UCRL-JC-125485 Sum

ON ESTABLISHING BENCHMARK STANDARDS FOR PARALLEL PROCESSING MONTE CARLO CODES

H.P. Alesso, C.E. Annese and S. Murty
Lawrence Livermore National Laboratory
Livermore, California 94550

SUMMARY

BACKGROUND

Assurance of subcriticality is the most important element in any nuclear operation. Today, this assurance is provided by using analytical computational tools to evaluate and analyze nuclear processes under all possible scenarios and geometry. The reliability of the evaluation results depends upon how accurately the computational tool represents the realistic condition of the operation in question. Proof of the accuracy of the computational tool also depends upon the proper benchmarking of the code against actual nuclear fission process experiments.

The principle limitation in using the Monte Carlo method for solving many complex shielding and criticality problems has been the computer time required. Parallel processing offers a vast saving in computational time that will likely become the general practice in the very near future.

Parallel applications of Monte Carlo codes have been successfully implemented in SIMD (single-instruction, multiple-data) and MIMD (multi-instruction, multi-data) architectures. Particle transport Monte Carlo was first applied to SIMD in 1981. It required changing the history-based scheme to an event-based scheme. However, the natural partitioning by particles is inherently parallel and ideally suited to MIMD parallel computers. The advent of multiple distributed computer software packages, such as, PVM (Parallel Virtual Machine) allows workstations, mainframes and SIMD computers to work together on a single job as a virtual supercomputer. This virtual supercomputer relies on relatively slow networks to pass information back and forth between processors,

but Monte Carlo codes rely on relatively little communication in comparison to computation. It is important to realize there are no algorithmic differences between Monte Carlo on MIMD computers and on distributed computers. Forward looking approaches today, seek to leverage the rapid advances in processor design, Internet connectivity, and the implementation of distributed computing embodied in such languages as Java.

For several years, we have been conducting criticality benchmarking calculations [1-3] of the LLNL developed Monte Carlo particle transport code COG.[4] As part of the benchmarking process, we have compared COG calculations to both critical experiments and other code calculations including MCNP, KENO, SAN, and MORSE (for various cross-section libraries including ENDL and ENDF/B).

In 1994, a version of COG was developed that executed concurrently on a set of networked workstations, or on a MPP, under the PVM control system. This version of COG runs a set of computational nodes using a master-slave relationship. The user starts COG as a master process. Using PVM message-passing and control system, the master COG starts up slave COG processes on specified workstations.

For shielding problems, each slave obtains a starting random number seed, a number of particles to run, and a reporting time interval. The slave reports its results to the master and is restarted as needed. Criticality problems have a different organization. Because the fission created in one batch are the neutron source for the next batch, the source evolves until it reaches an equilibrium source distribution. Then settling batches are run. These methods are efficient and robust.

The process of criticality benchmarking has traditionally included evaluating errors in the areas of; (1) the model, (2) the Monte Carlo code, (3) the point-wise data. With parallel processing, new issues arise including: (1) random number sequence cutting[5], (2) round-off error of individual CPUs, (3) statistical ($n^{-0.5}$) accuracy[6], (4) batch routing and timing, and (5) variance of the variance limitation.

OBJECTIVE

The objective of this paper is to report on COG results in the PVM parallel mode for criticality benchmark experiments and to use this experience to: (1) establish a set of

parallel benchmarks and (2) set standards for benchmarking methodology and results-reporting together with a controlled database .

RESULTS

Our results to date have demonstrated good compliance of COG in parallel mode to COG and MCNP in scalar mode. Also, we have begun to develop a criteria for benchmarking methodology for parallel mode criticality applications of Monte Carlo codes.

REFERENCES

1. Alesso, H.P., Pearson, J.S., and Choi, J.S., Criticality Benchmarks for COG: A New Point-Wise Monte Carlo Code, 9th International Symposium on Packaging and Transportation of Radioactive Materials (PATRAM), Washington, DC, June 11-16, 1989.

2. Alesso, H.P., Pearson, J.S., Nguyen, D.H., and Choi, J.S., Status of Benchmarks for Criticality Calculations Involving Fission Product Poisons and Special Actinides in Spent Fuel, RAMTRAN, Vol. 1, No. 1, pp 25-32, 1990.

3. Alesso, H.P., Annese, C.E., Buck, R., Lloyd, W., and Pearson, J.S., Monte Carlo Criticality Benchmark: Comparisons for Waste Packaging, Institute of Nuclear Materials, Safeguards, Transportation Waste Management, Orlando, Florida, July 19-22, 1992.

4. Wilcox, Jr., T.P., and Lent, E.M., COG: Particle Transport Code Designed to Solve the Boltzman Equation for Deep Penetration (Shielding) Problems, LLNL, April, 1989.

5/. W. R. Martin, J. A. Rathkopf, and F. B. Brown, The Impact of Advances in Computer Technology on Particle Transport Monte Carlo, UCRL-JC-109525, LLNL, Jan. 21, 1992.

6/. J. Hall, Monte Carlo Modeling of Neutron and Gamma-Ray Imaging Systems, COG User's Quarterly, Vol. 2 , No. 1, LLNL, March 1, 1996.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore Naitonal Laboratory under contract No. W-7405-Eng-48.