International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Survey on Task Allocation Algorithms in Wireless Sensor and Actor Networks

Sareh Beheshti, Mansoureh Ghiasabadi Farahani, Mona Sharifnejad and Mohsen Sharifi

***Abstract* – Tasks allocation refers to a group of tasks assigned to actor nodes in wireless sensor actor networks (WSANs).In these networks, sensor nodes gather information about environment while actor nodes make appropriate actions on the environment based on sensory information they receive from sensors. In order to provide effective assigning task, two factors namely time-aware and energy-aware are important. This paper presents a concise survey on some of existing task allocation algorithms in WSANs.**

***Keywords* – Wireless Sensor Actor Networks, Task Allocation, Energy-Aware, Time-Aware.**

I. Introduction

Over the last few years, technological advances have cause to the genesis of wireless sensor and actor networks (WSANs). Such networks include a set of sensor nodes and actor nodes connected via wireless links. They are able to monitoring the physical world, processing the data, making decisions and performing proper actions (tasks).Sensor nodes gather information about environment while actor nodes make appropriate actions on the environment based on sensory information they receive from sensors.

These networks are characterized by limited power and energy supplies. In many applications of them, reduction of completion times of tasks is very important. To make efficient use of WSANs capabilities, employing appropriate task allocation algorithms is indispensable [1]. Hence, many task-scheduling algorithms for distributed systems have been presented so far with the aim of minimizing the total completion time of tasks [2]-[3]-[8]. These algorithms allocate tasks to most suitable actors to minimize the overall completion time [8].

In many applications of wireless sensor actor networks (WSANs) that often run in harsh environments, the reduction of completion times of tasks is highly desired [8]. Researchers have begun focusing on task large number of exiting task scheduling algorithms for distributed systems in general that try to reduce the total task completion time and energy consuming of the system. The reminder of this paper is organized as follows. Section II presents WSANs Architecture. Section III describes Dynamic Mapping in Task Allocation. Section IV presents Task Allocation Algorithms in WSANs. Section V concludes the paper.

II. Wireless Sensor Actor Networks Architecture

There are basically three architectures depending on the strategies adopted by actors to send commands: 1) semi-automated, 2) automated, and 3) cooperative architecture.

In automated architecture, the network operates in a fully distributed way with the actors that autonomously undertake the appropriate actions upon receiving sensory information. In semi–automated architecture, sensors collect and transmit environmental information to a singleton network sink and the sink determines the proper actions that actors should do in response and allocates these actions (tasks) to appropriate actors. In cooperative architecture, sensors transmit sensing data to actuators in single hop or multiple hops. Actuators analyze data and may consult the sink(s) before taking any action. That is, actuators may use their peer-to-peer network to make decisions and take action, possibly informing the sink about the action taken, or could inform the sink and wait for further instructions from the sink [8].

III. Dynamic Mapping in Task Allocation

Dynamic mapping (matching and scheduling) heuristics for a class of independent tasks using heterogeneous distributed computing systems are studied. Two types of mapping heuristics are considered, immediate mode and batch mode heuristics.

There are five immediate mode heuristics [4]:

A. Minimum completion time (MCT)

It assigns each task to the machine those results in that task's earliest completion time. This causes some tasks to be assigned to machines that do not have the minimum execution time for them.

B. Minimum execution time(MET)

It assigns each task to the machine that performs that task's computation in the least amount of execution time(this heuristic is also known as LBA (limited best assignment) and UDA (user directed assignment).

C. Switching algorithm (SA)

It uses the MCT and MET heuristics in a cyclic fashion depending on the load distribution across the machines The MET heuristic can potentially create load imbalance across machines by assigning many more tasks to some machines than to others, whereas the MCT heuristic tries to balance the load by assigning tasks for earliest completion time. If the tasks are arriving in a random mix, it is possible to use the MET at the expense of load balance until a given threshold and then use the MCT to smooth the load across the machines. The purpose is to have a heuristic with the desirable properties of both the MCT and the MET.

