White Paper

SGI's Altix® Real-time Futures

Lori Gilbertson, Bill Mannel, Michael Raymond, Dimitri Sivanich, Leslie Tung

November, 2004

Table of Contents

1.0 Abstract

This paper describes the efforts underway to bring hard real-time (HRT) capabilities resident in SGI's IRIX® environment into the open source Linux® environment. The audience for this paper are programmers and system analysts already familiar to some extent with real-time systems and applications, whether in an embedded architecture or in a more generalized compute architecture.

2.0 SGI® Real-Time Applications

SGI has produced systems capable of hard real-time response for over ten years. Real-time systems can complete computational operations within stringent time constraints while obtaining accurate results. Applications that typically need real-time computation are many and varied, including:

- Man-in-the-loop simulators: Closed-loop simulators used for training simulators (aircraft, ship, tank), engineering simulators, and power plant simulators
- Hardware-in-the-loop simulators: Closed-loop simulators with high frame rates (thousands of Hz) often requiring high-speed, low-level I/O
- Telemetry, radar, and data acquisition: Open-loop systems for acquiring, processing and storing (or re-transmitting) data in real-time
- C4I applications (Command, Control, Communications, and Intelligence): Systems that combine a lot of the previous application needs and often include displays that need to be updated for the latest information
- Video applications: Systems that handle live video such as virtual sets, real-time broadcast, and video-on-demand (VOD)

There are two distinctions to make in real-time processing: hard real-time (HRT) and soft real-time (SRT). In hard real-time, the computation is successful only when the answer is accurate and it is delivered within the required time constraint. This time constraint is often referred to as the "guaranteed time." In soft real-time systems, the computation is successful if the answer is accurate and it is delivered within a threshold of time. Soft real-time systems frequently have very good "average" response times, but occasionally can have a few periods when response may be an order of magnitude or more than the average time. HRT systems require specific capabilities within the operating system that have guaranteed latencies and determinism; thus it is much more challenging to achieve than SRT systems.

3.0 SGI Real-Time Features

In the past SGI real-time systems have been exclusively based upon MIPS® processors and the IRIX operating system. For systems up to 64-processors, interrupt response times were guaranteed to 50 microseconds. SGI real-time features have typically been very useful for a class of applications that do not require extremely low interrupt response times (in the single digit microseconds) but which benefit from a powerful 64-bit processing, I/O, and high-end graphics environment. In some cases, other systems with lower interrupt response times may handle the interrupt and data ingest, while an SGI real-time system does key processing of the data received.

In addition to enabling low, guaranteed, interrupt response times, SGI has built a feature set useful to the real-time programmer:

- Programmatic access to a real-time clock (RTC) with high resolution
- Low latency primitives for process creation, control and coordination, both in a proprietary (sproc(2)) and standards-based paradigm (POSIX® threads, or pthreads). These include process synchronization features such as locks, semaphores, and message queues
- Programmatic and command-line access to features that allow memory locking, process pinning, and CPU restriction and isolation
- Additional paradigms for I/O handling including asynchronous I/O, direct I/O, and guaranteed rate I/O
- User level drivers that allow control of devices outside the kernel context (and subsequent overhead)
- High level capabilities such as the Frame Rate Scheduler (FRS) to handle frame-based applications, i.e. those applications that have a standard set of operations repeated continuously. Completion of each set or "frame" of operations within a given frequency is imperative
- Real-time implementations of the UNIX® signal capability including real-time priorities and queued signals
- Capabilities to both generate and receive real-time interrupts from outside the MIPS and IRIX system
- IRIXview™ that provides real-time application debugging via a graphical interface

The wealth of features and capabilities available on IRIX today represent many years of effort. As open source-based operating systems such as Linux become more popular, SGI has decided to bring the development of its class of real-time capabilities into that arena as well. The immediate target for this work will be the SGI Altix system, which is SGI's server platform combining Itanium® 2 processors and Linux.

4.0 Real-Time Systems and Linux

As with many systems, initial real-time systems began as proprietary operating systems, often extensions of other proprietary operating systems used for more generalized applications (such as IRIX). Over time, the open source advantages as well as the “lighter” initial weight of Linux favored its adoption as a real-time system. The favored use of Linux for internet-based applications, and the need for embedded systems in many internet and telecommunications hardware provided additional impetus to this trend.

Today a significant amount of embedded real-time applications rely upon various flavors of real-time operating systems built from Linux. Typical vendors are TimeSys (www.timesys.com), Concurrent Computer Corp. (www.ccur.com), Monta Vista (www.montavista.com), Lynuxworks (www.lynuxworks.com), FSMlabs (www.fsmlabs.com), and Wind River Systems (www.windriver.com).

