



Reducing Energy Consumption in Smart System through Mobilouds Framework

Mr. Muneshwara M.S

Dept. of CS&E, BMSIT&M,
Yelahanka, Bengaluru-566064,
Karnataka, India

Dr. Anil G.N

Dept. of CS&E, BMSIT&M,
Yelahanka, Bengaluru-566064,
Karnataka, India

Dr. Thungamani M

Department of CS&E , COH UHS
Campus GKVK, Yelahanka,
Bengaluru -560065, Karnataka

ABSTRACT

Mobile cloud computing (MCC) it's related energy implication are seen everywhere in large-scale. Offloading computationally raised endeavor's to the cloud datacentres being the basic thought driving MCC (Mobile Cloud Computing), a vast segment of the mobile terminal assets partaking in the MCC (mobile Cloud Computing) aggregate execution are wasted as they remain sit out of gear till the mobile terminals get response from the datacentres.. This is an additional wastage of assets near to the cloud assets are starting at now being tended to as colossal vitality customers. Despite the fact that the assets consumed of the site without moving mobile resources are unimportant in contrast with the cloud partner, such utilizations impact sly affects the mobile devices bringing about superfluous battery channels. Mobilouds which consolidate a multi-level processing architecture with various phases of process cluster limits and a product application to supervise vitality utilization. Mobilouds framework energizes the mobile device co-operation in the MCC (Mobile Cloud Computing) synchronized effort execution, there by lessens the weight of idle mobile resource and uses such idle resource in the actual job execution.

Keywords: *Cloud computing, mobile cloud computing, mobilouds*

I. INTRODUCTION

The cloud is an arrangement of various sort of software and hardware that works join to conveyance numerous things of registering to the end user as an

online services. Cloud computing (CC) is the utilization of hardware and software to reaction an administration over a system with cloud computing, users can get to any record and any application from any gadget that can get to the web. Mobile Cloud computing (MCC) is the consolidated approach of cloud computing (CC), Mobile Computing (MC) and wireless network (WN) to convey better computational asset to mobile user, network operators and in addition cloud computing suppliers (CCS). Mobile Cloud Computing (MCC) is an incorporated system that consolidates cloud datacenters, mobile devices and correspondence framework. Mobile Cloud Computing (MCC) organizations are broadly utilized as a part of different applications, for example, e-learning, Tele-checking, Tele-surgery, IT, business administrations and so forth.

Mobile Cloud Computing (MCC) advantage models assemble a complicated association between foundation suppliers, application and specialist organizations, designers and end-clients. Framework suppliers generally it will incorporate some additional components for equipment and programming organizations; application and specialist co-ops are responsible for executing customer requested organizations; engineers are all around who make applications being encouraged on the cloud datacenters; and end-clients are the shoppers of the Mobile Cloud Computing (MCC) organizations. The end-clients of the Mobile Cloud Computing (MCC) administrations they don't have the advantage of control over the subordinate foundations, for instance,

equipment, arrange, servers and so forth., yet they can have an aggregate control over the applications sent by them. The achievement of such a Mobile Cloud Computing (MCC) structure depends on upon the predictable blend of committed equipment and programming assets of the three core technologies.

II. LITERATURE REVIEW

Alzharani et al [1] discussed outline of Mobile Cloud Computing (MCC) rewards, drawbacks. The authors have likewise talked about significance of versatile cloud application and highlights of the portable distributed computing open difficulties. Mathew et al [2] investigates a portion of the specifics of these wellbeing and wellness application. And furthermore author presented advancement calculation as an instrument to effectively handle information point locally by mobile devices. This calculation can take focal points of the nearby preparing energy of mobile devices and diminishes correspondence taken a toll between versatile endpoint and cloud based long haul information administrations. Kitanov et al [3] discussed about a method to solve the problem related to throughput, low-latency (delay), high mobility (speed) and high capacity. The author have also discussed about importance of mobile cloud application, services and current research trends. Sanaei et al [4] discussed about a method to solve the problem related to heterogeneity in convergent figuring and systems administration (wired and remote systems) and separation it into two measurements to be specific vertical and horizontal. The author have also focused on issue related to impact of heterogeneity in MCC are researched and overwhelming heterogeneity dealing with approach like virtualization, middleware and benefit situated design. Abolfazli et al [5] discussed about problem related to mobile augmentation domain and present taxonomy of CMA approach and also main objective like effect of remote resource on the quality and dependability of increase prepare and utilizing changed cloud based asset in expanding mobile devices. The author likewise break down the condition of workmanship CMA approach. Lose et al [6] discussed about problem related to mobile device do not need high end resource (e.g., processing speed, storage and memory capacity) since all the data and complex computing can be offloaded to the cloud and cloud will perform action on that data and response will give back to user. The author also focused on cloud computing infrastructure to augment the use of

