Resource allocation scheme for Dew Computing paradigm using mobile grid

Amit Sadanand Savyanavar,   Vijay Ram Ghorpade

Abstract: With the widespread availability of smartphones and advancement in communication technologies, Dew Computing paradigm(DCp) has emerged as a state-of-the-art computing paradigm. DCp provides an ecosystem to execute computationally intensive tasks which comprise of several subtasks. Each subtask is allocated for execution to an available and capable mobile device by taking into consideration its features like mobility, processing power, remaining battery, etc. This kind of "on-the-spot" paradigm comprises of mobile devices only which are part of mobile grid and it doesn't use the fixed infrastructure based computing systems for computational purposes. Being resource constrained, such a paradigm needs an efficient scheme for allocation of resources. Here we propose a scheme called MGRA for allocation of computing nodes which takes into account challenging issues like mobility of users, inefficient resource allocation and handling of failure situations. Experimentation was carried out using a DCp testbed comprising Android devices connected with Wi-Fi Direct protocol. MGRA exhibited significant improvement in terms of time for application completion, amount of battery usage and time required for recovering from failure as compared to present-day approaches.

Index Terms: Dew computing, fog computing, resource allocation, rough set theory.

I.   INTRODUCTION

With the advancement of technology, several computing paradigms came into existence. Grid computing is an infrastructure to solve complex computationally intensive problems by sharing the computing devices which are members of the Grid. It invariably comprises of resourceful high processing devices usually connected through a wired network. The software is a vital component of this architecture ensuring optimal utilization of resources. Cloud computing provides service centric models to meet users requirements. The services are provided by remote service and data centers which requires that the user be connected via internet for availing the services. With intermittent loss of internet connectivity, services cannot be guaranteed. Also, there's significant latency involved in sending the request and receiving the reply especially when emergency or disaster situations occur.

Fog computing [1] came into existence to provide "on-the-fly" or "on-the-spot" services involving reduced latency. The computation is performed in the proximity at the client-side where it is required without using the cloud. Significant improvement is achieved in terms of savings on bandwidth usage, ceaseless availability of data and providing a secured environment which are contentious issues with Cloud computing. With the ubiquitous availability of smartphones and a variety of mobile devices, Dew Computing paradigm (DCp) [1, 2] or Mobile Grid(MG) computing [3] came into existence. DCp can form an adhoc network for computing and does not require any fixed or wired networked infrastructure.   Thus, it provides a novel opportunity of service and data accessibility without continuous internet connectivity. It works on the idea of anytime-anywhere computing paradigm and can be created on-the-fly. It can be deployed fast and so could be potentially used for disaster management and military applications. With the advent of high computing mobile devices, mobile devices can also be a potential computing resource for a Grid. There are umpteen applications in real world, which require devices with high processing capability. The extent of penetration of mobile devices in India, has made mobile phones a necessary commodity in daily life. Even with the availability of numerous devices in the vicinity, their computing power goes either unutilized or underutilized. Such devices are potential resources for computing in a Grid. DCp can include a diverse collection of mobile devices enabled to communicate using various wireless mechanisms ranging from miniature sensors, wearable computing devices, mobile robots, RFIDs, smartphones, tablets, laptop computers and any other mobile device with processing and storage capability. The advantages with such ubiquitous devices is their huge availability, variety, affordability, compactness, mobility and much more. DCp brings all such devices in the neighborhood willing to share their resources under its ambit to work as a resource provider to solve computationally intensive problems. On the flip side, these devices face a few serious challenges like limited processing power, mobility, storage capacity, battery life and communication reliability. DCp is a contemporary paradigm providing an entirely new perspective for on-the-fly computing. Hence there is a paucity of research work in this domain with very little literature available. DCp environment being highly dynamic and distributed, faces numerous challenges [4] namely node mobility, battery drain, security, connection reliability, node failure, etc. DCp provides a conducive ecosystem for executing sophisticated algorithms. A case in point is to execute image and video processing algorithms on mobile robots for

surveillance pertaining to military or non-military threats [5]. Such robots navigate in the surroundings using vision-based algorithms[6] for navigation. Such algorithms require computational capabilities beyond the capacity of an individual device and hence DCp is required. In the battlefield, soldiers can construct a 3D map [5] using DCp comprising of wearable computing devices and sensor nodes for vigilance of stationary and moving objects. DCp also aids in ubiquitous healthcare applications predominantly in rural or remote areas where there is an iota or unavailability of fixed infrastructure for remotely monitoring the health condition of a patient. One of the critical challenges faced by DCp is that of resource allocation which involves identifying the appropriate mobile nodes to which the tasks of an application can be assigned.

