New automated tool 'debugs' nuclear weapon simulations

June 1, 2010

Print Version

WEST LAFAYETTE, Ind. - Purdue University researchers, working with high-performance computing experts at Lawrence Livermore National Laboratory, have created an automated program to "debug" simulations used to more efficiently certify the nation's nuclear weapons.

The program, called AutomaDeD (pronounced like automated), finds errors in computer code for complex "parallel" programs.

"The simulations take several weeks to run, and then they have to be debugged to correct errors in the code," said Saurabh Bagchi, an associate professor in Purdue's School of Electrical and Computer Engineering. "The error might have occurred in the first hour of operation, and if you had known about it you could have stopped it then."

Because international treaties forbid the detonation of nuclear test weapons, certification is done using complex simulations. The simulations, which may contain as many as 100,000 lines of computer code, must accurately show reactions taking place on the scale of milliseconds, or thousandths of a second.

"Many times an error in a simulation code may not become evident until long after it occurs," said Bronis R. de Supinski, co-leader of the ASC Application Development Environment Performance Team at the U.S. Department of Energy's Lawrence Livermore National Laboratory. "These delays are challenging since they make the actual location of the bug unclear."

In parallel operations used for powerful simulation tools, a highly complex job is split into numerous smaller and more manageable processes that are handled by separate machines in large computer clusters. After the computers complete their individual processes, all of the parallel results are combined.

Conventional debugging programs, however, must be operated manually, with engineers navigating through a large number of processes.

"Debuggers have worked well for sequential applications," Bagchi said. "But when we extend these to large parallel applications, application developers are not very happy because it's very time consuming and difficult to do the manual debugging. It is just difficult for human cognitive abilities to keep track of what is going on simultaneously in many processes and determine what is

Featured News

- Purdue to welcome second Woodrow Wilson Teaching Fellows class
- Children from military families will participate in Operation Purple camp at Purdue
- New automated tool 'debugs' nuclear weapon simulations
- Purdue to expand capacity, improve facilities for health and noise research
- Trustees approve 10-year capital plan
- Purdue trustees ratify appointments, honor administrators and athletes, confirm retirement plan change, approve coal purchase
- Gasoline prices for summer: expect the unexpected

More News »

anomalous."

So, to enable the automatic debugging of the simulations, the researchers created AutomaDeD, which stands for automata-based debugging for dissimilar parallel tasks.

"The idea is to use AutomaDeD as the simulation is running to automatically monitor what's happening," Bagchi said. "If things start going wrong, AutomaDeD would stop and flag which process and which part of the code in the process is likely anomalous."

Errors in software code cause "stalls" and "hangs" that slow or halt simulations or give incorrect results. Another problem with parallel programs is interference from software that previously ran on the same computer clusters but were not properly expunged before the new job started running.

Recent research findings show AutomatDeD was 90 percent accurate in identifying the time "phase" when stalls and hangs occurred; 80 percent accurate in identifying the specific tasks that were the sources for stalls and hangs; and 70 percent accurate in identifying the interference faults.

The findings will be detailed in a research paper to be presented on June 30 during the 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks in Chicago. The paper was written by Purdue doctoral student Ignacio Laguna, Bagchi, and Lawrence Livermore scientists Greg Bronevetsky, de Supinski, Dong H. Ahn and Martin Schulz. The primary developers of the program are Bronevetsky and Laguna.

The same debugging approach could be used to find errors in other parallel applications, such as those used in climate modeling and high-energy particle physics.

AutomaDeD works first by grouping the large number of processes into a smaller number of "equivalence classes" with similar traits. Grouping the processes into equivalence classes keeps the analysis simple enough that it can be done while the simulation is running.

AutomataDeD also works by splitting a simulation into numerous windows of time, called phases.

"So our tool lets you know if the error occurs for task 1 and task 5 in phase 153 and allows you to zoom in and find the specific part of the code that is problematic," Bagchi said.

Large computer clusters operated by Lawrence Livermore containing thousands of processors have been used for the debugging operations.

Purdue researchers did not work with the actual classified nuclear weapons software code but instead used generic "NAS parallel benchmarks," a set of programs designed to help evaluate the performance of parallel supercomputers developed by the NASA Advanced Supercomputing division.

The work began a year ago, and the new debugging program is currently being used by the federal lab to detect certain types of errors. The researchers are continuing to improve the program. The work is funded by the Department of Energy.

Also involved in the ongoing work is Karu Sankaralingam, an assistant professor of computer sciences and electrical and computer engineering at the University of Wisconsin at Madison.

Writer: Emil Venere, 765-494-4709, venere@purdue.edu

Sources: Saurabh Bagchi, 765-494-3362, sbagchi@purdue.edu

Donald B. Johnston, senior editor, press officer, Lawrence Livermore National Laboratory, 925-423-4902, johnston19@llnl.gov

ABSTRACT

AutomaDeD: Automata-Based Debugging for Dissimilar Parallel Tasks†

Greg Bronevetsky§, Ignacio Laguna‡, Saurabh Bagchi‡,

Bronis R. de Supinski§, Dong H. Ahn§, , Martin Schulz§,

‡*Purdue University* *Lawrence Livermore National Laboratory*

{ilaguna, mailto:saurabh%7D@purdue.edu
{bronevetsky, bronis, ahn1, schulzm}@llnl.gov

Today's largest systems have over 100,000 cores, with million-core systems expected over the next few years. This growing scale makes debugging the applications that run on them a daunting challenge. Few debugging tools perform well at this scale and most provide an overload of information about the entire job. Developers need tools that quickly direct them to the root cause of the problem. This paper presents AutomaDeD, a tool that identifies which tasks of a large-scale application first manifest a bug at a specific code region and specific program execution point. AutomaDeD statistically models the application's controlflow and timing behavior, grouping tasks and identifying deviations from normal execution, which significantly reduces debugging effort. In addition to a case study in which AutomaDeD locates a bug that occurred during development of MVAPICH, we evaluate AutomaDeD on a range of bugs injected into the NAS parallel benchmarks. Our results demonstrate that AutomaDeD detects the time period when a bug first manifested with 90% accuracy for stalls and hangs and 70% accuracy for interference faults. It identifies the subset of processes first affected by the fault with 80% accuracy and 70% accuracy, respectively, and the code region where the fault first manifested with 90% and 50% accuracy, respectively.