Memory and CPU Power based Automatic Fault Tolerant Framework in Alchemi Computational Grid

Rajesh Kumar
Ph.d Research Scholar
Department of Cse
JJT University, India

Surender Jangra, PhD
Assistant Professor
Gtb Collage
Bhawanigarh, India

ABSTRACT

Computational Grid processing is defenseless against numerous manual dangers. On the off chance that the desktop is bargained, information misfortune, or even finish control of the desktop, could happen. The foundation presently sent in most network situations tends to a great degree complex. In past work heaps of routes proposed to taking care of the computational Grid if there should arise an occurrence of fault and failure in grid. The primary disadvantage is arrangement given worked physically. There was no effective solution which can deal with the fizzled computational grid automatically at failure time. We Selected the Alchemi grid middleware due to its support for Microsoft window platform. Our Research Paper is to give productive Framework which will ready to handle computational network automatically if there should be an occurrence of Failure in a computational grid middleware. This advancement of productive framework will change over manual path into programmed way based on the Processing power and Memory. It will therefore work to give programmed execution occupation even after middleware offer powerlessness to take care of issue. Execution procedure can begin on remote node in view of the present Memory status and Processing Power status. Straightforward memory and processing Power control makes the computerized structure more influential. Proposed mechanized system give reaction in milliseconds to control the failure in execution Process. This Research work is capable to maintain the existing processing power in computational grid middleware. Testing of proposed framework on Peer to Peer Node in Alchemi desktop middleware in Local Area Network.

Keywords

Computational grid, Alchemi, Automatic, Microsoft window

1. INTRODUCTION

"Conveyed" or "grid" registering as a rule is an extraordinary kind of parallel figuring which depends on complete PCs (with installed CPU, stockpiling, power supply, arrange interface, and so on.) joined with a system (private, open or the Internet) by a customary system interface, for example, Ethernet. This is as opposed to the conventional thought of a supercomputer, which has numerous processors joined by a neighborhood rapid PC bus. Grid frameworks are named Computational, Data and Service grids. Computational Grid class means the frameworks that have a higher total Computational limit accessible for single applications than the limit of any Constituent machine in the framework. Computational grid systems are tremendously complex and prone to failures [10]. This Research paper identified with Automated computational grid in the event of failure in execution process via Proposed programmed system.

Information Grids the asset administration framework oversees information appropriated over Geographical areas. Information Grid is for frameworks that give a foundation to Synthesizing new data from information vaults, for example, advanced libraries or Data Warehouses that are appropriated in a wide range system.

Service Grid classification is for the frameworks that give benefits that are not gave by any single machine. This class is further subdivided in On Demand, Collaborative and Multimedia Grid Systems.

ALCHEMI (.Net grid computing structure)

Alchemi is an open source structure that permits setting up a computational desktop grid to easily total the figuring force of arranged machines into a virtual supercomputer (desktop network) and to create applications to keep running on the grid[1,3] . It has been planned with the essential objective of being anything but difficult to use without relinquishing force and adaptability. Alchemi incorporates:

a) The runtime hardware (Windows executables) to build computational frameworks.

b) .NET API and devices to create .NET grid applications and grid empower legacy applications.

By utilizing alchemi to set up a computational framework environment where to test the casing work. Information base is kept up by sql fundamental server and also recuperation server. There are four sorts of circulated segments (hubs) included in the development of Alchemi frameworks and execution of grid applications: Manager, Executor, and User and Cross-Platform Manager [1,3]. A network is made by introducing Executors on every machine that is to be a piece of the framework and connecting them to a focal Manager segment. The Windows installer setup that accompanies the Alchemi conveyance and insignificant arrangement makes it simple to set up a framework [2,4]. An Executor can be designed to be committed (which means the Manager starts string execution specifically) or non devoted (which means string execution started by executor).Non committed agents can work through firewall and Nat servers since there is one and only route correspondence in the middle of agent and trough. Committed execution is more suited for intranet environment and non devoted for web environment. An assortment of nodes can execute applications that encourage can carry out on quick and costly supercomputer [9].

