

CONF-961103--29

Paper to be submitted to the 1996 International Conference and Embedded Topical Meeting,
November 10-15, 1996, Washington, D.C.

ANL/RA/CP--90183

RECEIVED

APR 14 1997

OSTI

Development of a Parallelization Strategy for the VARIANT Code

by

U.R. Hanebutte, H.S. Khalil, G. Palmiotti

Reactor Analysis Division
Argonne National Laboratory
Argonne, IL 60439

and

M. Tatsumi

Department of Nuclear Engineering
Osaka University
2-1, Yamadaoka, Suita, Japan

MASTER

The submitted manuscript has been created by the University of Chicago as Operator of Argonne National Laboratory ("Argonne") under Contract No. W-31-109-ENG-38 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, non-exclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED *ph*

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

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

**Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.**

Development of a Parallelization Strategy for the VARIANT Code

U.R. Hanebutte, H.S. Khalil, G. Palmiotti

Reactor Analysis Division
Argonne National Laboratory
Argonne, IL 60439

and

M. Tatsumi

Department of Nuclear Engineering
Osaka University
2-1, Yamadaoka, Suita, Japan

The VARIANT code^{1,2} solves the multigroup steady-state neutron diffusion and transport equation in three-dimensional Cartesian and hexagonal geometries using the variational nodal method. VARIANT consists of four major parts that must be executed sequentially: input handling, calculation of response matrices, solution algorithm (i.e. inner-outer iteration), and output of results. Within a single outer-iteration, odd-even sweeps are performed along the z-direction combined with an n-color iteration scheme for each x-y plane. The outer-iteration may be accelerated by a coarse mesh rebalance (CMR) procedure, which is performed after each outer-iteration step. The objective of the parallelization effort was to reduce the overall computing time by distributing the work of the two computationally intensive (sequential) tasks, the coupling coefficient calculation and the iterative solver, equally among a group of processors. Further, the parallelization strategy must ensure that the convergence properties of the iterative solver are not affected by the parallelization. Another objective was to preserve the original form (including data structure) of the code. Dependent on the given problem, either the coupling coefficient calculation or the iterative solver requires most of the computing time. To obtain maximum efficiency, the number of processors applied in the two parallel tasks should be allowed to differ. Due to the two-cyclic iteration scheme, an odd/even pair of planes is assigned to one processor. Typically, the parallel VARIANT code may utilize up to a few tens of processors. Examples of alternative, massively parallel response matrix algorithms can be found in Ref.^{3,4}.

A single program multiple data (SPMD) implementation for the parallel VARIANT is obtained by introducing two control-matrices which control the parallel execution of the coefficient calculation and the iterative solver. Each entry of the control-matrix denotes the processor which is responsible for the task associated with the entry. During an initialization step the mapping between processors and tasks is performed, taking into account load balance and communication patterns which minimize message passing overhead. Identical control-matrices are generated on each processor. The coefficient calculations are then performed in parallel as follows. Each processor executes the entire loop over all unique node types. However, only the node types indicated by the control-matrix as being under its responsibility are actually calculated on this processor. For all other node types the complex coefficient calculations are skipped. A data exchange step follows the computation of the coefficients in which each processor obtains the

non-locally computed coefficient data required for its part of the iterative solver. The parallel execution of the iterative solver is orchestrated in a similar fashion.

The parallel implementation of VARIANT is based on the Message Passing Interface (MPI) standard⁵. To date, we have successfully installed and used the parallel VARIANT code on several architectures including the 128 processing node IBM SPx Supercomputer^{6,7}, and heterogeneous networks of Sun and IBM/RS6000 workstations.

