Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 10 (3): 174-181, 2010
ISSN 1812-5654

A Centralized Location-Based Job Scheduling Algorithm for Inter-Dependent Jobs in Mobile Ad Hoc Computational Grids

S.C. Shah, S.H. Chauhdary, A.K. Bashir and M.S. Park
Internet Computing Lab., Department of Computer Science and Engineering,
Korea University, South Korea

Abstract: This study addresses the problem of job scheduling to inter-dependent jobs in mobile ad hoc computational grids. To maximize the utilization of shared computational resources and to improve application performance, an effective job scheduling algorithm plays a key role. Previously, numerous job scheduling algorithms have been proposed, but most of them are either targeted towards large scale infrastructure-based systems or they don't consider the characteristics of mobile devices and inter-job dependencies. As the performance of inter-dependent jobs is greatly affected by communication performance; therefore, to improve communication performance of inter-dependent jobs, we have proposed a centralized job scheduling algorithm that takes into account the location of nodes, inter-job dependencies and communication traffic among inter-dependent jobs to schedule them on closely located nodes. Simulation results demonstrated that our proposed algorithm performs well as compared to existing approaches and reduces the communication cost thus energy consumption.

Key words: Ad hoc computational grid, ad hoc networks, mobile grids, inter-dependent jobs

INTRODUCTION

Grid computing is an emerging technology that provides high performance and cost-effective solutions based on connect and share approach. It combines various distributed computing devices to form a single, unified computing resource with a key objective, to free the user from the system concerns, care and individual experience of the original PC and to foster resource and application sharing within a group (Baker *et al*., 2002). A grid can be used in different ways to provide various services ranging from data, information and knowledge services to application, storage and computational services. A computational grid is focused on setting aside resources specifically for computing power and used to solve computationally intensive problems (Shah *et al*., 2009; Baker *et al*., 2002; IBM, 2009).

Currently, large amount of work in this area is dedicated towards large scale infrastructure-based systems and applications. However, due to advancement in performance of mobile devices and wireless networks (Intel Corporation, 2009; Apple, 2009) this field has got considerable attention. Researchers are exploring new technologies and approaches to extend the implementation of grid computing on mobile devices to offer grid services anywhere, anytime (Maria *et al*., 2008; Markov, 2004; Senger *et al*., 2005; Millard *et al*., 2005).

Mobile ad hoc computational grid is integration of computational grid and mobile ad hoc network that allow autonomous mobile devices to form a single, unified computing resource without support of any fixed infrastructure. It has applications in disaster relief management, battlefield, defense, etc. (Marinescu *et al*., 2003; Wireless Grids Lab., 2009).

To maximize the utilization of shared computational resources and to improve application performance, an effective job scheduling algorithm plays a key role. Job scheduling algorithm is concerned with ordering the execution of jobs and assigning resources to jobs. In literature, this process is also referred as resource allocation, mapping and resource management (Feitelson *et al*., 2004; Sodan, 2005). Design of a robust and effective job scheduling algorithm for mobile ad hoc computational grids presents many challenges due to node mobility and infrastructureless network environment. For example, node may leave the grid anytime or may fail due to low battery power.

Earlier study on job scheduling algorithms for computational grids (Cao and Kwong, 2005; Arora *et al*., 2002; Maria *et al*., 2008; Markov, 2004; Senger *et al*., 2005; Shivle *et al*., 2006; Braun *et al*., 2008) is focused towards large scale and powerful computing systems connected through high performance communication networks. In order to support high scalability, most of these algorithms

Corresponding Author: Myong-Soon Park, Department of Computer Science and Engineering, Korea University, South Korea
Fax: +82-2-927-3797

such as (Cao and Kwong, 2005; Arora *et al*., 2002) use de-centralized approach that results in ineffective scheduling decisions due to lack of global view of network. These algorithms also don't take into account the inter-job dependencies in contrast to (Maria *et al*., 2008; Markov, 2004; Senger *et al*., 2005; Shivle *et al*., 2006; Braun *et al*., 2008) that use knowledge about parallel applications in order to improve scheduling decisions. All these algorithms are based on fixed infrastructure and are designed for geographically dispersed computational sites; therefore, not suitable for mobile ad hoc computational grids. A detailed analysis and comparison of job scheduling algorithms is given in (Feitelson *et al*., 2004; Sodan, 2005).

