INL/CON-05-00521
PREPRINT

Architectural Advancements in RELAP5-3D

ANS 2005 Winter Meeting

George L. Mesina

November 2005

The INL is a
U.S. Department of Energy
National Laboratory
operated by
Battelle Energy Alliance

Architectural Advancements in RELAP5-3D

Dr. George L. Mesina

Idaho National Laboratory, P.O. Box 1625, Idaho Falls, ID, 83415-3890, George.Mesina@inl.gov

INTRODUCTION

As both the computer industry and field of nuclear science and engineering move forward, there is a need to improve the computing tools used in the nuclear industry to keep pace with these changes. By increasing the capability of the codes, the growing modeling needs of nuclear plant analysis will be met and advantage can be taken of more powerful computer languages and architecture. In the past eighteen months, improvements have been made to RELAP5-3D [1] for these reasons. These architectural advances include code restructuring, conversion to Fortran 90, high performance computing upgrades, and rewriting of the RELAP5 Graphical User Interface (RGUI) [2] and XMGR5 [3] in Java.

DESCRIPTION OF THE ACTUAL WORK

Code Restructuring

Code restructuring is accomplished by applying the FOR_STRUCT [4] program. It reengineers the logic paths within subroutines to produce structured code with block-oriented, Fortran 90 constructs. Code restructuring has no impact on the calculated results; however, it does make the coding much easier to read and understand.

Code restructuring is complicated by the fact that RELAP5-3D has both pre-compiler directives (for CPP, OpenMP, Cray, etc.) as well as FORTRAN 90 language items that FOR_STRUCT cannot handle. Ways to work around these limitations in the restructuring tool are being developed, and restructuring is underway. Each modified subprogram is tested by recreating the RELAP5-3D executable and running a small set of test cases. After each group of 5 subprograms is converted, all normal test cases are run. Conversion is deemed successful when output from the modified code is identical to the output of the unconverted code for all test cases.

Conversion to FORTRAN 90

The FORTRAN 90 conversion is another kind of code reengineering. Its ultimate purpose is to keep pace with improvements in the computer industry and to extend the useful life of the program. It replaces many old, antiquated, and obsolescent coding constructs with more modern ones. It introduces modules, the forerunner of classes and object-oriented programming that will be available in FORTRAN 2003. The most fundamental changes being introduced by this effort are:

(1) Replacing all common blocks with modules.
(2) Converting all arrays to allocatable derived types.
(3) Eliminating all equivalence statements.
(4) Use of FORTRAN 90 pointers.

Conversion to FORTRAN 90 has no impact on the calculations; however, it is pervasive and affects tens of thousands of lines of code. This effort is complicated by the fact that a subprogram cannot be converted until all the subprograms in its calling tree are converted first. A number of programs and scripts have been written to automate processes. Testing is carried out exactly as described for code restructuring.

High Performance Upgrades

For the purpose of reducing wall-clock time for large input models, two high-performance computing upgrades have been implemented in RELAP5-3D, namely, conversion to OpenMP parallel and vectorization of key subroutines.

The vectorization work affects only the subprograms that take a significant amount of CPU. Two subroutines, PHANTV and PHANTJ, that typically account for between 10 and 33 percent of the code run time were vectorized. These could not vectorize prior to Cray's development of "streaming." Numerous vector inhibitors were rewritten so that these routines could vectorize [5].

A program was written that creates an OpenMP directive for each do-loop of a given subprogram. The developer must eliminate directives for loops that cannot parallelize and reassign scalar variables to private, shared or recursive clauses as needed. As with FORTRAN 90, all called subprograms must be converted before the calling subprogram can be parallelized.

Java RGUI

The decision to rewrite RGUI in Java was motivated by the success of the Java language. When RGUI was created in 1997, Java was but one of many alternatives and among the least mature. It is now clearly the industry leader and computer industry forecasts indicate that its longevity will be great. The goals of rewriting RGUI is to produce a Java version with the same functionality as the Tcl/Tk version, to create a more modern look and feel, and to respond to user requests for additional or modified features.

The actual rewrite is performed using J-Builder. Java RGUI makes use of widgets, such as multi-document interface, and application software not available in the original freeware language of RGUI.

RESULTS

The code restructuring effort will result in reduced the time and cost to develop the code, debug it, and maintain it.

The FORTRAN 90 conversion will eliminate the fixed memory limitation on input model size and will allow much larger problems to be run than is currently possible.

Initial OpenMP parallel work has been completed and RELAP5-3D runs reliably on one, two, and four processors and produces the exact same calculations on each for all problems in the parallel test set.

The vector improvements do alter answers due to replacement of bisection with algorithms that vectorize, but the changes are within controlled tolerances and speed-ups of over an order of magnitude were achieved in the modified routines with speed-ups of over 50% for RELAP5-3D.

CONCLUSIONS

These architectural changes will extend the lifetime of RELAP5-3D, reduce the costs for development and maintenance, and improve it speed and reliability.

REFERENCES

1. RELAP5-3D CODE DEVELOPMENT TEAM, “RELAP5-3D Code Manual,” INEEL-EXT-98-00834, Revision 2.3, Idaho National Laboratory, Idaho Falls, ID, USA (2005).
2. G. L. MESINA, "Visualization of RELAP5-3D Best Estimate Code,” Best Estimate Seminar 2004, Washington, D.C, November 14-18, (2004).
3. S. P. MILLER, G. L. MESINA, “Visualization of RELAP5-3D Best Estimate Code,” Proceedings of the 2004 International RELAP5-3D User Seminar, Sun Valley, ID, USA, August 25-27, (2004).
4. COBALT BLUE, INC., “FOR_STRUCT, Your FORTRAN Structuring Solution,” 11585 Jones Bridge Rd, Suite #420-306, Alpharetta, GA, 30005, USA, (1997).
5. G. L. MESINA, P. P. CEBULLl, “Extreme Vectorization in RELAP5-3D,” Proceedings of the Cray User Group 2004, Knoxville, TN, USA, May 16-21, (2004).