Linux® Scaling Linux to Extremes: Experience with a 512-CPU Shared Memory Linux System

Ray Bryant, John Baron, John Hawkes, Arthur Raefsky, and Jack Steiner
Silicon Graphics, Inc.
raybry@sgi.com, jbaron@sgi.com, hawkes@sgi.com, raefsky@sgi.com, steiner@sgi.com

Abstract

The SGI® Altix™ 3000 family of servers and superclusters are nonuniform memory access whose hardware supports up to 512 Intel® Itanium® 2 processors and 4TB of main memory in a single cache-coherent, shared memory domain. As originally announced in January 2003, the Linux system for Altix only supported up to 64 processors in a single Linux image. Since then, SGI engineering has extended Altix for Linux to support a full 512 processor system in a single Linux image. This paper discusses the changes to Linux that had to be made in order to do this. In addition, we summarize our experience with such large system and provide some application results that summarize the customer value of such systems.

1 Extended Abstract

The SGI Altix 3000 is a highly expandable platform that pushes the extremes of Linux in the dimensions of CPU count, main memory size, and I/O configurations. Its NUMA hardware architecture supports cache coherent memory accesses for as many as 512 Itanium 2 processors, 4 terabytes of main memory, and thousands of disk drives.

When the SGI Altix system was announced in January 2003, SGI supported a 64 CPU single-system Linux system (which at that time was the largest single-system Linux system supported in the industry). Larger systems were supported as 'super-clusters' of up to 8 such Linux systems connected via shared memory. Applications for a super cluster were programmed using MPT, SGI's implementation of MPI.

During the course of 2003, rapid progress has been made in scaling up the sizes of Altix machines running a single-system-image (SSI) Linux. Currently, we have a 512-cpu system routinely running an SSI Linux at our Eagan, MN engineering facility, and one early release customer (NASA Ames) has a 512-cpu system running a single Linux image. With our current hardware, the 512 processor system is the largest SSI Linux system that can be built, since cache coherency is only supported up to this number of processors. Systems with larger processor counts are constructed as clusters with each node in the cluster consisting of a SSI Linux with up to 512 processors. Communication betweeh the nodes of such a so called "super-cluster" is done via shared memory with software enforced cache coherence.

During the course of 2003, rapid progress has been made in scaling Linux both up and out

– sizes of Altix machines for larger clustered configurations continued to increase while demand for larger single-system-image configurations running a single copy of the Linux kernel have increased as well. For example, early adopter customers such as NASA Ames Research Center have already deployed a 512-cpu system running a single Linux image to solve real world problems. SGI continues to see strong, real world demand in pushing Linux to the extreme in total cluster size, node size, and single system image size.

Each of these dimensions of system configuration have challenged Linux in the areas of functionality, performance, and scaling. Over the past two years, SGI and the Linux Community at large have extended Linux to efficiently support such extreme configurations, beginning with the 2.4 kernel and continuing today with the 2.6 kernel. This presentation will describe the work that was necessary to support these extreme configurations in a Linux 2.4 kernel environment, the Community work that went into the 2.6 kernel, and the ongoing work necessary for the 2.6 kernel to achieve levels of scaling that SGI has accomplished for the 2.4 kernel.

We will also discuss our experience in using such a system to solve real world HPC problems and the day-to-day challenges that we have encountered in managing, using, and maintaining such a large system. We will explore the advantages and disadvantages of running an Altix system as a single huge SSI system versus running the same system in cluster mode. We will also discuss why our customers seem to prefer a single SSI system over a clustered system.

Functional changes to Linux for such extreme configurations include expanding kernel data structures, and extending user-kernel interfaces, to provide basic support for these significantly larger numbers of CPUs, main memory sizes, and I/O configurations. Since the Altix system (as well as many other large-scale, shared-memory systems) is a NUMA architecture, the kernel needed to learn to manage the NUMA aspects of these configurations, such as discontiguous physical memory addresses.

Efficient scaling to large configurations requires effective reductions in system bottlenecks. Some of these bottlenecks are inherent in hardware design. Others bottlenecks are found in software, such as contention on locks (both kernel locks and user locks) and contention on common memory locations (some intentional and others accidental). NUMA systems function most efficiently when CPUs access local memory, so performance is rewarded when care is taken to maximize local memory accesses and minimize remote memory accesses. For example using our NUMA support, we have recently achieved a world record result for the STREAM triad benchmark of over one terabyte per second on a 512 processor system.

Extreme main memory configurations may also lead to extraordinarily large allocations for various kernel data structures. Even though the allocations may be linearly proportionate to memory size, the extreme memory sizes may lead to unnecessarily huge data structures that not only waste memory, but also are noticeably cumbersome and time consuming to search and manipulate.

Huge I/O configurations increase the need for efficient ways to manage the persistent naming and organization of devices. For example, a system administrator needs to be able to quickly and correctly identify a failed disk drive in a configuration of potentially thousands of drives. Administrators and users also need tools to display throughput and performance statistics for I/O devices. Data structures to capture this information, and visualiza-

tion tools to display the information, that work well for a handful of devices may likely be ineffective in handling hundreds or thousands of devices.

Nonetheless, the overall number of changes necessary to support such large shared memory SSI Linux systems is relatively small. Leveraging on the existing Linux base of kernel and system code has allowed SGI engineers to concentrate on the particular issues of scaling Linux for the SGI Altix platform. In part, this is the reason for the phenomenal progress that SGI has made made in 2003 – from initial announcement of 64 processor SSI Linux systems in January 2003 to demonstrated support of a 512 processor SSI system.

If one considers the fact that at the present time, a 512-processor Itanium 2 Linux cluster is considered to be a large non-shared memory cluster, the fact that SGI has demonstrated a 512-cpu SSI Linux running as a shared memory system certainly should qualify the SGI Altix system as an example of Extreme Linux. SGI engineering is dedicated to solving the problems of supporting truly large SSI and supercluster versions of the Altix system and this presentation is intended to describe the challenges that had to be overcome to develop, support, and use such large computing systems running Linux.