2. RISK IN COMPUTATIONAL GRID MIDDLEWARE (ALCHEMI.NET)

Computational grids are an outstanding proposal for parallel computing, due to their cost/performance ratio. Dependability is a main question in this type of platforms [5]. As the dependability of individual Grid assets may not be able to be assured [6]. Physical security is a vital segment of a data frameworks general security stance. A venture Grid, similar to any data framework, needs to shield against physical dangers from people (either pernicious or unintentional) and in addition man-made and characteristic calamities. Having one endeavor framework rather than various storehouses. Frameworks may prompt some productivity in tending to physical security dangers. Social designing portrays how an unapproved client cons an approved client into giving data expected to get to a framework. A venture framework does not as matter of course present new types of social dangers, and may even lessen these dangers if an endeavor grid is taken off with a solid arrangement of controls (e.g. a reasonable administrator model, procedures, and strategy). Alchemi.net is middlware.It is capable to add the accessible Computational Power in the current computational resources. Problem in the Execution Process on remote hub can damage the accessible computational power [8]. So it is exceptionally important to put control on the execution process in Alchemi middleware.

Automatic Administrative system guarantees that the capacities being performed by the grid and its clients are not damaging any faults and failure in Alchemi Middleware. The automatic necessities that are inside of the regulations ought to be joined to framework approaches [7].

Presently, the primary center of the most generally sent automatic arrangements if there should arise an occurrence of failure of computational framework. Numerous answers for this issue are given in past exploration, yet constraint is that arrangement worked physically not automatically. We required a compelling arrangement which ought to works automatically rather than physically if there should be an occurrence of fault or failure of computational network. so in this paper we are going portray such kind of strategy which works absolutely programmed way, if there should arise an occurrence of failure in execution node of a computational grid(alchemi.net).If the focal supervisor comes up short, then the agents are advised to join with the new chief. Be that as it may, this assignment performs physically in past. There must be some procurement so that on failure of executor will automatically provide the executor. The improvement of a usable lightweight system that can be sent in existing fault situations in Alchemi.net. In computational grid middleware (Alchemi.net) then our proposed framework will take control on execution node automatically. The primary goal is our work to make framework completely programmed by evacuation of manual framework to control the execution faults in Alchemi desktop middleware.

3. DEVELOPMENT OF AUTOMATIC CONTROL SYSTEM

3.1 Development of Component for checking the Process on distant node

This module oversees checking of remote hub process. To begin with it will test to remote hub selected in execution. In case essential machine not reply specifically time then it will automatically show the present status of node on the GUI Interface of Proposed system. The graphical client interface will demonstrate the reasonable status of machines in Alchemi desktop middleware. It will show two sorts of asset. To start with is consistent and second is deficient Resource. Proposed System will repeatedly refresh the Status of Node in 30 seconds.

3.2 Development of Component for Automatic control on execution Node

This module will handle control on the remote execution point. Modified control on execution points is relentless to take control on the Existing taking care of Power in Alchemi desktop middleware. Execution system can start and stop in fraction of seconds.

3.3 Development of Component for Automatic status of Memory Available in Remote node

This module will handle control on the remote execution for discover the present memory status. On the premise on the memory accessible in node. Agent will begin execution process on remote node in milliseconds. Inclination of node relies on upon the accessible memory.

3.4 Development of Component for Automatic status of Processing Power Available in Remote node

This module will handle control on the remote execution for discover the present processing power status. On the premise on the Processing power accessible in node. Agent will begin execution process on remote node in milliseconds. Inclination of node relies on upon the accessible Processing Power.

