2014 IEEE 6th International Conference on Cloud Computing Technology and Science

Job Scheduling for Cloud Computing Integrated with Wireless Sensor Network

Chunsheng Zhu, Xiuhua Li, Victor C. M. Leung, Xiping Hu
Department of Electrical and Computer Engineering
The University of British Columbia
Vancouver, Canada
Email: {cszhu, lixiuhua, vleung, xipingh}@ece.ubc.ca

Laurence T. Yang
Department of Computer Science
St. Francis Xavier University
Antigonish, Canada
Email: ltyang@stfx.ca

***Abstract*—The powerful data storage and data processing abilities of cloud computing (CC) and the ubiquitous data gathering capability of wireless sensor network (WSN) complement each other in CC-WSN integration, which is attracting growing interest from both academia and industry. However, job scheduling for CC integrated with WSN is a critical and unexplored topic. To fill this gap, this paper first analyzes the characteristics of job scheduling with respect to CC-WSN integration and then studies two traditional and popular job scheduling algorithms (i.e., Min-Min and Max-Min). Further, two novel job scheduling algorithms, namely priority-based two phase Min-Min (PTMM) and priority-based two phase Max-Min (PTAM), are proposed for CC integrated with WSN. Extensive experimental results show that PTMM and PTAM achieve shorter expected completion time than Min-Min and Max-Min, for CC integrated with WSN.**

***Keywords*-Cloud computing; wireless sensor network; job scheduling; Min-Min; Max-Min; priority; expected execution time**

I. Introduction

Cloud computing (CC) is a novel computing model to enable on demand and scalable access to a shared pool of configurable computing resources (e.g., processors, storage, networks, applications, services) [1], while a wireless sensor network (WSN) interconnects spatially distributed autonomous sensors that cooperatively collect physical or environmental data (e.g., temperature, sound, vibration, pressure, motion) [2]. Incorporating the powerful data storage and data processing abilities of CC as well as the ubiquitous data gathering capability of WSN, CC-WSN integration is a new topic attracting increasing attention from both academia and industry [3] [4] [5] [6]. Figure 1 shows a CC-WSN integration scenario in which the powerful cloud receives and processes various sensory data (e.g., traffic, house monitoring, temperature, humidity) collected by different types of sensors (e.g., static sensor, mobile sensor and video sensor) in the WSN and transmit these data on demand in response to requests for sensory information from users. Thus the user can obtain desirable sensory information offered by the cloud anytime and anywhere with network connectivity.

Job scheduling [7] [8] [9] is one of the major and critical activities performed in CC. The goal of job scheduling is to spread the load on processors and maximize their utilization while minimizing the total task execution time. The main task of job scheduling is to schedule jobs onto the adaptable resources in time, which requires finding out a proper sequence in which jobs can be executed under transaction logic constraints. The main advantage of job scheduling is to achieve high computing performance and system throughput. However, to the best of our knowledge, job scheduling for CC integrated with WSN is unexplored.

To fill this gap, this paper first analyzes the characteristics of job scheduling in CC-WSN integration and then investigates two traditional and popular job scheduling algorithms (i.e., Min-Min [10] [11] [12] [13] and Max-Min [14] [15] [16] [17]). Further, this paper proposes two novel job scheduling algorithms, namely priority-based two phase Min-Min (PTMM) and priority-based two phase Max-Min (PTAM) scheduling, for CC integrated with WSN. In both PTMM and PTAM, cloud tasks related to WSN are executed first in phase 1, followed by the execution of other ordinary cloud tasks in phase 2. We present extensive experimental results for these four job scheduling algorithms in CC-WSN integration, and show that PTMM and PTAM achieve shorter expected completion time (ECT) than Min-Min and Max-Min.

The main contributions of this paper are summarized as follows.

- This paper is the first work that discusses and analyzes job scheduling for CC integrated with WSN. This clearly demonstrates the novelty of our work compared with other works on job scheduling.
- This paper further explores the ECT performance of Min-Min and Max-Min in CC-WSN integration and proposes two new job scheduling algorithms, PTMM and PTAM, for CC integrated with WSN. Experimental results are presented to show that these novel algorithms effectively improve the ECT of Min-Min and Max-Min.