Due to advancement in performance of mobile devices and wireless networks (Intel Corporation, 2009; Apple Computers, 2009) various research efforts have been made to extend the implementation of grid computing on mobile devices to offer grid services anywhere, anytime. In literature, two kinds of approaches have been proposed. In first, mobile devices are allowed to access fixed grid resources (Gonzalez *et al*., 2005; Millard *et al*., 2005; Grabowski *et al*., 2009; Fox *et al*., 2008; Robinson *et al*., 2005) while in second, they can be used as computational resources in a grid (Shah *et al*., 2009; Franke and Fernando, 2007; David and Humphrey, 2004; Bhagyavati, 2004). Based on second approach where mobile devices can access and offer services, few algorithms (Liu *et al*., 2005; Braun *et al*., 2008; Hummel and Jelleschitz, 2007; David and Humphrey, 2004; Stanislav and Bhagyavati, 2004) have been proposed for scheduling jobs on mobile ad hoc computational grids.

Hummel and Jelleschitz (2007) proposed a de-centralized job scheduler which supports the integration of mobile devices as computational grid resources. It allows mobile peers to perform mapping based on job's requirement and device capabilities autonomously. The key limitations of this approach are: it doesn't consider the inter-job dependencies and uses job replication technique for fault tolerance which is not appropriate for low constraint devices that have very limited resources in terms of storage, processing and communication. DICHOTOMY (Antonio *et al*., 2007) is another de-centralized job scheduling algorithm which schedule jobs among the most resourceful nodes in the mobile grid. It employs delayed reply mechanism to select most suitable nodes. However, it also has same problems as in (Hummel and Jelleschitz, 2007). David and Humphrey (2004) also provided the implementation of grid computing on mobile devices to promote resource sharing and collaboration. The scheme has a provision for hosting grid services on mobile devices and application controlled process migration and distributed execution. However, it also doesn't consider inter-job dependencies. The work presented by Braun *et al*. (2008) takes into account data dependencies among jobs, but it is based on static allocation of resources and targeted towards the minimization of energy consumption rather than application performance. Moreover, due to static allocation of resources, this algorithm is not adaptive to network changes and application behavior. Liu *et al*. (2005) designed a de-centralized, power-aware job scheduling algorithm to support adaptation needs of ad hoc applications such as changes in the network topology and application behavior. However, it doesn't consider the capabilities of nodes and only takes decision based on available energy. To reduce mean path length of data packets, it migrate jobs to topologically closer nodes, but only during runtime after analyzing the data communication patterns. Migrating jobs during runtime is costly operation and unsuitable especially for low constraint devices. Moreover, the movement of communicating jobs onto topologically closer nodes doesn't yield optimal reduction in communication distance.

All of these algorithms are based on de-centralized approach which is not effective due to lack of network wide view. Furthermore, to best of our knowledge, none of the algorithms proposed for mobile ad hoc computational grids takes into account the location of nodes and inter-job dependencies to make scheduling decisions.

As the performance of inter-dependent jobs is greatly affected by communication parameters such as communication overhead and end-to-end communication delay; therefore, to improve communication performance of inter-dependent jobs, we have proposed a centralized job scheduling algorithm that takes into account the location of nodes, inter-job dependencies and communication traffic among inter-dependent jobs to schedule them on closely located nodes. It is based on group mobility model (Hong *et al*., 1999) where all nodes work in a group. As compared to existing approach (Liu *et al*., 2005) it also avoids job migration due to long communication distance. To deal with job failure due to low battery power or movement of nodes across grid, it reschedules jobs to next available nodes and allows them to migrate to newly selected node. Simulation results demonstrated that our proposed algorithm performs well as compared to existing approaches and reduces the communication cost thus energy consumption.

PROPOSED ALGORITHM

System model: An ad hoc computational grid is represented by complete undirected graph G = (N, E, p, m, b, d) where, N is set of vertices representing the mobile nodes and E is set of edges representing communication links among them. The parameters p_i, m_i and b_i represents processing power, memory and remaining battery power of node n_i while d_{ij} represents the distance between two nodes. All the nodes within network are heterogeneous.

