Research Journal of Applied Sciences, Engineering and Technology 9(11): 982-989, 2015
DOI:10.19026/rjaset.9.2591
ISSN: 2040-7459; e-ISSN: 2040-7467
© 2015 Maxwell Scientific Publication Corp.
Submitted: October 29, 2014 Accepted: December 18, 2014 Published: April 15, 2015

Research Article

Intelligent Water Drop Algorithm Based Particle Swarm Optimization (IWDPSO) Towards Multi Objective Job Scheduling for Grid Computing

D. Thilagavathi and Antony Selvadoss Thanamani
Department of Computer Science, Nallamuthu Gounder Mahalingam College, Pollachi, Coimbatore, Tamilnadu, India

Abstract: The development of a huge amount of client's job for equivalent performance on open-resource grid system is the main reason of system failures or delayed process due to grimy hardware, software vulnerability, as well as shared confined policy. In this study we represent highly reliability conditions in grid work scheduling and present a new procedure for scheduling by hybridization of intelligent water drop algorithm and particle swarm optimization technique and compare it with earliest deadline in the basis of first come first served. The IWDPSO algorithm is tested with two datasets namely Numerical Aerodynamic Simulation (NAS) and Parameter Sweep Application (PSA) and the results are tested with performance metrics makespan, slowdown and failure rate and grid utilization. The proposed algorithms results in effective usage of grid computing resources with reduced makespan, slowdown and failure rate. The proposed algorithm is compared with Risky-MinMin (RMM), Preemptive-MinMin (PMM) and Delay Tolerant Space-Time Genetic Algorithm (DTSTGA).

Keywords: Grid, IWD, IWDPSO, job scheduling, NAS, PSA, PSO

INTRODUCTION

Foster and Kesselman (1998) introduced the concept of Grid Computing. It helps the clients to allow access the resources as provide in a crystal clear way. The Very Large Performance Computing (VLPC) Grids includes huge parallel processing systems. It is called as Grid Sites; it can able to run high demanding applications. Grid Computing is deployed over optical networks use to wavelength splitting up and multiplexing (Lehman *et al.*, 2006; Zheng *et al.*, 2005). Here node to node end connections called as light path. It is developed from starting node to ending node. In network, resources are focused to distribute by several clients. Scheduling and resource organization is mainly focused in optimizing several processors and it determines it's capable to distribute the Quality of Service requirements. However Grid computing has been overstated by the global area.

The main part of issues focused the efficient system integration by competently utilizing the already available tools which explore solutions that will make grid computing appropriate to many profitable scenarios. In this case several tools and procedures are developed to assurance the quality of service requirements tasks. In very famous and best toolkit for Grid Computing is Globus (Foster and Kesselman, 1997). Maximum and minimum fair sharing approach is used to provide fair accessibility to clients. If no shortage occurred then resource sharing or distributing complete the task within the deadlines, if there is shortage and then the CPU will reduce rates assigned to the tasks so the distributes of resources that each client will get proportional to the client's weight. Simple fair task order, adjusted fair task order and Maximum-Minimum fare share scheduling policy which use for fair scheduling:

- Simple fair task order is used to schedule the task according their individual fair close or conclusion times.
- Adjusted fair task order which use of adjusted fair conclusion time.
- Maximum-Minimum fair Share schedule method which will be used for address the problem of searching a fair task.

Hence it is capable enough to spot boundaries for the highest prospect of grid computing in a recent situation. Using risk, preemptive or replicated approach one can find the solution to increase the amount of performance when compared to normal rate. One of the most important challenges in current grid research is to find the scheduling strategies that propose a high-quality job to resource task. The huge performance computing grids helps to run extremely challenging applications. Such are simulations represents area of

Corresponding Author: D. Thilagavathi, Department of Computer Science, Nallamuthu Gounder Mahalingam College, Pollachi, Coimbatore, Tamilnadu, India

earth system science (Bernhold *et al*., 2005), biology (Ernst *et al*., 2006), or huge energy physics (Garzon *et al*., 2007). The common hypothetical and experimental mechanisms will not describe main problems individually. Heuristic strategies handle job transaction between computing sites as well as local development and presuppose an impracticable model by building a whole acquaintance concerning the system energetic presentation. Proportional integral differential control which helps adjust the number of responsibilities transmitted to a computing site. It is used to control the criticism information stored in the buffer.

Release of energetic light path process formulates a mixed logical dynamical model, i.e., Consequent transmitted tasks allow them only for the number of recognized light paths with a Quadratic Programming (QP). The Condor version of Grid called Condor-G, uses the Globus tools to deal with Grid jobs (Thain *et al*., 2003; Frey *et al*., 2002). Condor will run the jobs within a solitary administrative domain. The Globus tool makes jobs across several administrative platforms but Condor-G includes strengths of both. Next, Grid aware version introduced was Nimrod tool, extension of this was Nimrod-G (Abramson *et al*., 2000, 1997). Nimrod provides declarative parametric modeling language for expressing a parametric experiment. In the E-commerce architecture, dynamic grid resource allocation and sharing is adopted market economy (Wolski *et al*., 2001). This was categorized into two methods which represent frame work architecture like commodities and the auctions.

