White Paper

SGI® Reconfigurable Application Specific Computing: Accelerating Production Workflows

Table of Contents

Introduction

As a pioneer in application-specific acceleration techniques and technology, SGI was an early provider of related hardware and software innovations that have been adopted in oil and gas exploration, defense and intelligence, bioinformatics, medical imaging, broadcast media, and other data-dependent industries. The fourth generation of the SGI® Reconfigurable Application Specific Computing (RASC™) solutions continue to make field-programmable gate arrays (FPGAs) a cost-effective, low-power alternative to compute clusters based on general-purpose processors.

This paper overviews the trends that are driving requirements for today's application-specific acceleration solutions, and explains the defining capabilities of the new SGI® RC100™ RASC™ technology. An example of an acceleration solution is also described.

1.0 Industry Trends

Large-scale high-performance computing (HPC) FPGA systems continue to deliver attractive performance compared to supercomputer cluster systems. Lower cost and smaller form factors contribute to the benefits of this approach, and the lower power consumption has become a major advantage as users run into facility limitations when scaling traditional compute resources.

The increasing adoption of FPGA-based technology is driving further advancements into systems. The current 90nm processes will most certainly progress to 65nm, continuing to reinforce the advantages of these platforms. To date, FPGA semiconductor developments have proceeded at a faster rate than general-purpose CPUs, and are expected to continue to do so.
There are still many challenges related to the application of FPGA acceleration solutions. These include:

- Ease of use – Traditional HPC programmers are used to the simplicity and maturity of FORTRAN development tools, while FPGA-related tools are still emerging. The steep learning curve and low productivity of circuit design with engineering design automation (EDA) tools make them impractical for these users.
- Time to solution – Getting started with FPGA-based systems can require long project times.
- Double-precision floating-point performance – These operations can require assists from powerful microprocessors.
- Memory bandwidth – High rates of computation place high demands on system-to-memory interconnects, and require very large memory, ranging from tens of gigabytes (GBs) to terabytes (TBs).

2.0 SGI RC100 RASC Technology Overview

The recently released SGI RC100 system combines the high-performance SGI Altix™ architecture with leading-edge FPGA technology. Provided by industry leader Xilinx, the Xilinx® Virtex™ FPGA has been integrated into the blade-based, highly scalable SGI RC100 system design.

This system leverages the expertise gained with earlier SGI application specific computing solutions including those with Tensor Processing Units (TPUs). TPUs have been deployed in the SGI Origin 2000 and Origin 3000 systems.

2.1 Hardware

Basic HW characteristics of the RC100 include:

- Two Xilinx Virtex-4 LX200 FPGAs
- Very low power consumption per FPGA (~ 10W)
- Very high bandwidth into the system interconnect using NUMALink™-4 (6.4 Gbyte/sec)
- Expandable memory (up to 5 SRAM DIMMs for a total of 40 MB)
- Excellent price-performance ratio, even compared to PCIx-based solutions

2.2 Software

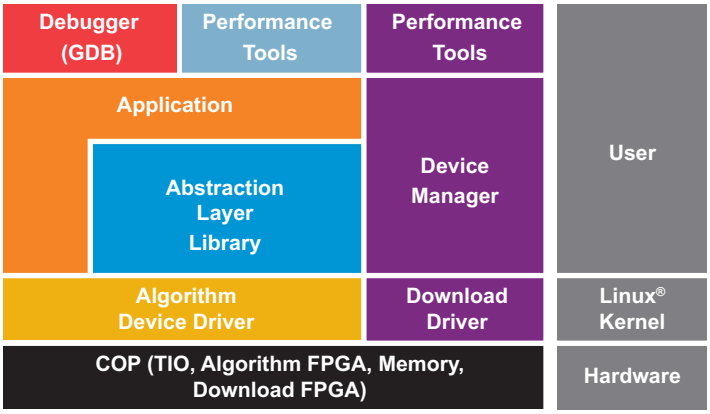
The RC100 HW solution is supported with a proven system software stack (see Figure 1):

- RASCLib API: This easy-to-use API for FPGAs includes unique features such as wide and deep scaling
- Debugging: The SGI RASC-aware debugger is based on gdb with FPGA-specific extensions
- Algorithms: VHDL or Verilog-based algorithms are fully supported, with example implementations

SGI is also a provider of high-level third-party tools including Celoxica™ Handel-C™ and the Mitrionics™ Mitrion™ platform.