mobile phones in information and communication technology for development and also services that cloud can offer to improve and support the use of mobile phone. Satyanarayanan et al [7] discussed about problem related to specialized snags to this change and versatile client misuses virtual machine innovation to quickly instantiate redid benefit programming on a close-by cloudlet and afterward utilizes that administration over a remote LAN the cell phone ordinarily work as a thin customer regarding the administration. Creator additionally talked about cloudlet is a put stock in, asset rich PC or group of PC that is very much associated with the web and accessible for use by adjacent cell phones and furthermore utilizing cloudlet likewise disentangles the difficulties of taking care of the pinnacle transfer speed demand of different clients intuitively producing and getting media, for example, superior quality video and high determination image. Oureshi et al [8] discussed about the problem related to the information handling, stockpiling and other escalated operation. The creator additionally centered on best in class versatile distributed computing and its execution strategies. Ravi et al [9] discussed about the problem related to the correspondence overhead, offloading of utilization execution to cloud customer more vitality than executing in the gadget itself. The creator additionally centered around the structure for vitality proficient consistent administration with highlight like, associating heterogeneous cell phone to frame portable impromptu cloud. Administration disclosure in versatile specially appointed cloud and offloading choices. Li et al [10] discussed about the problem related to the key security challenges confronted by green distributed computing condition and outline a virtualization security confirmation design named cyber guarder to address the security problem with consideration of energy efficient. The author also focused on Virtual Machine Security Service (VMSS) incorporating a numeral of novel technique including VMM-based integrity measurement approach for a Netapp isolation mechanism for as user isolation, VM (virtual machine) separation and virtual network separation of multiple Netapps according to dynamic energy efficiency and security needs. Al-Aqrabi et al [11] discussed about a method to solve the problem related to Business Intelligent (BI). The author also focused on cloud facilitating of BI has been proposed with the assistance of reenactment on Op-net which involving a cloud show with different OLAP application servers apply parallel question stacks a variety of servers facilitating social database. Li et al

[12] discussed about the method to solve the problem related to protection and safeguard of IaaS cloud environment and also resource usage. The author also focus on these challenges to overcome this problem, author proposed a new system called Cloud-Mon which empowers dynamic asset allocation. Cloud-Mon provides two kind of technique to maintain high resource efficiency. The first technique is by making use of fuzzy model it establish a complex relation amongst execution and asset requests of a NIDS-VA and builds up an online Fuzzy controller to organize asset distribution for NIDS-VA under shifting system activity. Second one is worldwide asset booking approach for improving the asset productivity of the cloud conditions. Lee et al [13] discussed about a method to solve the issue identified with pivot time and asset squander in cell phones. The author likewise centered on the proposed system can be connected to a more practical distributed computing and furthermore creator explained the calculation of structure and directed a broad arrangement of reproduction with different situations. Liu et al [14] discussed about a method to solve the problem related to extend battery lifetime, increase processing speed and approximately a few critical issue, for example, issue of dead spots or scope openings. The author additionally centered around the errand offloading utilizing self-composed criticality (TOSOC) utilized this technique to take care of the issue identified with dead spots or scope gaps and furthermore issue identified with benefit postpone imperatives. Nir et al [15] discussed about a method to solve the problem related to minimize computational time and energy consumption and also user defined constraints. The author also focused on centralized broker node approach, this approach will solve the problem related to undertaking task to limit the aggregate vitality utilization over.