An application is collection of jobs represented by a graph G_A = (J, L, rp, rm, rb) where J represent the set of jobs and L represent the communication link among them. A job j_i depends on job j_i if there is edge between them. Jobs within an application are divided into two categories: computation-bound jobs represented by $j_i^{cpu\text{-}bound}$ and communication-bound jobs represented by $j_i^{comm\text{-}bound}$. Computation-bound jobs have high processor utilization with low communication traffic, while communication-bound jobs have low processor utilization with high communication traffic. The parameters rp_i, rm_i and rb_i represents required processing power, memory and battery power by job j_i. An example job dependency graph is shown in Fig. 1.

Assumptions

- Mobile ad hoc computational grid has already been formed
- There is a resource discovery service that updates scheduler with mobile nodes' characteristics such as processing power, memory and remaining battery power. The updating occurs before the submission of job

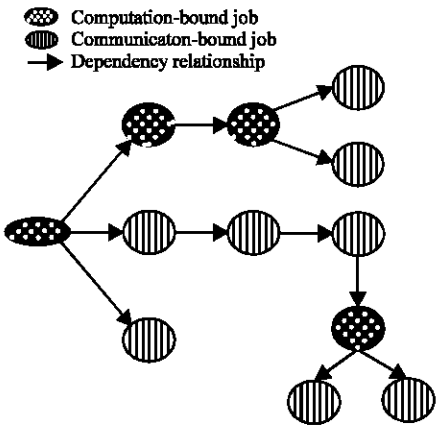

Fig. 1: Job dependency graph

- Each node is running a monitoring service which triggers scheduler when battery power touches threshold value
- Each node knows its location by using Global Positioning System (GPS) or other technique such as triangulation
- A node which intends to leave grid updates scheduler about decision

Algorithm: Job scheduler proposed in this study is based on centralized approach which results in effective scheduling decisions due to global view of network. As compared to de-centralized approach, it also reduces the communication cost associated with exchange of control information. This is because all the nodes report to central node. However, centralized approach suffers from scalability and single point of failure problems. We have chosen it by considering that ad hoc computational grids are temporally formed to conduct operations in emergency situation. These grids usually have small number of nodes; thus, wouldn't result scalability issue. However, to deal with single point of failure, an effective failure handling algorithm will be devised.

As ad hoc computational grids are temporarily formed to run specific application; therefore, once scheduler allocates jobs within an application, it remains idle. In this situation keeping it active during whole process is waste of valuable resources. To effectively utilize the resources, we use an event-driven job scheduling model in which scheduler is triggered in response to pre-defined events. For example, when application arrives, battery power touches threshold value, nodes sends message to leave grid, etc. Once jobs have been allocated, it deactivates itself making resources available for other applications. Figure 2 shows the scheduler's interaction with other components responsible for generating pre-defined events. The description of all the components is not scope of this study.

To keep control information, proposed algorithm maintains three tables: nodes information table, nodes distance table and job allocation table. Nodes information table records mobile nodes' characteristics such as processing power, memory and remaining battery power. Nodes distance table keeps distance information among mobile nodes which is calculated using location of nodes. Distance table is updated with distance information when mobile node changes the location. Job allocation table records the list of allocated jobs.

Whenever application arrives within a system, a job scheduler is triggered to schedule jobs within a mobile ad

