J Electr Eng Technol.2016; 11(4): 987-992
http://dx.doi.org/10.5370/JEET.2016.11.4.987

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

Energy Aware Task Scheduling for a Distributed MANET Computing Environment

Jaeseop Kim* and Jong-Kook Kim†

Abstract – This study introduces an example environment where wireless devices are mobile, devices use dynamic voltage scaling, devices and tasks are heterogeneous, tasks have deadline, and the computation and communication power is dynamically changed for energy saving. For this type of environment, the efficient system-level energy management and resource management for task completion can be an essential part of the operation and design of such systems. Therefore, the resources are assigned to tasks and the tasks may be scheduled to maximize a goal which is to minimize energy usage while trying to complete as many tasks as possible by their deadlines. This paper also introduces mobility of nodes and variable transmission power for communication which complicates the resource management/task scheduling problem further.

Keywords: Distributed mobile computing, Dynamic resource allocation, Dynamic task scheduling, Power-aware, Energy-aware, MANET

1. Introduction

The mobile ad hoc network (MANET) is a truly peer-to-peer network where each device is mobile, have limited battery, uses wireless communication, and the connection of devices can change dynamically. A distributed mobile computing (DMC) environment may be based on the MANET environment where users can share their computational load with other devices and the result can be sent back to the original user ([25]). The system-level energy and each of the mobile devices' energy must be used efficiently in the sense that when tasks are allocated resources, estimated energy use to complete the task and/or the remaining energy of the device must be considered to complete as many tasks as possible while also try and prolong the devices' life. The DMC environment is usually heterogeneous in the sense that various devices have different characteristics such as processing speeds, different battery lifetimes, and architecture while tasks or applications may have affinity to certain devices. The heterogeneity of the resources and tasks must be exploited to maximize the performance or the cost-effectiveness of a system. To exploit the heterogeneity of devices and tasks, an important research problem is how to assign resources to the tasks (match) and to order the tasks for execution on the resources (schedule) to maximize some performance criterion of the system. This procedure is called mapping, task scheduling, and/or resource allocation. A resource management system (RMS) takes care of allocating the resources of a system. The energy/power management of computation and communication and the device mobility further complicates the resource management problem. In this research, dynamic mapping is performed because tasks arrive at unpredictable intervals and are mapped as they arrive (workload is not known *a priori*). In general the mapping problem is proved to be NP-complete. Thus, the development of heuristic techniques to find near-optimal solutions for task scheduling is an active research area (e.g., [1-4]). The environment proposed in this paper is simulated by a simulation tool called energy-aware distributed mobile computing simulator EArDruM ([20]), which is based on network simulator 2 (NS-2) [16].

The power/energy management of the proposed DMC environment is accomplished by using dynamic voltage scaling (DVS) [5] and variable-range transmission power control (VTPC) [6] for the computation and the communication, respectively. DVS is based on exploiting the relationship between the CPU supply voltage of a device and the power usage. The relationship of power to voltage is a strictly increasing convex function, represented by a polynomial of at least second degree [7]. The VTPC technique can improve the overall network energy usage because different transmission power levels can be used to send data according to the distance between the source and the destination devices whenever communication occurs. The DVS and VTPC in this research is managed by the resource manager and is transparent to the user.

In this paper we assume an environment where a user can request a program (task) to be executed, receive data, and send data using the mobile device. A device performing a computation may receive input data from other devices. The resulting output will be sent back to the task requester. The scenario from [13] is adopted and extended by introducing factors such as device mobility

---

† Corresponding Author: School of Electrical and Computer Engineering, Korea University, Seoul, South Korea. (jongkook@korea.ac.kr)

* HMC Investment Securities, Seoul, South Korea.

Received: September 9, 2015; Accepted: December 19, 2015

and VTPC for this research. For the efficient use of the overall system, it may be best for certain tasks to be executed on a remote, rather than on the local device. The reasons are 1) limited energy remaining on the local device, 2) a remote device can execute the task using less energy, and 3) a remote device can complete the task by its deadline while the local device cannot. An RMS makes this decision of locating a "suitable" device and allocating the task to that device. In this research, tasks have deadlines and the primary goal of this research is to complete as many tasks by their deadlines as possible while trying to efficiently use the overall system energy during a given interval of time.