For the rest of this paper, Section II reviews the related work and Section III analyzes the characteristics of job scheduling in CC-WSN integration. Min-Min and Max-Min job scheduling algorithms are studied in Section IV. The proposed PTMM and PTAM job scheduling algorithms for

978-1-4799-4093-6/14 $31.00 © 2014 IEEE
DOI 10.1109/CloudCom.2014.106

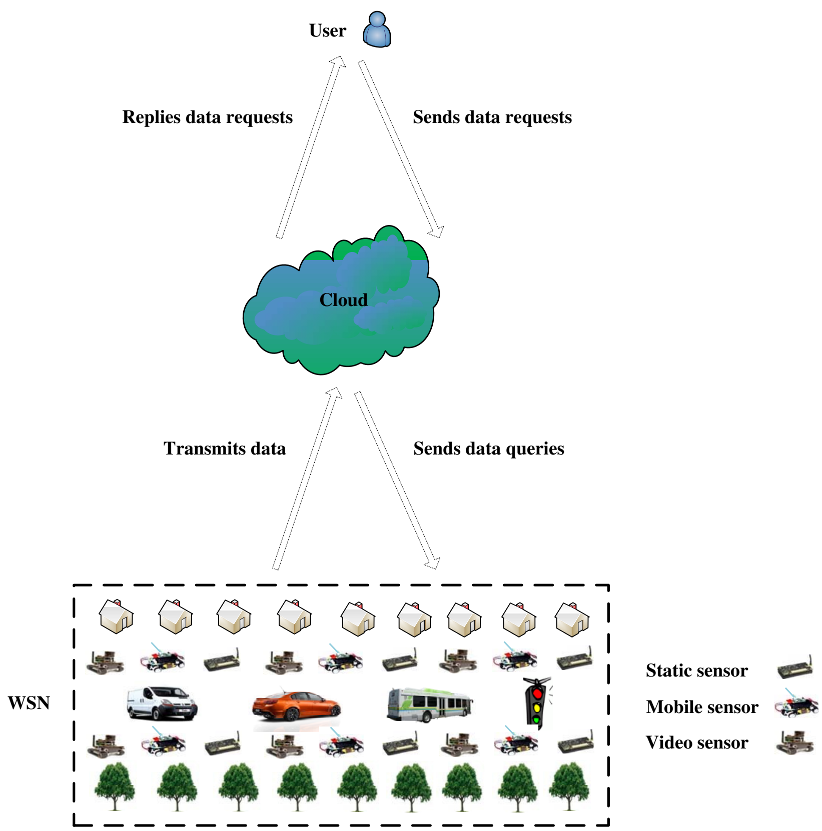


Figure 1. Example of CC-WSN integration

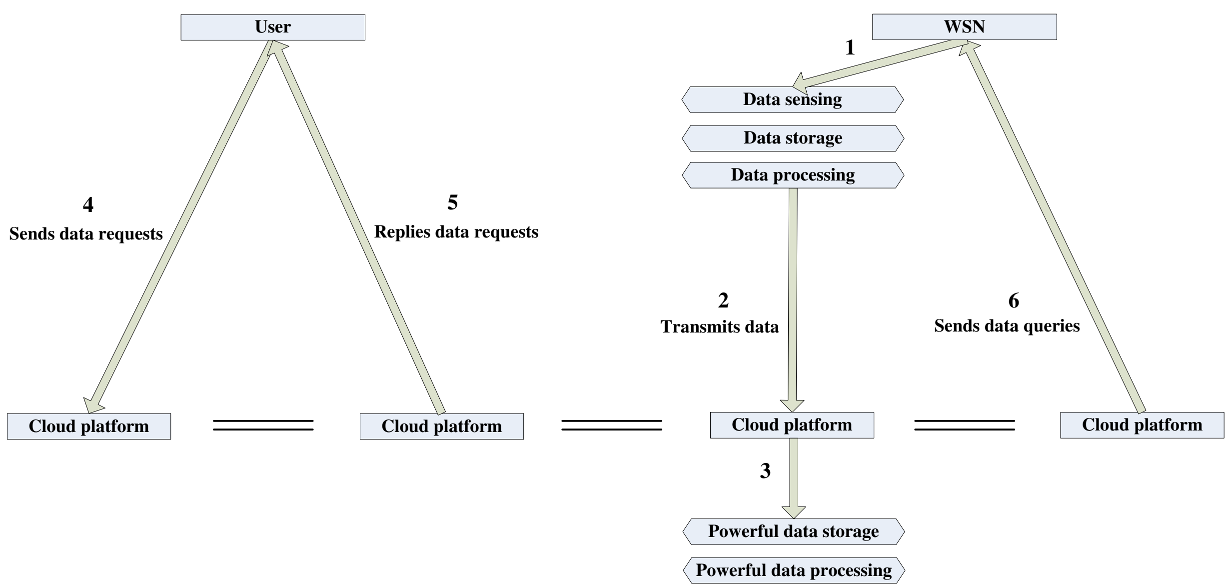


Figure 2. Flowchart of CC-WSN integration

CC integrated with WSN are presented in Section V. Section VI presents the evaluations of these four job scheduling algorithms. Section VII concludes the paper.