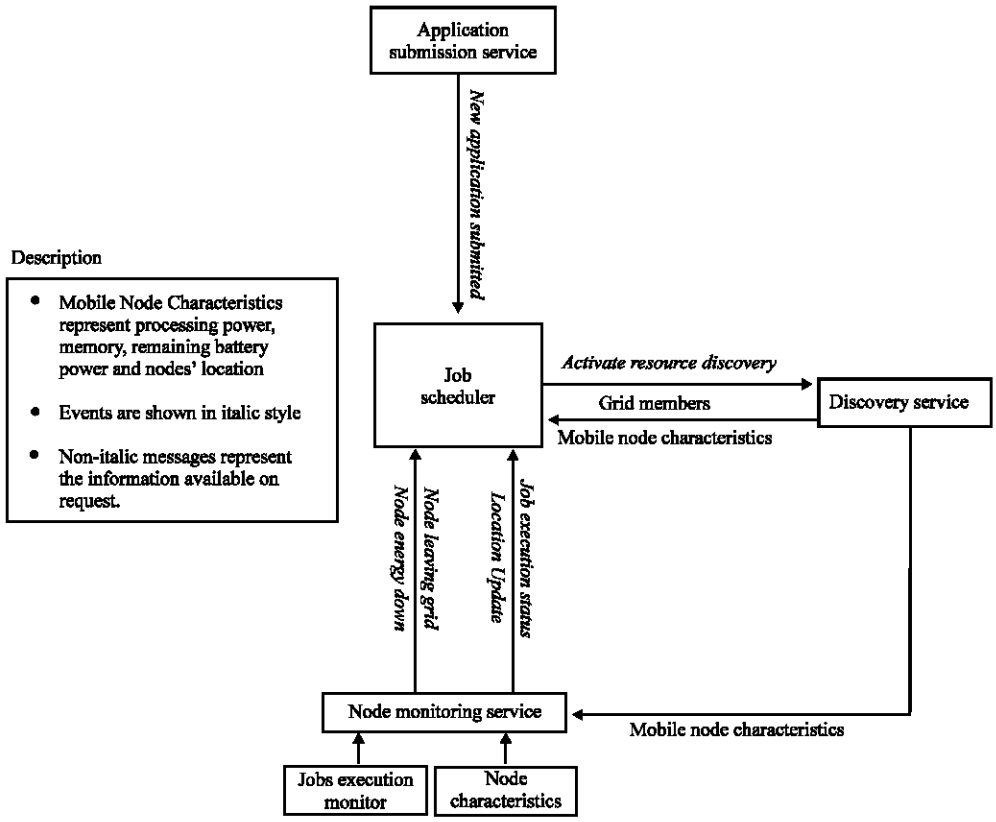

Fig. 2: Scheduler's interactions with other components

hoc computational grid. Before allocation of jobs, scheduler ensures that predecessor jobs have been allocated and finished execution. Later, it calls AllocateJobs function for the allocation of jobs. AllocateJobs function allocates inter-dependent jobs to nodes which are close to one another with respect to physical distance. It also ensures that nodes satisfy the requirements of jobs such as processing power, memory and battery power. In case of more than one dependent jobs with combination of communication-bound and computation-bound jobs, AllocateJobs gives priority to communication-bound jobs and allocate them to closer node. This is because communication-bound job generates more traffic; thus, having more affect on performance as compared to computation-bound job. Independent jobs are allocated to any node which satisfies the requirements of job such as processing power, memory and battery power.

To ensure that failure of node, either due to low battery power or mobility across coverage area, will not result failure of dependant jobs, our algorithm assume that when node leaves the grid it report scheduler regarding the decision. For failure due to low battery power, the monitoring service running on node keeps track of battery power and triggers scheduler when power reaches threshold value. In response to these two events, scheduler is activated, checks the dependency of running jobs and calls AllocateJobs function to reallocate jobs. When job running on a low battery power node or node which intends to leave grid, has a dependency, then it is reallocated close dependent job. Among dependents jobs communication-bound jobs have more priority as discussed earlier.

While allocating jobs to mobile nodes, scheduler puts information regarding the location of remote jobs, so they can communicate with one another directly. Whenever job is reallocated to different node, scheduler informs its dependant jobs about the new location. For details see algorithm 1.

