PERFECT FAULT TOLERANT GRID RESOURCE SELECTION FOR MULTI TASK PREDICTION BASED ON HYBRID GRP-PSO ALGORITHM AT REAL TIME

Surendran.R.[1], **Parvatha Varthini. B.**[2]

[1]Research Scholar, Department of Information Technology, Sathyabama University, Chennai, India
[2]Department of Computer Applications, St.Joseph's College of Engineering, Chennai, India
Email: [1]surendran.mtech.it@gmail.com, [2]parvathavarthini@gmail.com

Abstract

A powerful concept of grid computing is Resource sharing in the large level high performance computing network at world wide. Main focus of grid computing is perfect Fault tolerant Resource selection for a job then only receives the good result. Selecting the perfect Grid resource for task prediction is the major problem in the dynamic grid computing. In this paper, we have designed the new optimization algorithm presented and named as hybrid GRP-PSO. Hybrid GRP-PSO is combination of Particle Swarm optimization (PSO) algorithm and Grid Resource Prediction Pattern (GRP) from two or more grid networks. Since Particle Swarm Optimization algorithm is weak in Local search, Grid Resource Prediction pattern has been used to improve the quality and selecting the perfect Fault tolerant grid resource. The experimental results show the important of proposed system to select the perfect grid resource. Here the end User doesn't need the grid resource knowledge only submit task to the grid service. This proposed grid service (portal) will care of all knowledge about the perfect resource selection automatically with secure and efficient via.

Key words: Perfect Fault tolerant resource selection; Multi task prediction; Hybird GRP-PSO; Grid Manager & Grid Remote Executor; Experimental results

I. INTRODUCTION

In Real time Grid Computing application Environment, Job Scheduling and Allocation is the major and also difficult work. To allocate or Schedule for Grid Job/Task, find the Most Efficient resource on the Dynamic Grid Computing network. In this Research work tries to find the methodology which can fit to the Need and fulfill the most import need of selection of perfect Fault tolerant Gird Resource for Task Prediction in Grid Computing. In our Focused research, we have come up with the three different Methodology/ Functionality/ Pattern Approach to find the solution for our focused area. Before we could Analysis more on these area, let we discuss the below mentioned proposed Area are Proposed Grid Service provider, Resource provider & Remote Executor Service Architecture, Proposed Grid Resource Prediction Pattern, System Architecture design to find the perfect Fault tolerant Resource, Sequence Diagram to find the best grid node for Task.

II. LITERATURE REVIEW

In the Existing system Grid Computing System task get allocated to the grid Node with out even Predicting. Grid Computing has limitations in the existing system for resource monitoring and prediction strategies. Such as, grid technologies do not have efficient resource monitoring and prediction strategies. It does not co-operate with various components to achieve high performance [4], because it does not have sufficient resources. An Infrastructure for Monitoring and Management in Computational Grids Based on GMA (Grid Monitoring Architecture) technique with advantage of Monitoring and fault diagnosis for task. The Globus Heartbeat Monitor Specification, Based on HBM (Globus Heartbeat Monitor) is able to monitor with High accuracy. The Ganglia Distributed Monitoring System: Design, Implementation, and Experience is Based on host name, Identification, Memory capacity, Name and version of operating system, File system data , The processor load, and so on through carrying on the monitoring to the cluster with advantage of high level monitoring. In Performance Information Services for Computational Grids, Based on the concept of collects information from these sensors on computing nodes, and predicts resource usages in certain time interval ahead, using multiple models such as mean based one, median based one, and autoregressive method. Some system [9] follows the Case-Based Reasoning technique on scheduler machine. Executer will select useful rules and then nodes and then deliver the result to Resource analyzer (RA). After that, RA will analyze this results based-on fault tolerant criteria and some important job condition by executing several queries to get better decision. In this system's advantage is very effective for adaptive grid scheduling due to reliability, fault tolerance, and then decrease of job completion time, disadvantage is Not follow the dynamic and intelligent component in scheduling phase. Finding a perfect Grid Node for the Task allocation is the major problem which most of the Grid Infrastructure has in today's world.

