

Performance Comparison of MRTOS and VxWorks in MultiBench Benchmark Suite

Jeong-Woo Lee, Sang Cheol Kim, Duk-Kyun Woo, Yu-Seung Ma, Pyeongsoo Mah

ETRI(Electronics and Telecommunications Research Institute), Korea

jeongwoo@etri.re.kr, sheart@etri.re.kr, dkwu@etri.re.kr, ysma@etri.re.kr, pmah@etri.re.kr

Abstract— Recently, multiple core systems are widely used in various area. To meet the needs of mission-critical applications running on multicore systems, a realtime operating system (RTOS) and verification testing for multicore are absolutely necessary. We have developed the multicore RTOS (MRTOS) for supporting the symmetric multiprocessing. It includes multicore system BSP, multicore kernel, middleware, and IDE. To evaluate the developed RTOS, in this paper, we have compared the performance of the VxWorks and the developed RTOS by using the embedded MultiBench algorithms. We observed that the developed RTOS is able to achieve more than 90% of the performance of VxWorks on average.

Keywords—Multicore, RTOS, SMP, symmetric multiprocessing, VxWorks, benchmark, MultiBench

I. INTRODUCTION

In embedded computing system, the use of multicore CPU is increasingly becoming popular as applications demand higher and higher performance levels. Reasons for using more cores on embedded systems are application speedup and improvement of real-time performance [1], [2].

The well-known RTOSs for supporting multicore system are FreeRTOS and VxWorks. The former is developed as open source software [3] and the latter is developed as proprietary software [4]. In Korea, ETRI have developed the multicore RTOS (MRTOS) for supporting the symmetric multiprocessing [5] for the first time. It includes multicore system BSP, multicore kernel, middleware, and IDE. The developed RTOS supports Intel and PowerPC platform.

To evaluate the performance of the developed RTOS, it is necessary to benchmark the scalability and flexibility of an RTOS. EEMBC MultiBench is a multicore benchmark suite meant for measuring the throughput of multiprocessor system, including those built with multicore processors [6]. The user may change the number of threads running in parallel, the data size, as well as their binding to processing elements.

For evaluating our RTOS, in this paper, we have compared the performance of the VxWorks and the developed RTOS by using the embedded MultiBench algorithms. The rest of this paper is organized as follows. In Section II, we introduce the developed multicore RTOS. In Section III, we briefly present EEMBC MultiBench software. In Section IV, we compare the performance of the VxWorks and the developed RTOS by

using the embedded MultiBench algorithms. Finally, some conclusions are drawn in Section V.

II. MULTICORE RTOS

The layout of the developed RTOS is shown in Figure 1. As depicted in the figure, it consists of two parts: SMP RTOS and Program IDE. A SMP RTOS part is for functionality of operation system. A Program IDE part is for developing applications easily.

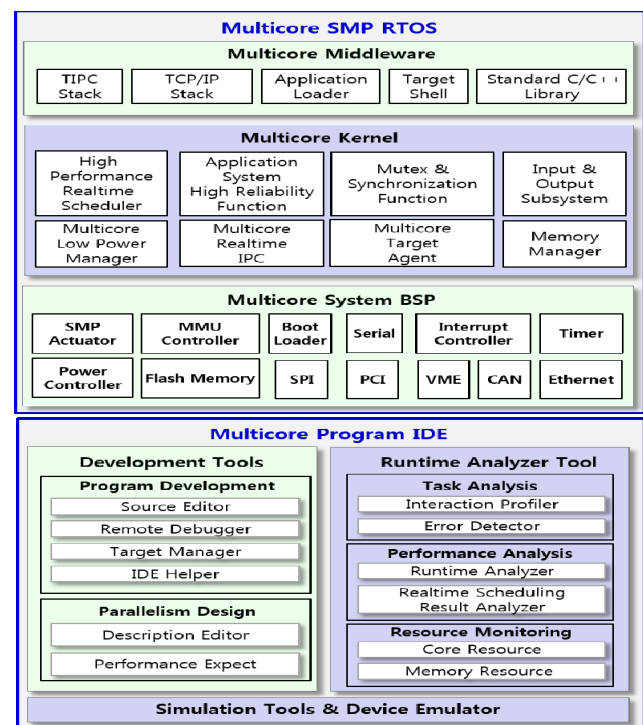


Figure 1. Multicore RTOS Layout

In SMP RTOS part, a system BSP supports MMU controller, Boot loader, and so on. A Multicore Kernel supports low power manager, realtime IPC, high performance realtime scheduler and so on. Especially, the developed scheduler is suitable for task scheduling for a fixed priority multicore embedded RTOS [5]. A Middleware supports TCP/IP stack, standard C/C++ library and so on. A Program IDE consists of development tools, runtime analyser tools and simulation tools.