4. TESTING OF FRAMEWORK

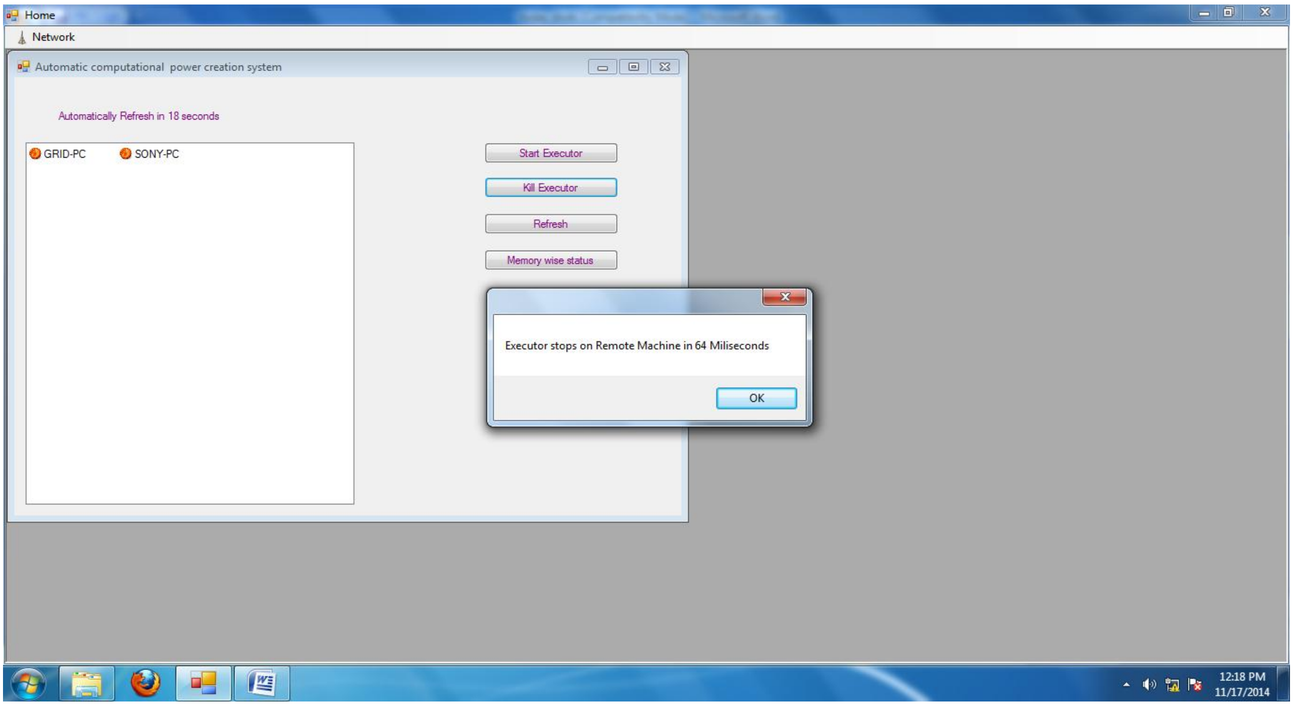
Testing of proposed work by make a computational desktop framework environment by using Alchemi on LAN. By make failure in remote execution node. System is tried On Peer to peer in LAN. Misfire by various ways we test that how framework handle computational grid (Alchemi.net).Proposed framework give Response in milliseconds. It will beat the manual needs to take control on execution Processes. In first case test is performed, when out of two nodes one node was blemished. Proposed Framework has finished it errand to begin execution process in portion of seconds. This Task is impractical in manual endeavor in portion of seconds. Second test is performed by making the both nodes imperfect. Framework extracts the present status of Available memory and processing power. On premise of estimations of memory and processing power, Framework performed proper require task in milliseconds.

Third case is performed by change the distance from the source node to remote node. Proposed framework finished its proper errand in part of seconds (milliseconds).There is no significant change in time when structure attempted to amend the inadequate node.I.e for node at a separation of 20 meter

Reacted in 56 milliseconds and for node at separation of 50 meter reacted in 53 milliseconds. It will uproot the manual

Prerequisites to evacuate the deficiencies and failure in Alchemi desktop grid middleware in local area Network.