III. GRID SERVICE PROVIDER, RESOURCE PROVIDER & REMOTE EXECUTOR SERVICE

3.1 Architecture

Proposed Grid Service provider, Resource provider & Remote Executor Service Architecture, is a Main Strategies of Grid Computing. Grid computing is distributed, large-scale cluster computing, as well as a form of network-distributed parallel processing. Main focus of Grid Computing is using software to divide and apportion pieces of a program among several computers, sometimes up to many more 1000's. To manage and handle pieces across multiple computing in Network-Distributed environment or Cluster Environment can use .Net Remote Concept.

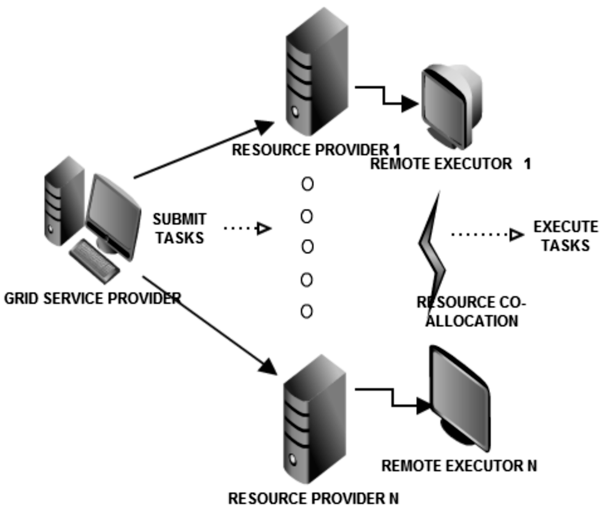

Fig. 1. Grid service provider, Resource provider & Remote Executor Service Architecture

This work uses the .NET Remoting for communication between a service provider, resource provider and the Executors. For getting away from firewalls and security boundaries, we are used the HTTP channel and the SOAP formatter through Grid security certified key. In this Grid computing network, one Grid service provider and N number of resource provider with *N* number of Remote executor Services are connected. We don't have any restrictions on the number of Grid Computing executor Services. They can grow up as much as till the resources let us. In "Fig.1", Grid service provider with multiple Grid Remote Executor service; it can be same Geographical location or Different Geographical Location. It almost like Electricity power grid, user from different country/ Geographical location will try to access the power without even knowing the where the power grid resist. People

Who looks for electricity power, they will never know where these Electricity power comes from and how does it connect from main station to their location. Likewise, our Grid service provider with 1000's Grid remote executor service will never know each other's location of grid user.

3.2 Communication Intensive via History Database

Each resource has a relationship with every other through communication intensive (Message Passing Interface (MPI) code) grid application, which has a lot of communication between the selected grid resources. Grid remote Executor Services all Activities and every execution are maintain and manage via History database. Each Grid remote Executor service will be on the different Cluster Environment, and linked with many or 1000's of Grid Node. In "Fig.2", to manage all the information about Grid Remote Executor Service, Grid Node and Grid Manager, we need to have the Grid History and Prediction Manager. Here Grid Manager is combination of Grid Service provider and Resource provider.

IV. GRID RESOURCE PREDICTION PATTERN

Grid Resource Prediction pattern has History, Grid Resource Prediction Model, Prediction for each Resource (rs). This History will become most important to have a prediction; This History always has information about any of the resource Historical data information of very least used Resource and most available resource. From this grid resource providers, will intelligently predict which the Fault tolerant resource for a task is through Grid Resource prediction. Resources are service, software, platform, infrastructure, supercomputer, cluster, server, personal computer, web cam, printer, scanner …etc. RSL (Resource Specification Language) code use to implement the prediction pattern. In "Fig.3", Grid resource Prediction model compare the data through

