A Low Level Analysis of the Realtime Mach Distributed Operating System

John Drummond, Michael Wu
NCCOSC RDT&E Division (NRaD)
San Diego, CA 92152-5000
(619) 553-4131, 553-4107
drummond@nosc.mil, mwu@nosc.mil

Abstract

The use of measuring techniques to assist in the process of evaluation has always been a speculative endeavor. There is the presence of doubt as to what should be measured, and once this measurable entity has been determined, then what constitutes proper measurement of the entity itself. Current work in the realtime operating system software development arena presents us with this same enigma (how to, and what should be measured). The domain we have been pursuing to this end is a segmented appraisal consisting of analyzing specific elements which are properties of realtime operating systems. Here, six candidate elements of primary importance in the evaluation of realtime operating systems have been suggested and interpreted by the Rhealstone benchmark Kar[1,2]. These elements include: Task Switch Time, Preemption Time, Interrupt Latency Time, Semaphore Shuffling Time, Deadlock Breaking Time and Interprocess Communication Latency Time. We have researched these elements in relation to the Realtime Mach Operating System and will present the results of our findings in this paper.

To minimize the variation on the end results, this appraisal was conducted under a specifically conditioned environment and we acknowledge the limitations of the results as such.

Introduction

We have been researching and investigating realtime operating systems for a good amount of time now. In an endeavor to determine various prominent features of these systems, we have and will continue to evaluate these real time operating systems at the application level as well as kernel level in addition to simulation and modeling levels. This work will be applicable in both military and commercial domains and hopefully provide some insight into the realtime operating system arena. Our goal is to provide a realtime operating system that will perform reliably under mission critical requirements.

The Rhealstone benchmark categorized as a "fine-grained benchmark"[3], examines low level kernel primitives, and as such, will provide some insight into specific RTMACH operating system aspects. The kernel primitives and their respective elements selected for this examination are all inclusive. In our measurement instance we will only be utilizing specific Rhealstone elements for the task of accomplishing future realtime operating system modeling efforts, however for the sake of completeness we have included the study of all of the Rhealstone elements in this paper.

Realtime Mach Operating System

Realtime Mach [4] is a microkernel based operating system offering a choice of scheduling algorithms (Rate Monotonic, Deadline Monotonic, Fixed Priority First Come First Serve, Earliest Deadline First). One of the foundations for development of this realtime operating system has been the Advanced Realtime Technology (ARTS) [5], another, of course, being the Mach [6] non-realtime operating system. Evolution of these systems has been the ongoing work of the School of Computer Science at Carnegie Mellon University (CMU) [7]. Current RTMACH versions are based upon Mach 3.0 which follows a multiple server model. Utilizing a lean microkernel, with limited primitives, the user can access full system functionality via emulation libraries, and various servers. From a program view, the emulation libraries and servers run outside the kernel, lending easy facilitation to the implementation of multiple operating system support. Additionally, the RTMACH operating system has the strength and flexibility of supporting diverse architectures such as uniprocessor and multiprocessor. From a distributed system perspective, the degree of shared memory
access is also variable across these architectures. This access ranges from Uniform Memory Access (UMA), Non-Uniform Memory Access (NUMA), to No Remote Memory Access (NORMA).

Generic and Weighted Average of Rhealstone Primitives

We have obtained, through Rhealstone measurements, results of individual examination of specific components within the systems kernel. These measurement findings have usefulness in their own right by providing deeper insight into RTMACH. In our instance, they will be used in the creation of a system model which will in turn be utilized for the development of a realtime analytic tool. The creators of Rhealstone[1] have suggested generic numerical interpretation representative of an average composed of the various kernel measurement primitives. The interpretation may also be weighted to offset realtime operating systems which are deemed application specific in the area of one or more kernel primitives. There has been some discussion within the realtime community as to the merit of such an interpretation. This not withstanding, we will present the generic numbers in the pursuit of completeness. The weighted average Rhealstone numbers will not be presented here. For the generic Rhealstone formula we utilized following listed[2] calculations:

With the following set of measured values, where\[ t_1 = \text{Task Switch Time}, t_2 = \text{Preemption Time}, t_3 = \text{Interrupt Latency}, t_4 = \text{Semaphore Shuffle time}, t_5 = \text{Deadlock Break time}, t_6 = \text{Inter task Communication latency}, \]

the arithmetic average of Rhealstone components is defined as:\[ t^* = \frac{(t_1 + t_2 + t_3 + \ldots + t_6)}{6} \]

and the system's consolidated realtime performance number is: \[ \frac{1}{t^*} \text{ Rhealstones/second} \]

The computation follows:\[ t^* = \frac{(25 + 40 + 60 + 46 + 94 + 3)}{6} = 44.6 \text{ us} \]

and the systems consolidated Rhealstone performance number is: \[ 22.4 \times 1000 \text{ Rhealstone/second}. \]

Conclusion

This work is, as was mentioned earlier, part of an ongoing effort to discern various aspects and functionality of the realtime arena. The "realtime requirement" is one which is fast becoming an exigency of all applications developed today. RTMACH is a microkernel based distributed operating systems developed by the School of Computer Science, Carnegie Mellon University (CMU). It provides the advantages of parallelism, prediction, preemptibility and extensibility. Benchmarks for the application level such as Hartstone have been used to measure and analyzed RTMACH. The results of this Rhealstone measurement provides us with some insight into the RTMACH's system level performance as well as its comparison with other operating systems. It is not our intention to claim that RTMACH has better throughput than others. In the future, we will cooperate with Carnegie Mellon University to conduct more experiments to utilize more powerful computers and operating systems. These experiments will provide higher performance, research data and new technology to industries.

References