LA-UR- 95-3110 CONF-960415--1

TITLE: INSTALLATION OF MCNP ON 64-BIT PARALLEL COMPUTERS

AUTHOR(S): Alan B. Meginnis
John S. Hendricks
Gregg W. McKinney

SUBMITTED TO: 1996 Radiation Protection and Shielding Topical Meeting "Advancements in Radiation Protection and Shielding," April 21-25, 1996, N. Falmouth, MA

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

FORM NO 836 R4
ST NO 2629 5/81

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Installation of MCNP on 64-bit Parallel Computers

Alan B.Meginnis
John S. Hendricks
Gregg W. McKinney
Applied Theoretical and Computational Physics Division
Los Alamos National Laboratory
Los Alamos, NM 87545

September 5, 1995

1 Introduction

Scientific computing is evolving toward 64-bit workstations with multiple processors. We report here for the first time the results of porting a major radiation transport code, MCNP^{TM}, to these computer architectures of the future. In this effort we have

- Enabled dynamic memory allocation for MCNP on 64-bit unix workstations;
- Determined that there is no advantage to 64-bit single precision mode over 32-bit double precision mode for MCNP;
- Gained significant performance improvements by running in parallel on multiple processors of a single workstation.

MCNP [1] is a general-purpose Monte Carlo N-Particle radiation transport code which transports neutrons, photons, and electrons in either continuous or multigroup/adjoint mode in three dimensions. MCNP is used for radiation shielding and protection, criticality safety, medical applications from boron neutron capture therapy to positron emission tomography, aerospace, physics, defense, environment, and other applications ranging from drug interdiction to bulk materials processing.

2 Dynamic Memory Adjustment for 64-bit Architectures

Until now we have been unable to implement dynamic memory allocation for MCNP on 64-bit workstations. A fixed-length memory had to be assigned at compilation time. Dynamic memory was only possible on 64-bit mainframe architectures such as Cray and on unix workstation 32-bit architectures.

In porting MCNP to a 64-bit 4-CPU Silicon Graphics, Inc., (SGI) computer [3] [4] [5] at the National Institute of Standards and Technology (NIST) we determined that the impediment to dynamic memory allocation is a data size mismatch occurring when passing a FORTRAN variable to a C subroutine and catching it with an int type variable. Other codes that combine both FORTRAN and C and presently run on 64-bit mainframes or 32-bit workstations are also likely to encounter the data mismatch problem when being installed on the new 64-bit workstations.

MCNP is a trademark of the Regents of the University of California, Los Alamos National Laboratory.

Careful attention must be paid to the size and type of variables passed, particularly pointers which have a different default size on 64-bit workstations. Correcting the Fortran - C data mismatch in MCNP also made dynamic memory allocation possible for the first time on the DEC Alpha 64-bit workstation. Now dynamic memory works for MCNP on virtually all mainframes and unix workstations.

3 Implications of 64-bit Architectures

A code that needs 64-bit precision will run faster on a 64-bit architecture than it will on a 32-bit architecture because the conversion from 4-bytes to 8-bytes is handled in software (or microcode), not directly in hardware, assuming that the compiler knows about the 64-bit floating-point instructions in a 64-bit architecture. A code that needs only 32-bit precision will usually run a bit faster on a 64-bit architecture than on a 32-bit architecture because, even though there are 64 bits available for computing and storing each word, only 32-bits are used, so there is less work to do.

We hoped that the 64-bit architecture platform would increase the efficiency of MCNP by eliminating the need for double precision data storage and arithmetic. It did not. We found that the 64-bit architecture ran MCNP problems equally fast regardless of how the 64-bit accuracy was achieved, that is, whether we used (1) no implicit double precision declaration and a compiler option to use the 64-bit word length; or (2) an implicit double precision declaration; or (3) real*8 declaration. This observation was made on both the 64-bit SGI and DEC workstations.

Thus we observed no advantage for MCNP in going from a double-precision 32-bit architecture code to a single-precision 64-bit architecture code other than the inherent improvement for all codes in going from 32-bit to 64-bit architectures.

4 Multitasking with PVM

The most important advantage of the new workstation architectures is the availability of multiple processing units. Significant performance gains were achieved by using MCNP's Parallel Virtual Machine (PVM) capabilities to multiprocess on the 64-bit SGI machine. PVM is a software package out of Oak Ridge National Laboratory that allows a heterogeneous network of parallel and serial computers to appear as a single concurrent computational resource [2] [6]. The PVM utility allows one to execute a problem simultaneously across several independent computers or to multiprocess a problem on a single machine with multiple processors. The NIST machine was equipped with four R8000 SGI processors. MCNP has also been ported to a Sun Sparc10 with four multiple processors.

On clusters of independent computers, PVM identifies individual machines and spawns tasks to specific processors to ensure that the number of tasks spawned is equal to the number of available machines. Unfortunately, on multiple-processor machines identifying individual processors is not possible. No matter how many processors a machine may have, PVM only recognizes the machine as having one processor. So, the only course of action with a multiple-CPU machine is to request PVM to spawn as many tasks to the machine as processors and hope that the machine is smart enough to send one to each processor. There is no guarantee that this will happen. In fact it is possible to actually get any combination from one task on each processor to all tasks on a single processor.

In our tests with the NIST multiuser SGI computer the use of PVM with four processes spawned increased the speed of processing by a factor of 1.8 to 1.9. These tests were done with a normal load on the machine. Theoretically if no other load is on a machine, the speed of processing should increase nearly proportionately with each additional processor used [7] [8]. So if four processors are used on a dedicated system MCNP should run nearly four times faster. We have also observed this on a 4-CPU Sun Sparc10 where in a dedicated environment MCNP ran nearly four times faster.

5 Conclusion

The Monte Carlo radiation transport code MCNP has been successfully ported to two 64-bit workstations, the SGI and DEC Alpha. We found the biggest problem for installation on these machines to be Fortran and C mismatches in argument passing. Correrction of these mismatches enabled, for the first time, dynamic memory allocation on 64-bit workstations. Although the 64-bit hardware is faster because 8-bytes are processed at a time rather than 4-bytes, we found no speed advantage in true 64-bit coding versus implicit double precision when porting an existing code to the 64-bit workstation architecture. We did find that PVM multitasking is very successful and represents a significant performance enhancement for scientific workstations.

6 Acknowledgement

References

[1] J. F. Briesmeister, Ed., "MCNP - A General Monte Carlo N-Particle Transport Code, Version 4A," Los Alamos National Laboratory report, LA-12625 (1995)

[2] A. Geist, A. Beguelin, "PVM3 User's Guide and Reference Manual," Oak Ridge National Laboratory, report ORNL/TM-12187 (Sept. 1994)

[3] "Preparing for the 64-bit Future of IRIX," System Software Marketing, Computer Systems Group, SGI.

[4] MIPSpro 64-bit Porting and Transition Guide, SGI

[5] Harvey Wasserman, "Benchmark Tests on a Silicon Graphics R8000-Based Workstation," Los Alamos National Laboratory, to be published (8/14/95)

[6] Richard Barrett, "Using PVM at LANL," Los Alamos National Laboratory Los Alamos NM, to appear as LANL technical report, May 12, 1995.

[7] G. W. McKinney, "Parallel Processing Monte Carlo Radiation Transport Codes," Proceedings of the 8th International Conference on Radiation Shielding, Arlington, Texas, American Nuclear Society (April 1994)

[8] G. W. McKinney et al., "Multiprocessing MCNP on an IBM RS/6000 Cluster," Trans.Am.Nucl. Soc., 68, 212 (1993)