The following are the contributions of the proposed IWDPSO algorithm:

- Grid work schedule strategies applied to allow declaration in intelligent water drop algorithm integrated with particle swarm optimization.
- The research findings propose that it is suppler to tolerating job delays by premeditated risky breaking instead of resorting to job preemption, replication, or assuming risky operations in Grid computing systems.

LITERATURE REVIEW

Today's research focused the problem of job scheduling which was shared between computing clusters. It has been continuously appeared in the analysis of grid computing performance scheduled services (Erwin and Snelling, 2001). Adaptive algorithm decides according to which based system state, it helps to split up a work and executes in different Grid sites (Ernemann *et al*., 2002). Identification of multiple jobs/objectives for well-organized job scheduling in Grid was discussed in Kurowski *et al*. (2006).

Figure 1 describes the grid job and computing resources which manage with grid allocation and management in a grid computing. Once after the client generates a job, which consists of autonomous tasks, the generated job is executed on a computing site according to the following procedure:

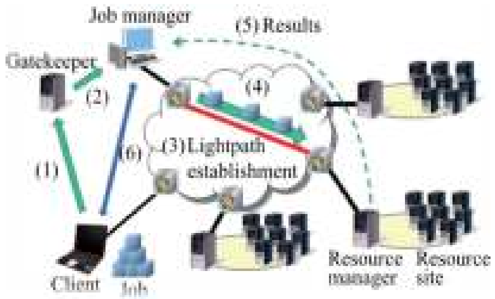

Fig. 1: Computing resource management

- The client requires a porter to perform the job and the porter authenticates the customer.
- The porter generates the job executive, which manages the job execution.
- The work executive establishes at least one light path to the site for the communication of tasks if there is no available light path.
- The resource administrator returns executed consequences to the job administrator.
- Job eradication and termination processes from the consumer are executed through the job executive.

Multiple resources shared from computing resources consequently compute the quantity of available in resources changes over time. Every job must be executed strongly at the time avoiding the lacking interruption by sharing the computing resources fairly. It also has small delay during the job execution (Osaki *et al*., 2005). Job scheduling is first and foremost optional for supercomputers, real-time computers and parallel computers (Hwang and Xu, 1998; Krauter *et al*., 2002; Kwok and Ahmad, 2000; Maheswaran *et al*., 1999). In security-aware job scheduling, the progression of scheduling becomes a toughest challenging (Welch *et al*., 2003; Xie and Qin, 2005). Scheduling approaches for Grid computing largely take no notice of this safekeeping thing, with only a handful of exceptions.

Use of dissimilar job distributes the strategy on two real workloads with no quantification of the resulting benefit (Hamscher *et al*., 2000). They measure up to their consequences to the non-cooperative state of affairs of the same machinery but do not give a quantitative estimation of possible collaborative benefits. Gupta *et al*. (2003) and Lin *et al*. (2004) have pointed two different notions faith and protected. Protection is a concept connected with the declaration of protected computing services by a grid site or by a group node, whereas conviction is reflected by the performance of a reserve node. Towards this description of both the protected demand of client's work and faith key of resource nodes it was describe that how they have an effect on the victorious execution of client's job. They used a self-protective and counteractive come within reach of preventing or avoiding security triggered failures from occurrence or destructive client works. In their new approach they have built a lead explore that tackles trust management was also built.

Thilagavathi and Thanamani (2014) proposed two such algorithms like Firefly Algorithm and Intelligent Water Drop Algorithm which outperforms the results of conventional algorithms and also some swarm intelligence algorithms like Ant Colony Optimization, Particle Swarm Optimization comparatively.

Xiong and Liu (2004) recommended a peer to peer standing system called Peer Trust, which maintains a amalgamated trust value for each peer. Other researchers have wished-for methods for the proliferation and management of trust and distrust. Lin *et al*. (2004) unspecified that trust in scheduling circumstance could be derived from improved grid safety measures. In their approach they have considered a difference of lies in an optimized matching of protection needs and its supports grid site mapping for clients works. So that the perceptibly transcends the continuation of standing values to make available feedback to grid sites. Grid job scheduling was considered as a direction of delay broadmindedness as well as the job replications. In their approach they made trade-offs between speed presentations. So, that they described a clients job to concern a Protected Stipulate (PS) to all obtainable resource sites.