Algorithm 1:

```
1:  IF newApplication_Arrived then
2:      For each job j_i ∈J
3:          IF j_i has pre (j_i) then
4:              IF pre (j_i) not allocated then
5:                  Allocate pre (j_i)
6:              Else IF pre (j_i) allocated and execution of pre (j_i) not
            completed then
```

Algorithm 1: Continued

```
7:                  Wait for pre (j_i) to complete execution
8:              Else IF execution of pre (j_i) completed then
                  /* Assign job j_i to mobile node closely located to
                predecessor job */
9:                  Invoke AllocateJobs pre (j_i ), (j_i)
                /* If independent job */
10:             Else IF j_i has no pre (j_i) then
11:               Invoke AllocateJobs (j_i)

        /* j_i is the job running on node with low battery power b_i or node
        n_i which is about to leave grid*/
12:  IF b_i < threshold or n_i is leaving the grid then
13:     IF j_i collecting results from pre (j_i) then
        Invoke AllocateJobs (pre (j_i) ,j_i)
        /* If independent jobs */
14:     Else IF j_i has no pre (j_i) then
15:       Invoke AllocateJobs (j_i)
16:  AllocateJobs (pre (j_i), j_i) {
        /* If there are more than one predecessor jobs */
17:     IF count ( pre (j_i) ) > 1 then
18:     Sort according to nature of job. j_i^comm-bound Followed by j_i^cpu-bound.
19:     Get location of node n_r executing pre (j_i)
20:     For each job j_i∈J
21:        Get node n_n close to n_r
22:        IF p_i >= rp_i and m_i >= rm_i and b_i >= rb_i then
23:           Allocate pre (j_i) to n_n
        /* If Independent job */
25:     IF ( pre (j_i) = null then
26:        Get node n_i
27:        IF p_i >= rp_i and m_i >= rm_i and b_i >= rb_i then
28:           Allocate j_i to n_i
29:  }
```

SIMULATION RESULTS

This study focuses on the communication performance and application completion time. For performance evaluation, we compare our algorithm with PeerPull (Liu *et al.*, 2005) and random allocation scheme. Random allocation scheme randomly allocate jobs to nodes without considering the inter-job dependencies. In PeerPull, only communication cost was calculated; however, migration cost has not been taken into account which will further increase communication cost, energy consumption as well as communication delay due to suspension and resumption of execution from one node to another node.

Simulation setup: We implemented our algorithm in ns-2 simulator. Group mobility model was used and 24 nodes were deployed randomly in groups of variable size. We deployed constant bit rate applications with varying number of jobs and communication traffic among them. The traffic was generated according to nature of job. Communication-bound jobs generated more traffic as compared to computation-bound jobs. Detailed simulation parameters are given in Table 1.

Table 1: Simulation parameters

Simulation time	1000 sec
No. of nodes	24
Transmission range	250 m
Area	700×700 m
No. of jobs	12-24-36-48
Transport protocol	TCP
Ad hoc routing protocol	AODV
MAC protocol	IEEE 802.11
Traffic type	Constant bit rate
Packet rate	2 Packets/s
Packet size	512 Bytes

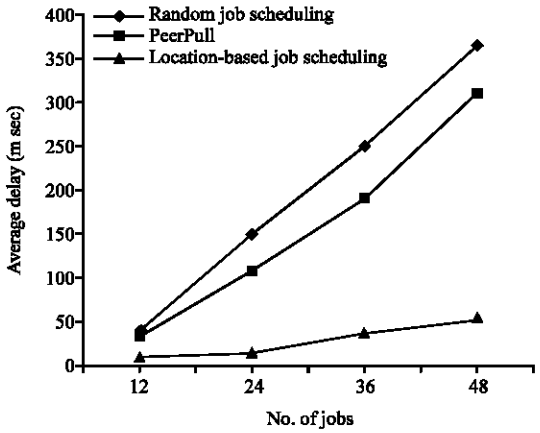

Fig. 3: Average End-to-end delay with increasing number of jobs

Simulation results: Figure 3 demonstrates the average end-to-end communication delay of three schemes. For given scenarios, the ratio of inter-dependent jobs and communication traffic were same.