Mobile Cloud Computing (MCC) can be viewed as an extension that fills the crevice between the limited computing resources of SMD (Storage Module Device) and processing necessities of escalated applications on SMDs. The Mobile Cloud Computing (MCC) Forum characterizes Mobile Cloud Computing (MCC) as takes after: "mobile Cloud Computing (MCC) in any occasion troublesome shape intimates a framework where both the information stockpiling and the information handling occur outside of the framework. Versatile cloud applications move the figuring power and information stockpiling far from framework and into the cloud, bringing

applications and portable registering to not simply mobile devices clients yet rather a liberally more expansive degree of portable endorsers". Mobile Cloud Computing (MCC) has pulled in the consideration of business specialists as a useful and valuable business solution that limits the development and execution costs of mobile applications, enabling mobile user to obtain most recent innovation advantageously on an on-demand basis. Fig. 4.1 demonstrates the general perspective of Mobile Cloud Computing which is made out of three primary parts: the mobile device, wireless communication implies, and a cloud infrastructure that contains data centers. These last give storage services, processing, and security instruments for both the cloud environment and mobile device.

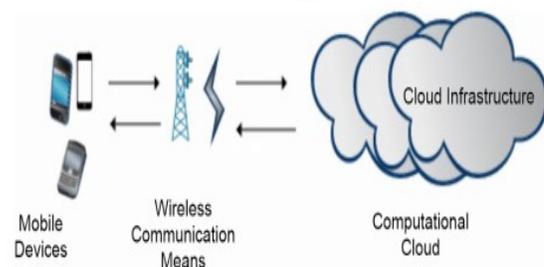


Figure-1: Mobile Cloud Computing

Computation offloading is the errand of sending computation intensive application components to a remote server. As of late, various computation offloading structures have been proposed with a few methodologies for applications on mobile devices. These applications are partitioned at various granularity levels and the components are sent (offloaded) to remote servers for remote execution keeping in mind the end goal to extend and improve the SMD's abilities. Be that as it may, the computation offloading mechanism are as yet confronting a few difficulties

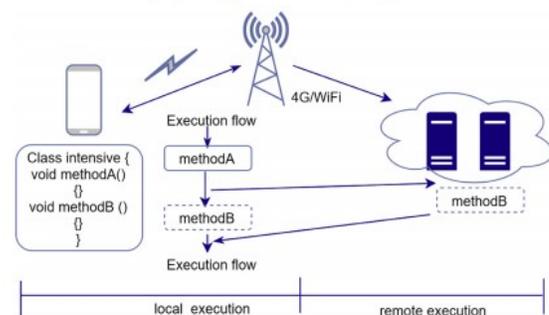


Figure 4.2: computational Offloading

III. PROPODED METHODOLOGY

The proposed solution addresses the issues of vitality consumption brought about by the sit out of gear

mobile assets and designed a Hybrid Energy Efficient MCC (HEE-MCC) framework, named Mobilouds, with the end goal of expanding the involvement of the mobile devices in the collaboration Mobile Cloud Computing MCC job completion at last to minimize the undesirable vitality utilizations of the sit out of gear mobile resource with reduced service delays.

The major contributions of this paper include,

1. A new multi-level process configuration named Mobilouds, made out of various sizes of process cluster for vitality productive Mobile Cloud Computing (MCC) consolidated execution. This Mobilouds framework can be both climbed to a higher limit cluster in the midst of asset deficiency and limited when there are wealth assets in the process cluster with the ultimate objective of limiting vitality usage.
2. The Mobilouds application which is a product procedure passed on to support the functionalities of the Mobilouds framework. This product procedure continues running in the mobile system for processing the asset accessibility in the portable terminals. This Mobilouds methodology serves to picks the ideal process cluster from the Mobilouds framework, and a vitality efficient Mobile Cloud Computing (MCC) collective execution is proficient in the picked process cluster by the technique for a passing on a dispersed offloading system among the available asset in the cluster.

IV. SYSTEM ARCHITECTURE

The architecture identifies the major modules and the functional interfaces between them.

The System architecture is shown below.

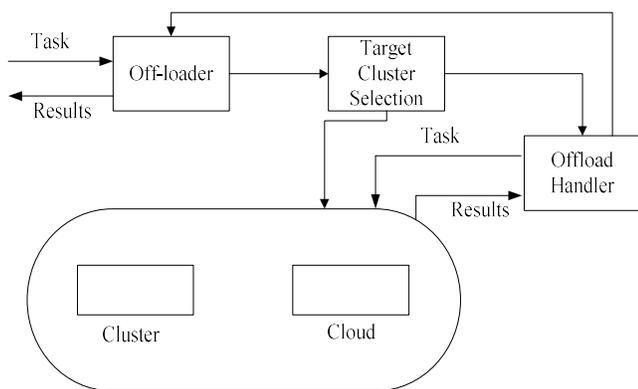


Figure 5.2: System Architecture

Offloader: This module accesses the task request for offloading and sent to Target cluster module. Also when the result of offloading arrives, it notifies to the User.