The trusted model requires assessing the resource site dependability which was called as Faith Point (FP) of a node. The FP quantifies how much a user can utilize a site for productively executing a given job. A job is expected to be productively carried out when PS and FP convince a safety measures assurance circumstance (PS< = FP) during the job mapping process. The procedure of corresponding PS with FP is comparable to the real-life circumstances where the Yahoo! threshold requires users to identify the safety measures level of the login session. We recommend hazard flexible strategies, like the follows as preemptive, replication and delay-tolerant strategies. The rationale is to decrease the menace concerned in job scheduling (Casanova *et al*., 2000). They draw from hazard flexible job scheduling algorithms, which were in particular mode in order for risky Grid environment.

Thilagavathi and Thanamani (2013) presented an investigation about the use of Heuristic algorithms to optimize the Job Scheduling Problem in Grid instead of the traditional optimization method.

In Braun *et al*. (2001), the authors have evaluated the performance of the minimum scheduling heuristic and genetic algorithm to illustrate the main concept of security binding. The safety measures determined by the authors have made to design a procedure transform several supplementary heuristics or hereditary procedure such as the maximum, minimum, suffrage, or insatiable.

Assessment of scheduling algorithms was discussed in Erwin and Snelling (2001). Assessment of different scheduling approach for Grid Computing was represented in (Ernemann *et al*., 2002), likely to first come first served and largest time first.

INTELLIGENT WATER DROP ALGORITHM

The IWD algorithm is inspired by the pressure group of ordinary water drops which flood in rivers, lakes and seas. It is an inhabitant based Meta heuristics where the IWDs create a enhanced explanation from end to end collaboration with each other. This algorithm can be functional to explain optimization problems (Shah-Hosseini, 2009). In the inventive IWD algorithm, the IWDs are connected with two attributes, namely, the quantity of top soil in a pathway and the rapidness of the IWDs. The rapidness enables the water drops to reassign soil commencement one position to a different. More rapidly water drop look hooked on get together and transport more top soil from the river beds. Above and beyond, the rapidity of the IWDs is also exaggerated by the path circumstance.

The quantity of top soil in a pathway has collision on the IWDs' soil collected works and pressure group. A pathway with a lesser amount of top soil allows the IWDs to be in motion more rapidly the length of that pathway and the IWDs can accomplish a advanced velocity and bring together supplementary top soil on or after that pathway while a pathway with supplementary top soil is the contradictory. In the IWD algorithm, the pressure group of IWDs on or after the source to the purpose is performed in separate predetermined distance end to end point in time stepladder. When an IWD moves on or after one position to the subsequently one, the augment in its rapidity is comparative (non-linearly) to the contrary of the top soil of the pathway between the two locations and the soils of the IWDs augment for the reason that the IWDs do away with a quantity of soil beginning their paths. The top soil augment is inversely comparative to the point in time required for the IWDs to get ahead of between the two locations. The point in time length to travel beginning one position to the subsequent position depends on the detachment between these two locations and the rapidity of the IWDs. In the inventive IWD algorithm, the undesirability of a pathway is reflected by the quantity of top soil in the trail. When an IWD has to decide a pathway in the middle of several contender paths, it would have a preference an easier path, i.e., a path with a lesser amount of soil than a path with supplementary soil. The IWDs decide on a path based on a probabilistic purpose.

The IWD calculation utilizes a parameterized probabilistic model to develop results and the estimations of the parameters are upgraded so as to expand the likelihood of build far over the ground predominance results. The IWD calculation has been accomplished utilizing more than a couple of ordinary advancement seat mark issues. It can go over great quality answers for the Traveling Salesman Problem (Duan *et al.*, 2008) and it can additionally get to the base of the robot way advancement the n-monarch riddle and the complex Knap sack Problems with most worthwhile or close most beneficial results. Well beyond, the IWD calculation has additionally been practical to different improvement issues in unique fields of learning and it gives upgraded or in any event undifferentiated from presentation in appraisal with other well-known optimization methods, for example, ACO.

These applications comprise the understanding of exchange and industry send off a demission transmit issues in control frameworks the Vehicle Routing Problem in the field of convey, allotment and logistics characteristics decision with Rough Sets textural gimmicks arrangement for expanding fastidiousness watering system framework most good information accumulation tree in remote sensor organizes In this exploration, the creative IWD calculation is extra improved to structure the MOJSS IWD calculation; the anticipated MOJSS-IWD expands the combination of the result space and in addition enhances the inquiry nature of the IWDs. A MOJSS problem can be represented as a disjunctive graph H_{dis}=$< M, N, P >$ where H_{dis} has a swelling set M, a disjunctive edge set N and a conjunctive edge set P. Every node in the node set N describes an operation. The node set N incorporates two mannequin operations (source node and sink node) with zero weights. Conjunctive edge set P contains heading for edges concerning the coterminous operations of the indistinguishable employment.