As shown in Fig. 3, in case of moderate number of jobs such as 12 and 24, our scheme outperforms existing ones. There is also steady increase in growth rate. However, with increasing number of jobs, there is significant increase in growth rate. There are many possible reasons for this. For example, with increasing number of jobs hence communication traffic, there is considerable increase in routing traffic. Moreover, as show in Fig. 5 and 6, the number of drop and lost packets also increase significantly. The unavailability of closer nodes and order in which jobs arrive within a system are also critical to performance. For example, when there are large numbers of jobs, some of them are allocated to closer nodes while others are allocated to distant nodes increasing the numbers of forwarded, drop and lost packets. Furthermore, if communication-bound jobs follow computation-bound jobs then computation-bound jobs are allocated to closer nodes making distant nodes available for communication-bound jobs. As communication-bound jobs generate more traffic thus

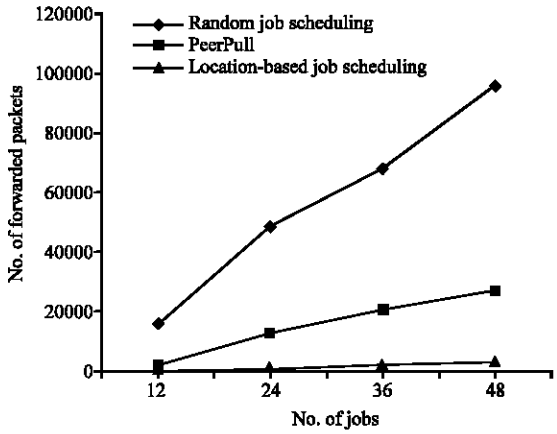

Fig. 4: No. of forwarded packets

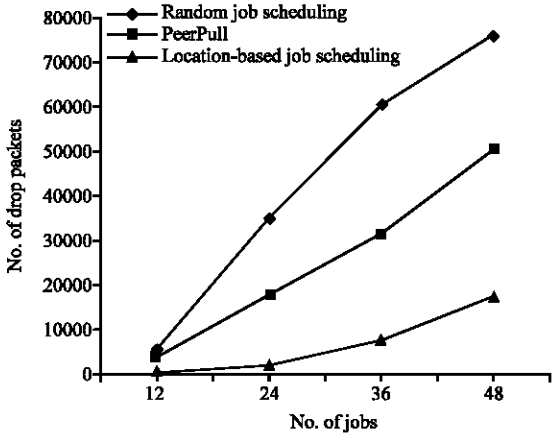

Fig. 5: No. of drop packets

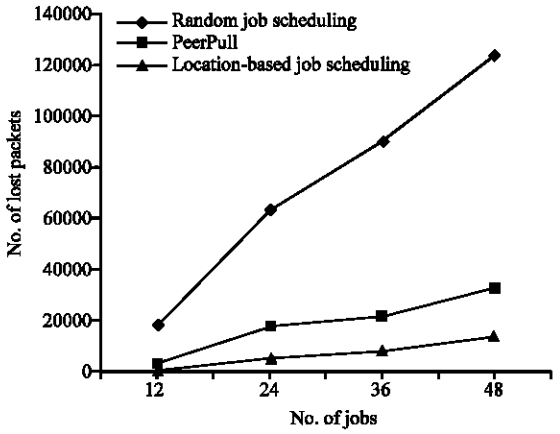

Fig. 6: No. of lost packets

allocation of these jobs to distant nodes has more impact on communication performance as compared to computation-bound jobs.

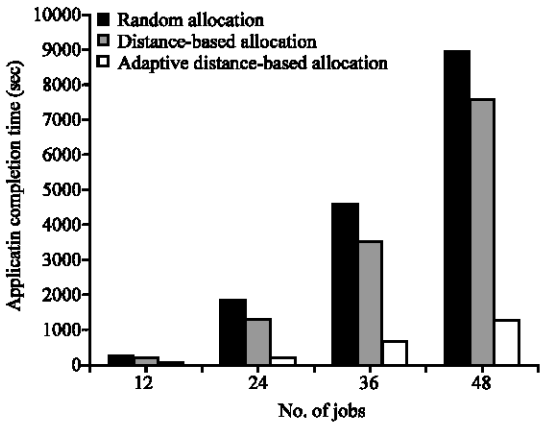

Fig. 7: Application completion time

As compared to our approach, random and PeerPull job schedulers result in poor performance. Both allocate jobs randomly; therefore, jobs may be allocated to distant nodes increasing communication distance. As shown in Fig. 4-6, the increased communication distance results increase in the number of forwarded, drop and lost packet. However, as compared to random scheduler, PeerPull improves the performance through monitoring the communication traffic of jobs. In case jobs communicate frequently, it migrates them to closer nodes reducing the communication distance thus communication cost.