D. K-percent best(KPB)

It considers only a subset of machines while mapping a task. The subset is formed by picking the m*(k/100) best machines based on the execution times for the task (m is number of machines), where 100/m<=k<=100. The task is assigned to a machine that provides the earliest completion time in the subset. If k=100, then the KPB heuristic is reduced to the MCT heuristic. If k=100/m, then the KPB heuristic is reduced to the MET heuristic. A "good" value of k maps a task to a machine only within a subset formed from computationally superior machines.

E. Opportunistic load balancing (OLB)

It assigns a task to the machine that becomes ready next, without considering the execution time of the task onto that machine. If multiple machines become ready at the same time, then one machine is arbitrarily chosen.

There are three batch mode heuristics:

A. Min-min

It considers the approximate execution and completion times of all tasks on each actor and only then assigns a task with lowest completion time to an actor with minimum execution time. it always gives priority to smaller tasks and allocates small tasks to faster actors [8].

B. Max-min

It selects the task with the maximum completion time and assigns it to the actor on which achieves minimum execution time. it always gives priority to larger tasks and allocates large tasks to faster actors [4].

C. Sufferage

It is based on the idea that better mappings can be generated by assigning a actor to a task that would "suffer" most in terms of expected completion time if that particular machine is not assigned to it [4].

IV. Task Allocation Algorithms in WSANs

Many task allocation algorithms for distributed systems have been presented so far with the aim of minimizing the total completion time of tasks. So there are some parameters that try to reduce the total task completion time and energy consuming of the system, such as deadlines of tasks, distances of actors to the event location, remaining energies of actors, and a balanced load on all actors.

In this paper, number of task allocation algorithms is described. They can use in two types of real time and non-real time system.

A. Task Allocation Algorithm in WSANs using Migration Algorithm

In [10] is presented an energy-efficient task assignment and distributed task migration algorithm for sensor networks. The node that has low energy level has responsibility to migrate to healtier node before it dies. To guarantee for the node to have time to migrate before it dies, a node starts searching neighbors for transferring task when it reaches a certain low energy level (e.g. 5% remaining energy). The node that reached low energy level initiates task migration process by sending REQUEST COST packets to its neighbors. Neighbors will respond with the cost if the tasks migrate to that node. After collecting all the cost values from neighbors, the node can determine which neighbor node is the best to transfer tasks. If the neighbors receive REQUEST COST packet, they calculate cost of transferring tasks to those nodes by requesting cost value to incoming and outgoing tasks of the originator task.

Because energy for communication is much expensive than computation and they are trade-offs each other, this paper tried to improve total energy consumption and maximum energy consumption in a node.

B. Task Allocation Algorithm in WSANs using graph transformation systems

In [6]-[7]-[11] is presented a two level task allocation mechanism. It first breaks end-to end periodic tasks into real-time jobs, and then uses appropriate algorithms for sensing tasks and acting tasks. It proposes a model for WSANs using graph transformation systems. Using this formalism it analyzes the correctness of algorithms. This algorithms guarantee that the tasks complete their activities before their deadlines expire.

The first algorithm is distributed and is run by cluster heads. According to this algorithm, cluster heads decide if periodic tasks that come from the sink are allocated to appropriate sensors or not. If a cluster head finds that there is a sensor node that is able to perform a requested sensing job, the cluster head allocates that job to that node; otherwise the cluster head rejects that job and informs the sink about the rejection. It details the operations that are run by each cluster head. These operations were modeled by different graph transformation rules.

After the sink receives environmental information from the cluster head, it detects the event region based on this information and sends appropriate acting task to the actors residing in that region.

The acting task allocation described in second algorithm is initiated when the sink receives environmental information from cluster heads. According to this information the acting relative task is performed. The first job of t his task is action that actors can perform it. Then, the sink uses the tuple space to retrieve information about status of actors such as type of acting job that actors can perform. The first actor that replies to the request and satisfies the above conditions performs the action and sends a signal to the sink. These two allocation mechanisms guarantee that sensor and actor nodes just accept tasks that can perform them in their deadlines.