II. RELATED WORK

Prior to task allocation, can we predict the next location i.e. mobility of the node is the intriguing question that needs to be answered. If we managed to do so, we can determine if a task can be assigned to it for computation and ensure that the node mobility will not disconnect it from DCp. The prediction of node mobility is thus crucial in reducing the communication cost and enhancing the lifetime of the network. Mobility of the node is synonymous to human mobility. Understanding and predicting node mobility is beneficial for a wide variety of applications. There is colossal demand for information regarding human mobility in applications like road traffic management, preventing epidemic of diseases, load balancing in mobile networks, etc. Mobile devices like smartphones are commonly used for mobility tracking of human beings[16] since they (1) are ubiquitous, widely used by everyone (2) are carried by human beings throughout the day, and (3) can be tracked continuously for mobility related data. The present day mobile devices are equipped with features like GPS and a variety of connectivity options to facilitate data collection related to locations visited by individuals. As a result, researchers are prolifically using these to carry out their research work in diverse fields.

LocP[17] system provides a modus operandi for energy efficient location monitoring and prediction. It involves periodic collection of location data of a mobile node with minimal energy consumption and then predicts the future location of that node. Human tendency is to visit locations with quite a similar pattern every day with only a small degree of variation. Such movement pattern could be recorded over a period of time. This can then be used to predict the next probable location based on the historically recorded data. The location tracking can either be carried out continuously or periodically. If done periodically, battery life can be improved.

A. Resource Allocation

Umpteen techniques existing in literature focus on the challenge of allocation of resources for computation but then they cater to systems with a fixed infrastructure [7, 8]. Beyond this, complex applications comprise of inter-dependent tasks and existing approaches don't consider this interdependency [8]. Further, the work can be categorized[3] into two: one,

mobile devices working as service consumers using service providers in the Grid for computation [9] and two, mobile devices also working as service providers i.e. computing resource within the Grid [10]. The Grid with mobile devices as service providers can be integrated with fixed-infrastructure based computing systems [11], whereas the Grid could comprise of only mobile devices without any fixed-infrastructure [12] which we are referring to as a MG or DCp. Furthermore, the existing methods lack focus on energy consumption preservation and that to pertaining to conservation of energy due to computation. Preserving energy due to transfer of data is still an open problem which is more demanding in the case of applications involving large data transmission. Computationally intensive applications comprise of numerous tasks, which may be independent or interdependent. This can be identified if we can represent the tasks using a directed acyclic graph. Independent tasks can be executed on any distant node of the DCp, but when it comes to interdependent tasks to preserve on communication cost determining the closest nodes would be vital. Scheme for allocation of resources based on distance (DRA) was proposed by S. C. Shah[14] for inter-dependent tasks in MG. It works on the principle that the distance between the nodes affects the cost required for communication and ultimately the total time required to complete the execution of the application. Hence the approach involves selecting nodes at least distance from the current node which needs computing resources and allocate interdependent tasks to these nodes. The current node will select the best nodes in its locality but will not consider the entire collection of nodes in MG. This being a centralized approach will not be scalable and is vulnerable to failure. A scheme called NLRA[14] considers mobility-based history while selecting nodes but doesn't consider the distance amongst nodes.

Shah S. C et al [15] proposed EERA - a new energy-efficient scheme for allocation of resources based on power levels was used for allocation of inter-dependent tasks to computational nodes in MG. For the inter-dependent tasks, nodes with minimal distance can form a group and then tasks are allocated to them. It thus reduces the communication cost amongst the tasks. But it lacks a well-defined failure handling mechanism.

