

Meeting Today's Science and Engineering Computational Needs



A NEW APPROACH TO MAKING EFFICIENT USE OF HPC RESOURCES

By **Sal Salamone**, Contributing Editor, *Bio•IT World*

Produced by Cambridge Healthtech
Media Group Custom Publishing

sgi[®]

www.sgi.com

Meeting Today's Science and Engineering Computational Needs

Simulation, modeling, data analysis, and visualization are at the heart of most scientific and engineering efforts today. In fact, when it comes to conducting basic research or developing commercial products, there is an ever-increasing need for more and more computing power to accelerate the work on critical projects. The challenge is how to meet that need for computing power.

Several factors continue to drive up the high performance computing (HPC) requirements. First, the volume of data that needs to be analyzed, incorporated into models and simulations, or visualized is exploding. Each successive generation of test and lab equipment, be it for oil exploration or new drug discovery, offers finer resolution and more detailed results. This yields greater insight into the problem being investigated, but also produces massive amounts of data.

Similarly, engineers and scientists keep evolving their models and simulations to include finer spatial or temporal resolution or they extend the simulations to cover greater time scales.

With these factors to consider, the challenge most organizations face is how to efficiently support the wide variety of applications they must support, all of which have particular computing requirements.

MATCHING APPLICATIONS AND HPC RESOURCES

Increasingly more powerful computing resources must be applied to today's mix of more realistic and complex life sciences research and development in-

vestigations. The challenge is how to accommodate the broad mix of applications that are typically used.

In particular, to ensure an application runs efficiently, there needs to be a matching up of an application's characteristics with suitable high performance computing (HPC) system components. For example, a life sciences application like BLAST or a computer-aided engineering (CAE) application like FLUENT can benefit from a cluster approach where each node has sizable memory to support the large databases that analysis samples are normally run against. With this type of configuration, a large run can be broken into many smaller jobs all of which can be run simultaneously on discrete nodes.

In contrast, a finite element model of, for example, a protein's interaction with a cell's membrane would require a system with a large shared memory and tightly coupled nodes. Similarly, many Gaussian and HMMER runs can benefit greatly from shared memory systems.

Essentially, an organization might devote a different computing platform to each type of application. To make the match, an application would be checked to see if it was memory-intensive, bandwidth-intensive, CPU-intensive, or a combination of all three. Other factors would also need to be taken into account. For example, how interdependent or loosely coupled are the nodes for a particular application? What volumes of data must be retrieved from storage to perform a calculation? How much output data is involved? Is the output data used in another application? Perhaps it needs to be combined with or compared to another data-

CAE HPC application traits

CAE Applications	SGI® Altix®	SGI® Altix® XE
ABAQUS		Preferred
NASTRAN	Preferred	
LS DYNA		Preferred
FLUENT		Preferred
ANSYS		Preferred
Pam CRASH		Preferred
RADIOSS		Preferred
Star CD	Preferred	

Many of the most commonly used CAE applications are ideally suited to run on HPC clusters such as the SGI Altix XE, while several are better served by a shared memory system like the SGI Altix.

base. Does the data need to be visualized?

Having determined an application's characteristics, it could then be assigned to a suitable computing platform. Obviously, it would be ideal if each application had its own perfectly optimized platform available when needed. Of course, that would be impractical. So what's needed is a computing platform that incorporates a variety of computing technologies. This is why HPC clusters have gained such wide appeal.

One point in favor of using HPC clusters is that, increasingly, a variety of newer generation components have been incorporated into these clusters.

Many cluster nodes today use multi-core processors to increase the raw processing power without increasing the server size, which is becoming an issue in crowded data centers. Multi-core processors often run at lower clock speeds than their comparable single-core processors and as a result, they often consume less electricity, which is also a growing consideration in many data centers. (And as a result of using less electricity, the cooling demands are also lower, thus saving additional energy.)

Multi-core processors also typically come with integrated memory controllers to improve the flow of data between a CPU and its associated memory. This can help improve the performance of many

classes of applications by eliminating the wait for data to be fed into today's high performance processors.

In many cases, even as CPU performance increased, some life sciences applications got an additional boost from co-processors. In particular, applications often were accelerated using application-specific integrated circuits (ASICs) and field-programmable gate arrays (FPGAs). And many industry experts believe researchers will soon be tapping the power of graphics processing units (GPUs), which due to the demand for more realistic gaming products, have been growing in power at a faster rate than traditional CPUs.

These approaches have the potential to deliver additional benefits. For example, as is the case with multi-core processors, FPGA systems are often very energy efficient on a performance/watt basis when compared to running the same job on a traditional server.