II. Related Work

There are a lot of works about job scheduling algorithms for CC, which can be categorized into the following two main groups [7] [18]: 1) Batch mode heuristic scheduling algorithms (BMHA) and 2) Online mode heuristic algorithms (OMHA).

A. BMHA

For BMHA, jobs are queued and collected into a set when they arrive in the system. The scheduling algorithm starts after a fixed period of time. Main examples of BMHA algorithms include: First Come First Served (FCFS) [19], Round Robin (RR) [20], Min-Min [11] and Max-Min [15]. Particularly, for FCFS, jobs that come first in the queue are executed first followed with jobs that arrive later. With respect to RR, the queue is in the shape of a ring and there is a fixed time quantum. If the execution of a job cannot be finished in one quantum, the job will be placed at the end of queue and wait for the next round to be executed. In terms of Min-Min and Max-Min, they are two traditional job scheduling algorithms. In short, Min-Min executes the task with the smallest size in the queue first, while Max-Min executes the task with the largest size in the queue first.

B. OMHA

With respect to OMHA, jobs are immediately scheduled when they arrive in the system. A very suitable example of OMHA is the Most Fit Task scheduling algorithm (MFTF) [21]. Specifically, it aims at executing the task fitting best in the queue. In other words, it assigns tasks to resources according to a fitness value and the task with the highest fitness value is executed first.

From our review of the state of art, we do not find any existing work on job scheduling for CC integrated with WSN. Our work is the first that explores and proposes job scheduling for CC integrated with WSN.

III. Characteristics of Job Scheduling in CC-WSN Integration

Figure 2 shows a flow chart of the information exchange in CC-WSN integration. Specifically, there are three main processes shown as follows in the flowchart of CC-WSN integration.

- 1) WSN senses, stores and processes the sensory data, and transmits these sensory data to the cloud.
- 2) Cloud stores and processes these sensory data next.
- 3) Cloud further transmits the processed sensory information to requesting users on demand.

From this flowchart, we can observe that the WSN functions as the data source, while the user is the data requester for CC-WSN integration. Motivated by this, we can further determine that *cloud tasks related to WSN (e.g., sensory data storage and processing by cloud) should be executed first, followed with the execution of other ordinary cloud tasks (e.g., historical data storage and processing or user behavior data storage and processing by cloud)*. Otherwise, there will be substantial execution time delays for users to obtain their required sensory information in CC-WSN integration.

IV. Min-Min and Max-Min

To introduce Min-Min [10] [11] [12] [13] and Max-Min [14] [15] [16] [17] job scheduling, we assume that we have a set of n tasks $(T_1, T_2, T_3, \ldots, T_n)$ that need to be scheduled onto m available resources $(R_1, R_2, R_3, \ldots, R_m)$. The ECT for task T_i $(1 \leq i \leq n)$ on resource R_j $(1 \leq j \leq m)$ is denoted as CT_{ij} which is calculated as in $CT_{ij} = ET_{ij} + rt_j$. In this equation, ET_{ij} represents the execution time of task T_i on resource R_j and rt_j represents the ready time of resource R_j.

A. Min-Min

Min-Min is a very simple and classical job scheduling algorithm. The main idea and pseudocodes of the Min-Min algorithm are shown as follows. Specifically, it starts with a set T of all unmapped tasks. Then the resource R_j that has the minimum completion time for each task is found. Next, the task T_k with the minimum size is selected and assigned to the corresponding resource R_j (hence the name Min-Min). Last, the task T_k is removed from set T and the same procedure is repeated by Min-Min until all tasks are assigned (i.e., set T is empty).

B. Max-Min

Max-Min is also a simple and popular job scheduling algorithm. It is very similar with the Min-Min algorithm but it focuses on balancing the load as well. Particularly, as the following pseudocodes show, it also starts with a set T of all unmapped tasks. Then the resource R_j which has the minimum completion time for each task is found. Next, the task T_k with the maximum size is selected and assigned to the corresponding resource R_j (hence the name Max-Min). Last, the task T_k is removed from set T and the same procedure is repeated by Max-Min until all the tasks are assigned (i.e., set T is empty).

From the above introductions about Min-Min and Max-Min, we can see that Min-Min and Max-Min schedule jobs based on the task size. In other words, *Min-Min and Max-Min do not consider that WSN related cloud tasks should be executed first for CC-WSN integration*. Starting from this point, we propose PTMM (priority-based two phase Min-Min) and PTAM (priority-based two phase Max-Min).

V. PTMM and PTAM

In this section, we introduce the proposed PTMM and PTAM for CC integrated with WSN.