However, the majority of the applications in use are for either single-processor or small (two to four processor) Shared Multiprocessor (SMP) architectures. In these applications open-source Linux is used for the basis of a special kernel to enable HRT performance. The HRT performance of these systems can be quite good (in the sub-10 microseconds response time); however, the systems are often limited to handling only interrupts with little additional processing. Although HRT is not possible within the open source Linux kernels available today, SRT is possible and frequently used for a variety of applications.

SGI and a number of makers of real-time kernels have begun to contribute improvements to Linux that will enable more HRT level performance in the future. The Linux 2.6 kernel contains the following HRT real-time capabilities:

- Latency improvements made to the kernel
- Preemptive kernel scheduling. This allows the kernel to interrupt (preempt) one process in favor of another and is important for real-time processing
- Memory locking, which allows key portions of a program to always be memory-resident, avoiding the additional latency of paging in when rapid response is desired
- Round-robin and FIFO real-time scheduling to allow more fine-grained control of process scheduling often necessary in real-time applications
- High-resolution clocks
- Interprocess communications via semaphores, real-time signals, shared memory objects, and message queues

5.0 SGI Altix and Real-Time Response

As the first step to developing real-time capability within the Altix platform, it was necessary to run some tests on the Altix platform in various configurations to understand a baseline of interrupt response time for an early Linux 2.6 kernel. These tests are similar to those used to determine the interrupt response time on IRIX. The tests were also intended to understand a direction of engineering effort in putting real-time capabilities in Altix. The tests would also indicate if any of the specialized real-time versions of Linux were really necessary.

Tests were run on both 4P and 32P Altix machines. A Linux 2.6 kernel was used because the 2.6 kernel has the important feature of preemptive kernel scheduling as well as kernel latency improvements that the current 2.4 kernel shipping on Altix lacks. (Time frame for shipment of the 2.6 kernel on SGI Altix is Q1, CY2005). Preemptive kernel scheduling was enabled for the purpose of the test.

For the 4P Altix, 98% of the response times were 30 microseconds or less. While for the 32P Altix, 87% of the response times were 30 microseconds or less. On the 32P system those response times not in the 30 microseconds or less range could range as high as hundreds of milliseconds.

In comparison, similar tests were run on a 32P Origin® 3000 server with CPU restriction enabled. CPU restriction allows specific processes or threads to run on a specific processor, and not be interrupted by another process needing to run. There, 100% of the interrupt response times ranged from 25-50 microseconds. When CPU restriction was not used, 87% of the response times were in the 27-50 microseconds range, while the 13% of responses had times out to 100 microseconds.

In conclusion, by enabling preemptive kernel scheduling and utilizing CPU restrictions, a 32P Altix server can deliver response times similar to that of a 32P Origin® server.

6.0 SGI Altix and Real-Time Futures

SGI Engineering has completed work to provide a real-time foundation within the Altix system. As evidenced in the baseline testing, CPU restriction is necessary to achieve HRT performance. High-resolution clocks are required as well, in order to get highly accurate real-time timings.

SGI is focussing future efforts on development of additional features including:

- A port of the Frame-rate Scheduler (FRS) to Linux
- User-level interrupt capability
- External interrupt capability (this is hardware plug-in to generate interrupt, not pure software)
- IRIXview capability, by taking advantage of Linux tools already available

7.0 Conclusion

Standard Linux 2.6 coupled with the Itanium 2 processor-based Altix platform will serve as a foundation for similar real-time capabilities that customers have enjoyed with MIPS and IRIX OS-based platforms over the last decade. 30 microsecond guaranteed interrupt response is available in late 2004 with early access software and general availability delivered in 2005. Some advanced features are in development and will be delivered in 2005.

8.0 References

1. http://www.linuxdevices.com/articles/AT6106723802.html
2. http://www.sgi.com/software/react/
3. http://www.sgi.com/support/custeducation/courses/sw_dev/real-time_programming.html
4. SGI® REACT™ Programmer's Guide, (document number 007-2499-010) available at http://techpubs.sgi.com

sgi®

Corporate Office
1500 Crittenden Lane
Mountain View, CA 94043
(650) 960-1980
www.sgi.com

North America +1 800.800.7441
Latin America +55 11.5509.1455
Europe +44 118.925.7500
Japan +81 3.5488.1811
Asia Pacific +1 650.933.3000

3606 [11.03.2004] J14747