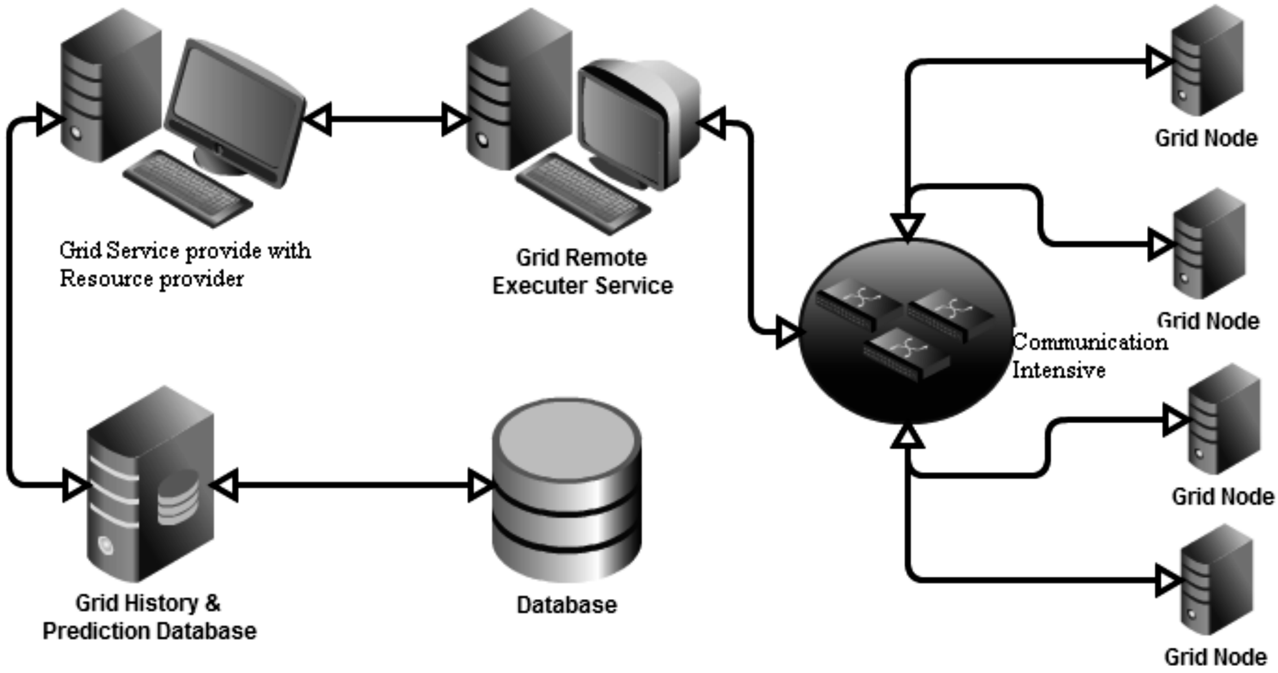

Fig. 2. Communication Intensive Grid Nodes

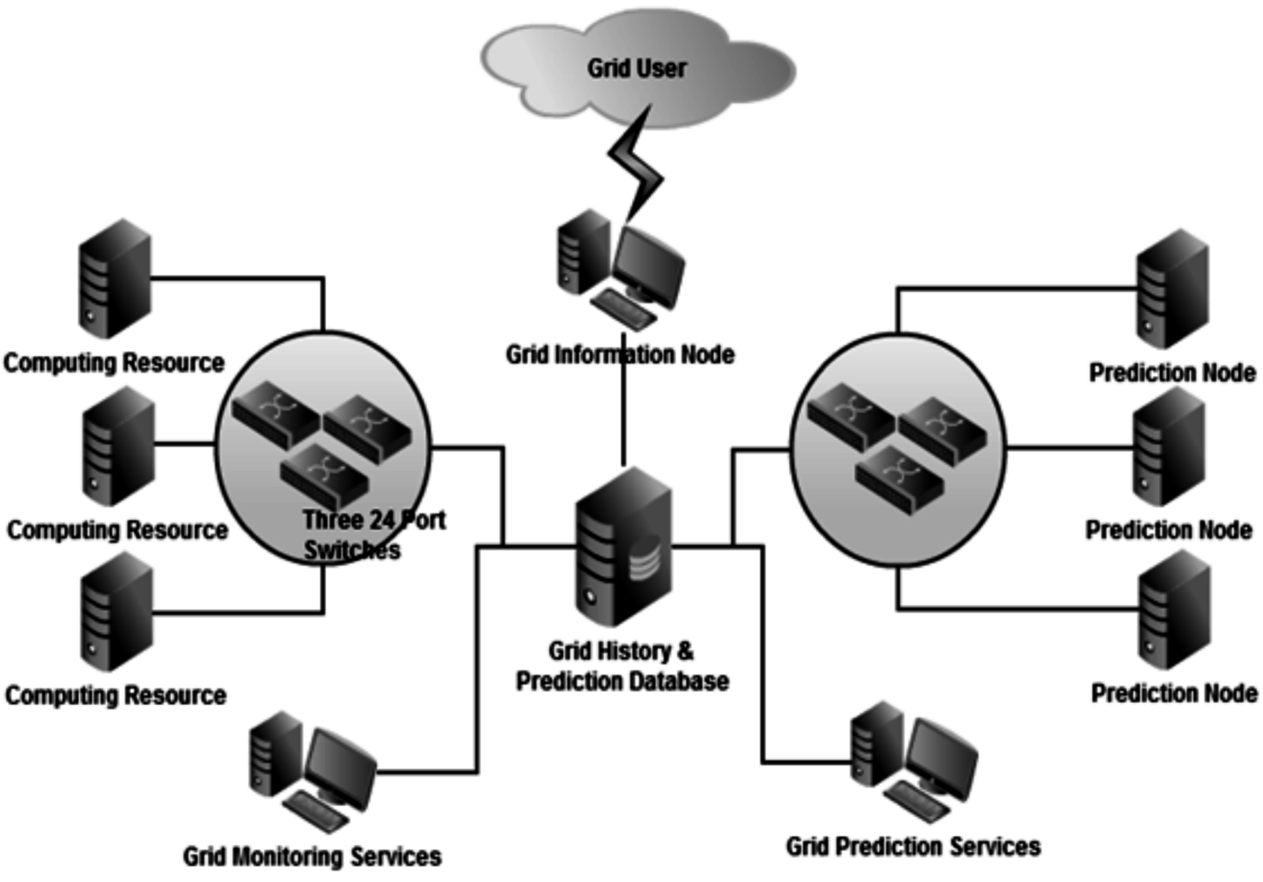

Fig. 3. Grid Resource Prediction pattern Architecture

the History and Most Prediction data for each Resource and find the future of most available Prediction Resource.

V. SYSTEM ARCHITECTURE

In our Proposed Gird Architecture system, we combine the Grid Resource Prediction Pattern (GRP) with perfect fault tolerant resource on most iteration from hybrid particle Swarm Optimization (PSO). This will give us to better quality approach to find the Best Grid Node. In "Fig.4", we have almost all the components are combined together to realize the proposed Grid Computing Architecture and experiment will be performed to measure its performance. Vendors of grid differ in their capabilities, product offerings, and maturity. In grid node selection, Grid balancing is the main aspect to avoid a situation where some nodes are heavily loaded while others are idle or doing little work.

A. GRP - POS Algorithm to find the perfect Grid Computing Node

Every particle has its own cost value, which will be evaluated and minimized through Function. Every Particle has velocity, which will be evaluated and Give a direction to particle that can fly through it.

Algorithm: 1 Calculation for Cost evaluation for Grid Computing Resource

Step 1:

Start and Initialize every Particle with Random Position and velocity vectors.

```
// Init a,b with random position in [-3,3]:
double a = Random.NextDouble() * 6 - 3;
double b = Random.NextDouble () * 6 - 3;
double[] particlePos = { a, b };
double va=Random.NextDouble()*6-3;
double vb = Random.NextDouble() * 6 - 3;
double[] particleVel= {va, vb };
```

Step 2:

Start finding the cost based on Each Particle using Random values like a,b.