Pseudocodes of Min-Min algorithm

Step 1: For all submitted tasks T_i in tasks set T
Step 2: For all resources R_j
Step 3: $CT_{ij} = ET_{ij} + rt_j$;
Step 4: End For;
Step 5: End For;
Step 6: Do while tasks set is not empty
Step 7: Find task T_k that costs the minimum ET_{ij}
Step 8: Assign T_k to resource R_j which offers minimum CT_{ij}
Step 9: Remove T_k from the tasks set
Step 10: Update ready time rt_j for selected R_j
Step 11: Update CT_{ij} for all T_i
Step 12: End Do

Pseudocodes of Max-Min algorithm

Step 1: For all submitted tasks T_i in tasks set T
Step 2: For all resources R_j
Step 3: $CT_{ij} = ET_{ij} + rt_j$;
Step 4: End For;
Step 5: End For;
Step 6: Do while tasks set is not empty
Step 7: Find task T_k that consumes maximum ET_{ij}
Step 8: Assign T_k to resource R_j which gives minimum CT_{ij}
Step 9: Remove T_k from the tasks set
Step 10: Update ready time rt_j for selected R_j
Step 11: Update CT_{ij} for all T_i
Step 12: End Do

Pseudocodes of proposed PTMM algorithm

Step 0: Divide tasks into two groups according to the task properties: WSN related cloud tasks group G_1 and ordinary cloud tasks group G_2

Phase 1:
Step 1: For all submitted tasks in group G_1
Step 2: For all resources R_j
Step 3: $CT_{ij} = ET_{ij} + rt_j$;
Step 4: End For;
Step 5: End For;
Step 6: Do while G_1 is not empty
Step 7: Find task T_k that costs the minimum ET_{ij}
Step 8: Assign T_k to resource R_j which offers minimum CT_{ij}
Step 9: Remove T_k from the tasks set
Step 10: Update ready time rt_j for selected R_j
Step 11: Update CT_{ij} for all T_i
Step 12: End Do

Phase 2:
Step 1: For all submitted tasks in group G_2
Step 2: For all resources R_j
Step 3: $CT_{ij} = ET_{ij} + rt_j$;
Step 4: End For;
Step 5: End For;
Step 6: Do while G_2 is not empty
Step 7: Find task T_k that costs the minimum ET_{ij}
Step 8: Assign T_k to resource R_j which offers minimum CT_{ij}
Step 9: Remove T_k from the tasks set
Step 10: Update ready time rt_j for selected R_j
Step 11: Update CT_{ij} for all T_i
Step 12: End Do

Pseudocodes of proposed PTAM algorithm

Step 0: Divide tasks into two groups according to the task properties: WSN related cloud tasks group G_1 and ordinary cloud tasks group G_2

Phase 1:
Step 1: For all submitted tasks in group G_1
Step 2: For all resources R_j
Step 3: $CT_{ij} = ET_{ij} + rt_j$;
Step 4: End For;
Step 5: End For;
Step 6: Do while G_1 is not empty
Step 7: Find task T_k that consumes the maximum ET_{ij}
Step 8: Assign T_k to resource R_j which offers minimum CT_{ij}
Step 9: Remove T_k from the tasks set
Step 10: Update ready time rt_j for selected R_j
Step 11: Update CT_{ij} for all T_i
Step 12: End Do

Phase 2:
Step 1: For all submitted tasks in group G_2
Step 2: For all resources R_j
Step 3: $CT_{ij} = ET_{ij} + rt_j$;
Step 4: End For;
Step 5: End For;
Step 6: Do while G_2 is not empty
Step 7: Find task T_k that consumes the maximum ET_{ij}
Step 8: Assign T_k to resource R_j which offers minimum CT_{ij}
Step 9: Remove T_k from the tasks set
Step 10: Update ready time rt_j for selected R_j
Step 11: Update CT_{ij} for all T_i
Step 12: End Do

A. PTMM

PTMM is a priority-based two phase Min-Min algorithm. Specifically, as the following pseudocodes show, the tasks are divided into two groups (i.e., WSN related cloud tasks group G_1 and ordinary cloud tasks group G_2), according to the task properties first. For all submitted tasks in group G_1 with higher priority, the Min-Min algorithm is executed first in phase 1. After that, for all submitted tasks in group G_2 with lower priority, the Min-Min algorithm is executed again in phase 2.

B. PTAM

PTMM is a priority-based two phase Max-Min algorithm. As the following pseudocodes show, for PTMM, the tasks are also divided into two groups according to the task properties: WSN related cloud tasks group G_1 and ordinary cloud tasks group G_2. In phase 1, for all submitted tasks in group G_1 with higher priority, the Max-Min algorithm is executed. With that, for all submitted tasks in group G_2 with lower priority, the Max-Min algorithm is also executed in phase 2.