The contributions of this research include 1) the modeling of dynamically mapping tasks with deadlines onto mobile wireless devices while managing power using the DVS and VTPC methods, and 2) the design, analysis, and comparison of six simple resource allocation methods for this environment. Section 2 introduces the heterogeneous DMC environment that was modeled for this research. In Section 3, the heuristics designed for this research are presented. Section 4 describes the simulation setup. The results are examined in Section 5 followed by a brief summary of the literature related to this paper. The last section gives a summary of this research.

2. Distributed Mobile Computing Environment

In this research, a single-hop distributed mobile computing (DMC) system is considered where the devices are mobile and communicate with each other by wireless means creating an *ad hoc* network. These devices have limited battery capacity (energy), use dynamic voltage scaling (DVS) for computation power management, use variable-range transmission power control (VTPC) for communication power management, and have movement (mobility) capabilities. The batteries are assumed to be recharged after a certain amount of time and the battery capacity is different for different devices. The number and value of the discrete voltage levels of DVS vary among the devices. But the number of the transmitted power levels (varied according to the distance between two communicating devices) and the transmission power of each of the levels for VTPC are identical for all devices. In this paper, a centralized resource management system (RMS) is used to maximize the performance goal which is to complete as many tasks by their deadlines as possible while trying to efficiently use the overall system energy during a given interval of time. After the task execution is completed (t0 → t1 in Fig. 1), the result is sent back to the task requester. More details can be found in [20].

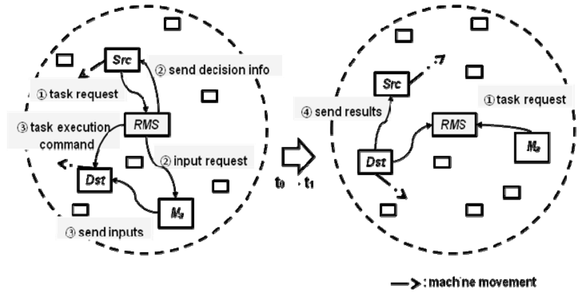

Fig. 1. Shows an example of a distributed mobile computing environment.

The tasks discussed here have a deadline. If a task cannot complete by its deadline, it has no value. It is assumed that the RMS knows all of the devices' status information and the tasks' execution times on those devices. Hence, the RMS can accurately decide a "suitable" device for a requested task. The estimated execution times of each task on each device is assumed to be known based on user supplied information, experiential data, task profiling, and analytical benchmarking, or other techniques (e.g., [8, 9]). This DMC environment was simulated using the EArDruM simulator [20] which is modified and extended from the network simulator 2 (NS-2) [16].

3. Resource Management Methods

3.1. Overview

In this environment, a mapping event occurs when a task is requested from one of the mobile devices and the task will be considered for mapping onto one of the devices in the DMC system. Before deciding on a device, all devices may be checked to verify whether the task can be executed on the device. For example, the remaining energy (i.e., the energy that is remaining after being used for completing tasks and after being drained of idle energy) of a device may not be enough to process the task or communicate with other devices. After the verification, all heuristics described in the following subsections map a task onto one of the selectable devices which are devices that are expected to complete the tasks by their deadline. If a task is already executing on a device when another task is assigned to the device, the new task is enqueued into the scheduling queue of the device. The task already executing on the device is not preempted. The position of devices are initially set and the direction (current position is known and the destination position is randomly selected and the velocity of a device is selected randomly from 0 to 10m/s. All data are communicated using VTPC to use as little energy as possible. The main difference of the task scheduling schemes described in this research is shown in Table 1.

3.2. Originator and random

The originator heuristic executes the new task on the original device itself when the task is requested. This heuristic will consume the smallest amount of communication energy as there will only be input communication when needed. The random heuristic maps the new task on a random device from the selectable devices. In both heuristics, if the selected device cannot complete the task by its deadline, the task will have no value. Even if the task fails, it will use the battery while it is being executed and the energy consumption is updated whenever there is a new task request. All tasks are executed using the highest DVS level which means using the fastest level and the highest energy consuming level.

