Shared Memory: Standards Scale Up with SGI Altix UV

Addison Snell

May 2010

White paper

EXECUTIVE SUMMARY

Shared-memory systems gave way to clusters in high performance computing (HPC) because of the perceived price/performance benefits of industry-standard processors and operating environment, leaving behind proprietary architecture that scaled well but which were based on more expensive RISC processors (or later, Itanium) and customized UNIX implementations. This cost savings came at a cost: the loss of shared-memory programming capabilities and performance benefits in the wholesale movement to MPI on distributed-memory clusters.

SGI, a long-time architecture differentiator, has kept true to that path through significant upheavals in the company's fortune, and now the vision is coming full circle with Altix UV, a scalable shared-memory model based on Intel® Xeon® processors and Linux. Some of the features of Altix UV are:

- Scalability to 2,048 cores and 16TB of globally shared memory, the architectural limit of Intel® Xeon® processor E7 family.
- NUMAlink 5 interconnect and hub, Intel QPI, and MPI Offload Engine provide an environment for scaling MPI codes in a shared-memory environment.
- Support for numerous topologies and programming models, allowing end users to suit the computer to the application.

Under new management, SGI has focused its roadmaps and remains committed to HPC, and the company is seeing an increase in its adoption rates. Ultimately SGI's fortunes with Altix UV will be tied to how far people are willing to push their codes and their imaginations.

MARKET DYNAMICS

The Emergence of Clusters – Lower Cost, at a Cost

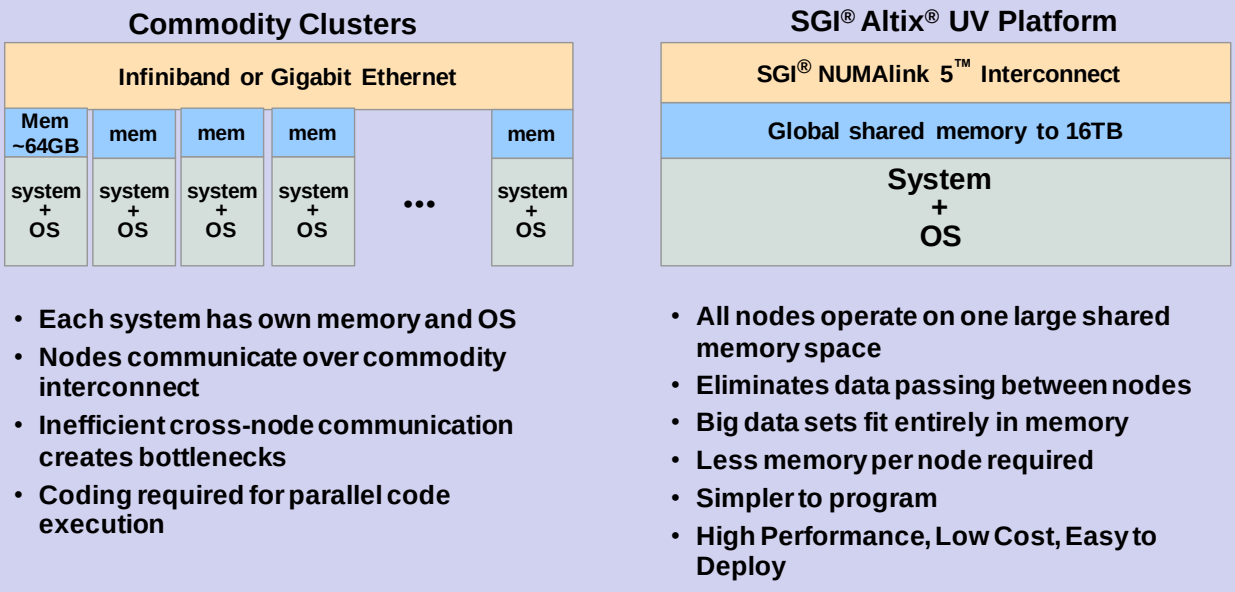
In the relentless pursuit of increased performance for a given budget, the HPC industry has undergone periodic technology disruptions. Over a ten-year period from the mid-1990's to the mid-2000's, HPC buyers made a large-scale shift away from symmetric multi-processing (SMP) systems based on RISC microprocessors and proprietary UNIX systems, to more loosely coupled clustered architectures made up of independent nodes of commercial-off-the-shelf (COTS) servers running x86 microprocessors and Linux. The industry-standard x86 processors were produced at high volume for lower cost than RISC, and most Linux distributions were free, resulting in superior price/performance

comparisons for simple metrics, such as dollars-per-megaflop and the Linpack benchmark for many applications. By the late "aughties," clusters were far-and-away the dominate architecture in the HPC industry.