VI. Evaluations

In this section, we evaluate the performance of Min-Min, Max-Min and PTMM as well as PTAM, in terms of ECT for CC integrated with WSN.

A. Evaluation setup

The evaluation is performed with Matlab. Particularly, the program in Matlab includes a resources-generator and a task-generator [12]. The resource-generator simulates the cloud environment and generates cloud resources in the specified range of processing speed. The task-generator generates random tasks in the specified range of task size.

1) Min-Min & Max-Min: For Min-Min and Max-Min, the number of resources is chosen to be 5 and number of tasks is chosen to be 10. The processing speeds of resources are set to be random values from 1 to 10 and the sizes of tasks

Table I
PROCESSING SPEEDS OF RESOURCES AND SIZES OF TASKS FOR MIN-MIN AND MAX-MIN

Resource	Processing Speed (MB/Sec)	Task	Task Size (MB)
R_1	5	T_1	58
R_2	7	T_2	37
R_3	4	T_3	59
R_4	10	T_4	41
R_5	8	T_5	65
		T_6	60
		T_7	42
		T_8	45
		T_9	7
		T_{10}	83

Table II
PROCESSING SPEEDS OF RESOURCES AND SIZES OF TASKS IN SCENARIO 1 FOR PTMM AND PTAM

Resource	Processing Speed (MB/Sec)	Task	Task Size (MB)	Task Group
R_1	5	T_1	58	1
R_2	7	T_2	37	1
R_3	4	T_3	59	2
R_4	10	T_4	41	2
R_5	8	T_5	65	2
		T_6	60	2
		T_7	42	2
		T_8	45	2
		T_9	7	2
		T_{10}	83	2

Table III
PROCESSING SPEEDS OF RESOURCES AND SIZES OF TASKS IN SCENARIO 2 FOR PTMM AND PTAM

Resource	Processing Speed (MB/Sec)	Task	Task Size (MB)	Task Group
R_1	5	T_1	58	1
R_2	7	T_2	37	1
R_3	4	T_3	59	1
R_4	10	T_4	41	1
R_5	8	T_5	65	1
		T_6	60	1
		T_7	42	1
		T_8	45	1
		T_9	7	2
		T_{10}	83	2

Table IV
EXAMPLE OF PROCESSING SPEEDS OF RESOURCES AND SIZES OF TASKS IN SCENARIO 3 FOR PTMM AND PTAM

Resource	Processing Speed (MB/Sec)	Task	Task Size (MB)	Task Group
R_1	9	T_1	30	1
R_2	1	T_2	73	1
R_3	7	T_3	100	1
R_4	2	T_4	18	1
R_5	10	T_5	22	1
R_6	6	T_6	55	1
R_7	8	T_7	81	2
R_8	3	T_8	46	2
R_9	5	T_9	65	2
R_{10}	4	T_{10}	90	2
		...	...	...
		T_{100}	5	2

are set to be random values from 1 to 100. The specific processing speeds of the 5 resources and sizes of the 10 tasks utilized are shown in Table I.

2) PTMM & PTAM: With respect to PTMM and PTAM, the evaluation is performed in the following three scenarios [12].

- Scenario 1: In this scenario, there is a low proportion (i.e., 20%) of cloud tasks that are WSN related (i.e., group 1). The number of resources is also chosen to be 5 and number of tasks is chosen to be 10. The processing speeds of resources are also set to be random values from 1 to 10 and the sizes of tasks are set to be random values from 1 to 100. The specific processing speeds of the 5 resources and sizes of the 10 tasks utilized are the same as that utilized for Min-Min and Max-Min, except that the tasks belong to two different groups as shown in Table II.
- Scenario 2: For scenario 2, the proportion of WSN related cloud tasks (i.e., group 1) is high (i.e., 80%). The numbers of resources and tasks are also chosen to be 5 and 10, respectively. The processing speeds of resources are also set to be random values from 1 to 10 and the sizes of tasks are set to be random values from 1 to 100. The specific processing speeds of the 5 resources and sizes of the 10 tasks utilized are the same as that utilized for Min-Min and Max-Min, except that the tasks belong to two different groups as presented in Table III.
- Scenario 3: About scenario 3, the proportion of WSN related cloud tasks (i.e., group 1) is different and random (i.e., 0%, 10%, ..., 100%). The numbers of resources and tasks are chosen to be 10 and 100, respectively. Specifically, the processing speeds of 10 resources obey a random distribution between 1 to 10 and the sizes of 100 tasks obey a random distribution between 1 and 100. One example of the processing speeds of the 10 resources and the sizes of the 100 tasks are presented in Table IV. In scenario 3, we analyze the mean ECT which is the total ECT divided by the number of tasks.