To calculate the application completion time, we used cost model given (Zomaya, 1995). In this study we are targeting applications in which communication cost is much more than execution cost. For simplicity, we assume same computation cost for all the jobs.

As shown in Fig. 7, our approach minimizes the application completion time as compared to random and PeerPull job scheduling approaches. This is because our approach improves the communication performance which is directly related to overall application performance. Thus, by minimizing the communication cost, our approach significantly reduces application completion time.

CONCLUSION

Due recent advancements in mobile computing and communication technologies, mobile ad hoc computational grids are emerging as a new computing paradigm, enabling novel applications through the sharing of computing resources among mobile devices in ad hoc manner. However, the integration of computational grid and ad hoc network is not straightforward and introduces many challenges. One of the key challenges in these systems is scheduling of inter-dependent jobs. As

the performance of inter-dependent jobs is greatly affected by communication performance; thus, in this paper, we have proposed a centralized location-based job scheduling algorithm that improves the communication performance of inter-dependent jobs. It exploits the location of computing nodes and inter-job dependencies to schedule them on closely located nodes. The performance of proposed algorithm was compared with random and PeerPull job schedulers through simulation. The simulation results demonstrated that proposed algorithm minimizes the communication cost and application completion time.

ACKNOWLEDGMENT

Professor Myong-Soon Park is the corresponding authors. We are thankful for his supervision to this research.

REFERENCES

Antonio, T.A.G., A. Ziviani, L.S. Lima and M. Endler, 2007. DICHOTOMY: A resource discovery and scheduling protocol for multihop ad hoc mobile grids. Proceedings of the 7th IEEE International Symposium on Cluster Computing and the Grid, May 14-17, Rio de Janeiro, Brazil, pp: 719-724.

Apple Computers, 2009. IPhone. http://www.apple.com/iphone/iphone-3gs/.

Arora, M., S.K. Das and B. Rupak, 2002. A de-centralized scheduling and load balancing algorithm for heterogeneous grid environments. Proceedings of the International Conference on Parallel Processing Workshops, Aug. 18-21, Vancouver, BC, Canada. IEEE Computer Society, pp: 499-505.

Baker, M., R. Buyya and L. Domenico, 2002. Grids and Grid Technologies for Wire Area Distributed Computing, Software Practice and Experience. John Wiley and Sons, Ltd., UK.

Braun, T.D., J.S. Howard, A.M. Anthony and Y. Hong, 2008. Static resource allocation for heterogeneous computing environments with tasks having dependencies priorities deadlines and multiple versions. J. Parallel Distrib. Comput., 68: 1504-1516.

Cao, J. and O.M.K. Kwong, 2005. A peer-to-peer approach to task scheduling in computational grid. Int. J. Grid Utility Comput., 1: 13-21.

David, C.C. and M. Humphrey, 2004. Mobile OGSI.NET: Grid computing on mobile devices. Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, Nov. 8, Pittsburgh, PA., pp: 182-191.

Feitelson, D.G., L. Rudolph and U. Schwiegelshohn, 2004. Parallel job scheduling a status report. Job Scheduling Strategies Parallel Process., 3277: 1-16.

Fox, G., A.R.W. Chu and E.I. Kwan, 2008. A collaborative sensor grids framework. Proceedings of the International Symposium on Collaborative Technologies and Systems, May 19-23, Irvine, California, USA., IEEE Conference Publishing Services, pp: 29-38.

Franke, H.A. and L.K. Fernando, 2007. Grid-M: Middleware to integrate mobile devices, sensors and grid computing. Proceedings of the 3rd International Conference on Wireless and Mobile Communications, March 4-9, Guadeloupe, French Caribbean, pp: 19-19.

Gonzalez, C.F., J.V. Alonso and M. Livny, 2005. Condor grid computing from mobile handheld devices. Mobile Comput. Commun. Rev., 6: 117-126.

Grabowski, P., B. Lewandowski and M. Russell, 2004. Access from J2me-enabled mobile devices to grid services. http://www.gridlab.org/WorkPackages/wp-12/res/docs/Mobility_Paper_GridLab.pdf.

