Future Generation Computer Systems 32 (2014) 253–259

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Effective real-time scheduling algorithm for cyber physical systems society

Sanghyuk Park [a], Jai-Hoon Kim [b,*], Geoffrey Fox [c]

[a] Department of Network Centric Warfare Engineering, Ajou University, South Korea
[b] Graduate School of Information and Communication, Ajou University, South Korea
[c] Pervasive Technology Institute, Indiana University, Bloomington, IN, USA

HIGHLIGHTS

- Computers and physical systems are tightly coupled in cyber physical society.
- Conventional systems only consider cyber space.
- CPS should also consider physical, socio and mental space.
- Proposed scheduling algorithm considering physical factors.
- Efficiency of algorithm is verified by mathematical analysis and simulation.

ARTICLE INFO

Article history:
Received 4 July 2012
Received in revised form
29 September 2013
Accepted 4 October 2013
Available online 12 October 2013

Keywords:
CPS (Cyber Physical System)
Real-time scheduling algorithm
LST (Least Slack Time First)
ELST (Effective Least Slack Time First)
H-ELST (Heuristic Effective Least Slack Time First)

ABSTRACT

CPS (Cyber Physical Systems) tightly couple their cyber factor and physical factor in distributed computing or Grids environments to provide real-time services such as avionics, transportation, manufacturing processes, energy, healthcare, etc. We need to consider not only the cyber space (CPU, network, storage systems, etc.) and the physical space (location, migration, etc.) but also the socio space and mental space for the precise analysis and useful services. In this paper, real-time scheduling algorithms, namely ELST (Effective Least Slack Time First) and H-ELST (Heuristic-Effective Least Slack Time First), are presented for CPS, where servicing node needs to move to serviced node for real-time services. We measure the real-time performance in terms of deadline meet ratio by mathematical analysis and simulations. The results show that our algorithms reduce a deadline miss ratio approximately up to 50% and 20% compared to the conventional real-time scheduling algorithm, FIFO (First In First Out) and LST (Least Slack Time First), respectively.

1. Introduction

Timing issues are critical in real-time systems such as robot control [1,2], NCO (Network Centric Operations) systems [3–6], flight control, on-line multimedia systems [7], and real-time stock trading system, etc. [8–10]. Many real-time scheduling algorithms such as RM (rate monotonic) [11,12], EDF (earliest deadline first) [12–14], and LST (least slack time first) [12,14] deal with resource (CPU and network bandwidth) scheduling to maximize real-time performance (e.g., deadline meet ratio) [7,14]. As CPS (cyber physical system [15–18] and cyber physical society [19–21]) such as avionics, transportation, manufacturing processes, energy, healthcare, in which computers and physical systems (also, society and mental) are tightly coupled and timing is critical, is fast growing, real-time scheduling for CPS becomes the new research issues in the real-time systems [22,23].

In other aspects, as real-time applications become complex and relevant tasks and resources are widely distributed, we have to study the real-time scheduling in distributed computing infrastructures and Grids. For examples, in Grids infrastructures (e.g., EGI (European Grid Infrastructure) [24], SEE-GRID (South Eastern European Grid-enabled e-Infrastructure) [25], and EELA (E-science grid facility for Europe and Latin America) [26]) many tasks concurrently request various types of distributed resources. Middleware has to coordinate the resource allocation to provide services and guarantee a SLA. In these distributed environments, the real-time scheduling must consider transfer delays as task and data migrations among nodes having computing resources are common. Red Hat Enterprise MRG (Messaging, Real time, and Grid) Real time [27]

* Corresponding author. Tel.: +82 31 219 2546; fax: +82 31 219 1614.
E-mail address: jaikim@ajou.ac.kr (J.-H. Kim).

0167-739X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.future.2013.10.003

Table 1
Real-time scheduling for CPS.

	Conventional real-time scheduling	Real-time scheduling for CPS
Scheduling resources	CPU, BW, memory, I/O	Servicing node
Scheduling environment	Cyber environment	Cyber and physical environment
Scheduling parameters	Cyber factors (Period, execution time, release time, deadline, etc.)	Cyber factors and physical factors (period, execution time, release time, deadline, migration delay time, etc.)
Migration	No migration is required for CPU and Job. * We consider CPU is serving node and Job is serviced node	Migration is required for CPU and Job.
Well-known Scheduling algorithm	RM, EDF, LST, etc.	None
Spatial issues	Do not consider spatial issues	Physical migration delay time (between servicing node to serviced node)
Considering issues	Execution time, release time, deadline, laxity	Execution time, release time, effective deadline (deadline−moving time), effective laxity (laxity−moving delay time)