Each edge connection speaks to the inclination imperatives of the operations of the comparative occupation. The disjunctive edge set P contains undirected edges which connect continuous operations prepared on the identical machine. These edges are undirected ones nearby every supplementary, which compare to the unanswered inclination of the operations. Both the disjunctive edges and conjunctive edges begin starting the strategy nodes, the weight w (e) of every one edge e is the passing out time of the operations where they start. Introduces the parameters utilized within the calculation. To help the multitude of the preparatory clarification space, disparate in creative IWD calculation, the preparatory amount of top soil on every one edge is an inadvertent amount and the preparatory quickness of every IWD is moreover favored aimlessly in MOJSS-IWD:

$$fi^{qzx}(\mathrm{k}) = \min\left(1, \begin{cases} \dfrac{f(soil(m,n))*\left(\frac{1}{f(j)}\right)}{\sum b\epsilon scheduablem(soil(i,j))}, \\ \left(\dfrac{f(soil(m,n))*\left(\frac{1}{f(j)}\right)}{\sum b\epsilon scheduablem(soil(i,j))} \right.\end{cases} + \ \mathrm{hu}\right)\Big),$$

$$> \varphi o, \varphi dec \leq \varphi o$$

where,

$$n_{(soil(j,v))}=\frac{1}{\varepsilon z+\ h(soil(i,j))}$$

and

$$l_{(soil(k,j))} = \begin{cases} soil(i,j) & if \min soil(i,k) \geq 0 \\ soil(i,k) - \min soil(i,k) else \end{cases}$$

Update velocity:

$$uve^{IWDh}(\mathrm{y+1}) = uve^{IWDh(y)+}\frac{bv}{cv+nv*soil\text{^}2(i,j)}$$

When the IWDs through any edge in the disjunctive graph. The edge (i, k) soil updating and IWD soil updating use the following formulas:

$$soil(i,k) = \begin{cases} (1-\rho L)*soil(i,k) - \rho l\Delta soil_{\min}\ if\ \Delta soil(i,k) <\ \Delta soil_{min} \\ (1-\rho L)*soil(i,k) - \rho l\Delta soil_{\max}\ if\ \Delta soil(i,k) <\ \Delta soil_{max} \\ (1-\rho L)*soil(i,k) - \rho l\Delta soil_{\mathrm{(i,k)\ otherwise}} \end{cases}$$

$$soil(IWD) = \begin{cases} \{soil^{IWD} + \Delta soil_{\min}\ if\ \Delta soil_{(i,k)}\ \leq soil_{min} \\ \{soil^{IWD} + \Delta soil_{\max}\ if\ \Delta soil_{(i,k)}\ \leq soil_{max} \\ soil^{iwd} + \Delta soil(i,k)\ otherwise \end{cases}$$

$soil\ (IWD)$ is the soil where $\Delta soil\ (i,k)$ = 0:9 is the local soil update parameter. The upper bound and the lower bound of the soil updates are set based on the soil on the edge and the soil in the IWDs.

Compute delta soil:

$$\Delta\ soil(i,k) = \frac{v_x}{b_s\ + cn * time\text{^}2(i,k; uve^{IWD})}$$

$$time(i,k; uveIWD) = \frac{P(K)}{MAX(\in N, uveIWD)}$$

The Pareto confined rummage around combines a wideness investigate method and with a distance downward investigate a scheme to search the explanation space, where the search is based on a scoring meaning to weigh up the schedules. For each schedule, the sum of the three purpose values is computed and this sum serves as the score to position the schedules.

MODIFIED PARTICLE SWARM OPTIMIZATION (MPSO)

The major position of the MPSO, compared to the unique MPSO, is that the client's first choice is in use into explanation. In MPSO, partiality based genus is furthermore in employment for organizing slot in the client's preference into the PSO brings up to date development. This is because the supremacy based come within reach of is not successful in a lot of purpose troubles since the quantity of non under enemy control solutions increases exponentially with the quantity of objectives. To estimate how a large amount the two objectives are interrelated, a distinction determines between them was in employment. The difference $D_{\{Aq,Av\}}$ between the two clusters Aq, Av is distinct as a standard detachment to other objectives as follows:

$$D_{Aq,Av}=\frac{\sum A,\epsilon\emptyset.a,\neq Aq,a,\neq Av[\varepsilon_{\{Aq,Ar\}-\varepsilon\{Av,Ar\}]2}}{|\emptyset|-2}$$

Two purpose groups of object that encompass the negligible distinction are compound keen on single and the communication degrees in the middle of will clusters are recalculated. The interaction degree calculated as:

$$\varepsilon_{\{Aq,Av\}}=\frac{\sum(u,n)\epsilon\big(D(Aq)\times E(Av)\big)\varepsilon(i,j)}{|E(Aq)\times E(Av)|}$$

