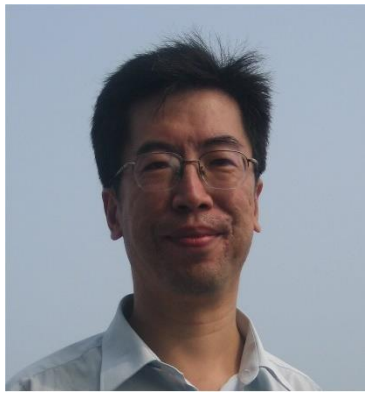
Using Linked List in Exact Schedulability Tests for Fixed Priority Scheduling

Jiaming Lv, Xingliang Zou, Albert M. K. Cheng, and Yu Jiang



Heilongjiang University and University of Houston

IEEE RTAS 2016 WiP

Outline

- The classic model (LL) and the response time analysis (RTA)
- Current best exact test
- Our linked list-based methods
- Conclusion

The classic model (LL)

- **The classic preemptive model**
  - **C. L. Liu and L. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of ACM*, 1973, 20(1): 46–61.**
- **For *fixed priority scheduling*, synchronous release defines critical instant of tasks.**
- **The *Rate-Monotonic* (RM) fixed priority scheduling is optimal for periodic/sporadic tasks that comply with a restrictive system model and that have implicit deadlines.**

The RTA-based exact test

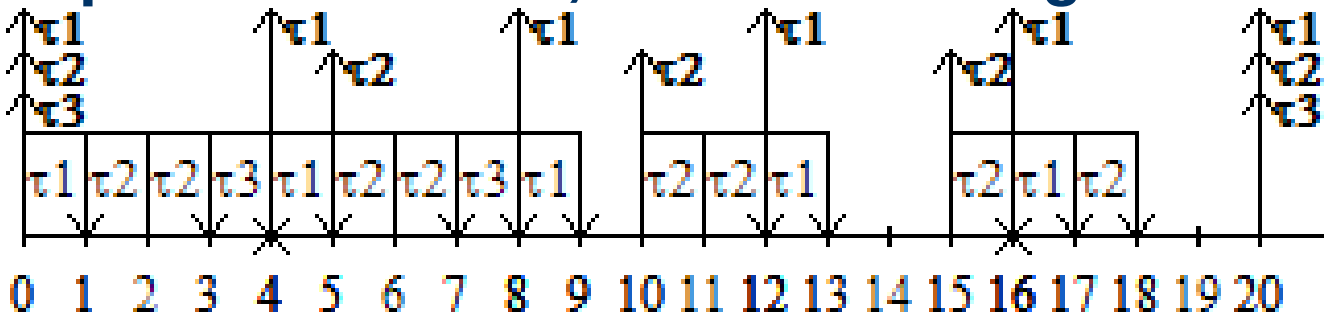
- Audsley, Burns, Richardson, Tindell, and Wellings presented the famous response time analysis (RTA) algorithm, calculating in an iterative way.
  - Applying new scheduling theory to static priority preemptive scheduling. *Software Engineering Journal*, 1993, 8(5): 284–292.
  - $R_i = C_i + sum_{0<j<i}(ceiling(R_i/T_j)*C_j)$
- The RTA-based test is a kind of exact (necessary and sufficient) schedulability test.

Current best exact test

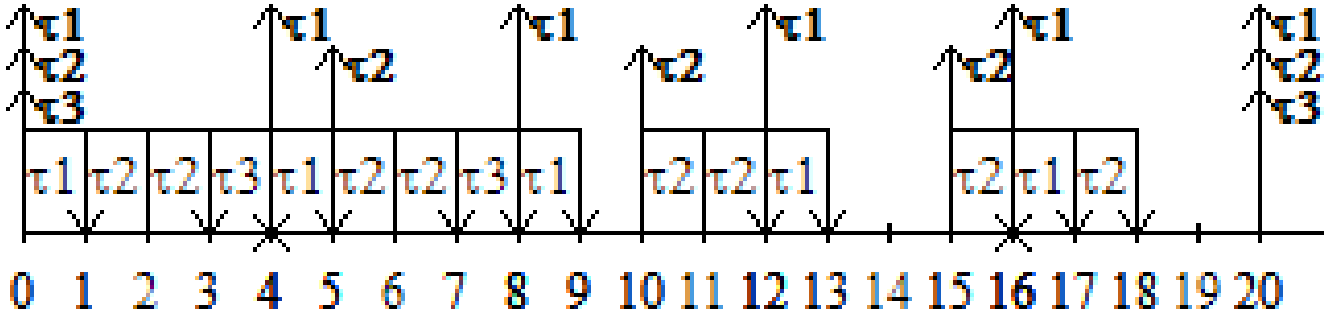
- Davis, Zabos, and Burns presented the current best RTA-based test with better initial values (In general sense, i.e., no paticular restrictions such as harmonic on task periods).
  - Efficient exact schedulability tests for fixed priority real-time systems. *IEEE Trans. on Computers*, 57(9): 1261–1276, 2008.
- Eisenbrand and Rothvoss regarded that response time computation is NP-hard.
  - Static-priority real-time scheduling: Response time computation is NP-hard. *IEEE RTSS 2008*, pages 397–406, 2008.