The pure increase in processing performance new CPUs and accelerators deliver is only one aspect that impacts the ability to run applications in a timely manner. A cluster's interconnection infrastructure must also be considered. This includes addressing the switching fabric and the performance of the storage systems that feed the data to life sciences applications.

For years, InfiniBand and proprietary interconnection technologies were used to reduce latency and increase overall performance. Increasingly, more systems today are incorporating standards-based interconnection technology that is based on gigabit Ethernet.

When choosing an interconnection technology, organizations try to find the best match performance-wise, but other factors may also come into play. For instance, IT staffing skills are often considered (many more IT people have experience with Ethernet than other technologies). Similar considerations are often made today for matching storage systems with applications. Namely, the systems are selected based on performance criteria and staffing expertise criteria.

And for some applications, shared memory systems offer additional benefits to increase the performance of an application.

Matching applications to HPC platforms

Life Sciences Applications	SGI Altix	SGI Altix XE	SGI® RASC™
BLAST-n		Preferred	Preferred*
Gaussian	Preferred		
Amber	Preferred		
VASP	Preferred		
CASTEP	Preferred		
GAMESS		Preferred	
Jaguar			
Dmol3	Preferred		
NAMD		Preferred	
Gromacs		Preferred	
HMMER	Preferred		Preferred*
ClustalW		Preferred	Preferred*
BLAST-p		Preferred	Preferred*
Smith-W		Preferred	Preferred*

* With some minor modifications to the application code.

SGI's shared memory Altix, HPC cluster Altix XE, and FPGA-based RASC systems are each ideally suited for different classes of life sciences applications.

WHY THINGS NEED TO CHANGE

Obviously, organizations have been mixing and matching these technologies for many years to reduce the time it takes to run their applications. But there is a growing realization that generic, white box HPC clusters are not keeping pace with the computing demands of today's life sciences applications.

What's changed recently is the scale of computational efforts including the need to support much more data, an increased mix of applications types, and more complex models. At the same time, the people who need access to HPC resources have also changed. Increasingly, everyone from the bench chemist or biologist to the lead scientist

needs to run applications before they can move to the next step of their research. Similarly, a designer who only works on a small part of larger product (such as an automobile) might now need to test his part in a CAE simulation before passing it along to the development team responsible for the entire product.

In the life sciences example, lab scientists might need a fast analysis of an experiment's data to help decide which experiment to do next. For the lead scientists, they might want to see a quick visualization of a complex dataset or modeling run to gauge the effectiveness of a drug candidate before it is goes into expensive clinical trials.

The point is the people who need access to increasingly more power HPC resources are not computer scientists. So even if they are given a choice of platforms to run their applications on, they most likely will not know which platform is the most suitable for a particular application.

Additionally, today there is more emphasis on workflow. What is important is not how fast a single job can run. The objective is to maximize the number of specific jobs, related to one problem that can be run in a given time. In other words, the goal is how fast a researcher can reach a decision about what to do next based on running a series of computational and visualization applications.

To date, matching applications to HPC resources has normally been done manually. There are several problems with this approach. First, as noted, many of users of HPC resources are not computer scientists and typically do not know what the best match is.

Second, dedicating a specific platform to each application can lead to an inefficient use of HPC resources if not done properly. For instance, while a large BLAST run could certainly be executed on hundreds of nodes of a cluster, it might be completed much faster if it were modified slightly and run on FPGAs.

Third, manual intervention impedes the automation of computational workflows. Scientists designating where (and on which platform) each job should run might not make the more efficient use of resources and it certainly does not take into account the sequential running of other applications.

INTRODUCING THE HYBRID COMPUTING ENVIRONMENT

What's needed to accommodate the varying demands for HPC resources is a new platform that supports all types of HPC hardware systems including shared memory, multi-processor clusters, and accelerators like FPGAs.

The hardware itself must include software that helps manage and automate the process of running jobs. The idea here is to take the decision making process that matches an application and specific job to particular computing resources out of the hands of the scientist and make the process automatic based on some prior knowledge and rules.

With these points in mind, SGI has developed the Hybrid Computing Environment. The SGI Hybrid Solution system incorporates SGI's shared memory Altix, HPC cluster Altix XE, and FPGA-based RASC built with Intel® Itanium® and Quad-Core Intel® Xeon® Processors .

The system uses Moab as its intelligent interface software. Moab can schedule jobs to run on multiple architectures and uses policies to automatically determine the best match of an application to an HPC architecture.

To run jobs using Moab, users access a Web portal. Managers can create custom templates for common jobs where users do not need to know the characteristics of an application and which HPC resource is well suited for it.

Moab allows researchers to run a variety of applications types from a single interface in a more efficient manner. In particular, it can accommodate jobs traditional run in batch mode. However, Moab can consider workloads and resource usage before running the jobs and schedule their running in a way that optimizes throughout.

