## Shared-Memory Multiprocessors -Introduction\*

## Charles Severance Kevin Dowd

This work is produced by OpenStax-CNX and licensed under the Creative Commons Attribution License  $3.0^{\dagger}$ 

In the mid-1980s, shared-memory multiprocessors were pretty expensive and pretty rare. Now, as hardware costs are dropping, they are becoming commonplace. Many home computer systems in the under-\$3000 range have a socket for a second CPU. Home computer operating systems are providing the capability to use more than one processor to improve system performance. Rather than specialized resources locked away in a central computing facility, these shared-memory processors are often viewed as a logical extension of the desktop. These systems run the same operating system (UNIX or NT) as the desktop and many of the same applications from a workstation will execute on these multiprocessor servers.

Typically a workstation will have from 1 to 4 processors and a server system will have 4 to 64 processors. Shared-memory multiprocessors have a significant advantage over other multiprocessors because all the processors share the same view of the memory, as shown in here<sup>1</sup>.

These processors are also described as *uniform memory access* (also known as UMA) systems. This designation indicates that memory is equally accessible to all processors with the same performance.

The popularity of these systems is not due simply to the demand for high performance computing. These systems are excellent at providing high throughput for a multiprocessing load, and function effectively as high-performance database servers, network servers, and Internet servers. Within limits, their throughput is increased linearly as more processors are added.

In this book we are not so interested in the performance of database or Internet servers. That is too passé; buy more processors, get better throughput. We are interested in pure, raw, unadulterated compute speed for *our* high performance application. Instead of running hundreds of small jobs, we want to utilize all \$750,000 worth of hardware for our single job.

The challenge is to find techniques that make a program that takes an hour to complete using one processor, complete in less than a minute using 64 processors. This is not trivial. Throughout this book so far, we have been on an endless quest for parallelism. In this and the remaining chapters, we will begin to see the payoff for all of your hard work and dedication!

The cost of a shared-memory multiprocessor can range from \$4000 to \$30 million. Some example systems include multiple-processor Intel systems from a wide range of vendors, SGI Power Challenge Series, HP/Convex C-Series, DEC AlphaServers, Cray vector/parallel processors, and Sun Enterprise systems. The SGI Origin 2000, HP/Convex Exemplar, Data General AV-20000, and Sequent NUMAQ-2000 all are uniformmemory, symmetric multiprocessing systems that can be linked to form even larger shared nonuniform

<sup>\*</sup>Version 1.3: Aug 25, 2010 11:16 am -0500

<sup>&</sup>lt;sup>†</sup>http://creativecommons.org/licenses/by/3.0/

<sup>&</sup>lt;sup>1</sup>"Shared-Memory Multiprocessors - Symmetric Multiprocessing Hardware", Figure 1: A shared-memory multiprocessor <http://cnx.org/content/m32794/latest/#id1165410787491>

memory-access systems. Among these systems, as the price increases, the number of CPUs increases, the performance of individual CPUs increases, and the memory performance increases.

In this chapter we will study the hardware and software environment in these systems and learn how to execute our programs on these systems.