Calculate Cost by passing a, b Param Value and return the cost from Random values a, b.

```
calculateCost (double a, double b)
{
return 100 * (a * b - b * b) * (a * a - b * b) + (1 - a) * (1 - a); }
public double CalculateCost1(double a, double y)
{ double c =
3 * (1 - a) * (1 - a) *
Math.Exp(-a * a) -
(b + 1) * (b + 1) -
10 * (a / 5 - a * a * a - b * b * b * b * b) *
Math.Exp(-a * a - b * b) -
1 / 3 * Math.Exp(-(a + 1) * (a + 1) - b * b);
return 1 - c; }
```

Step 3:

Use the history data and Grid Resource Prediction Pattern Data (GRP) from the Prediction database to find the pbest value.

Step 4:

Match this Current Cost (C) with Best value so far (pbest), When Cost (C) is better than pbest value assign it pbest using Cost(c).

```
For each Particle Position Evaluate cost(C)
If Cost (C) is better than pbest then
pbest = C
```

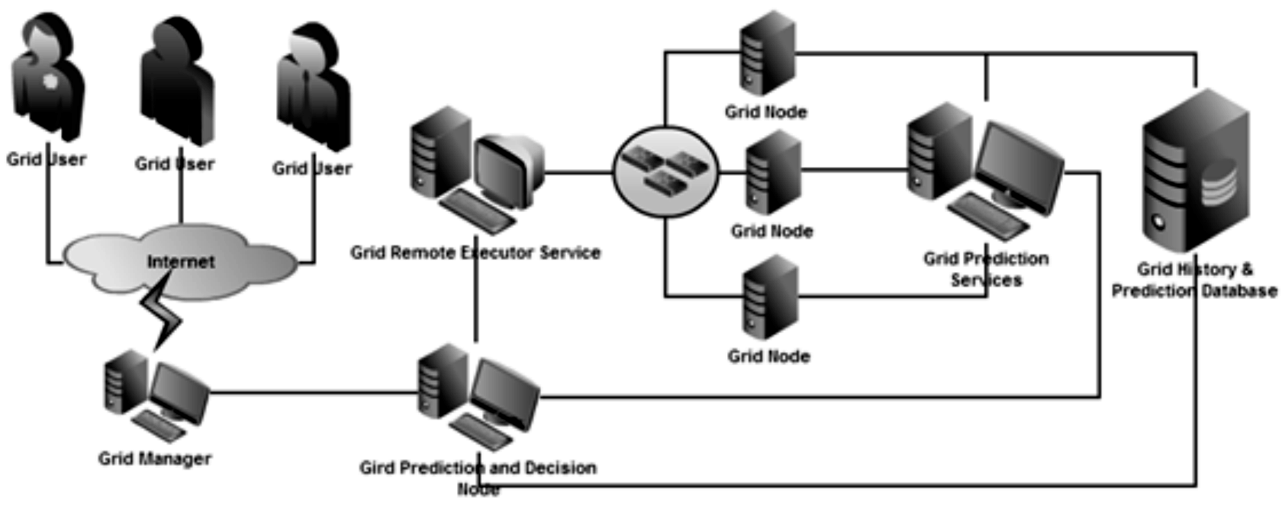

Fig. 4. Perfect fault tolerant resource selection grid architecture

End if
Loop

Step 5:

Result of pbest will be considered to be lowest cost of all particles. Assign the pbest lowest cost value to gbest.

Gbest = pbest

Step 6:

NewVel = New Velocity
CurVel = Current Velocity
Ran1, Ran2 = Random Number in the interval [0,1]
Coeff1, Coeff2 = Accelerated Coefficient
CurPos= Current Position
NewPos= New Position
Calculate the New velocity using below formula
NewVel = CurVel + Coeff1.ran1. (pbest-a) + Coeff2.ran2. (gbest - a)

Step 7:

Calculate the New Position using below formula
NewPos= CurPos+NewVel

Iterate the about formula to find the best position with Minimum cost value.