During the years of providing industry-leading FPGA-based platforms, SGI carefully analyzed feedback from early adopters. With this feedback, the SGI RASC solutions have been enhanced for usability and easy deployment. SGI has also created a strong cross-divisional team of application-specific acceleration experts and Professional Services personnel that are able to provide support and implementation assistance for FPGA-based deployments.

Figure 1. The SGI RC100 Software Stack



3.0 Benefits of the SGI Application-Specific Acceleration Approach

3.1 Scalability

To preserve investments in software, users can scale applications by taking advantage of a simple scalability parameter (see Table 1). The SGI RASC abstraction layer (RASCLib) processes this parameter and scales the algorithms across the specified number of FPGAs (wide scaling). The SGI RASC platforms deliver almost linear scaling of large HPC applications, with very high efficiency and the ability to preserve algorithm code.

Table 1. Code Example – Wide Scaling on SGI RASC

Using 1 FPGA	strcpy(ar.algorithm_id,"my_sort");
	ar.num_devices = **1**;
	rasclib_resource_alloc(&ar,1);

Using 32 FPGAs	strcpy(ar.algorithm_id,"my_sort");
	ar.num_devices = **32;**
	rasclib_resource_alloc(&ar,1);

RASCLib and core services execute transparently to provide this scalability to the user. SGI has showcased application scalability across more than 30 FPGAs in a single system, with total efficiency measured at greater than 90% for a single application.

3.2 Simplified Development of Portable Algorithms

Using traditional electrical engineering tools, an HPC user will find that designing circuits for algorithm implementation is a slow-going process. To optimize the development flow, SGI and Mitrionics have partnered to offer an integrated tool flow tailored for programmers familiar with the C language.

Based on the Mitrionics Mitrion platform, the algorithm development flow includes the start-to-finish generation of an FPGA application. The Mitrion platform is based on the Mitrion Virtual Processor, a configurable, fine-grained, massively parallel, non-von Neumann processor. Users without any hardware design expertise can develop algorithms for an FPGA by writing a software program for the Mitrion Virtual Processor and loading the processor into the FPGA. All hardware-related steps, such as synthesis and "place and route," are carried out with the push of a single GUI button. The Mitrion Virtual Processor is programmed in the intrinsically parallel programming language, Mitrion-C. The Mitrion platform includes a comprehensive simulation and performance analysis environment, as well as an easy-to-understand format for reviewing results from all steps in the flow.

This approach opens FPGA programming to all HPC users. Without requiring proficiency with electrical engineering tools, the combined SGI/Mitrionics solution lets users focus on HPC problem solving. The very short learning curve (less then one week) eliminates the delay typical with most FPGA-based systems, and is also a portable framework that will be ported to future SGI platforms to ensure consistency in the development flow. The combination of the unique Mitrion platform and the inherent scalability of the SGI RASCLib architecture provide unmatched performance and ease of use, while also reducing design cycles from months to days.

SGI is a key participant and sponsor of the openfpga group, working to evolve FPGA standards and ensure long-term portability of codes for current and future FPGA users.

3.3 Reliability, Availability, and Serviceability (RAS)

SGI's history for designing and building highly reliable systems goes back to the SGI MIPS® processors, which included fully fault-tolerant features such as "lock step." The company's long-term experience also includes delivering large and ultra-scale

systems with more then 10,000 processors and up to 13 TBs of main memory. RAS have been built into the SGI RASC solutions "from the ground up." The SGI RC100 blade includes parity or ECC error checking in all critical data paths, offline diagnostic capabilities, and full integration with the SGI system controller network.

In contrast, PCIx-based solutions do not offer these capabilities, making them unsuitable for large-scale technical computing systems or enterprise solutions that require continuous operations.

3.4 Turn-Key Systems

The SGI RC100 RASC blades are integrated and delivered as part of a turn-key system. The systems include fully integrated and tested applications, tuned to meet customer requirements. SGI integration efforts include programming the FPGAs and optimizing FPGA algorithm implementations. SGI can also provide solutions that interface with Star-P software.

SGI has RASC system integration expertise in a broad range of markets/applications, including:

- Bio-informatics
- Security (fingerprint identification, voice identification)
- Encryption
- Geo-physics
- Image processing

3.5 Rapid Proof of Concept

SGI maintains strategic relationships with the leading tool vendors, and can assist during customer evaluation cycles by providing access to systems and previously implemented algorithms. These "proof of concept" (POC) demonstrations can be used for benchmarking, or for exercising tools first hand. In most cases, SGI is able to accommodate proof of concept requests with results returned in two to five days, allowing customers to make better purchase decisions and minimize project risks. POC demonstrations are available free of charge to potential customers.