Target Cluster Selection: This class identifies the target cluster where the task has to offloaded and executed. It then transfers the code to the target machine for execution.

Offload Handler: It does the work of packing the task and sending to task machine for execution.

In this architecture mobile user will offload the task into offload handler and that task will processed by target cluster selection this will select the particular cluster and in that cluster there will be N number of nodes. Offload handler will select the particular node based on the node availability and then it will assign a task to the node and node will perform the task and then result will gives back to the offload handler and finally the offload handler will send the results to off-loader. The result will be notify to user with simple notification and finally user can view the results.

V. DATA FLOW DIAGRAM

Level 0 Data Flow Diagram (DFD) gives the overview of the data flow of the work. The above DFD provides us the flow of the data and gives us only the brief detail about the flow of data.

Offloading is overall process in the system.

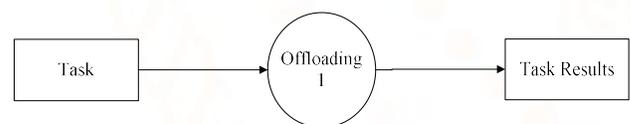


Figure 5.8: Level 0 Data flow Diagram

Offload process is split to sub process and drawn in the level 1 data flow.

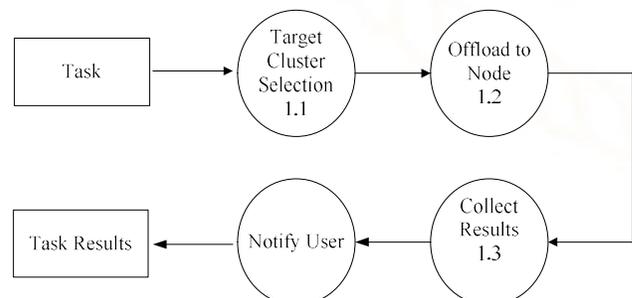


Figure 5.9: Level 1 Data flow diagram

VI. PSEUDO CODE

Pseudo Code is a partial code which explains the main content (explanation) of any program or any algorithm. It includes the aim of the function which we are designing.

The explanation behind using pseudo code is that it is less requesting for people to grasp than common programming language code, and that it is a gainful and condition free portrayal of the key norms of a count. It is routinely used as a piece obviously readings and consistent creations that are recording distinctive computations, and moreover in organizing of PC program change, for depicting out the structure of the program before the real coding happens.

The pseudo code for offloading at various modules is given:-

Node

Function: SubmitJob

Input: Task T

If resource need of T is available in system
Execute Job Locally

Else
Send offload request to cluster manager
End

Node

Function: Execute offload Job

Input: Task T, offloading Node X

Results←Execute T in local

Send Result to Node X

Cluster Manager

Function: SelectTargetNode

Input: Task T, offloading Node X

Node Y = Select Best Node in Local Cluster;

If Y==null

Forward Request to Next Level Cluster

Else

Send Task Execution Request to Node Y.

End

Function: connect

Input: Server IP, Port P, Guiist G

If server IP and Port P, Guiist G is matches with the cluster

Results←Node will connect to cluster

Else

Node will not connect to cluster

end

Function: Noderegister

Input: type 3

If the message type is 3

Results←Node is registered in the cluster manager

Else

Node is not registered in the cluster manager

End

Function: OffloadReq

Input: type 1

If the requesting to offload task into the cluster

Results← node is requesting to offload task into the cluster

Else

Node is not requesting

end

Function: Offloadres

Input: type 2

If the message type 2

Results← Node is responses to the Offloading task

Else

Cluster is not responding

end

VII. INTERPRETATION OF RESULT

The following snapshots describe the results or outputs that we will get after step by step execution of all the modules of the system.