III. EEMBC MultiBENCH

EEMBC, an industry alliance, develops benchmarks to help system designers select the optimal processors and understand the performance and energy characteristics of their systems. EEMBC has benchmark suites targeting cloud and big data, mobile devices (for phones and tablets), networking, ultra-low power microcontrollers, the Internet of Things (IoT), digital media, automotive, and other application areas. EEMBC also has benchmarks for general-purpose performance analysis including CoreMark, MultiBench (multicore), and FPMark (floating-point) [6].

The key element of MultiBench is the unique Multi-Instance Test Harness (MITH) that provides an abstraction layer to support portability of the benchmarks across an infinite number of platforms. MITH makes it easy to run on any platform, operating system, and tool-chain.

MultiBench is also comprised of a wide-variety of application-focused workloads encompassing the networking, consumer, and office automation domains. Regardless of the application-specific focus of these workloads, their flexibility makes them applicable to a whole host of other applications. And if that’s not enough flexibility, it allows us to compose custom workloads for testing specific system characteristics.

IV. PERFORMANCE COMPARISONS

For evaluating our RTOS, in this paper, we have compared the performance of the VxWorks and the developed RTOS by using the embedded MultiBench algorithms. VxWorks is a well-known real-time operating system developed as proprietary software by WindRiver Inc. Figure 2 shows how MultiBench algorithms are ported on MRTOS and VxWorks.

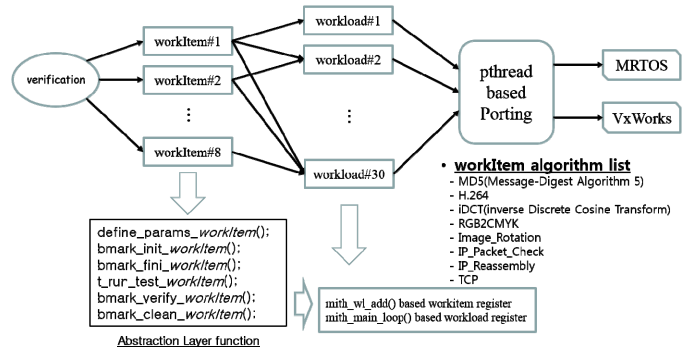


Figure 2. MultiBench Porting Diagram

The embedded MultiBench algorithms, which is called as workItem, include for example processing TCP/IP packets, image rotation, MD5 checksums, color conversion, iDCT, and H.264 video encoding. Every workitem is combined into each other. This combined algorithm is called “workload”. The workload is basic unit for testing RTOS function. Each workload is ported by using pthread functions (POSIX library) which are supported by RTOS.

The tests are performed on Intel Core i5 multicore system which support 4 CPU cores. In our simulation, we have combined each workitem to 30 kinds of workloads and tested every workloads with 4, 8, and 16 threads. To improve the

accuracy of the comparative evaluation for the developed RTOS and VxWorks, every workloads are repeated from 50 to 2500 times. To evaluate the performance, we measure how many each workloads are performed per second.

