TITAN: Paving the Way to Exascale

On the road to exascale systems (i.e., systems able to reach 1,000 petaflops), the Oak Ridge Leadership Computing Facility (OLCF) will be walking in uncharted territory. As on any great journey, the familiar will have to be left behind.

For more than 50 years computers have roughly doubled in speed every 2 to 2½ years, regularly increasing the performance of electronic devices and giving hardware and software designers a comfortable knowledge of what's ahead and how best to prepare for it.

Well, those days are over. Conventional processing architectures are quickly reaching their maximum potential, and if computers are to continue to increase in power and speed, a revolutionary change in strategy is necessary. Introducing Titan.

It's recently become clear that America's best chance of achieving exascale computing power rests with an architecture that utilizes more powerful nodes than today's systems. This will be achieved via a marked increase in the number and, likely, the types of processing cores.

Whereas recent gains have been achieved by adding more homogenous cores, essentially just ramping up the number of central processing units (CPUs), Titan's hybrid architecture will couple different types of processors, allowing each to do what it does best, thus increasing the power and efficiency of the overall machine. Specifically, Titan will feature the familiar AMD Opteron CPUs alongside general-purpose graphics processing units (GPUs), a more energy-efficient technology for crunching numbers.

The arguments for Titan's hybrid, multi-core architecture are twofold: (1) it seems to be the most straightforward way of increasing computational power, and (2) it accomplishes an exponential increase in computing power with only a slight increase in power consumption, a major expense when you're dealing with the most powerful machines in the world.

The ultimate goal of exascale computing is to achieve a thousandfold increase in delivered performance but within a power envelope that's roughly twice what Jaguar uses now, a 500-fold increase in power efficiency. Titan is the first step toward achieving this landmark metric.

But it's not all about the hardware. To achieve the required power savings, software designers and computational scientists are going to have to change the way they program. Specifically, the applications of the future need to minimize data movement, the most demanding aspect of large-scale computational science. The codes that best take advantage of Titan's enormous computing power will be those that maximize data locality.

Nothing about Titan will be easy, but the effort will be necessary if America is to continue to lead the way in high-performance computing (HPC), increasingly recognized as a vital part of a successful, competitive technological future. "We believe that we are taking the concrete first step towards a viable exascale architecture," said Bronson Messer, an astrophysicist at Oak Ridge National Laboratory (ORNL).

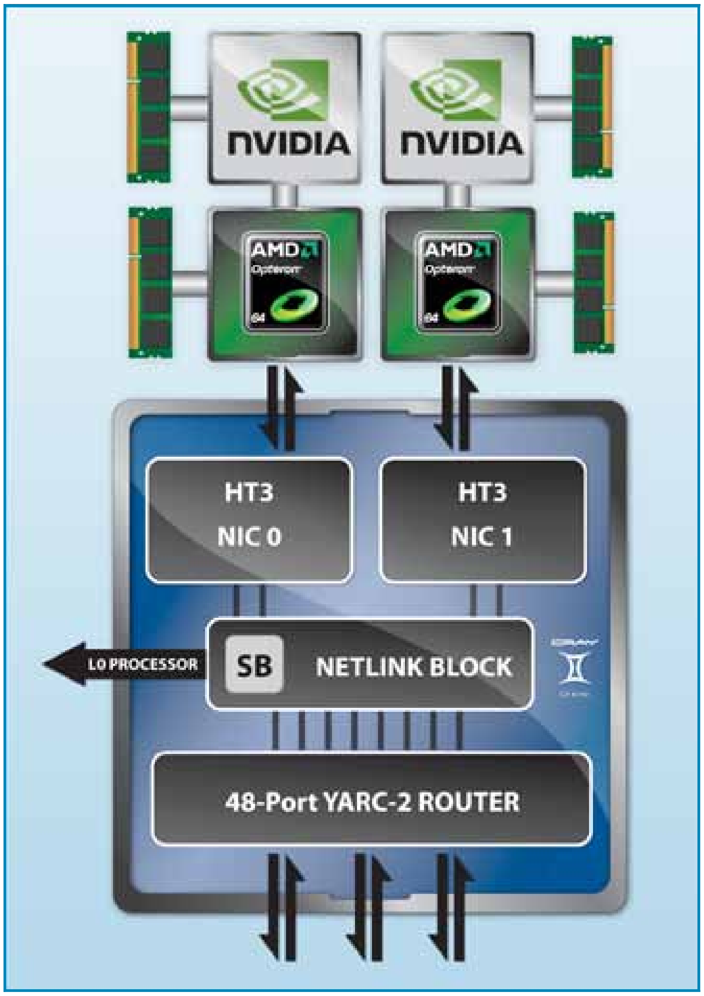
The aim of Titan is to achieve 20 petaflops, or a peak speed just under 10 times faster than Jaguar, while performing groundbreaking science. That will make Titan among the fastest machines in the world. While Titan will retain the same overall cooling and power infrastructure, just about everything else will have to change. It starts with the hardware.

