

## Speedup of Particle Transport Problems with a Beowulf Cluster

Zhongxiang Zhao† and G. Ivan Maldonado

Department of Mechanical, Industrial and Nuclear Engineering  
University of Cincinnati, PO Box 210072, Cincinnati OH 45221-0072

---

**Abstract:** The MCNP code is a general Monte Carlo N-Particle Transport program that is widely used in health physics, medical physics and nuclear engineering for problems involving neutron, photon and electron transport<sup>[1]</sup>. However, due to the stochastic nature of the algorithms employed to solve the Boltzmann transport equation, MCNP generally exhibits a slow rate of convergence. In fact, engineers and scientists can quickly identify intractable versions of their most challenging and CPU-intensive problems. For example, despite the latest advancements in personal computers (PCs) and quantum leaps in their computational capabilities, an ordinary electron transport problem could require up to several CPU-days or even CPU-weeks on a typical desktop PC of today. One common contemporary approach to help address these performance limitations is by taking advantage of parallel processing. In fact, the very nature of the Monte Carlo approach embedded within MCNP is *inherently parallel* because, at least in principle, every particle *history* can potentially be tracked individually in an independent processor. In practice, however, there are many issues that must be confronted to achieve a reasonable level of parallelization. First, of course, a suitable parallel computing platform is required. Next, the computer program itself should exploit parallelism from within by combining such tools as Fortran-90 and PVM, for example. This article describes the installation and performance testing of the latest release of MCNP, Version 5 (MCNP5), compiled with PGI Fortran-90 and with PVM on a recently assembled 22-node Beowulf cluster that is now a dedicated platform for the faculty and students of the University of Cincinnati's Nuclear and Radiological Engineering (UCNRE) Program. The performance of a neutron transport problem and that of a more challenging gamma-electron (coupled) problem are both highlighted. The results show that the PVM-compiled MCNP5 version with 20 tasks can execute roughly 12 times faster than the sequential version of the tested neutron transport problem, whereas another increasingly challenging gamma-electron (coupled) dose problem executed nearly 18 times faster than its serial counterpart, this performance on 20 processors is an impressive result relative to a linear (ideal) speedup.

**Kew words:** Beowulf cluster, speedup, particle transport problems

---

### INTRODUCTION

**Overview of the latest features of MCNP:** Version 5 of MCNP is the latest release of this code that includes some of the most significant (and modern) advancements to this program that is maintained by Group X-5 at Los Alamos National Laboratory (LANL). In recent times, MCNP5 has been tested at LANL as well as at other laboratories and institutions and it is increasingly being used on PCs, Linux clusters and Unix-based ASCII computers. The code was modernized to meet requirements of portability and standards-based coding (F90+MPI+OpenMP and PVM). As described by Dr. Forrest Brown, the Lead Scientist of the X-5 Monte Carlo Team, the effort to modernize MCNP was driven by the need to: adopt modern practices for software engineering and software quality assurance, adhere to current standards for

Fortran-90 and parallel processing, preserve all existing code capabilities, add flexibility for rapid introduction of new features and adaptation to advanced platforms<sup>[2]</sup>.

**Description of the UCNRE beowulf cluster:** Until recently, researchers at the UCNRE program could only execute serial versions of MCNP on personal PCs, as available. However, earlier this year the program assembled a 22-node Beowulf Cluster, which is a high-performance, scalable and potentially massively parallel computer built from off-the-shelf components and running the freeware operating system Linux. In essence, it is a cluster of PCs interconnected by a private high-speed network that can be dedicated to running high-performance computing tasks.