Additionally, Moab can be used to allocate appropriate resources for interactive jobs such as visualization applications where the researcher must examine a result to initiate the next computational or visualization step. This enables visualization to be brought into the same cloud of HPC resources as other applications. This is as opposed to having a dedicated system just for visualization as is often done. The benefit here is that it makes more efficient



SGI Altix



SGI Altix XE



SGI RASC

use of HPC resources and perhaps has the potential to expand the use of visualization as a dedicated system might not be cost effective on its own.

BENEFITS OF THE HYBRID COMPUTING ENVIRONMENT

There are a number benefits to using the SGI Hybrid Computing Environment.

First, organizations can improve their computational workflows. Applications can be assigned resources and scheduled to run based on computational demands.

For example, consider a workflow where the applications MD NASTRAN and ABAQUS must run sequentially. The former runs more efficiently on a shared memory system; the latter is better suited to run on an HPC cluster.

If both applications run on an SGI Altix shared memory system, the MD NASTRAN runs much faster than it would on a cluster and the ABAQUS job takes longer than it would if it were run on a cluster. The opposite is true if both are run on just an SGI Altix XE HPC cluster.

In contrast, if both are run on an SGI Hybrid Solution, the execution of each application is optimized and the total time to complete the workflow is significantly less than using only one platform or the other (See chart: **Optimization the key**).

Second, the SGI Hybrid Solution optimizes the use of HPC resources. As noted, batch and interactive jobs such as visualization applications can share the same computing resources in contrast to having separate systems for each. Additionally, applications that benefit from shared memory, FPGAs, or other technologies can automatically be allocated based on HPC requirements and schedule to run in a sequence that optimizes use of the available resources.

A third benefit is that the SGI Hybrid Solution lowers total cost of ownership. There are fewer discrete systems to manage, so the burden on IT staff is reduced. Additionally, the variety of HPC resources are all managed using a single console using similar commands.

Fourth, user training is significantly reduced.

Optimization the key

	Altix 450 Only	Altix XE 240 Only
MD NASTRAN	15,200 secs	28,400 secs
ABAQUS	8,900 secs	4,500 secs
Wall Clock Time	25,100 secs	32,900 secs
SGI Hybrid Solution	19,700 secs	19,700 secs

Optimized solution delivered based on application mix

Using the SGI Hybrid Solution optimized the execution of a mix of applications significantly reducing the total time required to run all of the jobs.

Researchers and engineers do not need to develop an expertise in computer science to run their applications in the most efficient and timely manner. Managers and IT staff can set policies for matching applications to resources, scheduling jobs, and other tasks. The researcher or engineer simply uses a Web portal to submit a job and the system takes it from there.

Since the SGI Hybrid Solution concentrates HPC resources under one system, it allows all those resources to share common data. This means an organization can vastly simplify its data infrastructure by storing all data in one place. This also simplifies administration of data storage devices and the management of tying data volumes to specific applications.

And finally, by virtue of all of these other aspects the SGI Hybrid Solution helps improve research and engineering workflows. Jobs are executed in a timelier manner making the best use of HPC resources. The real-world impact is that complicated and multi-step analysis routines, as well as interactive visualization jobs, are completed in shorter times helping researchers make more intelligent and faster decisions as to how to proceed.

All of these benefits can add up to a significant advantage propelling an organization past its competitors. ■

APPENDIX:

Focus on the life sciences

The quest to better understand disease mechanisms and find new treatments is being driven by new laboratory research technologies and ever-more sophisticated modeling and simulation efforts.

With the competitive nature of the research and the commercial implications of being the first to discover and market a new drug, life sciences researchers find they must quickly analyze and visualize large volumes of data and run increasingly complex (and larger) models.

For example, new mass spectrometers routinely generate 1 GB or more of data per experiment. Many labs run a handful to dozens of these devices around the clock, thus generating volumes of data that must be examined almost as soon as it comes off the equipment.

Similarly, improvements in microscopy, CAT scans, MRIs, and other imaging techniques are yielding large volumes of digital data for each test conducted by a researcher. To

get useful information out of the images, researchers often must process the raw image data for analysis and frequently combine multiple slices (or views) to form 3D images of the cell or organism they are investigating.

With regard to simulation, techniques that were once the domain of academic labs, are now routinely used by life sciences organizations to identify promising drugs early on by virtue of absorption, distribution, metabolism, excretion, and toxicity (ADME/Tox) modeling and to optimize clinical research trials.

Unfortunately, as organizations incorporate new lab equipment or expand their modeling efforts, there are two common challenges that must be overcome. First, there is an ever-growing need for additional computational resources. And second, a wide variety of applications, all of which have particular computing requirements, must be supported.