3.6 Application-Specific Libraries

In addition to turn-key systems and POC demonstrations, SGI application engineers also produce and offer customized sets of library functions tailored to specific applications. These libraries become solution building blocks and give customers the benefit of proven FPGA programming expertise.

3.7 Joint SGI-Mitrionics Training

SGI provides training for both Handel-C and Mitrion-C, either at SGI or at customer sites.

SGI and Mitrionics jointly deliver one-day workshops covering FPGA programming techniques. Attendees can learn enough to run a first FPGA program in less than a day. Seminars provide users with access to current SGI RC100 hardware and software.

The workshops are open to anyone interested in FPGA applications, and provide an excellent opportunity to jump-start application development projects. Some students come with FORTRAN program skeletons, and many have working FPGA applications at the end of the first day even without any prior FPGA knowledge.

For information about schedules and locations for SGI-Mitrionics seminars, visit www.sgi.com. SGI also sponsors easy information exchange between users and SGI's FPGA engineering and application teams.

3.8 VHDL and Verilog Developer Resources

For existing FPGA programmers and ASIC designers, SGI provides a full-function interface to VHDL and Verilog. This includes example implementations of basic algorithms as well as access to SGI's core service source code.

4.0 SGI RASC Solutions and BLAST

Advances in DNA sequencing techniques have lead to an exponential growth in the sizes of genetic and protein databases. Growth in these databases outstrips the progress of traditional processors, meaning that searching these databases actually becomes slower over time, even as the processors evolve. Therefore, the Basic Local Alignment Search Tool (BLAST) is an example of an application that would benefit from the order of magnitude performance increases available from an FPGA-based compute platform.

Developed by the National Center for Biotechnology Information (NCBI), BLAST gives researchers and industry users a tool for searching public gene and protein sequence databases. Depending on the size of the database being searched, BLAST can take many hours or even days to return results. Since the search process involves several highly repetitive operations, the BLAST algorithms can be greatly accelerated when executed on a system such as the SGI Altix equipped with SGI RC100 RASC blades.

4.1 The Processing Flow

The BLAST application consists of three stages, where the database being searched is streamed through a series of successive filters that have been configured by the search query. Each consecutive filter is more restrictive than the previous step. Patterns are either considered a match and passed on, or discarded in each filter. From analyzing and profiling the NCBI BLAST code, it is apparent that the vast majority of time is spent

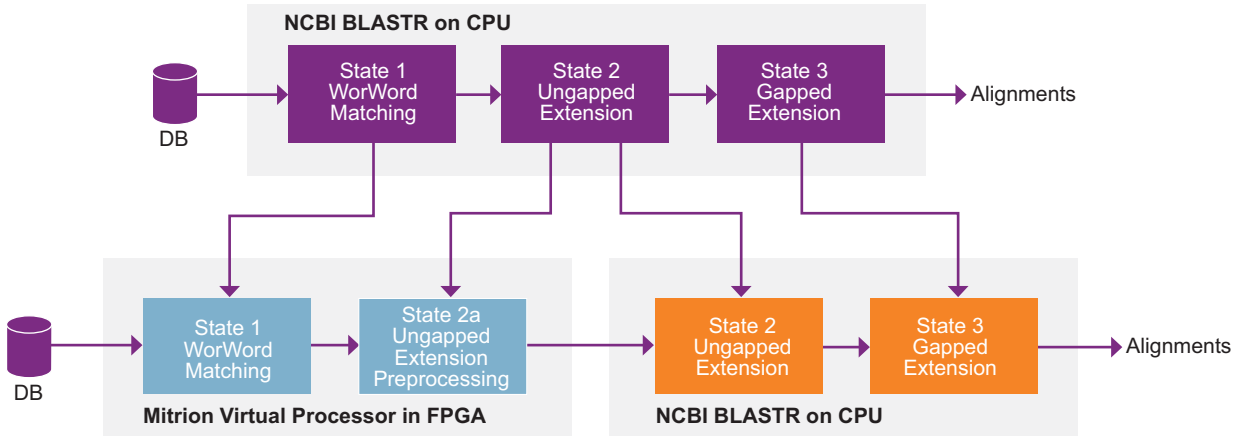


Figure 2. Three-stage deployment of BLAST

in the first two stages. As a result of these findings, Mitrionics' deployment focused on acceleration of the first two stages and left the third stage to the host processor. See Figure 2 for an architectural overview.