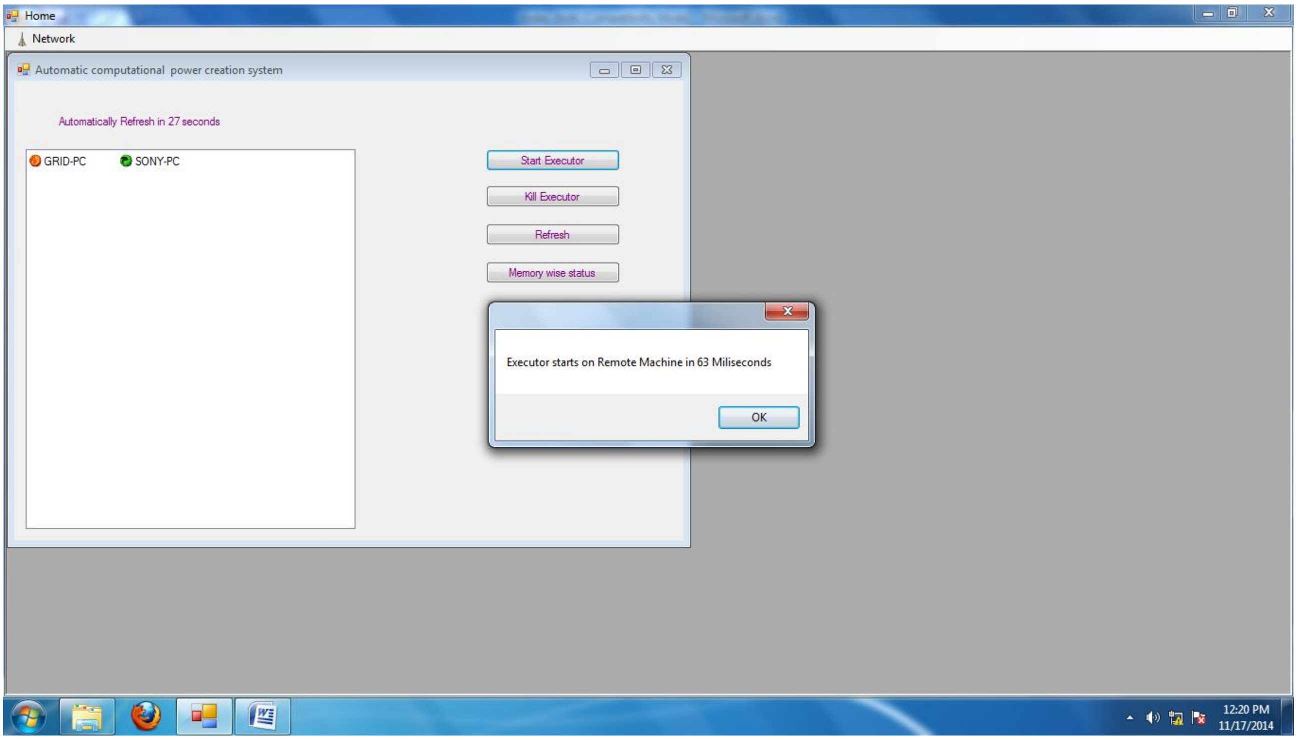
5. RESULTS



Automatically Refresh in 18 seconds

Executor stops on Remote Machine in 64 Miliseconds

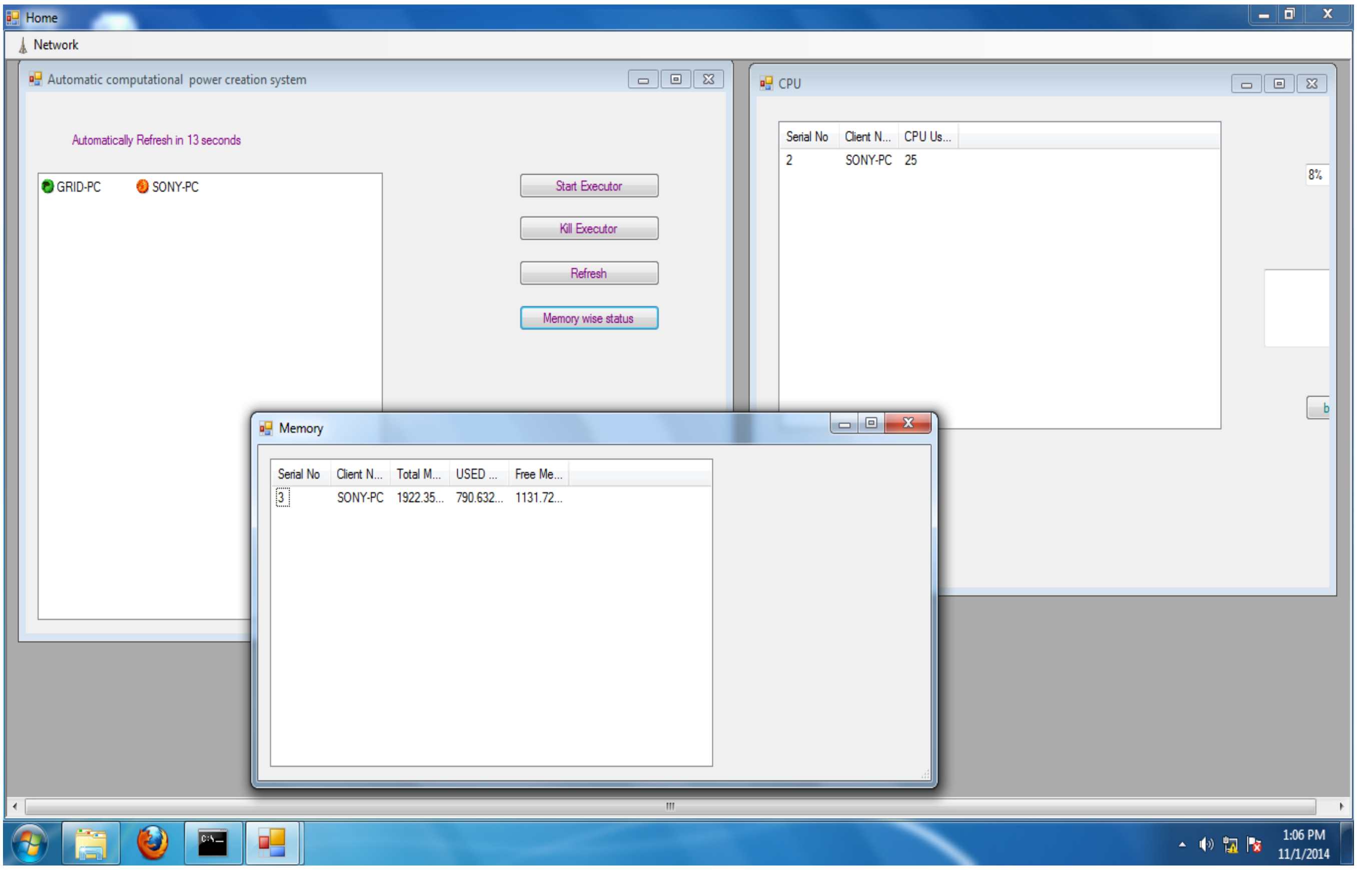
Fig 1: View of Executing machine executor stop by Proposed Framework on Local Area Network in sixty four Milliseconds