3.3 Current minimum communication energy (CMCE) and estimated minimum communication energy (EMCE)

The CMCE heuristic maps a task onto the device other than itself which will minimize the communication energy. The EMCE heuristic is similar with the CMCE heuristic. But, the EMCE heuristic estimates where the device will be when the result will be sent back to the task requester and calculates the amount of energy needed for the whole communication. All tasks are executed using the highest DVS level which means using the fastest level and the highest energy consuming level. Both methods follow the procedure described below.

```
For every mapping event (a new task is requested),
    Calculate the roundtrip communication energy to all selectable devices and the input communication energy to the selectable devices using VTPC.
    (CMCE: use the current position of the all devices for the calculation;
    EMCE: use the estimated location of the task requester device and selectable devices at the task completion time when calculating the communication energy when sending back the results.)
    Map the task onto the minimum total communication energy consumption device.
```

3.4 Current minimum total energy (CMTE) and estimated minimum total energy (EMTE)

In the CMTE heuristic, the task is executed on a device which will consume the minimum total energy (communication and computation) and can complete by the task's deadline. Similarly, the EMTE heuristic maps new task to the minimum total energy consumption device. But, instead of the current coordinates of the device, EMTE heuristic uses the estimated coordinates of the source and destination device after the task execution time to calculate the estimated total energy consumption for all selectable device. The DVS is applied such that the device can complete the task by its deadline while trying to minimize the energy being used in both heuristics. Both heuristics follow the procedure described below.

```
For every mapping event (a new task is requested),
    Calculate the total energy to all selectable devices.
    The communication energy is calculated using the distance using VTPC and the computation energy is calculated using the lowest possible DVS level while the task can complete by its deadline.
    (CMTE: use the current position of all the devices for the calculation;
    EMTE: use the estimated location of the task requester device and selectable devices at the task completion time when calculating the communication energy when sending back the results.)
    Map the task onto the minimum total energy consumption device.
```

Table 1. The differences between the schemes. (CE is the communication energy, CPU is the computation energy, EL is the estimated location of devices, D is the deadline of tasks, N means does not consider, and Y means it is considered.)

heuristics	CE	CPU	EL	D	DVS level
Originator	N	N	N	N	highest
Random	N	N	N	N	highest
CMCE	Y	N	N	N	highest
EMCE	Y	N	Y	N	highest
CMTE	Y	Y	N	Y	lowest possible
EMTE	Y	Y	Y	Y	lowest possible

4. Simulation Model

Eighteen different types of wireless computing devices, using information found in [20], and different task types are assumed to be used in the DMC system. Assuming the RMS knows the different types of devices/tasks and how fast each tasks run on the devices at the highest DVS level, it is assumed that the estimated time to compute (ETC) information of all tasks on the different devices are known. However, the arrival times or requested times of tasks and the task type is not known a priori.

A simplified DVS technique is used in this research that assumes each voltage level of a processor corresponds to a clock speed level for the processor. Details are in [20].

In each simulation of a system, eleven devices among the eighteen types are picked with equal probability. The RMS is placed on the first device among the eleven devices, situated in the middle of the simulated area, has no mobility, and the node does not execute any tasks. The simulation of the arrival of tasks or request of tasks is done using a Poisson distribution with mean six, eight, and ten seconds. (There are three scenarios.) The system is simulated for 28,800 seconds (i.e., assuming an eight hour work day). For all tasks, the ET values on eighteen types of devices taking heterogeneity into consideration is randomly generated using the gamma distribution method described in [17]. Three mean execution times of 60, 100, and 200 seconds, are used for the ET matrix. The mean execution time is chosen to represent applications such as processing data (such as maps or weather reports), generating strategies, etc. A trial is defined as one such simulation of the system. For each of the nine scenarios (three mean inter-task arrival time multiplied by three mean execution times), 30 trials are run.

For all devices, the transmission and reception power for each of the levels for VTPC is calculated. This research applies the IEEE 802.11b standard for wireless communication ([18, 19]). To utilize the VTPC technique, it is assumed that the LAN Card has seven discrete transmission power levels (for 10, 50, 100, 150, 200, 250, and 500 meters). The transmission power is calculated based on the two-ray ground reflection model in NS-2.