The position and orientation vector at the footstep were calculated as follows:

$$gxk = gx - 1 + \left(n_{off} + n_m\right).\cos\,(f\theta v_1 + (-1)^k.\frac{\pi}{2}$$

$$+fr.\sin\,(f\theta k - 1_{,} + (-1)^k.\frac{\pi}{2}$$

$$f\theta_k = f\theta(k-1) + \emptyset(k)$$

Calculate normalized weight as:

$$p_i=\frac{\sum_{j=1}^{k} p(j,v)}{\sum_{a=1}^{k}\sum j=1\, p_{l,j,i}}$$

The quality of the obtained solution set is high if this space is large. Diversity T is for evaluating the spread of non dominated solutions, which is defined as follows:

$$T = \sum_{W=1}^{V}(l_{x(\max)^{)}-}\, l_{y(^{\min})}) \div \sqrt{\sum_{i=1}^{|no|}(t(i)-t)\hat{}2}$$

To merge two highly related objectives into one, the dissimilar between them was accounted. In below dissimilarity between d_1, d_2 was calculated as:

$$t\{d_{1,}d_2\} = \frac{(\varepsilon 23-\varepsilon 15)^2 + ((\varepsilon 23-\varepsilon 19)^2}{|\emptyset|-2} = \frac{0.00}{2} = 0.00$$

Make a pair wise comparison matrix represents comparison matrix of objectives, P, which represents preference degrees between objectives, is define as follows:

$$\begin{bmatrix} p11 & p12\,....pm \\ p12 & p22\,....pm \\ & . \\ & . \\ & . \\ pm1 & pm2\,...pmm \end{bmatrix}$$

Pseudo code for Intelligent Water Drop Algorithm Based Particle Swarm Optimization:

```
If global scheduler solicits "execution time estimates" for a job, $J_k$ then
Solicit execution time estimates from all participating computers, $T_i$ ($J_k$)
Return max {$T_i$ ($J_k$)} to the global scheduler;
end if
if a job is assigned to the site then
Check the currently active number of participating computers, n;
Broadcast the value of optimal strategy s according to all the participating computers;
Apply IWDPSO
for round = 1 or more
        (i.e., a total of 2T units of time) do
    if a computer $M_i$ takes up the job
        (ties are broken randomly) then
      Send the job to $M_i$;
      Declare that the job is unavailable;
    end if
  end for
if no computer takes up the job then
  Declare that the job fails;
 end if
end if
```

RESULTS AND DISCUSSION

This research study uses GridSim simulator which installed on ALEA scheduler on the top of the GridSim. This study focus on the representation of parallel jobs and their execution.

NAS workload: This research work uses the three month records of Numerical Aerodynamic Simulation (NAS) Systems Division at NASA Ames Research Center. It includes the data of 7948800 sec (i.e., 92 days) collected in the year 1993. It has the job count of 16000 (Lo and Mache, 2002). To test the performance of job execution in a high throughput Grid environment, the 92 days trace data is compressed to

Table 1: Parameter settings for NAS workload

Parameter	Value setting
Number of jobs N	NAS:16000
Number of sites M	NAS:12
Job arrival rate	Given by trace
Job workloads	Given by trace
Site processing speed	8×8 nodes and 4×16 nodes
Job Security Demands (SD)	0.6-0.9 uniform distribution
Failure and delay coefficients	$\lambda = 3; \gamma = 2$

Table 2: Parameter settings for PSA workload

Parameter	Value setting
Number of jobs N	10000
Number of sites M	20
Job arrival rate	0.008 jobs/sec/site
Job workloads	20 levels (0-300000)
Site processing speed	10 levels (0-10)
Job Security Demands (SD)	0.6-0.9 uniform distribution
Failure and delay coefficients	$\lambda = 3; \gamma = 2$

46 days. Nodes to be simulated are 128 which to be plotted to 12 Grid sites; 16 nodes are to be treated in 4 sites and the others include 8 nodes i.e., 8 sites. The model of simulations is depends on job arrival time, size of the job and data runtime given by the trace. This trace was purified to uproot the client indicated data and preprocessed to right for framework downtime (Table 1).

PSA workload: The Parameter Sweep Application (PSA) model has emerged as a “killer application model” for composing high-throughput computing applications for processing on global Grids (Casanova *et al*., 2000). The parameter sweep application is defined as a set of independent sequential jobs (i.e., no job precedence). The independent jobs operate on different data sets. A range of scenarios and parameters to be explored are applied to the program input values to generate different data sets. The execution model essentially involves processing K independent jobs (each with the same task specification, but a different data set) on M distributed sites, where K is typically much larger than M (Table 2).

The following section discuss about the metrics used to measure the performance and to compare with RMM, PMM and DTSTGA.

Performance metrics: The performance metrics used for evaluation in this research work are discussed below.

Makespan: It is the aggregate execution time turnaround. From the Fig. 2 and 3 it is obvious that the proposed IWDPSO performs better than other methods.