B. Evaluation results

The evaluation results for Min-Min, Max-Min, PTMM and PTAM are shown from Figure 3 to Figure 6. Based on these figures, we analyze the ECT of these four algorithms.

1) Min-Min vs. PTMM: Comparing Figure 4(a) and Figure 3(a), we can observe that in scenario 1 in which task T_1 and task T_2 belong to group 1, the ECT of T_1 is around 7.2 seconds and the ECT for T_2 is around 3.5 seconds in PTMM, while the ECT for T_1 is around 11.5 seconds and the ECT for T_2 is around 4.5 seconds in Min-Min.

In addition, from Figure 5(a) and Figure 3(a), we can observe that in scenario 2 in which task T_9 and task T_{10} belong to group 2, the ECT of T_9 is near 1.7 seconds and the ECT for T_{10} is near 23.5 seconds in PTMM, while the ECT for T_9 is near 0.7 seconds and the ECT for T_{10} is near 20.5 seconds in Min-Min.

2) Max-Min vs. PTAM: Comparing Figure 4(b) and Figure 3(b), we can see that in scenario 1 where task T_1 and task T_2 belong to group 1, the ECT of T_1 is around 5.8 seconds and the ECT for T_2 is around 4.6 seconds in PTAM, while the ECT for T_1 is around 14 seconds and the ECT for T_2 is around 18 seconds in Max-Min.

Furthermore, based on Figure 5(b) and Figure 3(b), we can observe that in scenario 2 in which task T_9 and task T_{10} belong to group 2, the ECT of T_9 is near 12.2 seconds and the ECT for T_{10} is near 19.3 seconds in PTAM, while the ECT for T_9 is near 13 seconds and the ECT for T_{10} is near 8.3 seconds in Max-Min.

3) PTMM & PTAM: From Figure 6(a), we can see that with the increased ratio of WSN related cloud tasks in scenario 3, the mean ECT for PTMM is generally increased until the proportion of WSN related cloud tasks is 60%. Then the mean ECT for PTMM is decreased.

In contrast, from Figure 6(b), it is demonstrated that the increased ratio of WSN related cloud tasks decreases the mean ECT for PTAM, until the proportion of WSN related cloud tasks is increased to 40% in scenario 3. After that, the mean ECT for PTAM is increased.

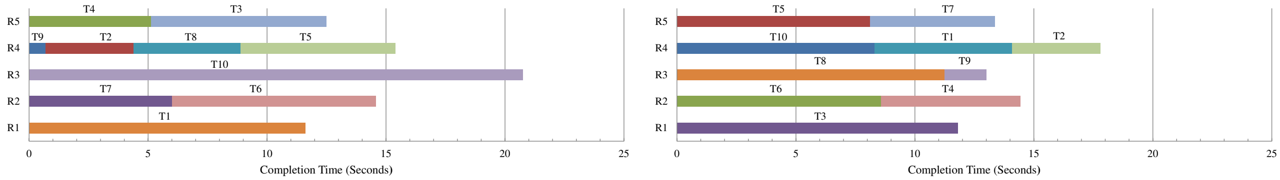
Based on the above analysis about ECT of Min-Min, Max-Min, PTMM as well as PTAM, we can make the following observations.

- ECT of WSN related cloud tasks which belong to group 1 with PTMM and PTAM are always much shorter than that with Min-Min and Max-Min in scenario 1.
- ECT of ordinary cloud tasks that belong to group 2 with PTMM and PTAM nearly are always longer than that with Min-Min and Max-Min in scenario 2. In other words, ECT of WSN related cloud tasks which belong to group 1 with PTMM and PTAM are generally shorter than that with Min-Min and Max-Min in scenario 2.
- Increased ratio of WSN related cloud tasks do not always increase the mean ECT for PTMM or decrease the mean ECT for PTAM in scenario 3.

In short, PTMM and PTAM outperform Min-Min and Max-Min, in terms of ECT of WSN related cloud tasks. Specifically, we obtain that *PTMM and PTAM achieve shorter ECT than Min-Min and Max-Min, for CC integrated with WSN.*

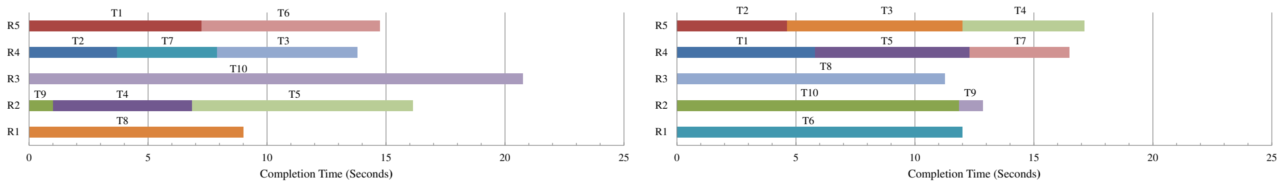
VII. Conclusion

This paper focuses on job scheduling algorithms for CC integrated with WSN, which is an important but unexplored topic for CC-WSN integration. Particularly, filling this gap, the characteristics of job scheduling with respect to CC-WSN integration have been analyzed. Further, the performance of two traditional and popular job scheduling algorithms (i.e., Min-Min and Max-Min) has been studied. We