For the simulation study, the maximum battery capacity (energy) of device *j*, BC(*j*), is set to the maximum CPU energy consumption energy plus the transmission energy consumption energy, multiplied by the maximum operation time. The maximum operation time is determined using a gamma distribution with a mean of two hours. For simulation purposes, the size of the task and output (result to the source) data was calculated using 100 Kbytes as the mean and a COV of 0.7 using a Gamma distribution. The size of the input data was calculated using 1 Kbytes as the mean.

This research assumes that when a task arrives, the deadline of the task is given. For our simulation studies, the deadline is chosen as to following to allow the new task to have a "good" chance of completing before a certain deadline. The deadline of task *i* is equal to its arrival time plus the overall mean execution time of all tasks using Gamma distribution (COV of 0.7) plus the median execution time of task *i* on all devices plus the average communication time of the task request and the result.

5. Results

The simulation results for the three mean inter-task arrival times and three mean execution times are shown (total of nine scenarios) in Figs. 2, 3, and 4. As the mean inter-task arrival times decrease, the number of tasks in the system also increases and the percentage of tasks completed decreases. Similarly, as the mean execution times increase, the percentage of tasks completed decreases as it takes more time and energy to complete the tasks. And sometimes because of the longer mean execution time, the tasks are more likely to be dropped. The average number of tasks per trial was 2837, 3579, and 4765 for the mean inter-task arrival time of ten, eight, and six seconds. In all the figures, the originator heuristic can be said to be the best among the four low performance methods. This is because the originator method executes the task on the device that requested the task and therefore there is almost no communication energy consumption. The performance of the CMCE and EMCE heuristics is poor because they do not consider the task processing or computation times and energy consumption. The CMTE and EMTE heuristics are the best performing methods out of these six methods that are introduced for all scenarios. The reason is that the heuristics consider task deadline, task computation time on various devices, and reduce energy consumption using DVS and VTPC. The simple estimation of the devices' location did not enhance the performance of the algorithm

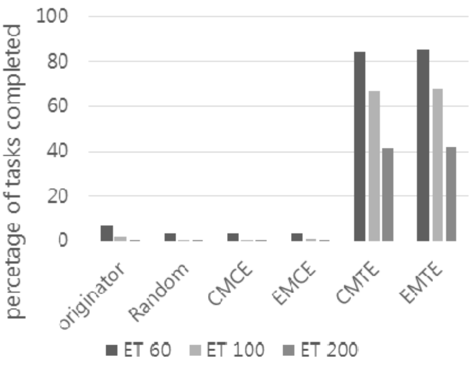

Fig. 2. The simulation result using the mean inter-task arrival rate of six seconds and mean estimated execution time (ET) of 60, 100, and 200 seconds.

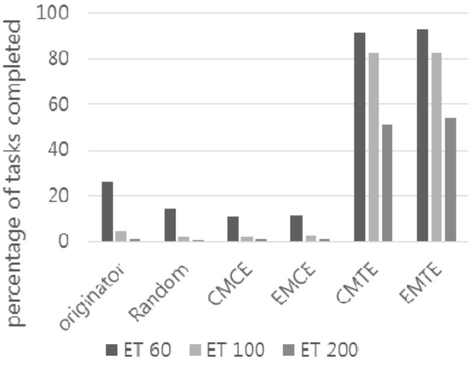

Fig. 3. The simulation result using the mean inter-task arrival rate of eight seconds and mean estimated execution time (ET) of 60, 100, and 200 seconds.

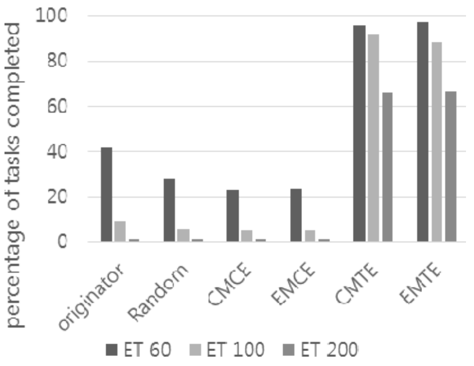

Fig. 4. The simulation result using the mean inter-task arrival rate of ten seconds and mean estimated execution time (ET) of 60, 100, and 200 seconds.