Interpretation:

The results are categorized in three categories

Case 1	Offloading within same Node (Do-Local)
Case 2	Offloading within sane cluster
Case 3	Offloading to Next Level Cluster

➤ Case 1 Screen Shots: Offloading within same Node (Do-Local)

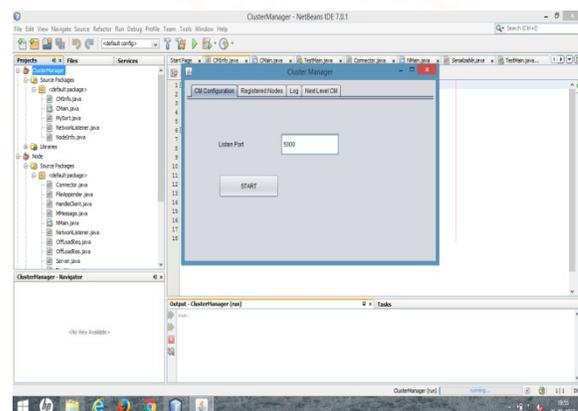


Figure 8.1: cluster port value

In this snap shot 8.1 we can see that Listen port as 5000 that means that in cluster, cluster manager will configures server at port value as 5000. Through this port value all node will get connected with the cluster and once if you give listen port as 5000 and press the start button if will starts the cluster manager and we can see either the cluster manager is started or not in Log window. And let's suppose if you want to add one more cluster we have go next level CM there we can add many number of cluster. And the Registered Nodes we can view the how many number of nodes are registered in the cluster.

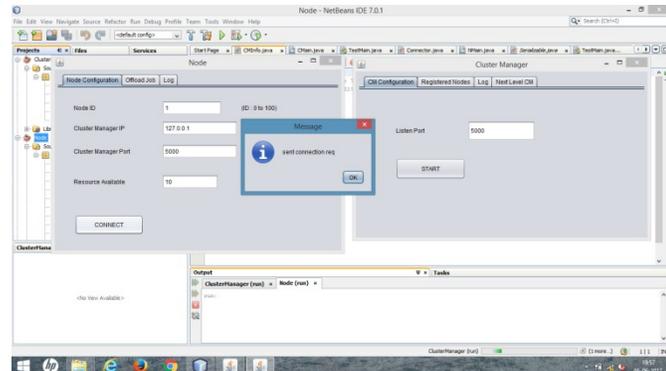


Figure 8.2: Node registration

In this window 8.2 we can see the node details that is before node is connected to the cluster we need to give node id and Node cluster manager IP address and also cluster manager port value so that node will get registered in that cluster manager and also Resource available in that particular node in this snapshot we can see available resource as 10KB and also we can see that one pop-up that tells that whether the node is connected to cluster or not.

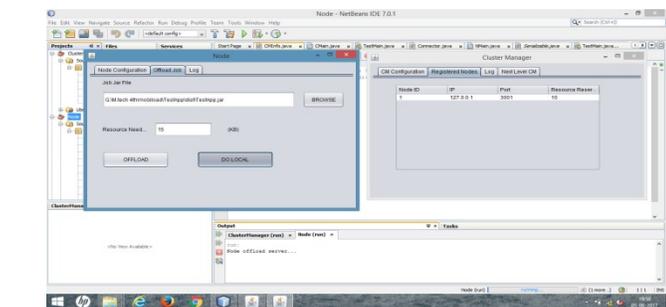


Figure 8.3: Do-local process

In this snapshot 8.3 we can see the registration of node ID along with IP address and port value and available resource in that node. In the pop-up that is node window we can see how exactly we will offload that task that is firstly we browse that task in the local disk and then we select the task. The task will be in the form of jar extension and also we need to give the

resource need to compute that task in the form of KB. After that we can see that two option that is offload button and one more is Do-Local in this case let's go for Do-local if you click that button it will compute task in that particular node itself.

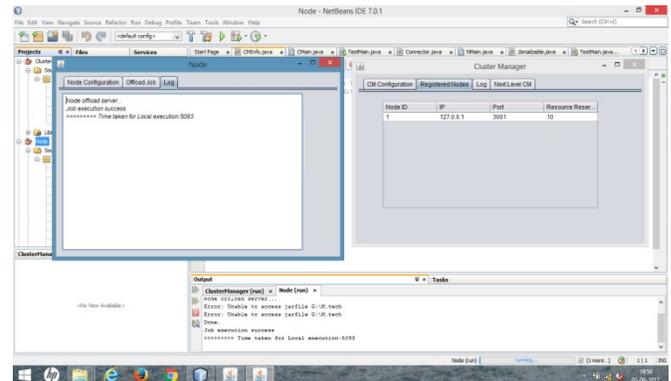


Figure 8.4: Result view in log

In this snapshot 8.4 we can see response time to compute that task. To compute that task in that same node it took 5093 mile second to compute that task and also we can see that one statement that is job is executed in locally. In other window we can view registered node in the cluster manager.