But for a large number of scientific and engineering applications, this transition came at a cost that belied the apparent price/performance superiority of clusters. SMP architectures offered single, flat, shared-memory spaces up to tens or even hundreds of processors, whereas clusters – largely 32-bit clusters at the time – were necessarily distributed-memory systems, with each processor limited to addressing the relatively small amount of memory that was local to the node. Thus began the transition in programming models to message-passing interface (MPI).

In an MPI model, each processing unit (typically a core or a node) is designated to be responsible for a cell of local computation within an application; the cells work in concert to produce the overall result. In most situations, the cells must exchange information about what is happening locally, as it will have an effect on what happens remotely. (In the extreme, consider the "butterfly effect" that describes chaos theory as applied to weather forecasting; a butterfly flapping its wings in California ultimately has a long-term effect on the weather in Beijing.) Observing an MPI application in action can then be vaguely like watching team juggling. During one beat the juggler has an internal action, tossing a pin to himself or switching hands, analogous to a core or node performing its own calculations. During the next beat the jugglers all throw (messages are sent), and on the third beat the jugglers all catch (messages are received). Note the importance of staying in synch; the jugglers must throw and catch at the same time, and so it is for MPI applications.

Comparison of Distributed-Memory Cluster to Shared-Memory SGI Altix UV
Source: SGI



The challenges for MPI have been multifold. Firstly, not all applications can easily be parallelized to accommodate a distributed-memory architecture, due to the high degree of data dependency between cluster nodes. It took some applications years to get ported to MPI, and some few still hold out on shared-memory programming platforms. Second, once an application is ported to MPI, optimization becomes a challenge, because of the need to keep messages in

Intersect360 Research
P.O. Box 60296 | Sunnyvale, CA 94088 | Tel. [888] 256-0124
www.Intersect360.com | info@InterSect360.com

synch, thereby tending to slow the performance of an application down to the speed of the slowest node. And finally, distributed-memory clusters tend to present more difficulties in management and administration than their shared-memory counterparts.

The fact that the market overcame these burdens is testament to what lengths the HPC industry will go in search of superior price/performance characteristics. The drive for the lower-cost, industry-standard, best-of-breed components – x86 processors and Linux – was reason enough to forsake the qualities of shared memory for a new, wild frontier of computing.

The New, More Capable COTS

With the market well in adoption of COTS components, an interesting and perhaps predictable thing happened; COTS components slowly and steadily became more capable. x86 processors added 64-bit capabilities, widening the range of addressable memory spaces, while the open-source community rallied around initiatives to make Linux more scalable. Various initiatives have even pursued virtual shared-memory capabilities for clusters.

On top of these enhancements, the HPC industry has begun to usher in yet another new technology wave with the advent of multi-core processors. Again we see a technological advancement that is destined to deliver more peak performance per dollar than other chip architectures have managed, and yet again there is an efficiency challenge. Putting multiple cores onto a socket is similar to putting multiple nodes on a switch. There is a new level of parallelism introduced, and it can be difficult for application programmers to efficiently locate data in memory relative to program threads on a core. This challenge can become increasingly daunting as data sizes scale with the complexity of simulations.

As a result of these new levels of COTS capability, combined with increasing parallelism difficulties, many HPC users are now showing a renewed interest in shared-memory programming on SMP architectures. If an SMP could be built on lower-cost components, some applications might see benefits in returning to shared-memory architecture. Shared-memory systems can hold large datasets in memory without breaking them up, allowing large-scale, in-core computations, and similarly, they offer flat I/O spaces, easing data managements for large data sets. But despite the improvements in both x86 and Linux, one final, necessary component has been absent: the interconnect.

For shared-memory models to work effectively, the interconnects must have sufficient routing and performance characteristics, including remote memory latency – the time it takes for a processor to send or receive data from memory on another node – that is low enough to maintain cache coherency, a consistent, stable map of the data resident in cache throughout the system. Generational enhancements in Infiniband and Ethernet have been impressive but not sufficient for managing the task. At this point, large-scale shared memory still requires a system vendor that it committed to building shared-memory scalability.