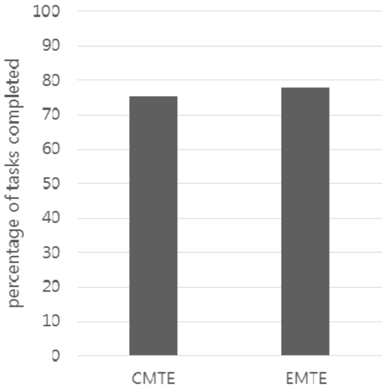

Fig. 5. The simulation result using the mean inter-task arrival of 6 seconds and mean execution time of 200 seconds.

much. But the EMTE method performed better when the system is congested and there are more tasks that cannot be completed by their deadline. As the system's resources are constrained, the location estimation plays a small role where some of the tasks are completed as expected.

To test for an environment where the number of devices is larger than ten, we assumed there are 20 devices for the worst scenario. (i.e., the mean inter-task arrival rate of six seconds and mean execution time of 200 seconds.) As shown in Fig. 5 the performance has improved but not as much as expected which would intuitively be two times. The reason is that many tasks wait for the communication of the results and when the final result is not sent before the deadline the task has failed. The reason for this can be failed communication between devices that did not occur when there were only ten devices.

The energy consumption graph is not shown because in the simulations, all of the energy is used and all of the devices died before the end of the simulation. Random and originator tried to complete tasks but most of them are not completed because a lot of tasks were completed after their deadlines. Thus using energy but only completing a small number of tasks.

6. Related Work

There are lot of research for minimizing or saving energy or power consumption of a mobile computing system ([10-12]). The difference is that our system includes computation times and the energy used for computation in the overall energy usage. Also, the tasks and devices are heterogeneous and devices use DVS and VTPC technique for power/energy management while tasks have deadlines.

Some research projects have explored the mobile ad hoc network (MANET) environment using the DVS technique [13], VTPC technique [14], or both techniques [15]. The difference is that our research considers mobility of devices, VTPC, the computation of tasks as well as the communication of data and tasks, heterogeneous nature of the environment, and the real-time aspect of the system.

7. Summary

This paper introduced an example distributed mobile computing environment and used the EArDruM simulator to simulate the environment. The mobile devices are modeled based on real specifications and have discrete DVS and VTPC levels for the power/energy management for device and communication respectively. Heterogeneity is introduced to the system via different DVS levels, VTPC levels, battery capacity, processing speed, and tasks with affinity to certain devices. Implementation of a simple function of calculating the expected coordinates of the mobile devices after a predetermined time is included for communication time and energy estimation. The tasks had heterogeneous execution times and deadlines. The devices are mobile with random speed and direction. Six simple methods for this environment are tested and compared.

Further investigation of the proposed environment can include the following: increasing the number of devices, increase the mean of the size of the task, input, and output data, and different calculation of the deadline of the task.

Acknowledgements

This research is funded in part by the National Research Foundation of Korea grant no. 2009-0076378.

References

[1] T. D. Braun, H. J. Siegel, and A. A. Maciejewski, “Heterogeneous Computing: Goals, Methods, and Open Problems,” PDPTA ’01, invited keynote paper, pp. 1-12, June 2001.

[2] H. Barada, S. M. Sait, and N. Baig, “Task Matching and Scheduling in Heterogeneous Systems Using Simulated Evolution,” HCW ’01, IPDPS ’01, Apr. 2001.

[3] T. D. Braun, H. J. Siegel, N. Beck, L. Boloni, R. F. Freund, D. Hensgen, M. Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, and B. Yao, “A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems,” JPDC, vol. 61, no. 6, pp. 810-837, June 2001.

[4] J.-K. Kim, S. Shivle, H. J. Siegel, A. A. Maciejewski, T. D. Braun, M. Schneider, S. Tideman, R. Chitta, R. B. Dilmaghani, R. Joshi, A. Kaul, A. Sharma, S. Sripada, P. Vangari, and S. S. Yellampalli, “Dynamically Mapping Tasks with Priorities and Multiple Deadlines in a Heterogeneous Environment,” JPDC,

vol. 67, no. 2, pp. 154-169, Feb. 2007.