B. Failure Handling

While executing an application in a DCp, there are possibly two types of failures that may occur: node failure and application failure. DCp has to find a way of overcoming different kinds of failures and restore the application computation. Checkpointing[18] is an approach which can be explored. In checkpointing, periodically the system state is preserved with the intention to restore to a previous state incase a failure occurs. This would save energy and time required for recomputing the task. When a fault occurs, the task need not restart again. It has 2 parts: firstly checkpointing periodically and rollback to a previous state when fault occurs. Checkpointing arrangement is crucial in DCp due to its dynamic nature. So the emphasis is on stability of the arrangement. In a DCp, a decentralized checkpointing

approach would be appropriate as each node can participate in the pairing arrangement without burdening a single node. A centralized approach will increase the amount of communication involved. A computational node will scout for a checkpointing partner in its locality which will not increase the cost of communication drastically.

Paul J DarbyIII and Nian-Feng Tzeng[19] proposed a middleware check-pointing arrangement called ReD, where the nodes take the dual roles of providers and consumers. ReD continuously looks out for the best possible checkpointing arrangement to ensure recovery from failure. Best pairing option is determined based on a calculated gain value. Connectivity and link reliability are used to calculate the gain. Here transmission of data over multiple hops is not required. ReD continuously searches for a better producer-consumer relationship and breaks existing one if the current relationship improves reliability.

Jack Dongarra et. al. [20] devised a technique involving multiple checkpointing iterations. It uses double checkpointing and also a triple checkpointing algorithm. Here the local memory is used to store checkpoints and then it is replicated remotely. A buddy node is identified and paired to store the remote checkpoint. When a failure occurs, data can be recovered either locally or from the remote buddy node. The possibility of another failure striking the buddy node is also addressed by a triple checkpointing algorithm. On the flip side, triple checkpointing results in consuming more energy while achieving a higher reliability. Triple checkpointing in DCp would put additional burden on the already resource constrained mobile nodes. Co-ordination messages between the nodes would further have an adverse impact on the battery.

III. PROPOSED SCHEME

Figure 1 depicts the architectural view of the proposed resource allocation scheme MGRA. Our system uses application metadata template (AMT) that describes the subtasks in the application. The subtasks may be computation or communication type and have parallel or precedence execution dependency. The resource allocation algorithm makes use of node mobility history to identify nodes that will provide long-term connectivity and uses failure handling mechanism. LocP[17] system is used to identify nodes that will provide long-term stability based on their mobility history. Only these nodes will be utilized for allocation of resources.
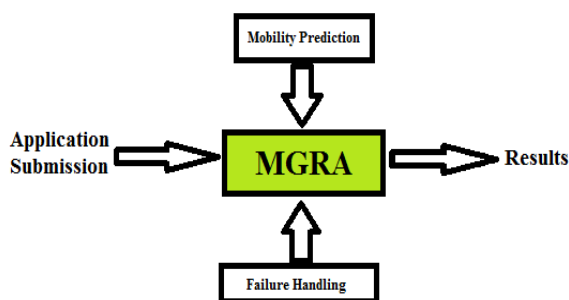

Fig. 1 System Architecture

MGRA SCHEME

Input: AMT

Step 1: Any node can initiate the application execution (initiator node)

Step 2: Use AMT for allocating each task

Step 3: Based on the AMT, assign a node to a task in the following manner:

For each task t_i in AMT

i) If tasks with parallel execution dependency exist, select k number of nodes based on number of dependent tasks as follows

a. Classify the nodes as Execution node or Checkpointing node using Rough Set Theory based model. The value sets for Execution node and Checkpointing node are as mentioned below

   E:Execution node→{High Battery %, High Connectivity, High Processing Power}

   C:Checkpointing node→{High Battery %, High Connectivity, High Storage}

b. The checkpointing arrangement is done by pairing an element E with C

$$E_1 \text{ ------ } C_1$$
$$E_2 \text{ ------ } C_2$$
$$........$$
$$E_k \text{ ------ } C_k$$

c. Nodes will keep parent nodes informed using waypoints periodically i.e. at every ρ seconds interval.

d. The execution nodes will store the execution results at every ρ seconds interval to it's checkpointing nodes.

e. After the task execution completes, each execution node will send the result to the parent node.

ii)If it's a computation task requiring a single node, then identify a checkpointing node closest to it and perform checkpointing process at every ρ seconds interval.

iii)If it's a communication task, checkpointing is not required and therefore assign a single node to it.