In the FPGA-accelerated version of NCBI BLAST, the RC100 unit is loaded with a Mitrion Virtual Processor that has been programmed with, and adapted to, stage 1 and parts of stage 2 of the BLAST program for nucleotide searches. The query is uploaded to the FPGA and the nucleotide database is streamed through. In stage one, the BLAST program looks for exact patterns of a certain length (11 bases by default for nucleotide searches). The accelerated parts of stage 2 then process the matches, or "seeds," by extending them within a fixed window. The alignments that pass this stage are passed from the Mitrion Virtual Processor to the NCBI BLAST software on the Altix host, where they are further extended, producing the final alignment sequences and the score based on the overall match.

4.2 Stage 1

The first stage is divided into two sub-stages (see Figure 3). First a fast probabilistic filter returns a list of probable exact matches. This stage is implemented using bloom filters, a space-efficient probabilistic data structure that is used to test whether an element in the database is a member of the set of query words.

Each bloom filter consists of a number of hash functions that check whether a specific bit is set or not in memory. This is where the FPGA architecture on the RC100 excels. Each FPGA has multiple independently accessible distributed memories that can all be used by the Mitrion Virtual Processor simultaneously each clock cycle. The flexibility of the Mitrion Virtual Processor allows these resources to be used where most needed and makes it possible to run a number of bloom filters in parallel, depending on the query size. The Mitrion Virtual Processor provides a sustained lookup rate of 16 loads per clock cycle for a 100k query and 64 loads per cycle for a 10k query. The throughput of the first stage is 400 megabases/second for a 100k query and 1.6 gigabases/second for a 10k query.

The bloom filters may produce false positives, but they will not produce any false negatives. The second part of stage 1 is a lookup in a hash table to filter out the false positives and to look up the position of the word within the query. The hash table can sustain one lookup per clock cycle to one of the SRAM memory banks connected locally to the FPGA on the RC100 blade, while the database is read simultaneously from another local SRAM memory bank. Since the pre-filter discards the significant amounts of the initial data, the performance requirements on the hash table lookup stage are much lower than on the bloom filters. The matches that pass the hash table look-up are forwarded to stage 2.

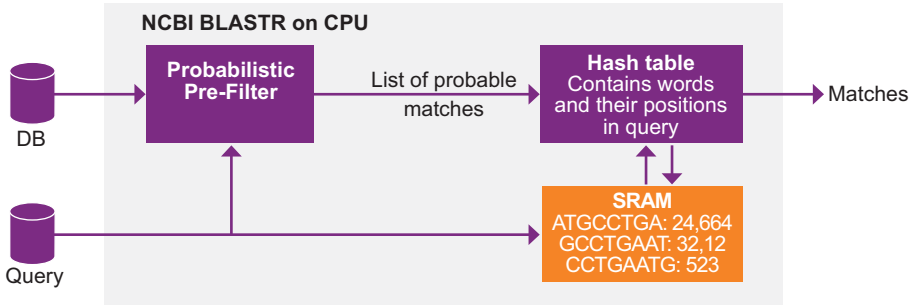
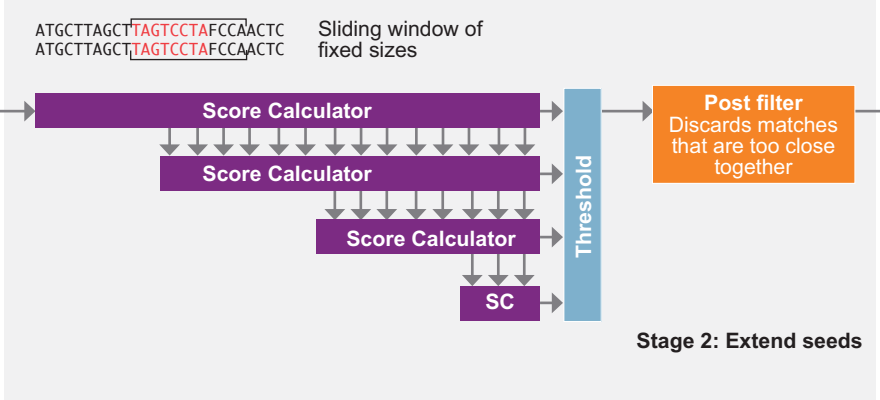


Figure 3. Stage 1 processing

Figure 4. Stage 2 processing



4.3 Stage 2

The first part of the second stage (see Figure 4) extends the seed within a fixed window, returning matches that pass the threshold score and discarding the rest. This part is implemented as an unrolled systolic array that is easily expressed as a Mitrion-C loop, which finds the best start position, stop position, and score of two 64-character words in a single clock cycle.