➤ **Case 2 Screen Shots: Offloading within sane cluster**

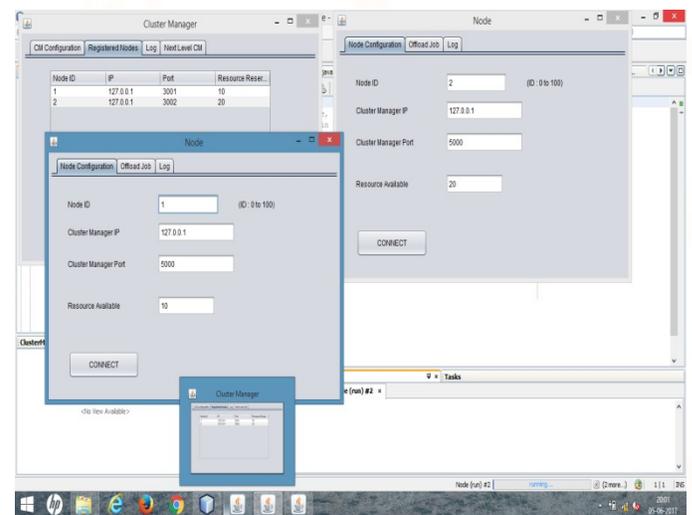


Figure 8.5: Two Node Registration

In this snapshot 8.5 we can view the two node are get connected to same cluster manager with same port value. And each node will have the unique node Id that is node number 1 and 2 followed by cluster manager IP address followed by cluster manager port value that is 5000 and available resource in that node in the node 1 it has 10KB of resource and node 2 has 20KB of resource. The available resource tells that it

can allocate 10KB of resource if any task comes to that node if any task come more than that available resource it cannot perform task.

statement that is found eligible node as 2. The cluster manager will scheduling job it first check whether the node has enough resource once if it got to know it has enough resource then only it assigns task to that node if not it will displays no eligible node are present in the cluster.

➤ **Case 3 Screen Shots: Offloading to Next Level Cluster**

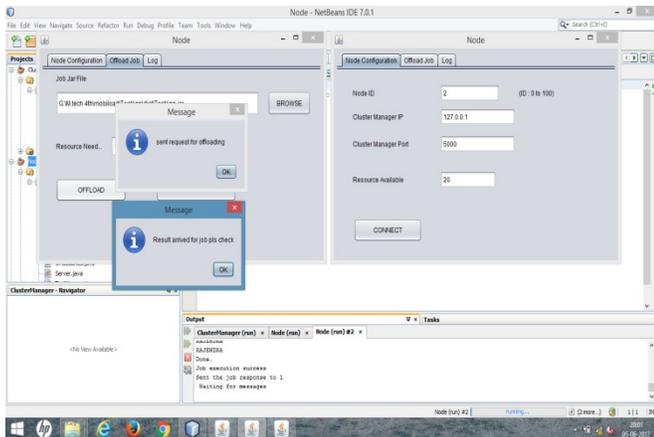


Figure 8.6: File offload to cluster

In this snap shot 8.6 we can see that we can see that node 1 is offloading task and we can see two pop-up that first one is offloading request is send to the cluster manger and other one is check the result. This how offloading of task will be done in cluster manager level. The offloaded task will be in the form of jar format that java executable form.

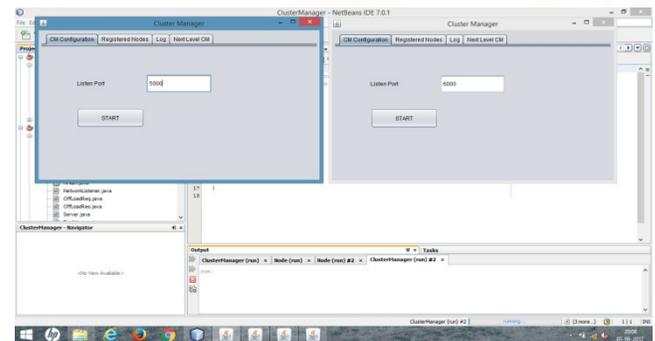


Figure 8.8: Multi level cluster

In this snapshot 8.8 we can see that how exactly multi-level cluster will create. Initially cluster manager 1 with port value 5000 and cluster manager 2 values with port value 6000. Once if you give port value and then press start button cluster manger will get started.