Step 4: Repeat step 3, for all the tasks in AMT.

IV. EXPERIMENTATION AND RESULT ANALYSIS

We created a DCp comprising of a variety of ubiquitously available Android smartphones, which form a network using Wi-Fi Direct protocol. The application that we deployed on our testbed is a multidocument(files) based summarization technique[21] for natural disasters like earthquakes, hurricanes, etc.. Several documents related to the disaster are inherently created. Invariably summarized information about the disaster aids in the recovery process. Disaster management can be subdivided into tasks which can be executed on our DCp using above mentioned scheme. Application metadata template(AMT) used in multidocument based summarization technique is as shown in Table I.

TABLE I. AMT

Task sequence	No. of smartphones used	Task type	Task Details
I	1	Communication-bound	Invoke application and distribute the files amongst the nodes
II	2	Computation-bound	Extract sentences from files
III	1	Communication-bound	Send sentence extracted files to nodes
IV	2	Computation-bound	Identify concept associated with each sentence & store in respective concept file
V	1	Communication-bound	Merge same concept related file from each node
VI	1	Communication-bound	Enter concept related query
VII	1	Communication-bound	Retrieve summary of queried concept

We repeated the experiments five times to ascertain the results. Here we considered the time required for computation as well as communication tasks of the application. For comparison, we are using the resource allocation schemes: DRA and NLRA. DRA has a problem that it takes into consideration only a single dependent task at any given point of time and then searches for a node which is at least possible distance to allocate a task. For a group of inter-dependent tasks, it does not consider a group of nodes which are at a shortest distance. DRA lacks a mechanism of group allocation for a collection of inter-dependent tasks. NLRA approach does not take into account the distance between the nodes and hence would possibly allocate nodes which are far away from the delegating node. This enhances the cost required for communication and ultimately results in a higher battery drain. Comparison is carried out on the basis of time required to completely execute the application, amount of remaining battery and time required to recover from failure. The application completion time T_{act} is determined as follows:

$$T_{act}=T_{execution}+ T_{communication} \qquad (1)$$

where, $T_{execution}$ is the time required for execution of all the tasks and $T_{communication}$ is the time required for communication amongst the tasks. For experimentation, 10 mobiles together formed a DCp: Micromax Canvas A2, Moto G1, Asus Zenphone, Coolpad, Micromax tab, Micromax Canvas A1, Moto G2, Redmi Note4, Sony Xperia and Samsung Galaxy Note with a remaining battery of 56, 53, 62, 63, 54, 44, 85, 63, 51 and 44 % respectively. DCp nodes were placed randomly varying from a distance of 3m to 18m to form a network. Wi-Fi Direct protocol is used for communication amongst the nodes.

Figure 2 depicts that MGRA comparatively requires lesser time for completing the execution of an application than the other methods. MGRA selects nodes based on the distance and hence the time for communication reduces. Even with the addition of checkpointing to MGRA, it requires only slightly more time as compared to other methods.
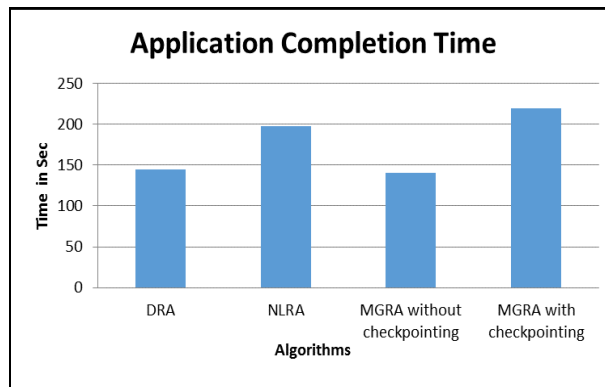