Slowdown: It is the distinction between the normal turnaround time and the normal holding up time. From the Figure 4 and 5 it is shown that the proposed IWDPSO achieves better performance.

Failure rate: It is the quantity of failed and rescheduled occupations. From the Fig. 6 and 7 can be observed the failure rate is dropped down to remarkable level in the proposed IWDPSO.

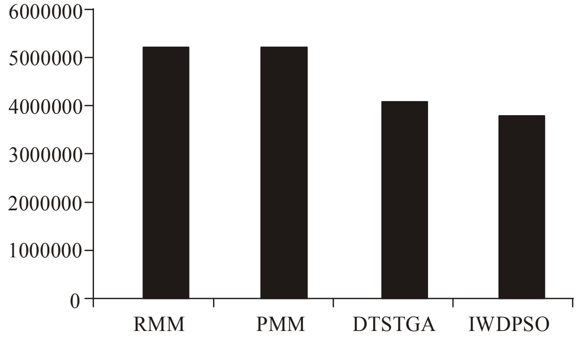

Fig. 2: Makespan in NAS jobs

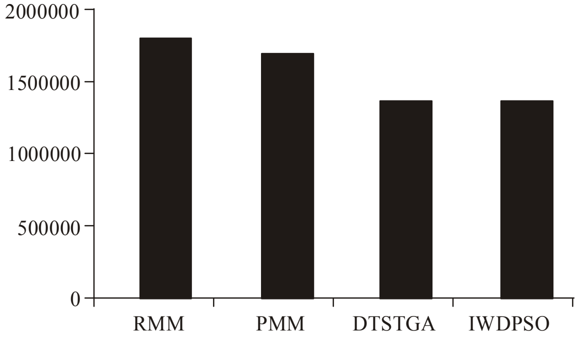

Fig. 3: Makespan in PSA jobs

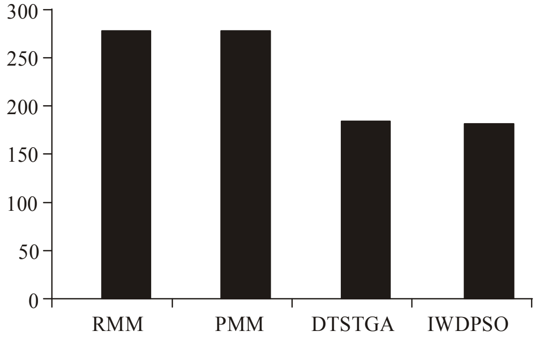

Fig. 4: Slowdown in NAS jobs

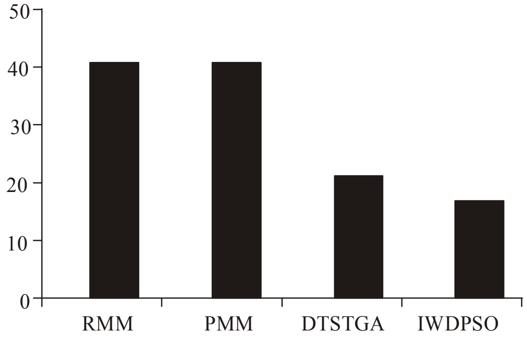

Fig. 5: Slowdown in PSA jobs

Grid utilization: Rate of transforming force allotted to effectively executed employments out of the aggregate