The UCNRE Beowulf cluster includes one server node and ten computing nodes. The server node is

---

**Corresponding Author:** G. Ivan Maldonado, Department of Mechanical, Industrial and Nuclear Engineering, University of Cincinnati, PO Box 210072, Cincinnati OH 45221-0072

† **Current Affiliation:** Zhongxiang Zhao, Aerospace Engineering Department, United States Naval Academy, 590 Holloway Road, Rickover Hall - MS 11B, Annapolis MD 21402-5042

equipped with dual Intel Xeon processors with 512k L2 cache at 2.4 GHz, a 2 Gigabyte PC2100 ECC RAM and a 73 GB 10K RPM Seagate SCSI hard drive. Each computing node has dual Xeon processors at 2.4 GHz, 1 GB RAM and a 40 GB IDE drive. The Supermicro Super X5DPI-G2 motherboard is used for all nodes, which features an Intel E7501 chipset and supports two CPUs with an up to 533 MHz Front Side Bus and a dual port Intel 82546EB Gigabit Ethernet Controller. Nodes are connected with two Asante Gigabit switches. The operating system Red Hat 9 (Linux kernel 2.4.20) is installed on the server node, with the file system residing only on the server node. Thus, the computing nodes mount the file system through the TFTP service and the local hard drive on each node is mounted as a temporary directory.

The off-the-shelf components for this *poor man's supercomputer* required a capital investment of approximately \$22,000, or roughly \$1,000 per node, which is consistent with the statistics reported by T. Sterling for parallelism of  $<100^{[3]}$ . However, it should be noted that this price tag rudely ignores the combined labor of a professor, talented graduate students and an exceptional departmental technician, all of whom handled the details of hardware and software acquisitions, installations and the troubleshooting that followed for several weeks and months.

**Software installation:** A Fortran 90 compiler is needed to compile MCNP5. In fact, according to the MCNP development team at LANL, Fortran 90 compilers like Absoft Pro-8.0 QF3, Lahey Pro\_6.1e and Portland Group PGF90 4.0-2 have been successfully tested on a Linux platform to compile MCNP5. Our preliminary compilations using the trial versions of the above compilers showed that, in a sequential mode, the executable code built by the PGI compiler was the fastest to execute. Therefore, we selected the PGI compiler for this application.

In addition to the PGI F90 compiler, the PVM 3.4.4 package, GCC C compiler, GNU make and X11 data libraries are also needed. The PGI compilers can be downloaded from the Portland Group's web site and the package we downloaded was the PGI workstation 4.0-3 that we could install thanks to the shared license that was obtained via a grant from the Ohio Supercomputer Center.

The source code of PVM 3.4.4 was downloaded via the Oak Ridge National Laboratory (ORNL) web site. The installation of PVM is prescriptive and straightforward, thus, after setting the PVM environment variables, the next step was to install MCNP5. The source code of MCNP5 1.14 first needs to be patched up to version 1.20 prior to installation. The installation script of MCNP5 is menu-driven, so within the multiprocessing options, "PVM with no shared memory" was selected to compile a PVM version. The number of PVM processes was set to 20, the Fortran 90

compiler was set to PGI F90 and the C compiler was set to GCC. Finally, it is also necessary to enter the appropriate directory paths to the MCNP5 cross section data and to the X11 graphics library. Both, the PVM (parallel) and sequential modes of MCNP5 were compiled.

**Performance test 1: neutron transport problem:** A neutron transport problem was used to test the parallel execution of MCNP5 and to verify the proper installation of all the supporting hardware and software of the Beowulf Cluster. In this particularly and purposely, challenging particle tracking problem, a californium-252 point neutron source is located 23 cm away from a planar polyethylene disk of 100 cm in diameter and 1.6 cm in thickness. A 60  $\mu\text{m}$  gadolinium foil is attached to the backside of the disk and so the neutron-gamma ( $n, \gamma$ ) "radiation capture" reaction in the gadolinium foil is simulated. The resulting reaction rate for this arrangement was of about  $6.54 \times 10^{-4}$  reactions/particle/cm<sup>3</sup>, which is notably low, thus the challenge involved. A side view of the geometry of this problem is schematically shown in Fig. 1, which is not drawn to scale.