Automatically Refresh in 27 seconds

Executor starts on Remote Machine in 63 Miliseconds

Fig 2: View of Executing machine executor start by Proposed Framework to provide the Processing power on Local Area Network in Sixty Three Milliseconds

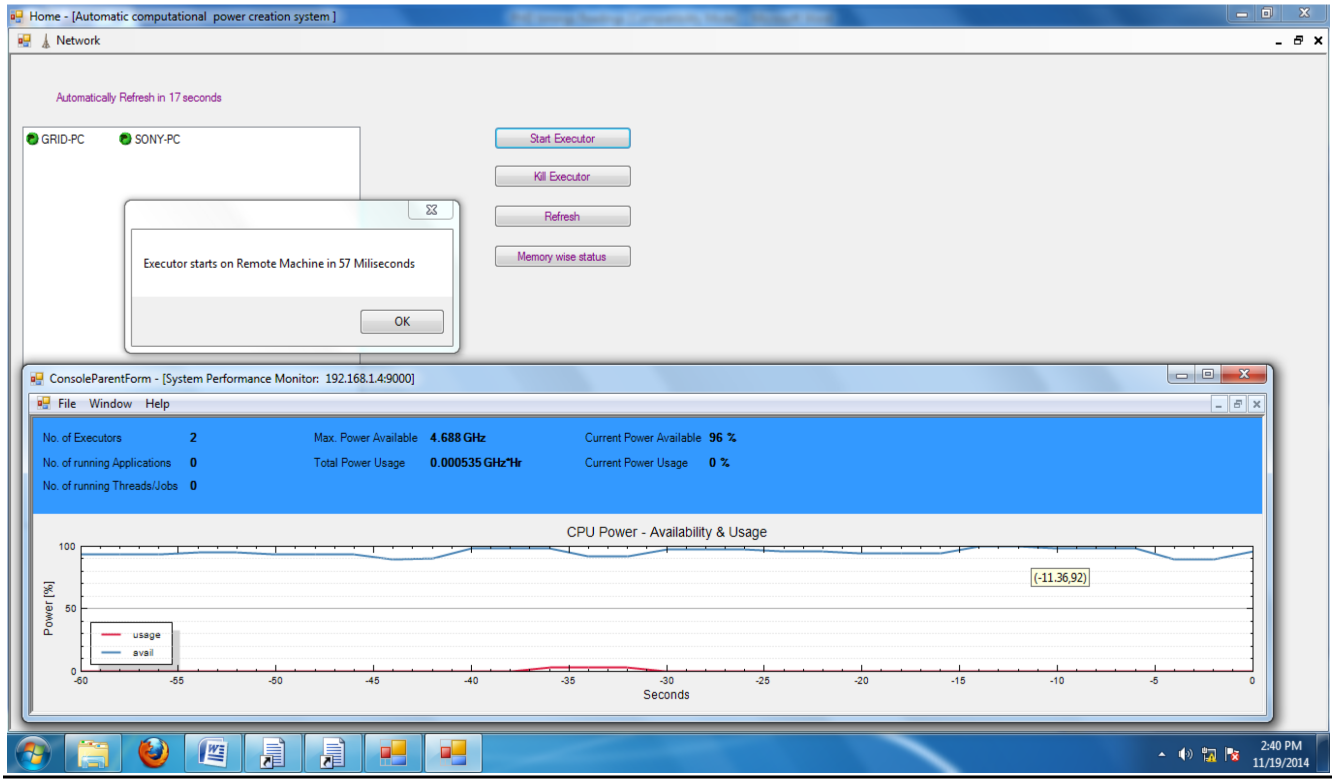


Automatically Refresh in 13 seconds

Serial No	Client N...	CPU Us...
2	SONY-PC	25

Serial No	Client N...	Total M...	USED ...	Free Me...
3	SONY-PC	1922.35...	790.632...	1131.72...

Fig 3: View of Automatic Fault Tolerant control system viewing the current Processing Power and Memory