All alignments that pass the threshold score are passed to a filter that discards matches that are so close to each other that they could not be extended without being joined. This part is done in stage 1 in the NCBI software implementation, but in the Mitrion implementation, it is executed after the un-gapped extension part of stage 2 processing. This is due to the fact that it enhances the performance of the algorithm since the speed of the un-gapped extension stage is so much better on the FPGA.

Matches that pass the final filter are returned from the FPGA to the Altix host. Here, the un-accelerated parts of NCBI BLAST may extend them even further.

4.4 SGI RC100 as a Platform for Mitrion-Accelerated BLAST

The final performance of an FPGA-accelerated application is determined by a number of factors, such as FPGA size, I/O speed, access to local storage for the FPGA, or the processing power of the host CPU. The critical factors depend on the characteristics of the application being accelerated.

The overall system architecture of the SGI Altix Server with RC100 blades (see Figure 5) combines high-performance CPU processing, high-speed NUMAlink data transfers, and large FPGAs with five local memory banks each. In addition, the applications running on the CPU have access to the RASCAL API to configure and communicate with the FPGA. On the FPGA, Core Services connect the Mitrion Virtual Processor (or other computing circuitry) to the RC100 memories and I/O channels.

The Mitrion-accelerated BLAST version relies on the RC100 architecture to provide simultaneous access to multiple SRAM memory banks to stream the nucleotide database through the

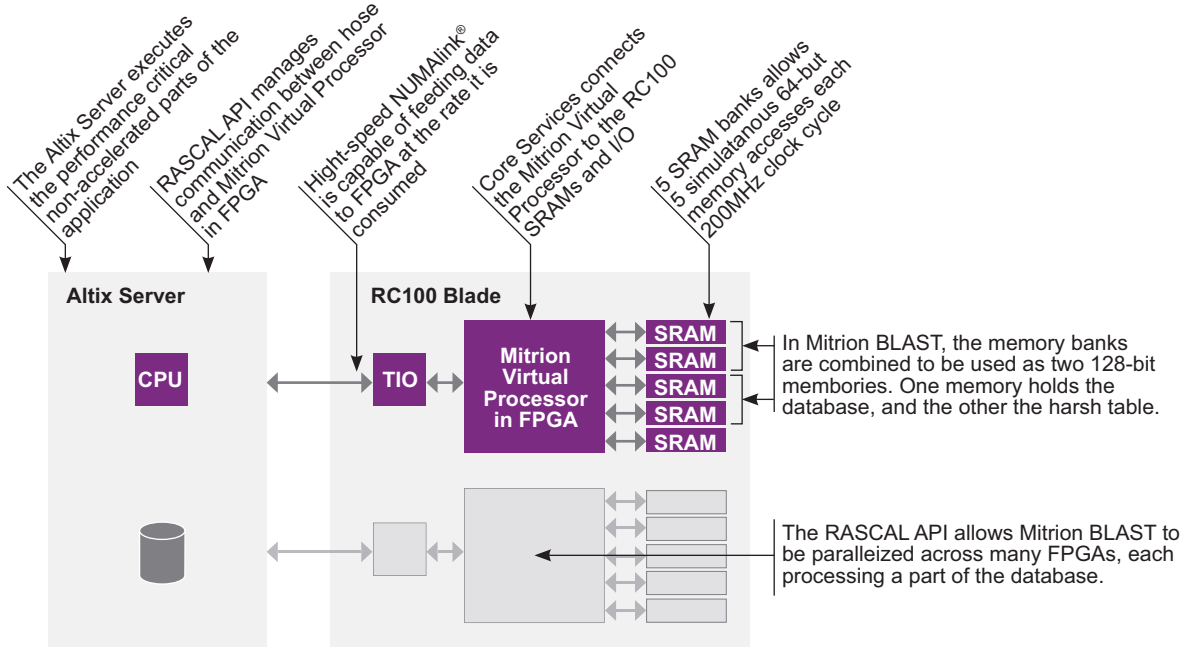


Figure 5. Mitrion-accelerated BLAST and the SGI RC100 system

FPGA and at the same time do continuous hash table look ups. The large Xilinx Virtex-4 FPGA enables extensive parallelization of the parts of the algorithm that are most critical for the performance. The tight integration of the RC100 FPGAs with NUMAlink provides the I/O performance needed by the Mitrion Virtual Processor to feed the 1.6 gigabases/s to the bloom filters.