SGI ALTIX UV

SGI HPC History: Remembering Memory

Throughout significant changes – both good and bad – in the company's fortunes, SGI has remained committed to building large-scale, shared-memory systems as a nexus of differentiation in its HPC strategy. For many years monolithic Origin and Altix systems that were driven by SGI's unique non-uniform memory access (NUMA) architecture offered the ultimate in big-memory scalability. (In November 2003, an SGI Altix 3000 was the first system to break the 1TB/sec memory bandwidth mark on the STREAM Triad benchmark, simultaneously becoming the first non-vector system to hold the Triad record.) But ultimately SGI suffered in price/performance comparisons of MIPS and Itanium processors

Intersect360 Research
P.O. Box 60296 | Sunnyvale, CA 94088 | Tel. [888] 256-0124
www.Intersect360.com | info@InterSect360.com

versus x86, and as its HPC market share dwindled, software providers and administrators became less likely to support SGI IRIX over various distributions of Linux.

SGI's dilemma was based on the fact that the company had differentiation that mattered for many HPC applications (NUMAlink and shared memory), but it sat on top of technologies that the market had gradually abandoned (MIPS, Itanium, and IRIX). SGI released distributed-memory products to round out its portfolio, but two tech recessions in six years (in 2002-03 and 2008-09) stood in the way of a turnaround.

In April 2009 the assets of SGI (including the brand) were acquired by Rackable, and SGI's product roadmap showed renewed strength, springing back to life as the new SGI organization moved quickly to consolidate strategies. SGI established quickly that HPC would continue to be core to SGI, with scalable architectures still forming the cornerstone of SGI's differentiation.

SGI Altix UV

With the Altix UV, the reemerged SGI is taking its tradition of scalability into a new dimension. Built on NUMAlink 5, SGI's newest, fastest interconnect, Altix UV has a latent HPC industry need in its crosshairs: a shared-memory system built on best-of-breed COTS components and a standard operating environment.

With roots in SGI NUMA architectures, SGI Altix UV is a blade-based family of systems designed to incorporate Intel® Xeon® processors, including the Intel® Xeon® processor E7 family. Scaling from 32 to 2,048 cores, Altix UV stretches the Intel® Xeon® processor E7 family to their architectural limit of 16 terabytes in a single, shared-memory image. Furthermore Altix UV follows the path blazed by the earlier SGI Altix systems in running either Red Hat or Novell SuSE Linux operating systems. The result is a scalable, open-platform, shared-memory environment designed for big-data across HPC and enterprise environments, including complex event processing, large-scale analytics, and database applications.

Altix UV comes in three sizes, depending on the level of scalability required by the application. The Altix UV 10 is a quad-socket, 4U rack-mounted server that can support up to 32 cores and 512GB of memory, delivering relatively large memory configurations to entry-level HPC and also usable as a powerful node in a cluster. Altix UV 100 targets midrange HPC applications with up to 768 cores and 6TB of shared memory in two industry-standard racks. And at the high end, Altix UV 1000 stretches the line into the supercomputing class, with up to 2,048 cores (over 18 teraflops with current processors) and 16TB of shared memory – the architectural limit of Intel® Xeon® processor E7 family – in four racks.

SGI Altix UV
Source: SGI

How an application makes use of this scalability will vary. Obviously many parallel applications will scale comfortably across the system fabric, but the potential benefits of the shared-memory architecture are not limited to that paradigm. Even a single-threaded application running on one core might take advantage of a very large data set in memory. In other cases, individual applications might not take advantage of the large memory space available on the Altix UV, but entire workflows using a number of applications could be run simultaneously, using the memory to communicate between applications.

Intersect360 Research
P.O. Box 60296 | Sunnyvale, CA 94088 | Tel. [888] 256-0124
www.Intersect360.com | info@InterSect360.com

Altix UV Architecture

The key enabling feature of SGI Altix UV is the NUMAlink 5 interconnect, with additional performance characteristics contributed by the on-node hub and its MPI Offload Engine (MOE), as well as the intra-node Intel QuickPath Interconnect (QPI). It is this combination that provides the low latency and interconnect capabilities that enable global shared memory.