B. Fault tolerant resource selection: Pi Number Calculation

An algorithm for calculating the digits of the pi number is "Fabrice Bellard". The algorithm has a method that gives an integer as an argument and returns a string containing 9 digits of the pi number digits from a given integer. Resource provider gives an array if IP addresses of executers, and after clicking the "Start" button, it sends the code for the pi calculation algorithm to each of them and requests to calculate some pieces of the pi number. QOS parameters are Dynamic pricing, Fault tolerance, Improved accuracy, Communication intensive, CPU speed, hard drive space, Distance , Bandwidth, Resource quality, Virtual machine capability, Good performance, User friendliness, Previous history, Latest version, Power capability, backup capacity, Grid network support, cooperation facility, High speed, Debugging tools, Alternate resource suggestion, Policy of resource, Customer's previous feedback. Based on this Qos values select the perfect resource for a multi task. Rank the available grid nodes based on their QoS values (mainly time & resource consuming), and selects the best performing ones. To improve the accuracy of grid ranking through following techniques, Data smoothing,Random walk,Matrix factorization Utilizing the content information. Failure-Aware Resource Selection managed by GRAM (Grid Resource Allocation Manager). In grid network, consider the communication performance between different grid nodes with Qos values for perfect grid selection.

Algorithm: 2 Faults Tolerant Grid Resource Selection

Input: Set of available resources {R}
Output: Resource selection from requirements

Step 1:

Detect the possible faults at the time of resource selection.

A fault tolerance service deals with the various faults like Resource failures, Process failures, Processor failures, Task failures & Network failures.

Step 2:

Design and implement a fault detector and a fault manager.

Fault detector detects the occurrence of resource failures and the Fault manager guarantees that the submitted jobs executed with optimal resources.

Step 3:

Compute the Resource based on high ranking ,high priority, failure aware, less bandwidth between submitting node to executing node, more user satisfaction, well hybrid grid network topology, perfect resource utilization, Reducing Energy Consumption, less resource consumption, minimum dynamic pricing and better policy.

Fault tolerances grid system gets high resource Availability.

VI. EXPERIMENTS

This work uses the .NET Remoting for communication between a service provider, resource provider and the Executors. For getting away from firewalls and Security boundaries, we are used the HTTP channel and the SOAP formatter through Grid security certified key. All resources in the grid network connection are design from weighted graph in data structure.

6.1 Evaluation 1

From the six different service select the perfect fault tolerant service based on Hybrid GRP-PSO Algorithm. Here X-axis represent the Number of Iteration process Y-axis represent the cost or dynamic price. Predict the Minimum dynamic price Value from the graph.

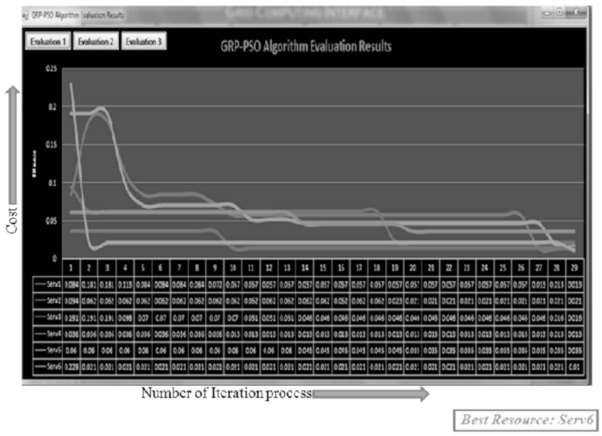

Fig. 5. Evaluation 1 for Minimum dynamic price Value prediction

Evaluation 1 for Minimum Cost Value:

Input Value:

Random Position, Random Velocity

No of Resources–6 (Serv1, Serv2, Serv3, Serv4, Serv5, Serv6)

pbest – Value from the History of Grid Resource Prediction Pattern (GRP)

Output:

No of Iteration – 29

Best/Minimum Cost – 0.01

Best Resource with Low Cost – Serv6

6.2 Evaluation 2

Same like Evaluation 1 predict the best cluster for a given task from the lot of available cluster.

