Nuclear weapon simulations show performance in molecular detail

June 5 2012, by Emil Venere

Employees at Lawrence Livermore National Laboratory work on a high-performance computer. Purdue researchers have collaborated with the national laboratory, using a similar high-performance computer to improve simulations that show a nuclear weapon's performance in precise molecular detail. Photo courtesy of Lawrence Livermore National Laboratory

U.S. researchers are perfecting simulations that show a nuclear weapon's performance in precise molecular detail, tools that are becoming critical for national defense because international treaties forbid the detonation of nuclear test weapons.

The simulations must be operated on supercomputers containing thousands of processors, but doing so has posed reliability and accuracy problems, said Saurabh Bagchi, an associate professor in Purdue University's School of Electrical and Computer Engineering.

Now researchers at Purdue and high-performance computing experts at the National Nuclear Security Administration's (NNSA) Lawrence Livermore National Laboratory have solved several problems hindering the use of the ultra-precise simulations. NNSA is the quasi-independent agency within the U.S. Department of Energy that oversees the nation's nuclear security activities.

The simulations, which are needed to more efficiently certify nuclear weapons, may require 100,000 machines, a level of complexity that is essential to accurately show molecular-scale reactions taking place over milliseconds, or thousandths of a second. The same types of simulations also could be used in areas such as climate modeling and studying the dynamic changes in a protein's shape.

Such highly complex jobs must be split into many processes that execute in parallel on separate machines in large computer clusters, Bagchi said.

"Due to natural faults in the execution environment there is a high likelihood that some processing element will have an error during the application's execution, resulting in corrupted memory or failed communication between machines," Bagchi said. "There are bottlenecks in terms of communication and computation."

These errors are compounded as long as the simulation continues to run before the glitch is detected and may cause simulations to stall or crash altogether.

"We are particularly concerned with errors that corrupt data silently, possibly generating incorrect results with no indication that the error has occurred," said Bronis R. de Supinski, co-leader of the ASC Application Development Environment Performance Team at Lawrence Livermore. "Errors that significantly reduce system performance are also a major concern since the systems on which the simulations run are very

expensive."

Advanced Simulation and Computing is the computational arm of NNSA's Stockpile Stewardship Program, which ensures the safety, security and reliability of the nation's nuclear deterrent without underground testing.

New findings will be detailed in a paper to be presented during the Annual IEEE/IFIP International Conference on Dependable Systems and Networks from June 25-28 in Boston. Recent research findings were detailed in two papers last year, one presented during the IEEE Supercomputing Conference and the other during the International Symposium on High-Performance Parallel and Distributed Computing.

The researchers have developed automated methods to detect a glitch soon after it occurs.

"You want the system to automatically pinpoint when and in what machine the error took place and also the part of the code that was involved," Bagchi said. "Then, a developer can come in, look at it and fix the problem."

One bottleneck arises from the fact that data are streaming to a central server.

"Streaming data to a central server works fine for a hundred machines, but it can't keep up when you are streaming data from a thousand machines," said Purdue doctoral student Ignacio Laguna, who worked with Lawrence Livermore computer scientists. "We've eliminated this central brain, so we no longer have that bottleneck."

Each machine in the supercomputer cluster contains several cores, or processors, and each core might run one "process" during simulations.

The researchers created an automated method for "clustering," or grouping the large number of processes into a smaller number of "equivalence classes" with similar traits. Grouping the processes into equivalence classes makes it possible to quickly detect and pinpoint problems.

"The recent breakthrough was to be able to scale up the clustering so that it works with a large supercomputer," Bagchi said.

Lawrence Livermore computer scientist Todd Gamblin came up with the scalable clustering approach.

A lingering bottleneck in using the simulations is related to a procedure called checkpointing, or periodically storing data to prevent its loss in case a machine or application crashes. The information is saved in a file called a checkpoint and stored in a parallel system distant from the machines on which the application runs.

"The problem is that when you scale up to 10,000 machines, this parallel file system bogs down," Bagchi said. "It's about 10 times too much activity for the system to handle, and this mismatch will just become worse because we are continuing to create faster and faster computers."

Doctoral student Tanzima Zerin and Rudolf Eigenmann, a professor of electrical and computer engineering, along with Bagchi, led work to develop a method for compressing the checkpoints, similar to the compression of data for images.

"We're beginning to solve the checkpointing problem," Bagchi said. "It's not completely solved, but we are getting there."

The checkpointing bottleneck must be solved in order for researchers to create supercomputers capable of "exascale computing," or 1,000

quadrillion operations per second.

"It's the Holy Grail of supercomputing," Bagchi said.

More information: Automatic Fault Characterization via Abnormality-Enhanced Classification, Greg Bronevetsky, Ignacio Laguna, Saurabh Bagchi, and Bronis R. de Supinski, PDF.

ABSTRACT
Enterprise and high-performance computing systems are growing extremely large and complex, employing many processors and diverse software/hardware stacks. As these machines grow in scale, faults become more frequent and system complexity makes it difficult to detect and diagnose them. The difficulty is particularly large for faults that degrade system performance or cause erratic behavior but do not cause outright crashes. The cost of these errors is high since they significantly reduce system productivity, both initially and by time required to resolve them. Current system management techniques do not work well since they require manual examination of system behavior and do not identify root causes. When a fault is manifested, system administrators need timely notification about the type of fault, the time period in which it occurred and the processor on which it originated. Statistical modeling approaches can accurately characterize normal and abnormal system behavior. However, the complex effects of system faults are less amenable to these techniques. This paper demonstrates that the complexity of system faults makes traditional classification and clustering algorithms inadequate for characterizing them. We design novel techniques that combine classification algorithms with information on the abnormality of application behavior to improve detection and characterization accuracy significantly. Our experiments demonstrate that our techniques can detect and characterize faults with 85% accuracy, compared to just 12% accuracy for direct applications of traditional techniques.

Provided by Purdue University