The BLAST algorithm, like many other algorithms, has parts that are a good fit for FPGA acceleration and other parts that run more efficiently on a CPU. The accelerated BLAST application makes use of the Altix Itanium2 CPU to perform parts of un-gapped extension stage, all of the gapped extension stage, and management of the setup of the FPGA. The power of the Altix CPU and the RASCLib API brings high levels of efficiency to the non-FPGA-accelerated parts of the BLAST application.

4.5 Scaling

The RASCLib API allows for straightforward wide scaling of the Mitrion-accelerated BLAST application to take advantage of the FPGAs available in a system. With a simple RASCLib API call, Mitrion accelerated BLAST can run in parallel on any number of FPGAs, processing the same query on segments of the database, thereby almost linearly decreasing the execution time with the number of FPGAs.

4.6 Results

The acceleration that has been obtained with the Mitrion Virtual Processor on the SGI RC100 system provides an order of magnitude leap in performance over a traditional CPU-only solution. With the Mitrion BLAST implementation, the speedup varies with the queries used and the databases being searched.

As one measure of the acceleration, a set of 6 query sequences (average length of 115,000 letters) from the Drosophila Genome was searched against the GenBank Mouse EST Database (4,720,060 sequences; 2,193,794,199 total letters) using the original and the FPGA-accelerated version of NCBI BLAST 2.2.13 with default options (see Figure 6). Both versions were run on a 1.6GHz 6MB L3 SGI Altix 4700 system. The average improvement for the FPGA accelerated version of BLAST was a factor of 17.5 compared to the original version of BLAST.

Several methods exist to increase performance even further. Re-arranging the memory access to four 64-bit memories has the potential of increasing performance another factor of two, in particular for shorter queries. Duplicating the hash table to make use of the fifth memory bank could increase performance by a factor of two for long queries.

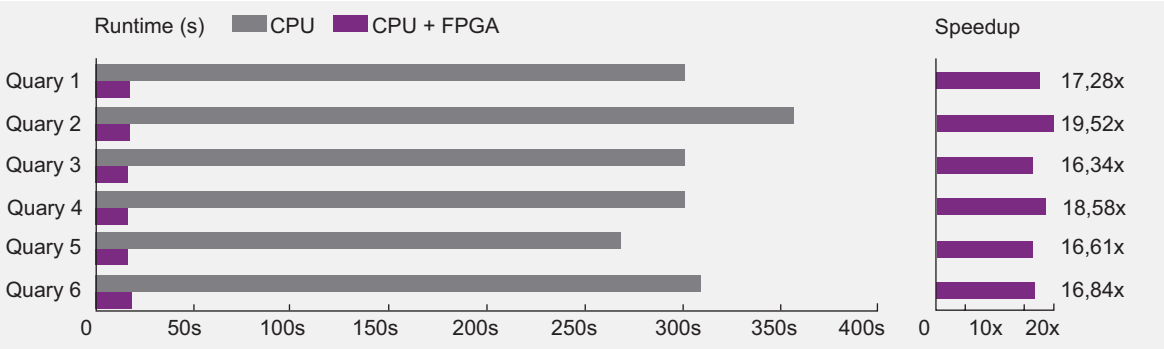
Conclusion

The SGI RC100 RASC solution offers a time saving alternative for accelerating very compute-intensive programs. With the industry's largest FPGA, very large configurable local memory, and a highly scalable system, solutions can be architected rapidly and the investment easily leveraged across multiple projects.

The fully integrated and tested SGI solution, with a comprehensive software tool and application environment, offers users a commercially robust platform that is easier to manage and support over the life of the system. Configurations can be created with up to 256 FPGAs and up to GBs of memory.

For more information about the SGI RC100 RASC solution, please visit www.sgi.com.

Figure 5. Mitrion-accelerated BLAST and the SGI RC100 system



Corporate Office
SGI
1140 East Arques Avenue
Sunnyvale, CA 94085-4602
650.960.1980

North America +1 800.800.7441
Latin America +55 11.5185.2860
Europe +44 118.912.7500
Japan +81 3.5488.1811
Asia Pacific +1 650.933.3000

3984 [12.2006]

J15222