[5] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for Reduced CPU Energy," Usenix Symposium OSDI '94, pp. 13-23, Nov. 1994.

[6] J. Gomez and A. T. Campbell, "A Case for Variable-Range Transmission Power Control in Wireless Multihop Networks," INFOCOM'04, pp. 1425-1436, Mar. 2004.

[7] I. Hong, G. Qu, M. Potkonjak, and M. Srivastava, "Synthesis Techniques for Low-Power Hard Real-Time Systems on Variable Voltage Processors," 19th IEEE Real-Time Systems Symp. (RTSS '98), pp. 95-105, Dec. 1998.

[8] A. Ghafoor and J. Yang, "A Distributed Heterogeneous Supercomputing Management System," IEEE Computer, vol. 26, no. 6, pp. 78-86, June 1993.

[9] J. Yang, I. Ahmad, and A. Ghafoor, "Estimation of Execution Times on Heterogeneous Supercomputer Architectures," ICPP '93, pp. I-219-I-226, Aug. 1993.

[10] R. Cohen and B. Kapchits, "An Optimal Algorithm for Minimizing Energy Consumption while Limiting Maximum Delay in a Mesh Sensor Network," INFOCOM '07, pp. 258-266, May 2007.

[11] E. Altman, K. Avrachenkov, G. Miller, and B. Prabhu, "Discrete Power Control: Cooperative and Non-Cooperative Optimization," INFOCOM '07, pp. 37-45, May 2007.

[12] W. Chen, M.J. Neely, and U. Mitra, "Energy Efficient Scheduling with Individual Packet Delay Constraints: Offline and Online Results," INFOCOM '07, pp. 1136-1144, May 2007.

[13] J. -K. Kim, H. J. Siegel, A. A. Maciejewski, and R. Eigenmann, "Dynamic Resource Management in Energy Constrained Heterogeneous Computing Systems Using Voltage Scaling," IEEE TPDS, vol. 19, no. 11, pp. 1445-1457, Nov. 2008.

[14] T. L. Wong, T. Tsuchiya, and T. Kikuno, "An Energy-Efficient Broadcast scheme for Multihop Wireless Ad Hoc Networks Using Variable-Range Transmission Power," IEICE Trans. Inf. & Syst., vol. E90–D, no. 3, Mar. 2007.

[15] G. S. A. Kumar, G. Manimaran, and Z. Wang, "End-to-End Energy Management in Networked Real-Time Embedded Systems," IEEE TPDS, vol. 19, no. 11, pp. 1498-1510, Nov. 2008.

[16] K. Fall and K. Varadhan, The ns manual. Available at http://www.isi.edu/nsnam/ns/

[17] S. Ali, H. J. Siegel, M. Maheswaran, D. Hensgen, and S. Ali, "Representing Task and Machine Heterogeneities for Heterogeneous Computing Systems," Tamkang J. Science and Eng., special 50th anniversary issue (invited), vol. 3, no. 3, pp. 195-207, Nov. 2000.

[18] Nokia Mobile Phones, "Nokia C110/C111 wireless LAN card," User guide, http://nds1.nokia.com/phones/files/guides/C110-C111_usersguide_en.pdf, 2009.

[19] Socket Communications, "Low Power Wireless LAN Card," Datasheet, http://www.quad.de/Datashe/Socket_WLAN. pdf, 2009.

[20] J. S. Kim, "Energy-Aware Distributed Mobile Computing for Real-time Single-hop Ad hoc Mobile Environments", Korea University, Master's thesis, 60 pp., Feb. 2010.

Jaeseop Kim is currently a Senior Software Engineer at the Ministry of National Defense. He received his M.S. degree in electrical engineering from Korea University in February 2010. His research interests include heterogeneous distributed computing, real-time mobile computing, distributed compilers, and financial mobile application development.

Jong-Kook Kim is currently an Associate Professor at Korea University, Seoul Korea. He received his M.S. and Ph.D. degrees in electrical and computer engineering from Purdue University in May 2000 and August 2004, respectively. His research interests include heterogeneous distributed computing, real-time mobile computing, evolutionary heuristics, energy-aware computing, efficient computing, machine learning and distributed compilers. He is a senior member of the IEEE and ACM.