C. Task Allocation Algorithm in WSANs using Queue system

In [9] is taken both energy awareness and reduction of actor tasks' times to completion in WSANs into account and is proposed a two-phase task allocation technique based on Queuing theory. The main goal in both proposed algorithms is to reduce task completion time, while load balancing is also considered in the second algorithm.BS and each actor have their own independent queue. Tasks are initially inserted in the queue of BS and later put in the queue of the actor chosen by BS to perform the task.

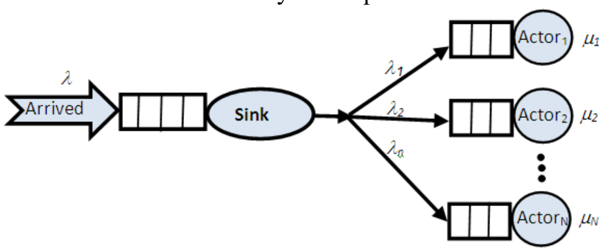

Fig.1. A queuing networks model of WSAN [9].

In the first phase, tasks are equally assigned to actors just to measure the capability of each actor to perform the assigned tasks. Tasks are then allocated to actors according to their measured capabilities in such a way to reduce the total completion times of all tasks in the network.

Following a Poisson process, BS inserts its configured tasks into its queue with λrate. As BS is usually more rapid than the actors with fewer faults (or ideally with no faults at all), it works like a gate whose output rate is the same as its input rateλ.

The sole objective of reducing tasks completion time and ignoring in Algorithm 1, may lead to overloading of some actors and idling of some other actors, resulting in the partitioning of the WSAN. To avoid this problem, a second algorithm is proposed that considers load balancing too. Then by using the average amount of λi that was derived in the first and second levels, the total tasks completion time is reduced and load balancing is taken care of too. In the other words, this algorithm makes a proper tradeoff between load balancing and reducing total tasks completion time and tries to optimize both load balancing and reducing total tasks completion time.

D. Task Allocation Algorithm in WSANs using Auction protocol

In [5] is proposed a distributed effective real time auction protocol. Sensors send their readings of events to their nearest actors using the location service. The task assignment problem is for each actor receiving sensory information to find a cost effective set of actors to perform a set of tasks in response to the event.

In this protocol is assumed that the frequency of event occurrences is not high and it initiates and auction whenever it wants to decide to which actors to assign a new task. There is just one event at the time requiring reply from actors. Actors that receive sensory information may not be the best actors to perform tasks if their current loads are high or their remaining energies are low; actor-actor coordination is required to select a cost effective set of actors. Tasks are non-preemptive, more than one task can be assigned to one actor, and each actor schedules its own assigned tasks using the Earliest Deadline First (EDF) policy. Actors participate in auction main and may invitetheir child actors too bid in auction. There is no overlapping area between actor nods and actors can search the wool network without any restriction in routings hops. A bid contains the energy of actor, the number of previously assigned tasks and its distance to event location.

E. Task Allocation in WSANs using Scate Algorithm

[8] have presented a new time- and energy-aware starvation-free algorithm called Scate for assigning tasks to actors while satisfying the scalability and distribution requirements of WSANs with semi-automated architecture. The proposed algorithm allows concurrent executions of any mix of small and large tasks and yet prevents probable starvation of tasks. To achieve this, it estimates the completion times of tasks on each available actor and then takes the remaining energies and the current workloads of these actors into account during task assignment to actors.

According to the assignment policy of min-min that is based on allocating small tasks to fast actors, because min-min assigns small tasks prior to large ones, it leads to higher makespan compared to max-min. In contrast, assigning the largest task to the fastest actor, as in max-min, provides a better chance for simultaneous execution of small tasks on other actors. Furthermore, max-min leads to a better balance of loads on actor that is critically desired in WSANs in order to prevent network partitioning.

Scate builds a matrix c' of ordered pairs whose first element Cij denotes the expected completion time of task Ti by actor Aj and the second element eij- remain denotes the expected remaining energy of actor Aj after performing the task Ti. To start with, all actors have their maximum energies and their workload is zero. Hence, in the first round, the largest task in the set of tasks requiring the highest execution time is allocated to an actor that can complete the task earliest than others. In the next rounds, the next largest and smallest tasks in the set of unallocated tasks (UTi) are assigned to the proper actors that can complete the tasks in the minimum time and also save maximum energy compared to other actors.