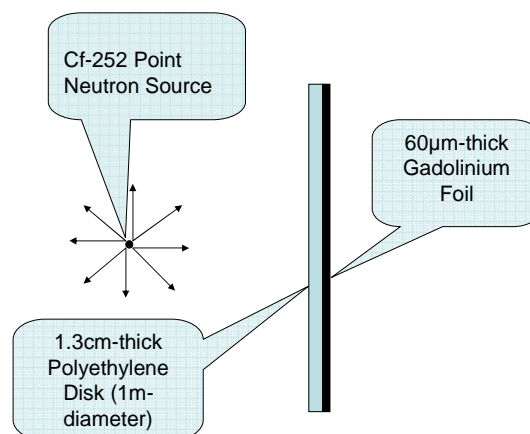


Fig. 1: Schematic of neutron-gamma problem being simulated

First, the sequential execution of MCNP5 was tested with the same input file to establish a reference point. The number of particle histories (nps) for this problem was set to 200,000,000 (two hundred million). Then MCNP5 was run in a parallel mode by assigning an increasing number of PVM tasks, from 2 to 20. The computing rate, which is the number of particle histories divided by computation time, was calculated as a function of PVM tasks was calculated and is plotted in Fig. 2 alongside the computing rate for the sequential execution.

As expected, the results in Fig. 2 shows that the computational speed increases with PVM tasks. In fact, at the upper end with 20 PVM tasks, the parallel execution of MCNP5 was 11.5 faster than the

sequential mode, in other words, a one order of magnitude increase in computational efficiency was achieved. A plot of the *linear* speedup increase (based on the computing rate of the sequential mode) is

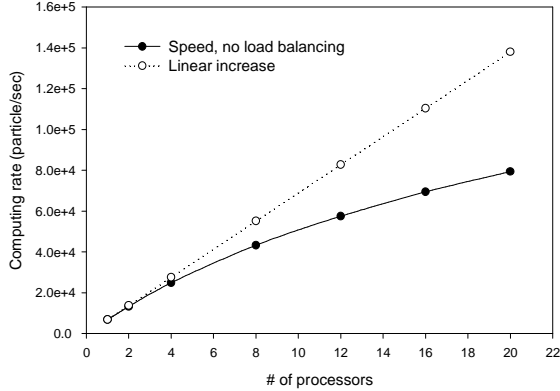


Fig. 2: Neutron transport problem performance (nps = 200,000,000)

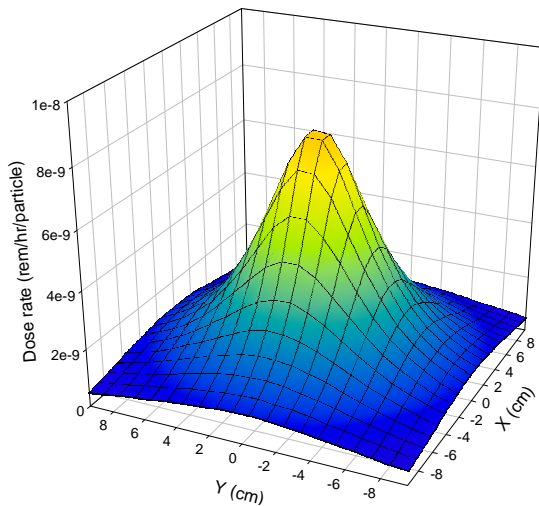


Fig. 3: Dose distribution due to point source

included in the plot for contrasting purposes. During the tests performed, MCNP5 was the only active running process on the Beowulf Cluster, thus, the wall clock time was recorded as the actual computing time. Given the fact that all cases tracked the same total number of particles, these cases all achieved the same level of statistical variance as one another.

**Performance test 2: Gamma-electron coupled problem:** In this performance test, we simulated the gamma dose distribution on a surface at a depth of 3 cm on a 20×20×10 cm<sup>3</sup> water volume. Two types of sources were employed in this problem; a 1 MeV isotropic gamma point source centered within the 20×20 cm<sup>2</sup> plane and located 1 cm above the water surface; and a 1 MeV gamma beam source of cross sectional diameter  $\phi=2$  cm that is centered within and

impinging on the surface of the water volume. For this problem, a new MCNP5 tally specification feature, the FMESH card, was employed to facilitate three-dimensional graphic rendering of the results.