The fifth-generation NUMA architecture is SGI's latest scalability engine. Each NUMAlink 5 router supports 15GB/sec of system bandwidth and can be combined in a number of topologies to expand to a full 15TB/sec of bisection bandwidth for the largest systems. Single-rack systems are connected in a switched dual-plane topology with no more than three NUMAlink hops across the system. Larger configurations can be reached with fat-tree implementations, as well as groups of fat-trees in a torus.

Within the server node, Altix UV leverages the Intel QPI interconnect, which delivers traffic from the Intel® Xeon® processor E7 family to the NUMAlink hub. In addition to the direct connections to on-node memory, each socket has two QPIs to the other socket, one QPI to the NUMAlink hub, and one to the I/O hub, for optional I/O attachments to the node. Together the four QPI connections have an aggregate bandwidth over 100GB/sec.

MOE is designed to take MPI process communications off the microprocessors, thereby reducing CPU overhead and lowering memory access latency, and thus improving MPI application performance and scalability. MOE allows MPI tasks to be handled in the hub, freeing the processor for computation. This highly desirable concept is being pursued by various switch providers in the HPC cluster arena; SGI implements it as part of the NUMAlink architecture.

Accessing MOE is straightforward for applications that use SGI's Message Passing Toolkit (MPT). MPT packs MPI implementation and optimization features for tuning MPI applications to the NUMAlink architecture. MPT and MOE are important components for SGI because they recognize the current market dynamics: today, even a shared-memory machine will need to run MPI jobs well.

INTERSECT360 RESEARCH ANALYSIS

For a long time the market ran away from SGI's strengths, not because they weren't appreciated, but because they sat on top of component technologies that were no longer competitive on price/performance metrics. Altix UV is designed to bring SGI shared memory systems – still remembered and loved throughout the industry – back into the ball game.

But if the technology is easier to adopt now than it has been in the past, SGI has another significant market dynamic to face. Generations of clusters have made MPI the dominant programming model, shortchanging some of SGI's differentiation. It is therefore critical for SGI to promote features like MPT and MOE to demonstrate that the NUMAlink architecture is nevertheless a strong performer for those applications.

MPT, MOE, and NUMAlink do make a concrete difference here, mostly in optimizing how a message is passed. In the global shared memory environment, each hub has access to all of the system memory, so if one node needs to send or receive data from another, that line can be read directly from or written directly to the memory of the remote node. SGI claims these MPI "gets" and "puts" are up to three times faster in NUMA shared memory than in competing cluster architectures.

Furthermore it is worth noting that even though most existing applications today are run in MPI, most software development begins in a desktop SMP environment. In the medium to long term we could see a renewed interest in programming for shared-memory, for example with OpenMP in place on MPI. According to a 2010 Intersect360

Intersect360 Research
P.O. Box 60296 | Sunnyvale, CA 94088 | Tel. [888] 256-0124
www.Intersect360.com | info@InterSect360.com

Research Study, nearly as many software developers for cluster environments use OpenMP (26% of respondents) for part of their development environment as there are using MPI (30%).[1] Additionally some SGI users are implementing a programming model known as PGAS (partitioned global address space), through the use of the Unified Parallel C (UPC) language and compiler. Altix UV gives developers more opportunity to explore programming model options.

But fundamentally, HPC is about what the user can achieve, and it is this holy quest that SGI has always strived to enable with its architectures. In the past, external factors drove the market away from components packaged within SGI systems, and now SGI wants to recapture its audience's imagination – big data for big dreams. For now SGI is recapturing some market share in addition to mindshare; in our 2009 survey, SGI saw an increase in the number of sites reporting the use of SGI systems.[2]

How far SGI will penetrate with Altix UV is tied to how many users buy into the renewed vision of SGI, which in turn is tied to the number of end users who buy into a forward-looking vision of shared memory. If SGI gains a toehold and a groundswell of support, the company could quickly find that bringing a differentiated, scalable architecture to HPC never goes out of fashion for long.

[1] Intersect360 Research HPC Software Environments Survey, May 2010.
[2] Intersect360 Research HPC Market Advisory Service, HPC User Site Census data, 2010.

Intersect360 Research
P.O. Box 60296 | Sunnyvale, CA 94088 | Tel. [888] 256-0124
www.Intersect360.com | info@InterSect360.com