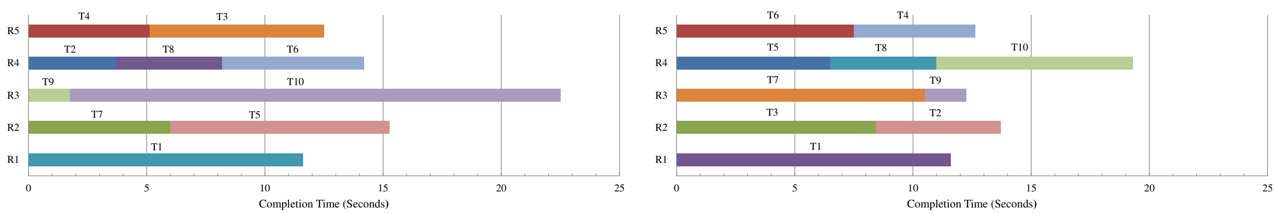
(a) (b)

Figure 3. ECT of Min-Min (a) and ECT of Max-Min (b).



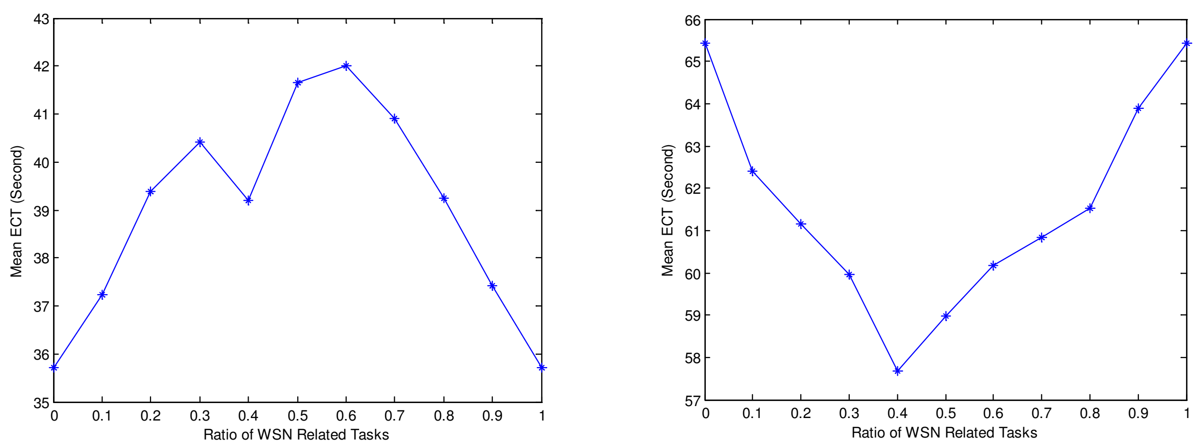
(a) (b)

Figure 4. ECT of PTMM in Scenario 1 (a) and ECT of PTAM in Scenario 1 (b).



(a) (b)

Figure 5. ECT of PTMM in Scenario 2 (a) and ECT of PTAM in Scenario 2 (b).



(a) (b)

Figure 6. Mean ECT change of PTMM in Scenario 3 (a) and mean ECT change of PTAM in Scenario 3 (b).

have proposed the novel PTMM and PTAM job scheduling algorithms for CC integrated with WSN. In these algorithms, the WSN related cloud tasks are executed first in phase 1 and ordinary cloud tasks are executed later in phase 2. Extensive experimental results that evaluate these four job scheduling algorithms have been presented to show that PTMM and PTAM achieve shorter ECT than Min-Min and Max-Min, for CC integrated with WSN.

ACKNOWLEDGEMENT

This work was supported by a Four Year Doctoral Fellowship from The University of British Columbia and in part by funding from the Natural Sciences and Engineering Research Council of Canada, the ICICS/TELUS People & Planet Friendly Home Initiative at The University of British Columbia, TELUS and other industry partners.

REFERENCES

[1] C. Zhu, V. C. M. Leung, X. Hu, L. Shu, and L. T. Yang, "A review of key issues that concern the feasibility of mobile cloud computing," in *Proc. IEEE International Conference on Cyber, Physical and Social Computing (CPSCom)*, 2013, pp. 769–776.