Performance studies have been carried out for several published benchmark problems on the IBM SPx system at Argonne National Laboratory. For brevity, only results for an EBR-II 3D hexagonal-z benchmark¹ (Table 1) are given here. The EBR-II problem contains 24 unique nodes, 14 axial planes and 9 energy groups. In the coupling coefficients calculation, data swapping between primary and secondary memory (core-to-disk) occurs due to the large memory requirement, which decreases the efficiency. Utilizing CMR acceleration, the iterative solver performs 17 outer iterations with 70 inner iterations for each outer iteration. The 14-axial node structure of the benchmark guarantees only perfect load balance during the solver stage for parallel execution on 7 processors. The communication costs for both parallel tasks are functions of the frequency and the size of the message exchange. For the first parallel task, an all-to-all data exchange of the calculated node coefficients represents an upper limit for the messages count. The number of message exchanges in the parallel solver is directly related to the number of performed inner iterations. To determine the message sizes for the various data exchanges, one may note the following. To store the coefficients for a unique node type requires 21,632 bytes and 1,392 bytes are needed to transmit the outgoing axial currents at one interface (i.e. between an odd and an even plane).

The performance analysis on the IBM SPx system shows good efficiency for well-load-balanced problems. Even for relatively small problem sizes, respectable efficiencies are seen for the SPx. A higher degree of parallelism can be achieved by assigning a group of processors rather than a single processor to the n-color iteration scheme for each plane. This extension will be addressed in future work.

References

1. Carrico, C.B., Lewis, E.E., Palmiotti, G. (1992), "Three-Dimensional Variational Nodal Transport Methods for Cartesian, Triangular and Hexagonal Criticality Calculations," Nuclear Science and Engineering, 111, 168 (1992).
2. Palmiotti, G., Carrico, C.B., Lewis, E.E. (1993), "Variational Nodal Methods with Anisotropic Scattering," Nuclear Science and Engineering, 115, 233-243 (1993).
3. Hanebutte, U.R, and Lewis, E.E. (1992), "A Massively Parallel Discrete Ordinates Response Matrix Method for Neutron Transport," Nucl. Sci. Eng. 111, 46-56 (1992).
4. Hanebutte, U.R, Laurin-Kovitz, K. and Lewis, E.E. (1992), "Massively Parallel Red/Black Algorithm for x-y-z Response Matrix Equations," Trans. Am. Nucl. Soc. 66, 275 (1992).

5. Gropp, W., Lusk, E., Skjellum, A. (1994), Using MPI, MIT Press, Cambridge, MA, 1994.
6. Gropp, W., and Lusk, E. (1995), "Draft: User's Guide for the ANL IBM SPx," available on the WWW at <http://www.mcs.anl.gov/sp1/guide-r2/guide-r2.html>, Argonne National Laboratory Report (Revised January 1995).
7. Gropp, W., Lusk, E., and Pieper, S.C. (1994) "User's Guide for the ANL IBM SP1," Argonne National Laboratory Report, ANL/MCS-TM-198 (October 1994).

Table 1: Performance of the EBR-II Hexagonal-Z 3D Benchmark on the IBM SPx

Number of Processors	1	2	3	4	5	7
Coef. Calculation (s)	131.04	68.13	47.92	42.27	34.52	28.53
Speed Up Factor		1.92	2.73	3.10	3.80	4.59
Efficiency (%)		96.2	91.2	77.5	75.9	65.6
Solver (s)	836.55	485.48	377.43	289.21	261.05	144.34
Speed Up Factor		1.71	2.22	2.89	3.20	5.80
Efficiency (%)		86.2	73.9	72.3	64.1	82.8
Coef. + Solver (s)	967.59	553.61	425.35	331.48	295.57	172.87
Speed Up Factor		1.75	2.27	2.92	3.27	5.60
Efficiency (%)		87.4	75.8	73.0	65.5	80.0
Total VARIANT (s)	1008.3	593.94	470.57	372.76	332.51	216.97
Speed Up Factor		1.70	2.14	2.70	3.03	4.65
Efficiency (%)		84.9	71.4	67.6	60.6	66.4

Speed Up = time (1 processor) / time (N processors)

Efficiency = Speed Up / Number of processors *100%