Our Method

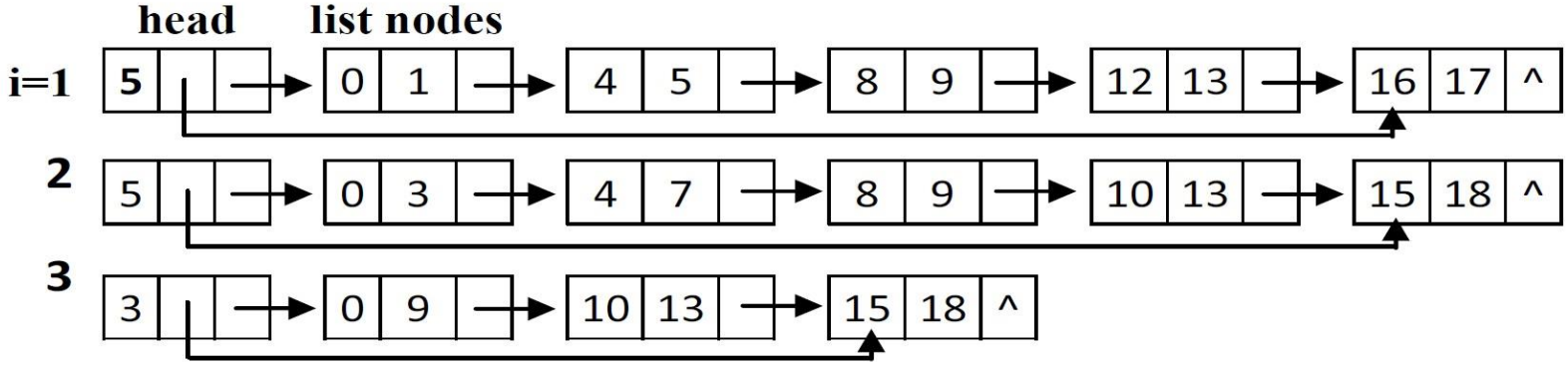
- We employ the linked list structure in the schedulability test and RTA, and
- use a linked list node to represent a *busy period* of a time interval.
- We use also a memory management in user space.
- An example
  - 3-task set: C_1=1,T_1=4; C_2=2,T_2=5; C_3=2,T_3=20; implicit deadlines; RM scheduling



An Example



C_1=1, T_1=4; C_2=2, T_2=5; C_3=2, T_3=20; implicit deadlines



The number of total jobs: 10 (=5+4+1). Response Time: RT_1=1， RT=3， RT_3=8

Performance - 1

TABLE I
HET, RTA, AND LLIST-BASED ALGORITHMS, EXECUTION TIME IN COUNTER CLOCK CYCLES × 1,000

Algorithm	Orders of magnitude spanning tasks periods						
	1(a)	1(b)	2(a)	2(b)	3(a)	3(b)	4
HET	6.9	9.4	26.6	30.8	112.8	127.4	464.9
RTA	29.56	29.82	32.47	32.48	34.63	34.49	36.67
LList	1.01	1.17	4.12	4.35	25.19	25.94	208.9
RTA/LList	29.3	25.5	7.9	7.5	1.4	1.3	0.2

The range of periods starts from 10 and 10^4 for (*a*) and (*b*), respectively.