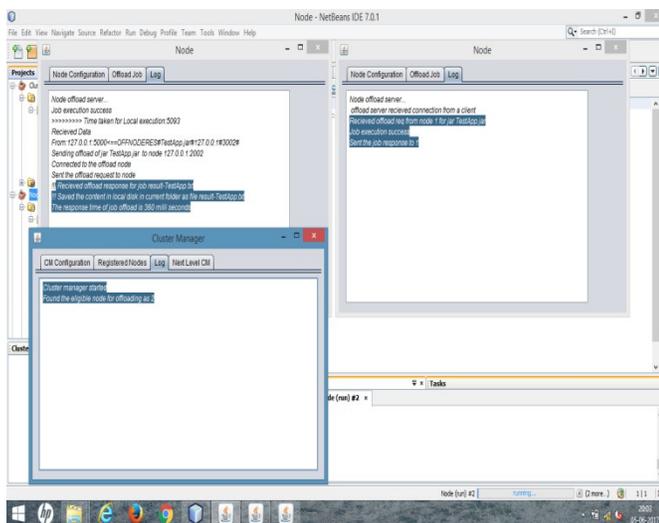


Figure 8.7: Result view in log

In this snap shot 8.7 we can see the result in log window. In the node 1 window we can see response time to compute that task and also we can see that result will be send to local disk of the system there we can view the results. And in the node 2 window we can see that two statement that is it got offload request form node 1 with task file name that is test-app.jar. Once the node is received request node will perform the task and then it will sends results to node 1. In the cluster manager window we can see that one

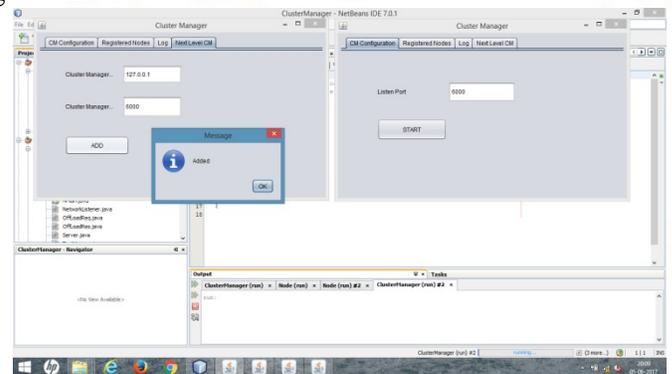


Figure 8.9: Connecting cluster levels

This snapshot 8.9 show how exactly two cluster manager that is cluster manager 1 and 2 will get connected that is in the cluster manager window we can see one option that is next-level CM in that window if you give cluster manager IP address followed by next-level cluster manager port value if you press add button it will add the next level cluster manager.

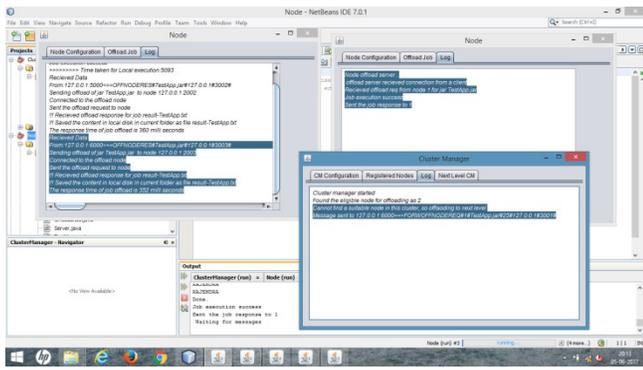


Figure 8.10: Results view in log

In this snapshot we can view how exactly task will execute in the next-level cluster manager. Initially job will offloaded from the node 1 into the cluster manager 1 and cluster manager 1 will check for available resource in same cluster if any resource present in the cluster then job will assigned to available resource if not cluster manager will forward to the next level cluster manager that is cluster manager 2 once again cluster manager will check for the available resource in the cluster manager 2 if it found resource in the cluster manager 2 the it will assigns that job to particular node based on that availability of resource in that node. We can see result in the node 1 log window to compute that job it took 352 mile seconds and also how exactly cluster manager will forward offload request to the next level cluster and we can see IP address of cluster manager and also node port value. This is how task will performed in the next level cluster manager.