Fig. 2: Application Completion Time comparison

Figure 3 exhibits that MGRA can achieve significant savings in terms of battery usage. This is again based on the modus operandi followed while assigning the tasks to the appropriate nodes.
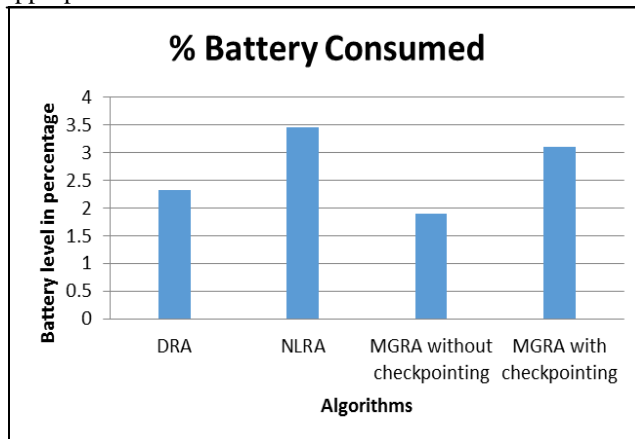

Fig. 3: Battery Consumption comparison

Here we have defined recovery time as the application resurrection time when an execution failure occurs. Figure 4 shows that MGRA with checkpointing uses the checkpointing node for speedy recovery from a task or node failure.
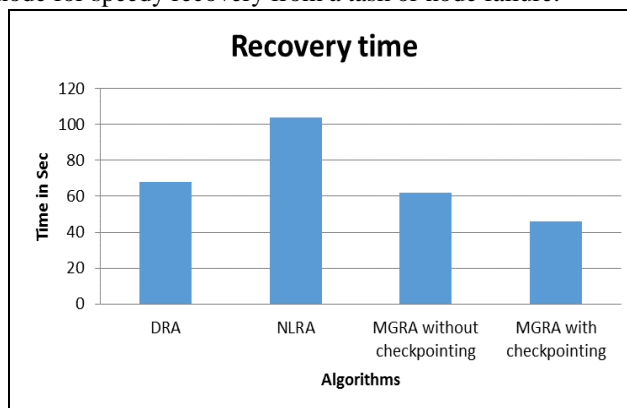

Fig. 4: Recovery Time Comparison

V. CONCLUSION

Dew Computing is truly emerging as a new paradigm for anytime, anywhere computing. We managed to create a DCp comprising of a variety of Android-based mobile devices using Wi-Fi Direct. Resource allocation scheme MGRA

investigates as to how the application completion time can be reduced i.e. communication cost can be reduced. We introduced a failure handling mechanism with the help of checkpointing, making a vulnerable network more robust. The results obtained were compared with the existing techniques. MGRA reduces the application completion time as compared to NLRA by almost 40% for the experimental setup. The battery consumption for MGRA is lesser than that required by DRA and half of what is required by NLRA. With checkpointing, MGRA reduces the recovery time by 32% and a whopping 55% as compared to DRA and NLRA respectively. As an extension to our work, a variety of applications can be deployed on the DCp and the performance of the approach can be verified.