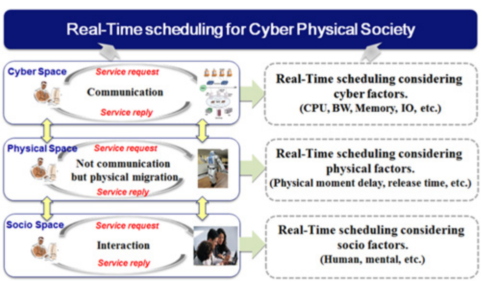

Fig. 1. Real-time scheduling for CPS.

provides a high level of predictability for consistent low-latency response times to meet the requirement of time-sensitive workloads. Many large-scale distributed applications require real-time responses to meet soft deadlines. Ref. [28] design and implement the real-time volunteer computing platform called RT-BOINC to schedule the real-time task and execute on the volunteer resources.

Many real-time scheduling algorithms have been proposed and widely used [11,12,14]. However, in cyber physical systems society, we need to consider not only cyber space (CPU, network, storage systems, etc.) and physical space (location, migration, etc.) but also socio space and mental space [19–21]. Fig. 1 shows the real-time scheduling model for cyber physical systems society [19–21]. Effective release time and deadline of real-time tasks may be different depending on the location and physical migration delay time of nodes participating in CPS. Real-time scheduling algorithms have to be modified to include spatial factors. Conventional cyber real-time system schedules CPU or network bandwidth. However, in real-time scheduling for CPS, location is matter. Location of nodes in CPS affects the effective release time and deadline.

In this paper, we propose new real-time scheduling algorithms for CPS, where the servicing node needs to move to serviced node for real-time services. If we assume, for an example, there are many scattered customers requesting real-time services but only one servicing staff exists in the area, real-time scheduling is necessary to maximize the performance (e.g., deadline meet ratio). In this case, the conventional real-time scheduling algorithm is not proper because the real-time scheduling does not consider physical factors (e.g., locations of customer of servicing staff, migration delay between the locations, etc.). In CPS, the physical factors, however, are not entirely predictable or easy to change [10], leading to problems such as missed task deadlines, faults of cyber systems, and faults of physical systems [11,12]. Such problems are very serious in CPS and could cause widespread social upheaval, as well as huge inconvenience and economic loss for individuals and industry alike. We propose a method of solving such problems by introducing new real-time scheduling algorithms for CPS.

Real-time scheduling for CPS differs from conventional real-time scheduling in many aspects. Table 1 highlights the key differences between the conventional real-time scheduling and the CPS real-time scheduling. As in many kinds of CPS, where servicing nodes must move to the location to perform real-time services, time required for moving has to be included in the real-time scheduling. In some CPS cases, servicing node cannot move to serviced nodes. As a future work, we will consider another case where serviced nodes move to servicing node. Also, we will make real-time scheduling algorithms considering social factors mentioned in Refs. [19–21] such as socio space, mental space, etc.

The remainder of this paper is organized as follows. Section 2 presents the real-time scheduling algorithms for CPS. Our algorithms are evaluated in Section 3. Finally, we conclude the paper and point out the future works.

2. Real-time scheduling model in CPS

In this section, we propose a real-time scheduling for CPS and compare the real-time performance (deadline meet ratio) between the conventional real-time scheduling and the proposed real-time scheduling for CPS. We assume parameters for real-time systems as follows:

- l_i: slack (laxity) time of task i (exponential distribution of average $1/\lambda$)
- e_i: execution time of task i (evenly distributed on $[0, E]$)
- m_i: migration time of servicing(computing) node to task(serviced node) i (evenly distributed in $[0, M]$).

Deadline meet ratio (DM) of task A without confliction against other tasks is the probability of the slack time l_A being greater than the moving time m_A (servicing node moving to serviced node (task A) within slack time l_A). As distribution of l_A is $\lambda e^{-\lambda t}$, the deadline meet ratio of a task A ($\mathrm{DM}_A(\lambda, m)$) is computed as follows:

$$\mathrm{DM}_A(\lambda, m) = \int_m^{\infty} \lambda e^{-\lambda t} dt = e^{-\lambda m}. \quad (1)$$

As m is assumed to evenly distributed $[0, M]$, an average deadline meet ratio is:

$$\mathrm{Mean}(\mathrm{DM}_A(\lambda, m)) = \frac{1}{M} \int_0^M e^{-\lambda m} dm = \frac{1}{\lambda M}(1 - e^{-\lambda M}). \quad (2)$$

For a simple demonstration, we compute a deadline meet ratio when two tasks conflict each other. (We also perform simulation in more realistic scenarios as described in Section 3.2.) We compute deadline meet ratios for three different scheduling algorithms: FIFO (First In First Service), LST (Least Slack Time First), ELST (Effective Least Slack Time First for CPS) scheduling algorithms.