[2] C. Zhu, L. Shu, T. Hara, L. Wang, S. Nishio, and L. T. Yang, "A survey on communication and data management issues in mobile sensor networks," *Wireless Communications and Mobile Computing*, vol. 14, no. 1, pp. 19–36, Jan. 2014.

[3] C. Zhu, V. C. M. Leung, H. Wang, W. Chen, and X. Liu, "Providing desirable data to users when integrating wireless sensor networks with mobile cloud," in *Proc. IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2013, pp. 607–614.

[4] C. Zhu, H. Nicanfar, V. C. M. Leung, W. Li, and L. T. Yang, "A trust and reputation management system for cloud and sensor networks integration," in *Proc. IEEE International Conference on Communications (ICC)*, 2014, pp. 557–562.

[5] C. Zhu, H. Wang, X. Liu, L. Shu, L. T. Yang, and V. C. M. Leung, "A novel sensory data processing framework to integrate sensor networks with mobile cloud," *IEEE Systems Journal*, 2014.

[6] C. Zhu, V. C. M. Leung, L. T. Yang, and L. Shu, "Collaborative location-based sleep scheduling for wireless sensor networks integrated with mobile cloud computing," *IEEE Transactions on Computers*, 2014.

[7] P. Salot, "A survey of various scheduling algorithm in cloud computing environment," *International Journal of Research in Engineering and Technology*, vol. 2, no. 2, pp. 131–135, 2013.

[8] B. Yang, X. Xu, F. Tan, and D. H. Park, "An utility-based job scheduling algorithm for cloud computing considering reliability factor," in *Proc. International Conference on Cloud and Service Computing (CSC)*, 2011, pp. 95–102.

[9] M. D. Assuncao, M. A. S. Netto, F. Koch, and S. Bianchi, "Context-aware job scheduling for cloud computing environments," in *Proc. IEEE 5th International Conference on Utility and Cloud Computing (UCC)*, 2012, pp. 255–262.

[10] D. D. H. Miriam and K. S. Easwarakumar, "A double min min algorithm for task metascheduler on hypercubic p2p grid systems," *International Journal of Computer Science Issues*, vol. 7, no. 5, pp. 8–18, Jul. 2010.

[11] G. Liu, J. Li, and J. Xu, "An improved min-min algorithm in cloud computing," in *Proc. International Conference of Modern Computer Science and Applications (MCSA)*, 2012, pp. 47–52.

[12] H. Chen, F. Wang, N. Helian, and G. Akanmu, "User-priority guided min-min scheduling algorithm for load balancing in cloud computing," in *Proc. 2013 National Conference on Parallel Computing Technologies (PARCOMPTECH)*, 2013, pp. 1–8.

[13] B. Wang, "Qos min-min scheduling strategy," *International Journal of Distributed Sensor Networks*, vol. 5, no. 1, pp. 1–13, Jan. 2009.

[14] S. Devipriya and C. Ramesh, "Improved max-min heuristic model for task scheduling in cloud," in *Proc. International Conference on Green Computing, Communication and Conservation of Energy (ICGCE)*, 2013, pp. 883–888.

[15] U. Bhoi and P. N. Ramanuj, "Enhanced max-min task scheduling algorithm in cloud computing," *International Journal of Application or Innovation in Engineering & Management*, vol. 2, no. 4, pp. 259–264, Apr. 2013.

[16] X. Li, Y. Mao, X. Xiao, and Y. Zhuang, "An improved max-min task-scheduling algorithm for elastic cloud," in *Proc. International Symposium on Computer, Consumer and Control (IS3C)*, 2014, pp. 340–343.

[17] B. Santhosh and D. H. Manjaiah, "An improved task scheduling algorithm based on max-min for cloud computing," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 2, pp. 84–88, May 2014.

[18] Y.-H. Lee, S. Leu, and R.-S. Chang, "Improving job scheduling algorithms in a grid environment," *Future Generation Computer Systems*, vol. 27, no. 8, pp. 991–998, Oct. 2011.

[19] U. Schwiegelshohn and R. Yahyapour, "Analysis of first-come-first-serve parallel job scheduling," in *Proc. 9th annual ACM-SIAM symposium on Discrete algorithms (SODA)*, 1998, pp. 629–638.

[20] S. Mohapatra, S. Mohanty, and K. S. Rekha, "Analysis of different variants in round robin algorithms for load balancing in cloud computing," *International Journal of Computer Applications*, vol. 69, no. 22, pp. 17–21, May 2013.

[21] S.-D. Wang, I.-T. Hsu, and Z.-Y. Huang, "Dynamic scheduling methods for computational grid environments," in *Proc. 11th International Conference on Parallel and Distributed Systems*, 2005, pp. 22–28.