REFERENCES

1. P. P. Ray, "An Introduction to Dew Computing: Definition, Concept and Implications", IEEE Access, vol. 6, pp. 723–737, 2018
2. Hirsch M, Mateos C, Rodriguez JM, Zunino A, "DewSim: A trace-driven toolkit for simulating mobile device clusters in Dew computing environments", Softw: Pract Exper. 2019, 1–31, https://doi.org/10.1002/spe.2696
3. Savyanavar Amit S., Dr. Vijay R. Ghorpade, "A survey on resource allocation in mobile grid computing," CiiT-International journal of wireless communication, vol. 5, no. 5, 2013
4. Li Chunlin, Li Layuan, "Design and implementation of economics-based resource management system in ad hoc grid," Advances in Engineering Software, vol. 45, pp. 281–291, 2012
5. J. Falcou, J. Serot, T. Chateau, F. Jurie, "A parallel implementation of 3D reconstruction algorithm for real-time vision," Parallel Computing, vol. 33, pp. 663–670, 2005
6. S. Se, D.G. Lowe, J.J. Little, "Vision-based global localization and mapping for mobile robots," IEEE Transactions on Robotics, vol. 21, no. 3, pp. 364–375, 2005
7. R. McClatchey, A. Anjum, H. Stockinger, A. Ali, I. Willers, M. Thomas, "Data intensive and network aware (DIANA) grid scheduling," Springer, Journal of Grid Computing, vol. 5, no. 1, pp. 43–64, 2007
8. D. Villela, "Minimizing the average completion time for concurrent grid applications," Springer, Journal of Grid Computing, vol. 8, no. 1, pp. 47–59, 2010
9. J. Zhang, A. Hamalainen, J. Porras, "Addressing mobility issues in mobile environment," Proceedings 1st Workshop on Mobile Middleware: Embracing the Personal Communication Device, 2008.
10. Fox G., Ho A.,Wang R., Chu E., Kwan I., "A collaborative sensorGrids framework," IEEE International symposium on Collaborative Technologies & Systems, pp. 29–38, 2008
11. Tadeu A., Gomes A., "DICHOTOMY: a resource discovery and scheduling protocol for multihop ad hoc mobile Grids," Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07), pp. 719-724, 2007
12. Preetam G., Nirmalya R., Das S.K., "Mobility-aware efficient job scheduling in mobile Grids," Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07), pp. 701-706, IEEE Press 2007
13. Shilve S., Siegel H.J., Maciejewski A.A., Sugavanam P., Banka T., Castain R., Chindam K., Dussinger S., Pichumani P., Satyasekaran P., Saylor W., Sendek D., Sousa J., Sridharan J., Velazco J., "Static allocation of resources to communicating subtasks in a heterogeneous ad hoc Grid environment," Elsevier, Journal of Parallel and Distributed Computing, vol. 66, no. 4, pp. 600–611, 2006
14. Shah S. C., Chauhdary S.H., Bashir A.K., Park M.S., "A centralized location-based job scheduling algorithm for interdependent jobs in mobile ad hoc computational Grids," Springer, Journal of Applied Sciences, vol. 10, no. 3, pp. 174–181, 2010
15. S.C. Shah, M.-S. Park, "An energy-efficient resource allocation scheme for mobile ad hoc computational grids," Springer, Journal of Grid Computing, vol. 9, no. 3, pp. 303–323, 2011
16. Yohan Chon, Elmurod Talipov, Hyojeong Shin, and Hojung Cha, "SmartDC: Mobility Prediction-based Adaptive Duty Cycling for Everyday Location Monitoring," IEEE Transactions on Mobile Computing, vol. 13, no. 3, pp. 512–525, 2014
17. Amit S. Savyanavar, Dr. Vijay R. Ghorpade, "Node Mobility Prediction in Mobile Grid," International Journal on emerging trends in technology, vol. 2, no. 2, 2015
18. Amit S. Savyanavar, Dr. Vijay R. Ghorpade, "Application checkpointing technique for self-healing from failures in mobile grid computing," International Journal of grid and high performance computing, vol. 11, issue 2, article 3, 2019
19. Paul J. Darby III and Nian-Feng Tzeng, "Decentralized QoS aware Checkpointing Arrangements in Mobile Grid Computing," IEEE transactions on mobile computing, vol. 9, no. 8, 2010
20. Jack Dongarra, Thomas Herault and Yves Robert, "Revisiting the double checkpointing algorithm," IEEE 27th International Symposium on Parallel and Distributed Processing Workshops and PhD Forum 2013, IEEE DOI 10.1109/IPDPSW 2013.11
21. Li, L., & Li, T., "An Empirical Study of Ontology-Based Multi-Document Summarization in Disaster Management," IEEE Transactions on Systems, Man, and Cybernetics, vol. 44, no. 2, 2014

AUTHORS PROFILE

Amit S. Savyanavar completed his Master's degree in CSE from Shivaji University, Kolhapur. Currently, he is a PhD candidate in CSE at Shivaji University. His area of interest includes distributed systems, mobile computing and operating systems.

Prof. Dr. Vijay Ram Ghorpade completed ME and PhD in Computer Engineering. He is a Professor in Computer Engineering department and Principal at BVCOE, Kolhapur. He is an eminent academician and researcher with papers published in various reputed journals. His area of research includes information security, mobile computing, ad-hoc networks, etc.