Hardware

Despite its revolutionary promise, Titan is in many ways evolved from Jaguar's existing XT architecture. This heritage is a great advantage for a system that must be rapidly and efficiently deployed. "The cabinets will be reused as part of the upgrade, but the electronics will be replaced," said Messer.

The main game changer is the introduction of the GPUs. These application accelerators, spawned from the video game industry, will afford an enormous increase in floating-point performance, which means simulations that would take weeks or months on Jaguar might run in days on Titan. When it comes to productive computational science, time is vital, and the introduction of Titan's GPUs will allow researchers to achieve faster breakthroughs via more simulations over time.

The beauty of GPUs lies in their massively parallel nature. Unlike conventional CPUs, which perform operations serially, or one-by-one, GPUs are capable of performing many different operations at once. In Titan, they will be soldiers, and the CPUs the generals. In other words, the GPUs will be marshaled to perform the most intensive calculations because of their ability to rapidly crunch numbers, while the AMD Opterons will be responsible for "command and control" in various scientific applications.



Titan's architecture represents the latest in adaptive hybrid computing. The node above combines AMD's 16-core OpteronTM 6200 Series processor and NVIDIA's Tesla X2090 many-core accelerator. The result is a hybrid unit with the intranode scalability, power-efficiency of acceleration, and flexibility to run applications with either scalar or accelerator components. This compute unit, combined with the Gemini interconnect's excellent internode scalability, will enable Titan's users to answer next-generation computing challenges.

Specifically, Titan and Jaguar will differ as follows: Each compute node on the current Jaguar XT architecture has two 6-core Opteron CPUs, and every node is connected via Cray's Seastar custom router chips. Titan will consist of one 16-core Opteron and one GPU. Also, the Seastar chips will be replaced by a new interconnect chip dubbed Gemini.

Gemini is Titan's second major breakthrough. Essentially, Gemini increases the computer's ability to do one-sided communication, allowing one processor to share data with another processor without the need for time-consuming "handshaking." This improvement is key, as communication between nodes is one of the most expensive elements of HPC.

As Titan is deployed, there will also be some changes in the computational ecosystem, or the attendant hardware and software surrounding the machine. Perhaps first and foremost among these is the Spider file system. In order to keep up with the demands of Titan, the OLCF will add hundreds of gigabytes per second of bandwidth and tens of petabytes of storage to Spider. Because Titan will be capable of producing more data at any given moment than Jaguar, the new system will have to keep up in terms of storage and file input/output.

These changes, though incremental, represent an enormous leap when taken together. As Titan is deployed there will no doubt be trials and tribulations, but in the end the machine will most likely represent the future of HPC.

But the hardware is just one dimension of the challenge to come. No matter how powerful Titan becomes, it's only as good as the programs it runs.

Software

The primary software challenge on Titan is to uncover and exploit three levels of parallelism: distributed, in which a task or physical domain is divided among a number of nodes; thread level, in which each task from the distributed workload is divided into discrete threads of execution on local processors; and vector level, in which the threads are further divided and sent to the GPUs for increased performance.

Currently, the software on Jaguar does a great job of distributed parallelism. Tools like the Message Passing Interface (MPI) and Global Arrays allow various applications to communicate between processors efficiently. These same tools also allow application writers to distribute work over lots of processors, one of Jaguar's best assets.

Further, Jaguar allows ways to expose the thread level of parallelism via shared memory processing in the form of methods like OpenMP and Pthreads, standards for creating parallel threads in a shared memory environment.

The most immediate software challenges reside in the third layer of parallelism, i.e., the vector. Ultimately, Titan's potential lies in optimizing the use of the GPUs through effective vector-level programming, where the real power and speed of the GPUs is found.

Take climate simulation, for instance. These complex exercises rely heavily on distributed parallelism, essentially dividing the surface of the Earth into grid cells, with each cell living on an individual processor. At each of these grid points, however, there is a lot going on; for instance, each contains numerous variables to describe various components of the Earth system, e.g., the atmosphere, the ocean, and the Sun's radiation, adding up to plenty of physics at individual grid points. On current systems these calculations are usually done serially, one-by-one, at each grid point.