Hong, X., G. Mario and C.C. Chiang, 1999. A group mobility model for ad hoc wireless networks. Proceedings of the 2nd ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems Seattle, (AIWMASWMSS'99), Washington, United States, pp: 53-60.

Hummel, K.A. and G. Jelleschitz, 2007. Robust de-centralized job scheduling approach for mobile peers in ad hoc grids. Proceedings of the 7th IEEE International Symposium on Cluster Computing and the Grid, May 14-17, Rio de Janeiro, Brazil. IEEE Computer Society Washington DC., pp: 461-470.

IBM, 2009. Grid computing. http://www.ibm.com/developerworks/grid.

Intel Corporation, 2009. Ultra-wideband (UWB technology): Enabling high-speed wireless personal area networks. http://www.intel.com/technology/comms/uwb/download/Ultra-Wideband.pdf.

Liu, H., T. Roeder, K. Walsh, B. Rimon and E.G. Sirer, 2005. Design and implementation of a single system image operating system for ad hoc networks. Proceedings of the 3rd International Conference on Mobile Systems Applications and Services, June 6-8, Seattle WA., pp: 149-162.

Maria, C., H.A. Flip and D.T. Flip, 2008. Scheduling of dependant grid jobs in absence of exact job length information. Proceedings of the EVGM2008 4th IEEE/IFIP International Workshop on End-to-End Virtualization and Grid Management in Conjunction with the 5th International Workshop on Next Generation Networking Middleware, (EIIWEVGMCIWNGNM'08), TW05 Department of Information Technology, pp: 185-196.

Marinescu, D., G. Marinescu, Y. Ji, L. Boloni and H.J. Siegel, 2003. Ad hoc grids: communication and computing in a power constrained environment. Proceedings of the Workshop on Energy-Efficient Wireless Communications and Networks, Apr. 9-11, Phoenix, Arizona, pp: 113-122.

Markov, L., 2004. Two stage optimization of job scheduling and assignment in heterogeneous compute farms. Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems, May 26-28, Suzhou, China. IEEE Computer Society Washington DC., pp: 119-124.

Millard, D., A. Woukeu, F.B. Tao and H. Davis, 2005. Experiences with writing grid clients for mobile devices. Proceedings of 1st International ELeGI Conference on Advanced Technology for Enhanced Learning, March 14-16, Vico Equense, Italy, http://eprints.ecs.soton.ac.uk/10671/.

Robinson, J., G.F. Jeremy, J.S. Andy, A.D. Reynolds and B.V. Bedi, 2005. Sensor networks and grid middleware for laboratory monitoring. Proceedings of the 1st International Conference on e-Science and Grid Computing, Dec. 5-8, Melbourne, Australia. IEEE Computer Society, pp: 569-577.

Senger, L.J., R.F. de Mello, M.J. Santana, R.H.C. Santana and L.T. Yang, 2005. Improving Scheduling Decisions Using Knowledge about Parallel Applications Resource Usage. In: Lecture Notes in Computer Science, Yang, L.T. *et al*. (Eds.). Springer, Berlin, Heidelberg, ISBN: 978-3-540-29031-5 pp: 487-498.

Shah, S.C., A. Kashif, S.H. Chauhdary, C. Jiehui and M.S. Park, 2009. Mobile ad hoc computational grid for low constraint devices. Proceedings of the International Conference on Future Computer and Communication, April 3-5, Kuala Lumpur, Malaysia. IEEE Conference Publishing Services, pp: 416-420.

Shivle, S., H.J. Siegel, A.A. Maciejewski, P. Sugavanam and T. Banka *et al*., 2006. Static allocation of resources to communicating subtasks in a heterogeneous ad hoc grid environment. J. Parallel Distributed Comput., 66: 600-611.

Sodan, A.C., 2005. Loosely coordinated co-scheduling in the context of other approaches for dynamic job scheduling: a survey. Concurrency Comput. Practice Exp., 17: 1725-1781.

Wireless Grids Lab, 2009. Wireless grids working paper I. http://wirelessgrids.net/people.html.

Zomaya, A.Y.H., 1995. Parallel and Distributed Computing Handbook. 1st Edn., McGraw-Hill UK.