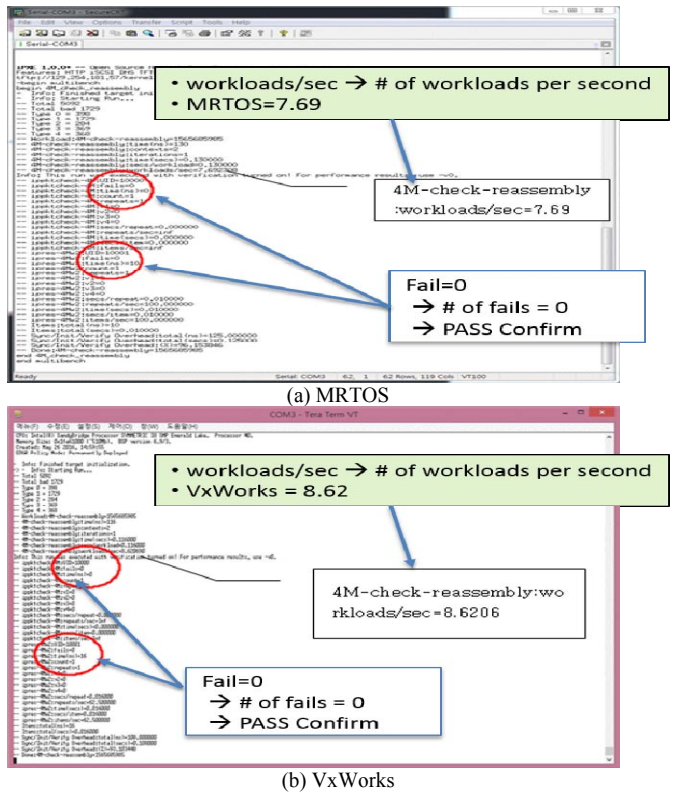


Figure 3. Simulation Results for 4M-check-reassembly workload

Figure 3 shows the simulation results for “4M-check-reassembly” workload. This workload is the combination of IP-packet-check and IP-reassembly workItems. Each workItem is performed sequentially. As shown in Figure, we can confirm the pass for this test on MRTOS and VxWorks. That is, the number of fails on both system is zero. Figure 3(a) and Figure 3(b) shows the performance result for MRTOS and VxWorks, respectively. The number of executed workloads per second is 7.69 and 8.62 for MRTOS and VxWorks, respectively. A simulation result shows that the developed RTOS supports about 90% of the performance of VxWorks.

Table 1 shows the performance comparison with MRTOS and VxWorks for various workloads. We have tested 20 kinds of workloads on both RTOSs. We observed that the performance of MRTOS is at least 80% higher than the performance of VxWorks. However, it has good performance in most cases. We observed that the performance is poor when many mutex and semaphore functions are used in applications comparing to VxWorks.

Table 2 shows the simulation results with the different number of threads. W4, W8 and W8 represent the number of threads with 4, 8 and 16, respectively. As shown is Table 2, the developed RTOS provides constant performance regardless of the number of threads. As a results, we can easily support the time limitation of the embedded system.

TABLE 1. PERFORMANCE COMPARISON WITH MRTOS AND VxWORKS

workload	MRTOS	VxWorks	Ratio
4M-check	21.50	7.52	285%
4M-check-reassembly	7.69	8.62	89%
4M-check-reassembly-tcp	2.50	2.40	104%
4M-check-reassembly-tcp-cmykw2-rotatew2	2.33	2.61	89%
4M-cmykw2	25.00	15.15	165%
4M-cmykw2-rotatew2	25.00	20.41	122%
4M-reassembly	7.69	8.62	89%
4M-rotatew2	100.00	30.3	330%
4M-tcp-mixed	2.56	3.16	81%
iDCT-4M	1.52	1.88	81%
ippktcheck-4M	50.00	7.52	665%
ipres-4M	7.69	8.62	89%
rgbcmyk-4M	25.00	20.41	122%
rgbcmyk-4Mw1	50.00	20.41	245%
rotate-34kX128w1	5.26	0.47	1132%
rotate-4Ms1	100.00	30.30	330%
rotate-4Ms1w1	100.00	30.30	330%
rotate-4Ms64	100.00	30.30	330%
rotate-4Ms64w1	100.00	30.30	330%
rotate-color1Mp	100.00	30.30	330%

V. CONCLUSIONS

EEMBC MultiBench is a good benchmark test suite for multicore kernel and system. From our experiences on Multibench test suite, we found out that the developed kernel architecture is efficient with multiple core perspective. When the developed RTOS is used, it is expected that developers can shorten development time of multicore applications. For the future works, we will develop the MRTOS with performance similar to the VxWorks.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. 10047067, Development of SMP RTOS technology for high performance and realtime multicore embedded systems)