Automatically Refresh in 17 seconds

Executor starts on Remote Machine in 57 Miliseconds

No. of Executors 2
No. of running Applications 0
No. of running Threads/Jobs 0
Max. Power Available 4.688 GHz
Total Power Usage 0.000535 GHz*Hr
Current Power Available 96 %
Current Power Usage 0 %

Fig 4: View of Automatic Fault Tolerant control system viewing with Alchemi console Inhance existing processing power by 2.195 GHz

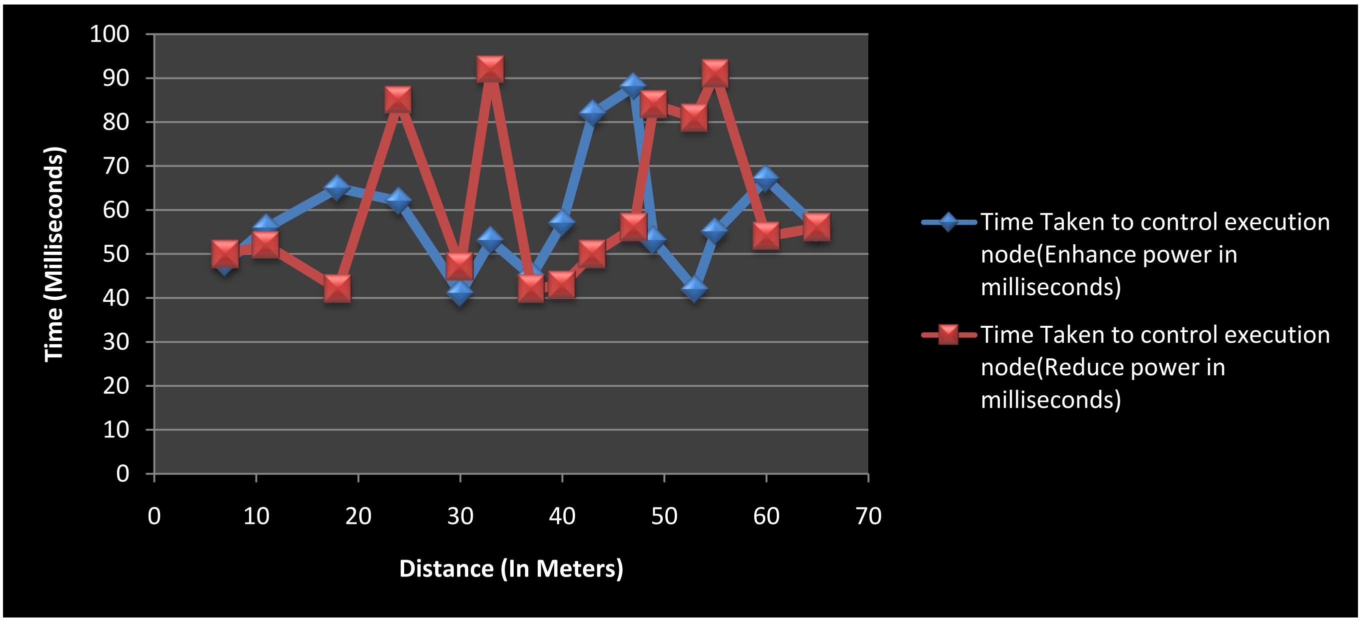


Fig 5: Graphical View of Automatic Fault Tolerant control system Distance versus Response time in Alchemi Desktop Middleware.