However, they can also be tackled at the thread level by exploiting the fact that at each grid point you have multiple layers of parallelism for each calculation. Tapping this unrealized parallelism can increase the fidelity of the simulations, speeding them up and allowing the use of more sophisticated approximations, resulting in greater overall accuracy.

On Titan, the additional availability of vector-like parallelism via the GPUs means even more optimized simulations. For example, a researcher might use an atmospheric chemistry package on Jaguar that had previously been out of reach because it took months to run, said Messer, adding that researchers at ORNL are already reducing months-long wallclock times on Jaguar to a handful of days with GPUs.

"Ultimately we hope the applications on Titan will be full-scale simulations where every layer of parallelism available is exploited," he said.

After much trial and error, the supercomputing community is now discovering ways to do just that. The current champ is CUDA, a language extension from GPU-maker NVIDIA. The fact that it's simply a language extension, and not a new language, is a very good thing for programmers and scientists alike.

That they don't have to learn an entirely new language and can simply plug CUDA into their existing codes is an enormous convenience, particularly with codes that have evolved over years. Unfortunately, effective use of CUDA requires an in-depth knowledge of the physical structure of the GPUs and the surrounding hardware. Few programmers possess this kind of knowledge, and because GPUs are relatively new it could be a while before the community absorbs a sound understanding of their architecture. Programming that demands hardware knowledge is known as low-level programming, and the OLCF is doing its best to avoid this model. CUDA is almost there, said Messer, hovering between high-level, in which less knowledge of a machine's hardware is required, and low-level.

"We need a higher-level description of doing this," said Messer, adding that several of the OLCF's partners, including Cray, The Portland Group (PGI), and CAPS, a leading provider of multi-core software tools, are looking at a compiler-based approach to this kind of software. The products will in all likelihood be language extensions that operate at a higher level, i.e., that don't require as much hardware knowledge. In essence, these new extensions will tell the compiler what the user wants to happen and the compiler will decide how best to accomplish the task during runtime.

The great benefit of a higher-level extension, besides not requiring as much low-level programming, is that the code will work on any platform if the directives, or instructions in the compiler (a computer program that transforms a programming language such as Fortran into another computer language such as binary), are done right. The directives describe what needs to happen. The compiler will implement that on GPUs or CPUs, depending on what is available.

Of course, directives present challenges as well, said Messer. Because the compiler is making the decisions, the user isn't necessarily guaranteed the highest performance from an application. Compilers, it turns out, are not optimal decision makers. The brunt of the work on the new language extensions involve making this penalty as small as possible. Currently, Cray, PGI, and CAPS are working on new compiler technologies that will allow the compiler to make better decisions and help researchers get the most from their codes and their time on Titan.

As a testament to this work, Cray's compiler on S3D, a scientific software developed at Sandia National Laboratories for simulating combustion, has gotten performance numbers that are equivalent to hand-coded CUDA. This is an enormous breakthrough for one simple reason: Programmers may no longer need to learn the minutiae of GPU architecture to reach their maximum potential on Titan.

A hallmark of scientific software is that it is constantly under development, said Messer. The cutting-edge applications running on leadership-class platforms like Jaguar are necessarily always changing, becoming more efficient and realistic. This constant development leads to a consistent barrage of problems and bugs that must be solved before the software can be used to perform numerical experiments. Because Titan will be a new platform with complex hardware, it will be difficult to know what is broken and exactly where the bug is.

That's where the next challenge lies: tools.

Tools

When it comes to Titan, two types of tools will be critical for success: debuggers and performance analysis tools.

Software is never simple, and when it comes to world-class supercomputers with mission-critical programs, it's at its most complex. Finding errors is essential to ensuring that simulation results are as accurate as possible.

The OLCF is working closely with Allinea to extend its DDT debugging product, a leading tool for debugging parallel MPI and Open MP programs, to include GPU debugging. Because GPUs are so new, figuring out how to program a debugger to accommodate their presence is a monumental task, and one that is being met head-on by the OLCF/Allinea partnership.

On the performance analysis front, the OLCF is also working with Technical University Dresden on its Vampir package, a modeling tool used to measure performance and discover bottlenecks. By helping to identify inefficient sections of code and allowing researchers to view the applications in progress, Vampir lets programmers and researchers optimize their applications and get the optimal performance from Titan's hybrid architecture.

If Titan is to be the productive machine envisioned, debugging and performance analysis will be absolutely crucial to realizing its potential and implementing the OLCF's future program plan and philosophy.

"Titan is the first step towards ensuring America's exascale future. With Titan, the OLCF will embark on a new era in simulation, one sure to contribute immensely to science and America's competitive technological future," said OLCF Director Buddy Bland.

olcf.ornl.gov/titan