ANL/RA/SUMM--88954
CONF-960616--35-Samm.

A summary to be submitted to the ANS Summer Meeting
Reno, NV, June 16 - 20, 1996

Performance Studies of the Parallel VIM Code

by

Baolan Shi and Roger N. Blomquist

Reactor Analysis Division
Argonne National Laboratory
9700 South Cass Avenue
Building 208
Argonne, IL 60439
(708) 252-9676

* Work supported by the U. S. Department of Energy, Nuclear Energy Programs under Contract W-31-109-ENG-38.

Performance Studies of the Parallel VIM Code

Baolan Shi and Roger N. Blomquist
Reactor Analysis Division, Argonne National Laboratory, 9700 South Cass Avenue,
Argonne, IL 60439

1. Introduction

In this paper, we evaluate the performance of the parallel version of the VIM Monte Carlo code[1] on the IBM SPx at the High Performance Computing Research Facility at ANL. Three test problems with contrasting computational characteristics were used to assess effects in performance. A statistical method for estimating the inefficiencies due to load imbalance and communication is also introduced.

VIM is a large scale continuous energy Monte Carlo radiation transport program and was parallelized using history partitioning, the "master/worker" approach, and p4 message passing library[2]. Dynamic load balancing is accomplished when the master processor assigns "chunks" of histories to workers that have completed a previously assigned task, accommodating variations in the lengths of histories, processor speeds, and worker loads. At the end of each batch (generation), the fission sites and tallies are sent from each worker to the master process, contributing to the parallel inefficiency. All communications are between master and workers, and are serial. The speedups of the code on the RA workstation network and the former IBM SP1 parallel supercomputer with up to 40 processors were previously evaluated[3].

The SPx is a scalable 128-node parallel supercomputer with high-performance Omega switches of 63 μsec latency and 35 MBytes/sec bandwidth[4]. For uniform and reproducible performance, we used only the 120 identical regular processors (IBM RS/6000) and excluded the remaining eight planet nodes, which may be loaded by other's jobs.

2. Speedup Measurements

The three test problems covered a range of computational intensity (computation to communication ratio). The reference problem was a realistic thermal TREAT reactor mockup with 3.41 MBytes of tally data/batch and 400 histories/batch/processor. The same problem was repeated with 4,000 histories/batch/processor. Finally, a fast bare system with 40.13 kBytes of tally data/batch and an unrealistically low computational intensity (12 times lower) tested the inefficient extreme.

All tests were run with batch sizes proportional to the number of workers (N). Each chunk had 100 histories, and each run had three batches. To test the reproducibility of speedups, three identical runs were taken for each number of workers. The speedups (S(N)) were calculated by $S(N)=N\times T_1/T_N$[3]. Here, T_N is the wall clock execution time for the scaled problem

with N workers.

3. Inefficiency Estimation

The total inefficiency, *ineff*, is inferred from speedup measurements as *ineff=(N-S(N))/N*. It can be caused by communication, load imbalance among the workers (uneven work assignments or unequal external processor loads), and other factors. VIM does not measure the time spent by each worker, so a statistical method was developed to estimate the inefficiency caused by load imbalance alone.

Assuming the (i)th worker finishes the batch within time period t_i and the slowest process took a maximum time period of t_{max}, the load imbalance inefficiency can be expressed as

$$ineff_{load} \approx 1 - \frac{\sum_{i=1}^{N} t_i}{t_{max} N}$$

t_i and t_{max} were estimated statistically from a empirical distribution of batch cpu times compiled from surrogate serial calculations. In our study, the load imbalance is mainly caused by the statistical nature of Monte Carlo because the SPx workers are identical and because the queuing system reserves the processors for a single job. Consequently, the variability of batch computation times observed in a serial calculation is analogous to the variability among the workers in a parallel calculation. The probability that a processor consumes less time than t_{max} is calculated as $P(t<t_{max})=(N-1)/N$. Once P is obtained, t_{max} is extracted from the sample order statistics[5]. The total and load imbalance inefficiencies for two problems with 400 histories/batch were evaluated.

4. Results, Discussion, and Conclusions

As shown in Figure 1, for the TREAT reactor problem, the 120-node parallel speedup improves significantly from 60 to 80 when the batch size per worker is increased from 400 to 4,000. The speedup also increases monotonically with N, but for the less computationally intensive bare reactor problem, the speedup peaks at 20 workers. As shown in Figure 2, the load imbalance is much more severe in the bare reactor problem. This is consistent with our observations of larger variations in the accomplished histories/worker. With 120 workers, the total inefficiency reaches 99% for the bare reactor problem but is less than 52% for the TREAT reactor problem. This points out the importance of selecting problem parameters to ensure domination by computation.

References

1. Blomquist, R. N., "VIM-A Continuous Energy Monte Carlo Code at ANL", *A Review of the Theory and Application of Monte Carlo Methods*, Proceedings of a Seminar-Workshop, ORNL/RSIC-44, April 21-23, 1980.

2. Butler, R. and Lusk, E., *User's Guide to the p4 Parallel Programming System*, ANL-92/17, April, 1994.

3. Matsuura, S., Brown, F. B., and Blomquist R. N., "Parallel Monte Carlo Eigenvalue Calculations", *1994 ANS Winter Meeting*, Washington D. C., Nov. 13-17, 1994.

