

TOTALVIEW® Case Study

TotalView® Debugger Helps Applied Research Associates' Southwest Division Reduce Development Time by at Least a Factor of Three

End-User

Applied Research Associates (ARA) provides innovative, in-depth and diversified research, engineering and technical support to clients in government and industry. The Southwest Division of ARA produces software and calculations/analyses for customers that include Defense Department agencies, primarily the Defense Threat Reduction Agency (DTRA).

The Challenge

With reduced federal funding for military research and competition from other defense contractors trying to “catch up” with ARA’s capabilities, it is more important than ever that its programmers have access to the most scalable, reliable and effective tools available to help them to develop highly-specialized software applications efficiently and effectively. Without an advanced multi-core debugging tool like the TotalView Debugger, the company would face ramifications such as slower development times and potential quality issues, which could lead to unhappy customers, budget overruns, and frustrated developers.

The Solution

The TotalView Debugger was designed from the ground up to handle the complexities of the world’s most demanding multi-processing applications that scale to thousands of processes and threads with applications distributed over multiple machines or processors. TotalView offers many advanced features, including multi-language support and built-in source code and optional memory debugging capabilities, which streamline and simplify the development process. TotalView has made it possible for ARA developers to significantly speed their development times, as well as to improve the quality of their software products.

“Once I determined that TotalView was the only effective debugger for handling MPI applications, using it was a no-brainer and I was hooked,” said Hank Happ, principal scientist at Applied Research Associates.

Background

The application codes used by ARA are inviscid CFD codes (primarily SHAMRC), with an emphasis on airblast. Once a set of changes is received to compile, ARA developers use TotalView to go through the modifications line by line to be sure that the modifications are performing as expected. More often than not, this exercise uncovers something. ARA also uses TotalView heavily to debug the development of the parallel version of its CFD code SHAMRC.

Development of parallel (MPI) code is extremely difficult without an interactive debugger.

Interactive debuggers are productivity enhancers and are not only useful for post-delivery debugging, but for “line-by-line” code behavior observation during development.

How the TotalView Debugger Helps

TotalView has helped ARA to reduce development times by at least a factor of three. When going through debugging sessions, ARA sometimes uncovers long-standing bugs that were not previously noticed, but are made clear when examining variable values in the debugger.

The benefits that ARA has realized by utilizing the TotalView Debugger include the ability to find and fix bugs quickly, so that an unhappy customer does not have to be unhappy for very long! Second, its ability to handle MPI makes it feasible to develop code, especially parallel code, with the budget and resources that are available. Without TotalView, ARA believes that customer estimates for the cost of code development would be about twice as much than what it is with TotalView.

“Once I determined that TotalView was the only effective debugger for handling MPI applications, using it was a no-brainer and I was hooked.”

TOTALVIEW® Case Study

"I really like that I can restart with an updated executable without leaving the debugger," continued Happ. "It sometimes takes a few minutes to get all of the variable windows up and situated the way I like. Being able to run the debugger in the background, find a problem using TotalView, fix it, recompile, and then restart within the debugger — allowing me to retain my windows — is a definite plus."

"We have also found the TotalView support team to be top-notch."

Additionally, the company has found that TotalView has made installations relatively straightforward. Even though ARA uses the FORTRAN 77 computer language, they have found that new versions of TotalView have useful features, especially the latest release, 8.3. One feature that has been helpful to ARA is the TotalView Debugger's support for the GNU compilers (which may not be true for all debuggers) and all of the platforms on which they run. Finally, TotalView is the debugger of choice for the HPC MSRC's, so it's possible for ARA to debug code running on those machines with TotalView as well.

"We have also found the TotalView support team to be top-notch," concluded Happ. "Questions are answered quickly, accurately and completely — and with a dash of appropriate, much-needed humor!"

About TotalView Technologies

TotalView Technologies is the world's leading provider of scalable debugging and analysis software solutions for the multi-core era. TotalView Technologies products enable software developers to quickly, easily and effectively debug UNIX, Linux, and Mac OS X applications running on development machines with single, dual-core, multi-core, or multiple processors.

For more than 20 years, TotalView Technologies products have been at work in research institutions, government laboratories, and technical computing centers, as well as commercial enterprises in the financial services, telecommunications, biotech, aerospace, weather prediction, film special effects and animation, oil and gas exploration, and computer-aided engineering markets. Recognized worldwide as the gold standard for debugging in high-performance, distributed or cluster computing environments, TotalView Technologies' award-winning technology is used to solve the world's toughest computing problems on many of the world's largest supercomputers.

For more information, visit www.totalviewtech.com.

TotalView Technologies

24 Prime Park Way
Natick, MA 01760
P.508.652.7700
F.508.652.7701