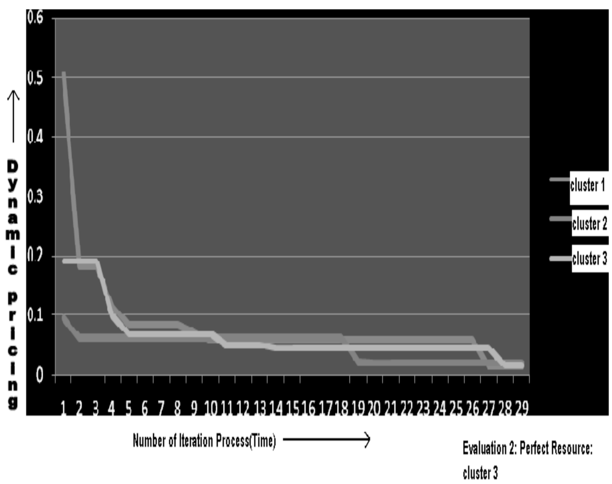

Fig. 6. Evaluation 2 for Minimum dynamic price Value prediction from clusters

6.3 Evaluation 3

Same like Evaluation 1 predict the best resource for a given task from lot of resources in the grid network.

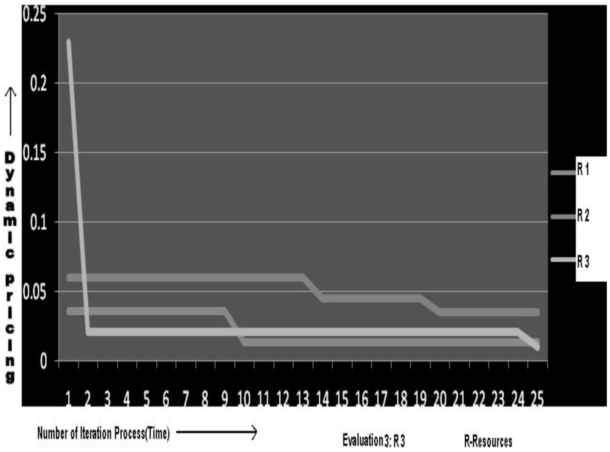

Fig. 7. Evaluaion 3 for Minimum dynamic price Value prediction from resources

6.4 Performance comparision of resource Executor:

Performance of algorithm based on two factors such as Time & Space Complexity, Same like Grid Application/algorithm based on time & Space Complexity. Recently one more factor added for performance evaluation such as Energy complexity. In grid computing, Energy is dependence on High QOS Value for grid resource. Based on the task 1, compare the performance of executor 1, executor 2, and executor 3 in the graph. Here x axis represent Time Complexity & y axis represents Energy complexity. Find the High energy with low time for task 1.

VII. CONCLUSION

Selection of the perfect Grid node for task prediction in the dynamic grid computing provides a best Resource selection for jobs and hence yielding the best result. In this research, we have used Grid Resource Prediction Pattern and hybrid Particle Swarm Optimization Algorithm & effort starts from analysis and evaluation to representative of Grid service provider, Grid resource provider, Remote Executor service and hybrid Grid resource Prediction service. Evaluation result are indicate the finding the best Grid Node to perform the task on the better prediction. Resource manager finds the optimal set of resources to