Table 1. Response Time by Proposed Automated Framework

S.no	Distance in Peer to Peer node (In Meters)	Time Taken to control execution node(Enhance power in milliseconds)	Time Taken to control execution node(Reduce power in milliseconds)
1	7	48 ms	50 ms
2	11	56 ms	52 ms
3	18	65 ms	42 ms
4	24	62 ms	85 ms
5	30	41 ms	47 ms
6	33	53 ms	92 ms
7	37	45 ms	42 ms
8	40	57 ms	43ms
9	43	82 ms	50 ms
10	47	88 ms	56 ms
11	49	53 ms	84 ms
12	53	42 ms	81 ms
13	55	55 ms	91 ms
14	60	67 ms	54 ms
15	65	56ms	57ms

6. CONCLUSION

Today the utilization of conveyed computational Power to make a virtual parallel machine is an unavoidable certainty. However the utilization of such frameworks acquaints new issues with the superior group. This exploration work means to add usefulness to the computational network environment utilizing automatic framework to control the faults. In the field of computational grid, programmed strategy give a productive path rather than manual methods .When a running computational grid get to be down due to execution faults ,jobs execution will affected, network information will likewise in danger around then. So such proficient automatic methods which can take control upon execution faults consequently where jobs execution can again start quickly is better choice for computational grid. Proposed automatic framework performs each and every task automatically rather than manually. It decreases the recuperation time when contrasted with manual way. Automatic fault free working of running computational grid can be kept up by productive proposed programmed system. Loads of conceivable outcomes by which our work can be stretched out in future. We actualize system in Alchemi.net middleware environment. Programmed casing work can use in another network environment. We tested this Proposed Framework in window based environment using alchemi.net. Different automated Developments made for control the faults and failure in execution processes in Alchemi Desktop Middleware. Proposed Automated Framework will valuable for control of the faults and failures in Alchemi middleware. Quick Response time (i.e. milliseconds) for control faults & failure will overcome the manual needs in computational grid.

7. REFERENCES

[1] Akshay Luther, Rajkumar Buyya, Rajiv Ranjan & Srikumar Venugopal, Alchemi: A .NET-based Grid Computing Framework and its Integration into Global Grids, University of Melbourne, Australia, December 2003

[2] Akshay Luther, Rajkumar Buyya, Rajiv Ranjan & Srikumar Venugopal, Peer-to-Peer Grid Computing and a .NET-based Alchemi Framework , High Performance Computing: Paradigm and Infrastructure, Laurence Yang and Minyi Guo (editors),ISBN: 0-471-65471-X, Wiley Press, New Jersey, USA, June 2005

[3] Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal, Alchemi: A .NET-Based

Enterprise Grid Computing System, Proceedings of the 6th International Conference on Internet Computing

[4] http://www.gridbus.org/~alchemi/files/1.0.beta/docs/AlchemiManualv.1.0.htm

[5] Fault Tolerant MPI for Clusters and Grid Environments Hernani Pedroso Critical Software SA Rua Pedro Nunes, 3030-397 Coimbra Portugal

[6] Paul Townend and Jei Xu, "Fault Tolerance within a Grid Environment"Proceedings of AHM2003

[7] K.Rajesh, J.surender, "Automated Fault Tolerated system for control computational Power in Desktop Grid""IEEE International Advance computing Conference, Banglore", Conference No-35547

[8] K.Rajesh, J.surender, "High Computational Power: A Research Challenges in Desktop Grid Computing", AEMDS, publisher Elsevier India, isbn-9789351070573 chapter 75, PP489-492.2013

[9] V.Monika, C.zoran, N.Catalina, "Availability of computational Resources for Desktop grid computing".Vol.LXI no 1/2009, 71-76.

[10] M. Raissa, C. Walfredo, "faults in grid, why they are so bad and what can be done for it", @dsc.ufcg.edu.br