Performance - 2

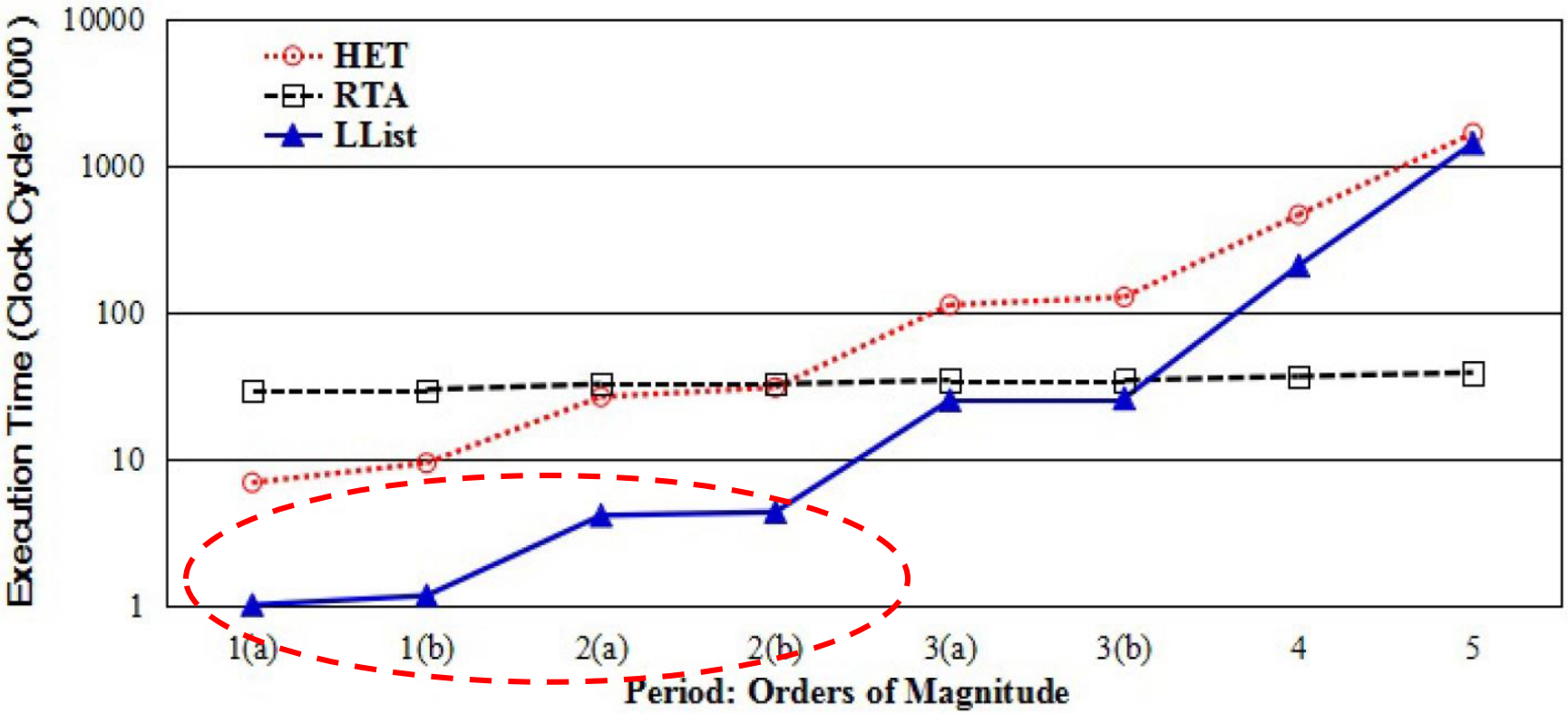


Fig. 1. Average execution time required by the HET, RTA, and LList-based exact schedulability tests versus number of orders of magnitude range of task periods. The range of periods starts from 10 and 10^4 for (a) and (b), respectively. Note both x and y-axes are logarithmic scales.

Performance - 3

- **For an *n*-task set under fixed priority RM scheduling, the total number of jobs, *N*, in the test time interval $[0, T_n)$ is at most $N = \Sigma^{n}_{i=1} ceiling(T_n / T_i)$ needed to be simulated in the LList-based exact test.**
  - **When $T_n / T_1 \leq 10$ (one order of magnitude), there will be at most 9*n*+8 jobs needed to be simulated, and our method needs only *O*(*n*) time for simulating an *n*-task set, regardless of the length of the maximum period of the task set as well as the total utilization factor.**

Performance - 4

- **Within three orders of magnitude spanning task periods, the memory space needed by the linked list is completely affordable.**

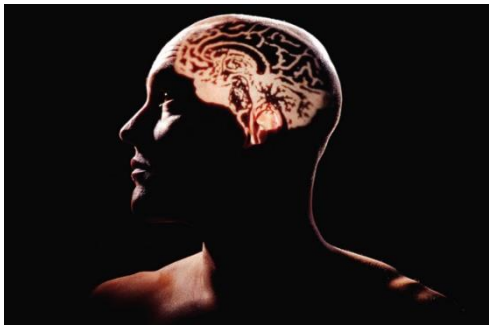
TABLE II

THE MAXIMUM NUMBER OF NODES AND CORRESPONDING MEMORY SPACE

	Orders of magnitude spanning tasks periods						
	1(a)	1(b)	2(a)	2(b)	3(a)	3(b)	4
max # nodes	28	58	236	441	2302	3246	20996
KB(int32)	0.3	0.7	2.8	5.2	27.0	38.0	246.0

Conclusion

- We provide an LList-based method for the response time analysis and exact scheduability test for fixed priority preemptive systems.
- Our preliminary results have shown that the LList-based exact test is a better candidate in exact RTA-based tests when task periods span no more than three orders of magnitude.



Thanks for your interest!

April 2016