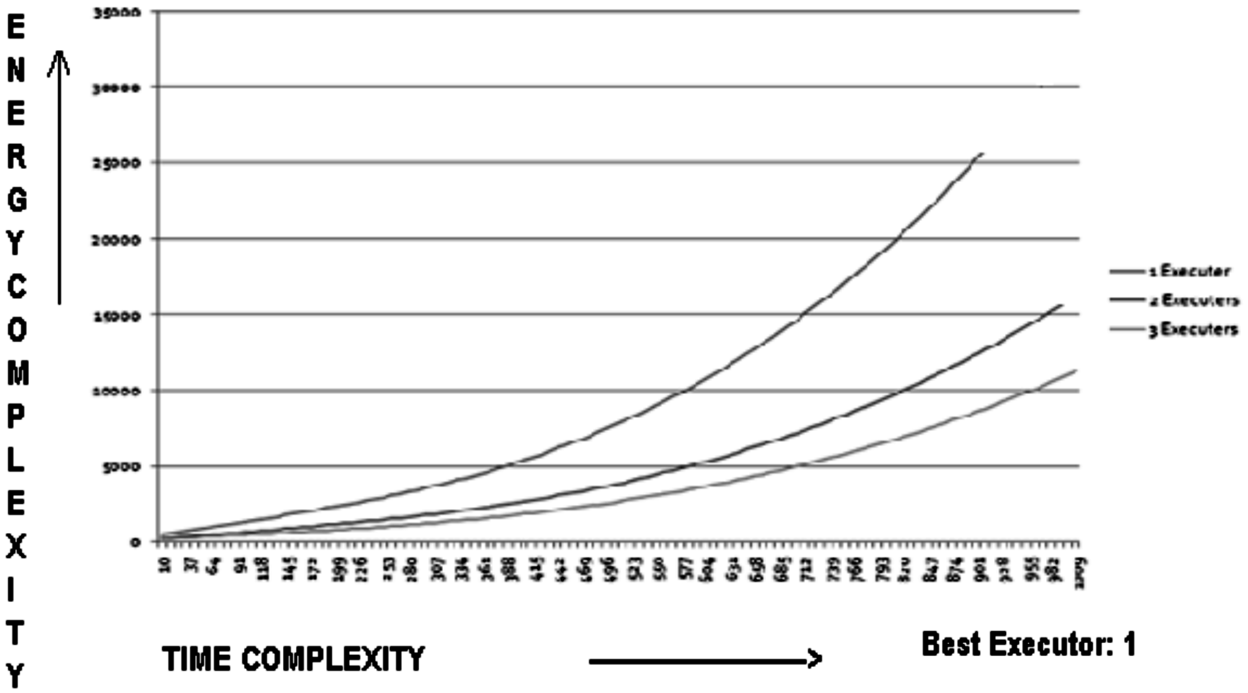

Fig. 8. Performance comparision of resource Executor

guarantee efficient task execution, the fault tolerance service guarantees that the submitted tasks are completed, and task execution is perfect even if some failures occur. High user satisfaction and correct resource utilization based on dynamic Energy-Efficient grid balancing technique. Iteration results indicate the best level of prediction of node can be achieved using Hybrid GRP-PSO.

REFERENCES

[1] Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal," Alchemi: A .NET-based Grid Computing Framework and its Integration into Global Grids", IEEE, 2004

[2] Ciprian Dobre , Ramiro Voicu , Iosif Legrand 2011., Monitoring Large Scale Network Topologies", IEEE.

[3] Jingbo Yuan, Shunli Ding, Jiubin Ju Liang Hu 2005., Prediction of the Running Time of Tasks Based on Load, IEEE.

[4] Liang Hu, Xi-Long Che, and Si-Qing Zheng, 2011.Online System for Grid Resource Monitoring and Machine Learning based Prediction, Parallel and Distributed Systems, IEEE Transactions on VOL. 23, Issue: 1, pp. 134 – 145.

[5] P. A. Dinda, and D. R. O'Hallaron, 2000. Host load prediction using linear models, Cluster Computing, vol. 3, no. 4, pp. 265-280.

[6] Pei Fan, Ji Wang, Zibin Zheng, Michael R. Lyu. 2011., Toward Optimal Deployment of Communication-Intensive Cloud Applications, IEEE, 4th International Conference on Cloud Computing, pp. 460-467

[7] T.H. Cormen. Introduction to algorithms. The MIT press, 2001.

[8] Z. Pooranian,A. Harounabadi , M. Shojafar and J. Mirabedini, 2011., Hybrid PSO for Independent Task scheduling in Grid Computing to Decrease Makespan, International Conference on Future Information Technology ,IPCSIT vol.13 IACSIT Press, Singapore, pp.327-331

[9] Asgarali Bouyer, Mohd Noor MD SAP. 2010., A Prediction-based Fault Tolerance on Grid Resources scheduling by using Optimized Case-based Reasoning, pp.1-9, IEEE