CONCLUSION

In this work we implemented a proof of concept offloading server. The offloading based on the resource usage. The offload decision was made in hierarchy till the best resource matching to the task request is available. Through this work, we demonstrated that if the task is executed in place where resource is not available, it takes much longer time or may also fail, but due to offload and matching to best resource the time was able to execute successfully in less time. We also offloaded to live Microsoft azure cloud and proved that the concept was able to resort to cloud at top level hierarchy.

REFERENCES

1. A. Alzahrani, N. Alalwan and Mohamed Sarrab, "Mobile Cloud Computing: Advantage, Disadvantage and Open Challenge," Proc. 7th Euro American Conf. Telematics and Information Systems, Chile, 2014.
2. G. Mathew and Z. Obradovic, "Improving Computational Efficiency for Personalized Medical Applications in Mobile Cloud Computing Environment," IEEE Int. Conf. Healthcare Informatics (ICHI), pp. 535540, Philadelphia, Spetember 2013.
3. S. Kitanov and T. Janevski, "State of the Art: Mobile Cloud Computing," 6th Int. Conf. Computational Intelligence, Communication Systems and Networks, Tetova, pp. 153-158, May 2014.
4. Z. Sanaei, S. Abolfazli, A. Gani and R. Buyya, "Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges," IEEE Communications Surveys & Tutorials, vol. 16 (1), pp. 369-392, Feb 2014.
5. S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani and R. Buyya, "CloudBased Augmentation for Mobile Devices: Motivation, Taxonomies, and Open Challenges," IEEE Communication Surveys & Tutorials, vol. 16 (1), pp. 337-368, February 2014.
6. T. Lose and M. Thinyabe, "A Cloud Computing Platform to Augment Mobile Phone Use in Marginilized Rural Areas," SATNAC, Western Cape, 2012.
7. M. Satyanarayanan, P. Bahl, R. Cáceres and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," IEEE Pervasive Computing, vol. 8 (4), pp. 14-23, October 2009.
8. S-S. Qureshi , T. Ahmad, K. Rafique and Shuja-ul-islam, "Mobile cloud computing as future for mobile applications - Implementation methods and challenging issues," IEEE Int. Conf. Cloud Computing and Intelligence Systems (CCIS), pp. 467-471, Beijing 2011.
9. A. Ravi and S-K. Peddoju, "Energy Efficient Seamless Service Provisioning in Mobile Cloud Computing," IEEE 7th Int. Symp. Service-Oriented System Engineering, pp. 463-471, Redwood City, March 2013.
10. J. Li, B. Li, T. Wo, C. Hu, J. Huai, L. Liu and K-P. Lam, CyberGuarder: A Virtualization Security

Assurance Architecture for Green Cloud Computing, Future Generation Computer Systems, Elsevier Science, Vol. 28(2), pp 379–390, February 2012.

11. H. Al-Aqrabi, L. Liu, R. Hill, and N. Antonopoulos, Cloud BI: Future of Business Intelligence in the Cloud, Journal of Computer and System Sciences, Elsevier, Vol. 81 (1), pp. 85-96. February 2015.
12. B. Li, J. Li and L. Liu, CloudMon: A Resource-Efficient IaaS Cloud Monitoring System Based on NIDS Virtual Appliances, Concurrency and Computation: Practice and Experience, Vol. 27(8), Wiley, June 2015, pp 1861–1885.
13. W. Lee J. Jung, and H. Kim, "Analyzing Extent and Influence of Mobile Device's Participation in Mobile Cloud Computing," Int. Conf. ICT Convergence (ICTC), pp. 14-16, Jeju, October 2013.
14. X. Liu, C. Yuan, Z. Yang, and Z. Hu, "An Energy Saving Algorithm based on User-Provided Resources in Mobile Cloud Computing," IEEE 78th Vehicular Technology Conference (VTC Fall), pp. 1-5, Las Vegas, Spet 2013.
15. M. Nir, A. Matrawy and M. St-Hilaire, "An Energy Optimizing Scheduler for Mobile Cloud Computing Environments," IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS), pp. 404-409, Toronto, May 2014.