Scate tries to allocate a new task to an actor that is expected to have more energy after performing that task than other actors and at the same time can perform the new task earliest than other actors. Some applications may have higher priority for energy saving, some applications may have higher priority for shortening the completion times of tasks, while some applications may be neutral and consider both as being equally acceptable.

In fact, the very strength of Scate in comparison with other related algorithms lies in weighting the remaining energy over completion time ratio allowing changes in precedence of applications running concurrently in the network. So Scate can be adaptable to different applications and to different runs of the same application. Then Scate is scalable.

V. Conclusion

In WSANs, There is group of tasks assigned to actor nodes. So optimal allocation of tasks is highly desired. This paper has discussed concise survey some of algorithms for task allocation. Many task-scheduling algorithms for distributed systems have been presented so far with the aim of minimizing the total task completion time and energy consuming of the system. These algorithms allocate tasks to most suitable actors to the overall completion time.

References

[1] I. F. Akyildiz and I. H. Kasimoglu, "Wireless Sensor and Actor Networks: Research Challenges", Ad Hoc Networks, vol. 2, 2004, pp. 351-367

[2] R. Armstrong, D. Hensgen, and T. Kidd, "The Relative Performance of Various Mapping Algorithms is Independent of Sizable Variances in Run-Time Predictions," Proc. IEEE Int.

[3] T.D. Braun et al., "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems," Parallel Distrib. Comput., vol.61, 2001, pp. 810-837.
[4] M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen and R. F. Freund, "Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems", Journal of Parallel and Distributed Computing 59(1999), 107-121.
[5] H. Momeni, M. Sharifi and M. Okhovvat, "Distributed Assignment of Real-Time Tasks in Wireless Sensor Actor Networks", IEICE Electronics Express Journal , ISSN: 1349-2543, Vol. 8, No. 7, pp. 429-435, April 2011.
[6] H. Momeni, M. Sharifi and S. Sedighian, " A New Approach to Task Allocation in Wireless Sensor Actor Networks", First International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), Indore, India (2009).
[7] H. Momeni, V. Rafe, M. Sharifi and A. T. Rahmani, "A Graph Transformation-Based Approach to Task Allocation in Wireless Sensor Actor Networks", 21th IEEE International Conference on Software Engineering and Knowledge Engineering (SEKE09), USA(2009).12.
[8] Mohsen Sharifi and MortezaOkhovvat, "Scate: A Scalable Time and Energy Aware Actor Task Allocation Algorithm in Wireless Sensor and Actor Networks" ,ETRI Journal, ISSN (Print): 1226-6463, Vol. 34, No. 3, June 2012.
[9] M. Okhovvat, M. Sharifi, and H. Momeni, "Task Allocation to Actors in Wireless Sensor Actor Networks: An Energy and Time Aware Technique", Procedia Computer Science, vol. 3, 2011, pp.484-490.
[10] H. Park and M. B. Srivastava, "Energy-Efficient Task Assignment Framework for Wireless Sensor Networks", Center for Embedded Networking Sensing (CENS), Technical Report (2003).
[11] M.Sharifi, H.Momeni, V.Rafe, (2010). "Optimal real-time task allocation in wireless sensor actor networks". Australian Journal of Basic and Applied Sciences, Vol. 4, No. 8, pp. 3455-3466.

AUTHOR'S PROFILE

Sareh Beheshti
Department of Computer Engineering Arak Branch, Islamic Azad University, Arak, Iran.
Email: s-beheshti @ iau-arak.ac.ir

Mansoureh Ghiasabadi Frahani
Department of Computer Engineering Arak Branch, Islamic Azad University, Arak, Iran.
Email: m-ghias @ iau-arak.ac.ir

Mona Sharifnejad
Department of Computer Engineering Arak Branch, Islamic Azad University, Arak, Iran.
Email: m-sharifneiad@ iau-arak.ac.ir

Mohsen Sharifi
Iran University of Science and Technology.
Email: msharifi@iust.ac.ir