REFERENCES

- [1] D. Bautista, J. Bautista, H. Hassan, S. Petit, and J. Duato, "A simple power-aware scheduling for multicore systems when running real-time applications," in *Proc. of 22nd IEEE/ACM international parallel and distributed processing*, pp. 1-7, 2008.
- [2] J. Mistry, "FreeRTOS and Multicore", Master's thesis, University of York, 2011
- [3] FreeRTOS website. [Online]. Available: <http://www.freertos.org/>
- [4] Wind River VxWorks Platforms 6.9 documentation. [Online]. Available: <http://www.windriver.com>
- [5] S. C. Kim, and S. Lee, "Decentralized Task Scheduling for a Fixed Priority Multicore Embedded RTOS", *Computing*, vol. 97, pp. 543-555, Jun., 2015.
- [6] "Multibench 1.0 multicore benchmark software," in *The Embedded Microprocessor Benchmark Consortium*. [Online]. Available: <http://www.eembc.org/>

TABLE 2. PERFORMANCE COMPARISON WITH MRTOS AND VxWORKS WITH DIFFERENT WORKERS(M: MRTOS, V: VxWORKS)

workload	W4		W8		W16	
	M	V	M	V	M	V
4M-check	100.0	7.51	50.00	7.52	50.0	7.52
4M-check-reassembly	7.69	7.52	7.69	7.52	7.69	7.52
4M-check-reassem-tcp	2.38	2.74	2.44	2.31	2.44	2.74
4M-check-reassembly-tcp-cmykw2-rotatew2	2.33	2.61	2.33	2.61	2.33	2.61
4M-cmykw2	25.00	15.15	25.00	15.15	25.0	20.41
4M-cmykw2-rotatew2	25.00	20.41	25.00	20.41	25.0	20.41
4M-reassembly	7.69	7.52	8.33	7.52	8.33	7.52
4M-rotatew2	100.0	30.31	100.0	30.31	100.0	30.31
4M-tcp-mixed	2.56	3.16	2.56	3.16	2.56	3.16
iDCT-4M	1.64	1.94	1.64	1.94	1.64	1.88
ippktcheck-4M	50.00	7.52	50.00	7.52	50.00	6.71
ipres-4M	8.33	8.62	7.69	8.62	7.69	8.62
md5-4M	50.0	1.82	50.0	3.53	33.33	6.71
rgbcmyk-4M	25.00	15.15	25.00	15.15	25.00	15.15
rotate-4Ms1	100.0	30.30	100.0	30.30	100.0	30.30
rotate-4Ms64	100.0	30.30	100.0	30.30	100.0	30.30
rotate-color1Mp	100.0	30.30	100.0	30.30	100.0	30.30



Jeong-Woo Lee (BS'96-MS'98-Ph.D.'03) received the Ph.D. degree in the Info. and Comm. Department from Gwangju Institute of Science and Technology (GIST) in 2003. He is currently working in Electronics and Telecommunications Research Institute (ETRI). His research interests include digital video coding algorithms, gpu-based algorithms, parallel processing, embedded software, and big data computing.



Sang Cheol Kim (BS'99-MS'01-Ph.D.'06) received the Ph.D. degrees from the Pohang University of Science and Technology (POSTECH) in 2006. He is currently a senior researcher in the Embedded Software Research Department in ETRI. He performed a lot of projects and has abundant experiences related to embedded systems, sensor networks and realtime operating systems.



Duk-Kyun Woo (BS'93-MS'95-Ph.D.'01) received the Ph.D. degrees in computer engineering from Hongik University in 2001. In January 2001, he joined the Embedded Software Research Department in ETRI, where he is currently a principal researcher. His research interests include compiler, real-time operating system, embedded software development tool, and wearable device development kit.



Yu-Seung Ma (BS'98-MS'00-Ph.D.'05) received the B.S., M.S., and Ph.D. degrees in computer science from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1998, 2000 and 2005, respectively. In 2005, she joined in the Electronics and Telecommunications Research Institute (ETRI), Korea. Her research interests include program testing, mutation testing, and embedded software engineering.



Pyeongsoo Mah is a principal researcher of Electronics and Telecommunications Research Institute (ETRI). He received his B.A. from Seoul National University, M.S. from City University of New York, and Ph.D. in Computer Science from Wright State University, USA. His research interest includes embedded software, operating systems, and big data computing.