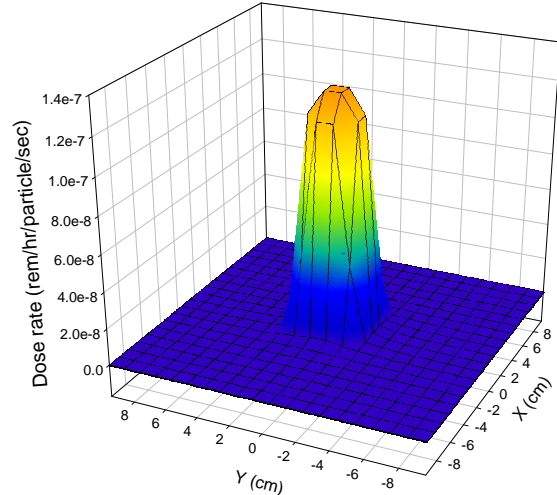


Fig. 4: Dose distribution due to beam source

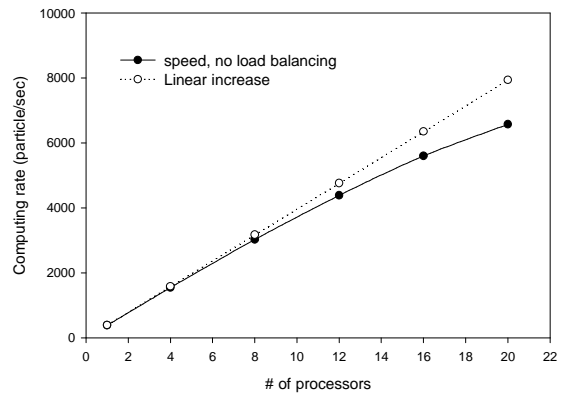


Fig. 5: Gamma-electron coupled problem performance (nps=2,000,000)

For this problem, the water volume was divided into 20×20×3 boxes, so that the radiation dose on the surface can be reported as a 20×20 matrix. What makes this problem computationally challenging is that gamma rays lead to photoelectric and Compton reactions, which generate electrons, so in a “coupled” photon-electron problem, MCNP5 is instructed to follow all photons and electrons produced from their birth until their “death,” which can be by absorption, leakage out of the water volume, or by slowing down below the lower energy cut off of 0.001 MeV.

Figure 3 and 4 are three-dimensional representations of the dose distributions due to the point source and beam source, respectively. As expected, it can be observed that although the point source is only 1 cm from the water surface, it yields a much larger exposure area relative to the beam source, whose

exposure area is limited to the beam's cross-sectional area ( $\phi=2$  cm).

With regard to computational performance, the simulation of the beam source case, which on a single CPU required approximately 2,100 minutes, was able to complete within 112 minutes on 20 PVM tasks on the Beowulf Cluster tasks. In other words, the computing rate of MCNP5 with 20 PVM tasks was 17.8 times faster than the equivalent single-CPU case. Figure 5 tabulates the parallel performance as a function of PVM tasks and contrasts it to an ideal linear speedup.

### CONCLUSION

MCNP5 was successfully compiled with the Portland Group F90 and PVM. Performance tests were carried out on a Beowulf cluster that was recently assembled by the UCNRE program<sup>[5]</sup>. The performance tests showed speedups ranging between 12 and 18, for the cases examined. Thus, at least a one-order of magnitude improvement has been achieved in computational speed for particle transport problems run with MCNP5 and with a relatively small capital investment. This clearly enables students and researchers in the program to pursue considerably larger and more complex particle transport applications than in the recent past.

### ACKNOWLEDGMENTS

The authors would like to acknowledge and thank the support and software licenses received from the Ohio Supercomputer Center. Likewise, the hard work by Prakash Velayutham, Vineet Mokashi and Larry Schartman from the Mechanical, Industrial and Nuclear Engineering Department is herein acknowledged.

### REFERENCES

1. Brown, F.B., *et.al*, 2003. X-5 Monte Carlo Team, Los Alamos National Laboratory. MCNP-A General Monte Carlo N-Particle Transport Code. Version 5.
2. Brown, F.B., 2003. MCNP5: Parts I-IV. MCNP5 DOE 2000 Electronic Notebook.
3. Sterling, T., 2002. Beowulf Cluster Computing with Linux. The MIT Press, Cambridge, Massachusetts.
4. Zhao, Z. and G.I. Maldonado, 2005. Combining a beowulf cluster and MCNP5 to solve particle transport problems. *Trans. Am. Nucl. Soc.*, 92: 522-524.