Supplement of "Proof and Evaluation of Improved Slack Reclamation for Response Time Analysis of Real-Time Multiprocessor Systems"

Hyeongboo Baek[1], Donghyouk Lim[2] and Jinkyu Lee[1]

[1]Department of Computer Science and Engineering, Sungkyunkwan University (SKKU), Republic of Korea.

[2]RTST, Republic of Korea.

1. Task set generation method

We randomly generate 100,000 constrained-deadline task sets for each $m \in \{2, 4, 8\}$, based on a technique proposed in [1] and also used in many studies, e.g., [2]–[6]. For task set generation, we consider two input parameters: the number of processors (m = 2, 4 or 8) and task utilization parameter for bimodal distribution or exponential distribution. For a task τ_i, T_i is uniformly chosen in [1, 10], and C_i is determined by a bimodal and exponential task utilization parameter. For a given bimodal parameter p, a value for C_i/T_i is uniformly chosen in [0, 0.5) with probability p, and for a given exponential parameter $1/\lambda$, that is chosen according to the exponential distribution whose probability density function is $\lambda \cdot exp(-\lambda \cdot x)$, where each parameter can be a value of 0.1, 0.3, 0.5, 0.7 or 0.9. Then, D_i is uniformly chosen in $[C_i, T_i]$ as we consider the constrained-deadline task model.

For each task utilization, we conduct the following steps until 10,000 task sets are generated.

1. We generate a task set containing $m + 1$ tasks.
2. We check whether the generated task set passes a necessary feasibility condition [7].
3. If it passes the necessary feasibility condition, we include the task for evaluation. Then, we generate a new task set by adding a new task into the old task set and return to Step 2). Otherwise, we discard the task and return to Step 1).

We create 10,000 task sets for each task utilization model (bimodal or exponential model with a given parameter value chosen among 0.1, 0.3, 0.5, 0.7 and 0.9), in total 100,000 task sets for a given m.

2. Fairness of simulation environment

Considered task set generation method was used in many previous studies since it covers unbiased task sets for a fair evaluation [2]–[6]. As shown in Tables 1 and 2, each task utilization distribution shows distinct average number of tasks in a task set (denoted by $\overline{n}$) and average task utilization (denoted by $\overline{C_i/T_i}$). Note that we used the exactly same task utilization distribution setting for T_i in [1, 10] exploited in [6] for a fair evaluation. Also, the similar values for T_i in [1, 1000] and $m \in \{2, 8\}$ are explicitly shown in [5]; $\overline{n}$ of 3.1, 11.6 and 4.3, and $\overline{C_i/T_i}$ of 0.55, 0.09 and 0.32, respectively for $m = 2$, and $\overline{n}$ of 10.1, 43.1 and 14.5, and $\overline{C_i/T_i}$ of 0.68, 0.1 and 0.39, respectively for $m = 8$.

Table 1: $\overline{n}$ of each task utilization distribution for a given input parameter $p \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$

m	Bimodal distribution					Exponential distribution				
	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
2	7.4	6.6	6.0	5.5	5.1	10.5	7.8	7.0	6.7	6.5
4	14.3	12.5	11.2	10.3	9.5	20.5	15.0	13.3	12.7	12.3
8	28.0	24.3	21.7	19.9	18.4	40.5	29.2	25.9	24.5	23.9

Table 2: $\overline{C_i/T_i}$ of each task utilization distribution for a given input parameter $p \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$

m	Bimodal distribution					Exponential distribution				
	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
2	0.39	0.47	0.55	0.62	0.71	0.24	0.37	0.43	0.46	0.48
4	0.4	0.49	0.58	0.67	0.77	0.25	0.37	0.44	0.48	0.5
8	0.41	0.5	0.6	0.69	0.79	0.25	0.38	0.45	0.49	0.51

Due to page limit, the main body of our paper presents evaluation results of bimodal distribution with 0.9, and exponential distribution with 0.1 and 0.9 for $m = 4$ (Fig 2. in the main body of the paper) among all evaluation results under 10 different task utilization distributions.

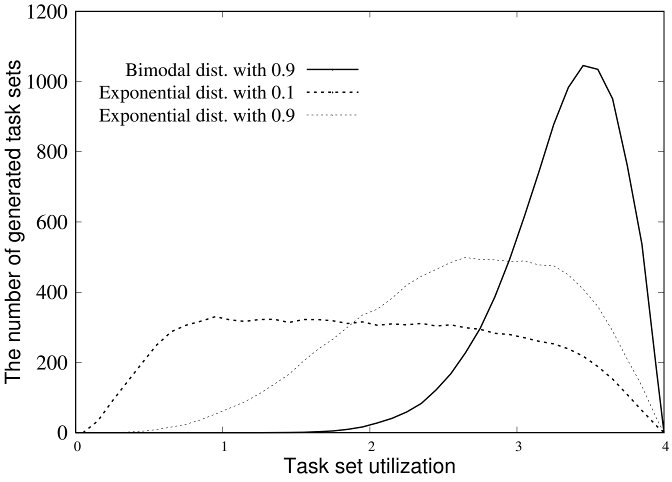

Fig. 1: The number of generated task sets under each task utilization distribution for $m = 4$

Fig. 1 plots the number of generated task sets having different task set utilization under bimodal distribution with 0.9, and exponential distribution with 0.1 and 0.9. As shown in Fig. 1, the maximum number of generated task sets picks at different task set utilization under different task utilization distributions, implying generated task sets' utilization covers various cases.

References

[1] T.P. Baker, "Comparison of empirical success rates of global vs. partitioned fixed-priority EDF scheduling for hard real-time," Tech. Rep. TR–050601, Department of Computer Science, Florida State University, Tallahassee, 2005.

[2] T.P. Baker, “Comparison of empirical success rates of global vs. paritioned fixed-priority and EDF scheduling for hard real time,” Tech. Rep. TR-050601, Dept. of Computer Science, Florida State University, Tallahasee, 2005.

[3] M. Bertogna, M. Cirinei, and G. Lipari, “Schedulability analysis of global scheduling algorithms on multiprocessor platforms,” IEEE Transactions on Parallel and Distributed Systems, vol.20, pp.553–566, 2009.

[4] J. Lee, A. Easwaran, and I. Shin, “LLF schedulability analysis on multiprocessor platforms,” Proceedings of IEEE Real-Time Systems Symposium (RTSS), pp.25–36, 2010.

[5] J. Lee, A. Easwaran, and I. Shin, “Contention-free executions for real-time multiprocessor scheduling,” ACM Transactions on Embedded Computing Systems, vol.13, no.69, pp.1–69, 2014.

[6] J. Lee, “Improved schedulability analysis using carry-in limitation for non-preemptive fixed-priority multiprocessor scheduling,” IEEE Transactions on Computers (TC), vol.66, pp.1816–1823, 2017.

[7] T.P. Baker and M. Cirinei, “A necessary and sometimes sufficient condition for the feasibility of sets of sporadic hard-deadline tasks,” Proceedings of IEEE Real-Time Systems Symposium (RTSS), pp.178–190, 2006.