4. Gropp, W. and Lusk, E., *User's Guide for the ANL IBM SPx Draft*, ANL/MCS-TM-00, February 20, 1995.

5. Hogg, R. V. and Tanis, E. A., *Probability and Statistical Inference*, Macmillan Publishing Company, 4th ed., New York, 1993.

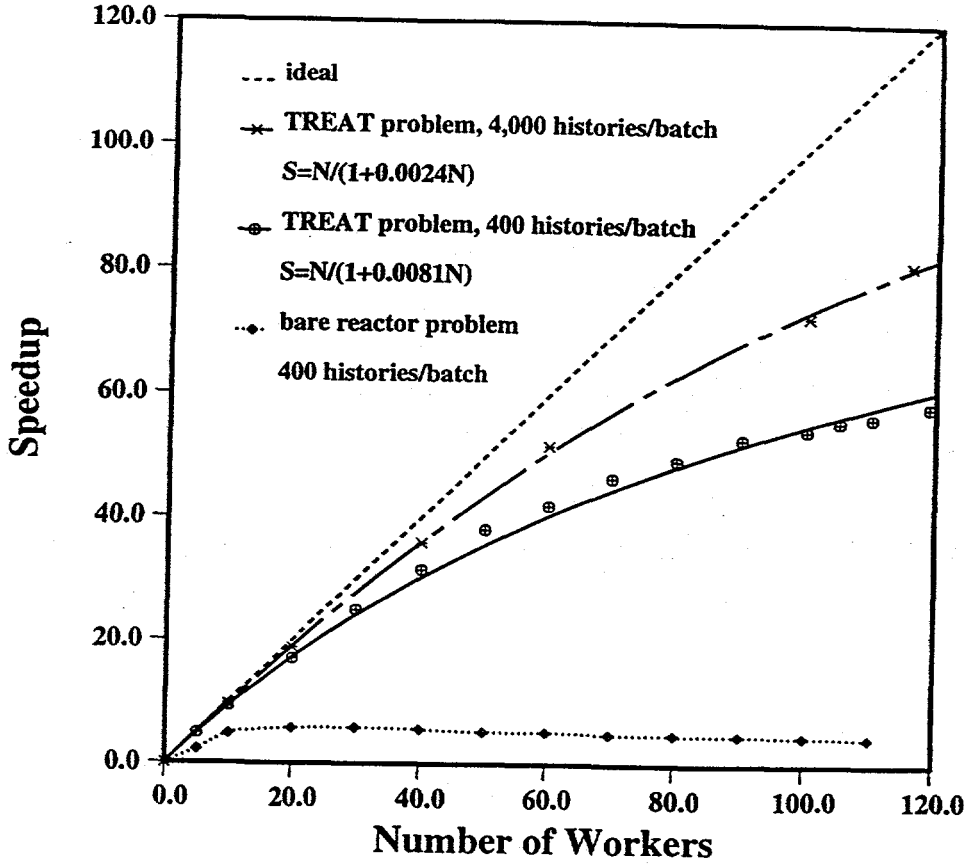

Figure 1. Speedups measured on SPx.

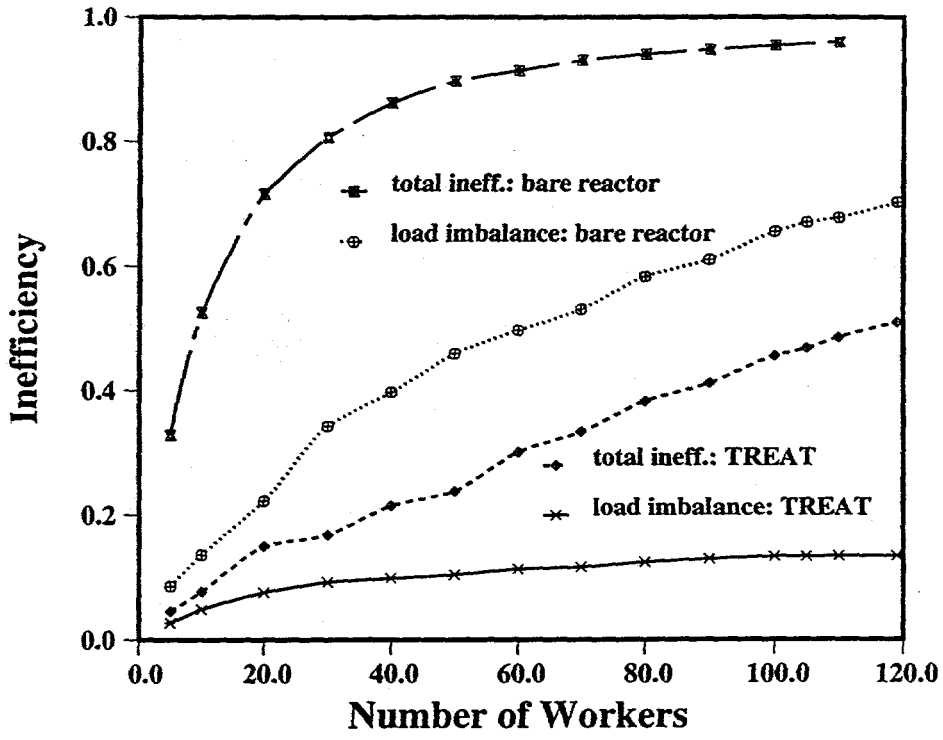

Figure 2. Inefficiency vs. number of workers.