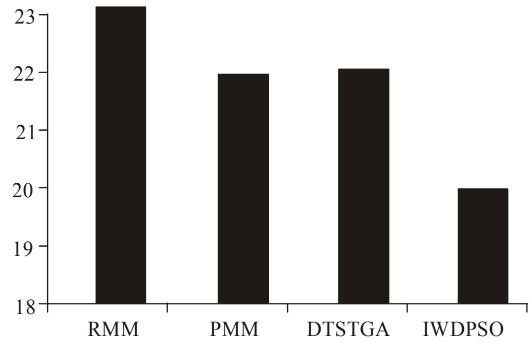

Fig. 6: Failure rate in NAS jobs

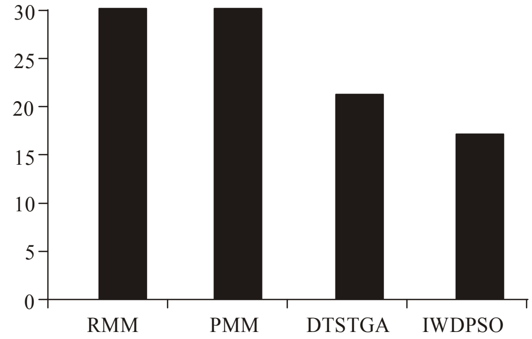

Fig. 7: Failure rate in PSA jobs

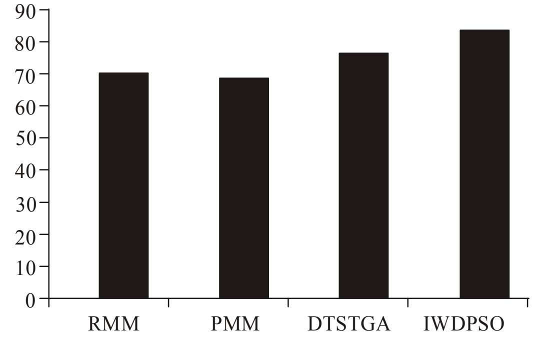

Fig. 8: Grid utilization in NAS jobs

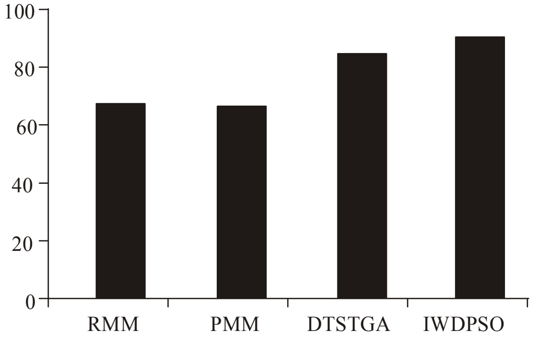

Fig. 9: Grid utilization in PSA jobs

preparing force accessible of a worldwide Grid. It is remarkable that the proposed IWDPSO achieves better grid utilization which is showed in Fig. 8 and 9.

Figure 2, 4, 6 and 8 shows the performance Analysis of Risky MinMin (RMM), Preemptive MinMin (PMM), Space-Time Genetic Algorithm based on Delay Tolerant (DTSTGA) and the proposed algorithm IWDPSO in 16,000 NAS jobs over 12 simulated Grid sites.

Figure 3, 5, 7 and 9 shows the performance Analysis of Risky MinMin (RMM), Preemptive MinMin (PMM), Space-Time Genetic Algorithm based on Delay Tolerant (DTSTGA) and the proposed algorithm IWDPSO in 10,000 PSA jobs over 20 simulated Grid sites.

CONCLUSION

This research study focuses on effective grid work scheduling and utilization. This research study proposed IWDPSO algorithm which is a hybridization of intelligent water drop algorithm and particle swarm optimization. The performance of the proposed algorithm was compared with existing algorithm namely RMM, PMM, DTSTGA with two datasets namely NAS and PSA. The results shows that IWDPSO outperforms than the existing algorithm in the two datasets with the performance metrics makespan, slowdown, failure rate and grid utilization.

REFERENCES

Abramson, D., J. Giddy and L. Kotler, 2000. High performance parametric modeling with nimrod/G: Killer application for the global grid. Proceeding of 14th International Parallel and Distributed Processing Symposium (IPDPS ’00), pp: 520-528.

Abramson, D., I. Foster, J. Giddy, A. Lewis, R. Sosic, R. Sutherst and N. White, 1997. Nimrod computational workbench: A case study in desktop metacomputing. Proceeding of Australian Computer Science Conference (ACSC ’97).

Bernhold, D., S. Bharathi, D. Brown and K. Chanchio, 2005. The earth system grid: Supporting the next generation of climate modeling research. P. IEEE, 93(3).

Braun, T.D., H.J. Siegel, N. Beck, D. Hensgen, R.F. Freund, L.L. Boloni, M. Maheswaran, A.I. Reuther, J.P. Robertson, M.D. Theys and B. Yao, 2001. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. J. Parallel Distr. Com., 61(6): 810-837.

Casanova, H., A. Legrand, D. Zagorodnov and F. Berman, 2000. Heuristics for scheduling parameter sweep applications in grid environments. Proceeding of 9th Heterogeneous Computing Workshop (HCW), pp: 349-363.

Duan, H., S. Liu and X. Lei, 2008. Air robot path planning based on intelligent water drops optimization. Proceeding of International Joint Conference on Neural Networks (IJCNN, 2008), pp: 1397-1401.

Ernemann, C., V. Hamscher, A. Streit and R. Yahyapour, 2002. Enhanced algorithms for multi-site scheduling. In: Parashar, M. (Ed.), GRID 2002. LNCS 2536, Springer-Verlag, Berlin, Heidelberg, pp: 219-231.

Ernst, M., P. Fuhrmann, A. Papaspyrou, M. Radicke, L. Schley and R. Yahyapour, 2006. A computational and data scheduling architecture for HEP applications. Proceeding of the Conference on High Energy Physics (CHEP, 2006). Mumbai, India.

Erwin, D.W. and D.F. Snelling, 2001. UNICORE: A grid computing environment. In: Sakellariou, R. *et al*. (Eds.), Euro-Par 2001. LNCS 2150, Springer-Verlag, Berlin, Heidelberg, pp: 825-834.

