, BehrouzTorkLadani, Kamran Zamanifar, and Ajith Abraham. "A novel particle swarm optimization approach for grid job scheduling." In Information Systems, Technology and Management, pp. 100-109. Springer Berlin Heidelberg, 2009.

[17] Ismail, Leila. "Dynamic resource allocation mechanisms for grid computing environment." In Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007. TridentCom 2007. 3rd International Conference on, pp. 1-5. IEEE, 2007.