

the Exascale Report



The voice of the emerging exascale community

<http://TheExascaleReport.com>

What good are extreme scale systems without extreme scale development environments? The answer is obvious to all of us. Progress in scalable debugging is already starting to change HPC software development capabilities. New, innovative approaches are demonstrating encouraging results and give encouraging evidence that a generation of extreme scale development tools are on the way.

Development tools heading for the Exascale future

By: David Lecomber

“May you live in interesting times” is known as the Chinese curse. For HPC software developers, life is becoming very interesting - and likely to continue to be for a long while.

There are many changes and challenges on the road to Exascale that have been identified in the output of the IESP (International Exascale Software Program): step changes in the form of greater concurrency, hierarchical concurrency and a strong need to exploit data locality are amongst them. These are changes that we can anticipate - and are already starting to have impact at the high end: Gone are the simple carefree(!) days of the one size fits all MPI code!

For example, Oak Ridge's Cray XK7 Titan system provides most of its 20 Petaflop

performance via over 18,000 GPUs. Users develop codes that are aware of the CPU/GPU memory hierarchy with explicit transfer of data from host to accelerator using OpenACC directives.

Another multi-petaflop system, TACC's Stampede uses Intel Xeon Phi coprocessors and sees users adopting a number of programming models - from an offload model similar to OpenMP and OpenACC, through to a symmetric mode of MPI and OpenMP on both the coprocessors and host processors simultaneously.

Some changes are more fundamental as limits of certain algorithms are reached. Weak scaling - increasing the problem size in order to eventually exploit the available parallelism - is not working any more. Many computational scientists are gearing up to rewrite codes to adopt paths that will actually scale, rather than pushing for incremental changes that have an eventual ceiling. The CRESTA project (<http://cresta-project.eu>), for example, is bringing together systems, tools and application developers to explore the Exascale challenge - and is rethinking the whole approach to simulation problems in domains such as weather, CFD, biomedical modelling and molecular dynamics.

So, victory over today's multi-petaflop and future extreme scale systems will be won or lost by the software that is developed. With software development at the centre of the challenge, it is vital that the development environment is available: we will still need to fix bugs and to fix performance: machines are

the Exascale Report



The voice of the emerging exascale community

already too expensive to waste on broken or inefficient code.

Allinea Software is one company leading the charge for development tools at scale. Its debugging tool, Allinea DDT, is already used for debugging multi-petaflop applications at full scale. Scalable debugging has already changed software development capability at some high-end sites - and shows extreme scale development tools are possible.

Debuggers are an incredibly important part of the system: reducing wasted machine time and developer time. In many cases debuggers make science possible by enabling a code to run.

The Oak Ridge National Laboratory relies on Allinea to debug software on their leadership facility, Titan. When a bug strikes at 100,000 processes there is no other way to discover what is happening. "You need high-grade software tools that can scale along with your code - a debugger in particular - because when problems arise at scale, you are in a totally different universe," says Joshua Ladd, Tools Project Technical Officer during the OLCF3 Project at ORNL.

Debugging complex applications and environments involves two equally critical parts. The first is seeing application state in the most effective way possible. The second is to be able to debug regardless of the size or architecture of the machine.

It is self evident that both parts will be requirements for Exascale. Whilst automatic

tools can offer hope of some defect detection for obvious crashes or regressions, there will always be behaviour that only the developer knows is wrong.

In debugging, handling the vast complexity program state is all about how it is presented. In Allinea DDT, techniques such as showing small graphs of variable values and automated data change detection and merging identical processes make the complexity of concurrency vanish. The scale of data sets are also a challenge computational scientists face - particularly when a simulation starts to diverge from the norm. Recently visualization of vast data sets during debugging was added by enabling interoperability with the scalable VisIt tool (<https://wci.llnl.gov/codes/visit>).

Debugging operations are as quick on a machine like Titan applied to over 100,000 cores as they are on a laptop - fractions of a second. Behind the speedy performance of Allinea DDT is a scalable tree-based topology used to communicate between daemons across the system.

A challenge for the future is to ensure that tool performance is maintained. The extra nodes are probably not the greatest challenge: the tree architecture in Allinea DDT can handle that. The primary concern will be to ensure that the step-change in chip-level parallelism is handled well by the tools and that will lead to interesting questions for chip and device vendors and for the operating systems developers as well as tool vendors.

the Exascale Report™



The voice of the emerging exascale community

Performance of applications is the other field in which tools hold the key to success - and developers will consider performance issues as much a bug as application crashes and expect closer interaction of previously disparate tools. Allinea's focus on the integrated tool environment using a scalable backbone is changing the outlook for performance profiling for the better here too.

In conclusion, tools today are able to serve developers well - but there is always work to do - and with changes underway such as increased hierarchical programming styles we can expect to see substantial refinements to really embrace the multi-level approaches, along with changes to better include programming models such as task-based programming.

If we note that 100,000 processes is already a vast number ordinarily beyond human comprehension ("one, two, many"!)- and that development tools already help developers to handle that, it is an encouraging start for the billion-way challenge of Exascale!

###

Talk to the Emerging Exascale Community

Don't miss out on your opportunity to talk to the emerging Exascale community by submitting a sponsored editorial opinion article.

Talk to the community in your own words. Address the issues you feel most strongly about. Raise your concerns or ideas, or present a challenge to policy makers, funding sources, vendors or end users.

For more information:

<http://theexascalereport.com/pdf/advertorial.pdf>

For subscription information, go to:

<http://theexascalereport.com/user/register>

The Exascale Report

The Voice of the Emerging Exascale Community.

© 2012, The Exascale Report™