Foster, I. and C. Kesselman, 1997. Globus: A met computing infra-structure toolkit. Int. J. Supercomput. Ap., 11(2): 115-128.

Foster, I. and C. Kesselman, 1998. The Grid: Blueprint for a Future Computing Infrastructure. 1st Edn., Morgan Kaufman, San Francisco.

Frey, J., T. Tannenbaum, I. Foster, M. Livny and S. Tuecke, 2002. Condor-G: A computation management agent for multi-institutional grids. J. Cluster Comput., 5: 237-246.

Garzon, J.I., E. Huedo, R.S. Montero, I.M. Llorente and P. Chacon, 2007. Adaptation of a multi-resolution docking bioinformatics application to the grid. J. Software, 2: 1-10.

Gupta, M., P. Judge and M. Ammar, 2003. A reputation system for peer-to-peer networks. Proceeding of ACM International Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), 2003.

Hamscher, V., U. Schwiegelshohn, A. Streit and R. Yahyapour, 2000. Evaluation of job-scheduling strategies for grid computing. Proceeding of the 1st IEEE/ACM International Workshop on Grid Computing (GRID '00), pp: 191-202.

Hwang, K. and Z. Xu, 1998. Scalable Parallel Computing: Technology, Architecture, Programming. McGraw-Hill, San Francisco, New York.

Krauter, K., R. Buyya and M. Maheswaran, 2002. A taxonomy of grid resource management systems for distributed computing. Software Pract. Exper., 32(2): 135-164.

Kurowski, K., J. Nabrzski, A. Oleksiak and J. Weglarz, 2006. Scheduling jobs on the grid-multicriteria approach. Comput. Methods Sci. Technol., 12(2): 123-138.

Kwok, Y.K. and I. Ahmad, 2000. Link-constrained scheduling and mapping of tasks and messages to a network of heterogeneous processors. Cluster Comput., 3(2): 113-124.

Lehman, T., J. Sobieski and B. Jabbari, 2006. DRAGON: A framework for service provisioning in heterogeneous grid networks. IEEE Commun. Mag., 44(3): 84-90.

Lin, C., V. Varadharajan, Y. Wang and V. Pruthi, 2004. Enhancing grid security with trust management. Proceeding of International Conference on Services Computing (SCC, 2004), pp: 303-310.

Lo, V. and J. Mache, 2002. Job scheduling for prime time vs. non-prime time. Proceeding of International Conference on Cluster Computing (Cluster ’02), pp: 488-493.

Maheswaran, M., S. Ali, H.J. Sigel, D. Hensgen and R.F. Freund, 1999. Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems. Proceeding of 8th Heterogeneous Computing Workshop (HCW '99), pp: 30-44.

Osaki, H., S. Watanabe and M. Imase, 2005. On dynamic resource management mechanism using control theoretic approach for wide-area grid computing. Proceeding of IEEE Conference on Control Applications (CCA, 2005), pp: 891-897.

Shah-Hosseini, H., 2009. Optimization with the Nature-inspired Intelligent Water Drops Algorithm. In: Santos, W.P.D., (Ed.), Evolutionary Computation. Vienna, Austria, pp: 297-320.

Thain, D., T. Tannenbaum and M. Livny, 2003. Condor and the Grid. In: Berman, F., A.J.G. Hey and G. Fox (Eds.), Grid Computing: Making the Global Infrastructure a Reality, John Wiley, Hoboken, NJ.

Thilagavathi, D. and A.S. Thanamani 2013. Heuristics in grid scheduling. Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET), 2(8).

Thilagavathi, D. and A.S. Thanamani, 2014. Scheduling in high performance computing environment using firefly algorithm and intelligent water drop algorithm. Int. J. Eng. Trends Technol. (IJETT), 14(1).

Welch, V., F. Siebenlist, I. Foster and J. Bresnahan, 2003. Security for grid services. Proceeding of 12th International Symposium on High Performance Distributed Computing (HPDC-12), pp: 48-57.

Wolski, R., J.S. Plank, T. Brevik and J. Bryan, 2001. G-commerce: Market formulations controlling resource allocation on the computational grid. Proceeding of 15th International Parallel and Distributed Processing Symposium (IPDPS ’01).

Xie, T. and X. Qin, 2005. Enhancing security of real-time applications on grids through dynamic scheduling. Proceeding of 11th International Conference on Job Scheduling Strategies for Parallel Processing (JSSPP, 2005), pp: 219-237.

Xiong, L. and L. Liu, 2004. PeerTrust: Supporting reputation-based trust for peer-to-peer Electronic communities. IEEE T. Knowl. Data Eng., 16(7): 843-857.

Zheng, X., M. Veeraraghavan, N.S.V. Rao, Q. Wu and M. Zhu, 2005. CHEETAH: Circuit-switched high speed end-to-end transport architecture testbed. IEEE Commun. Mag., 43(8): S11-S17.