

# **A Flexible Software Architecture for High Performance Synthetic Aperture Processing**

***Mr. Brian Markle, Chief Technology Officer***

Array Systems Computing Inc.

1120 Finch Avenue West, 7<sup>th</sup> Floor, Toronto, Ontario, Canada M3J 3H7

Phone: (416) 736-9000 Fax: 416-736-4715

Email: [www.array.ca](http://www.array.ca), [bmarkle@array.ca](mailto:bmarkle@array.ca)

In 1968 R.W. Hamming stated “Perhaps the central problem we face in all of computer science is how we are to get to the situation where we build on top of the work of others rather than redoing so much of it in a trivially different way.” This remains the central problem in computer engineering to this day, and in this article we show that the emergence and growing dominance of Open Source and Open Systems provide a logical foundation on which to build a flexible software architecture that promotes the reuse of existing software through the use of Generic Signal Processing Blocks (GSPBs).

## **The Need for a Scalable and Flexible Software Architecture**

When considering replacing their existing acoustic data analysis systems at the Swedish Naval Underwater Sensors Analysis Centre (MUSAC), a need was seen for a flexible and scalable high performance general-purpose signal processing system that could process both sonar and non-sonar data.

Array Systems Computing, of Toronto Canada, was awarded a contract by Försvarets Materielverk (FMV) of Sweden to design and build the SAPPS (Sonar Acoustic Post Processing System) system, based on our winning design ideas:

- Generic Signal Processor (GSP)
- Universal Beam Former

The Generic Signal Processor (GSP) needed to be scalable and flexible. The scalability would allow processors to be added or removed, without needing to recompile the existing software. This would give the user the ability to boost system performance when new high performance processors are available in the marketplace. A flexibility requirement of the GSP design is that new applications need to be easily constructed by the user, through the use of a toolbox of “Lego style” building blocks, called Generic Signal Processing Blocks (GSPBs). New GSPBs can also be constructed by the user, and added to the system without needing to recompile.

In determining a platform for the Generic Signal Processor, we looked at trends in High Performance Computing (HPC) for direction.

## **Practical Impact of Trends in HPC**

Current trends in HPC for the world’s top supercomputers give us insight into the future.

Although Charles Galton Darwin wrote: “Anyone who attempts to predict the history of the next ten years is a rash man”, it did not prevent him from writing his book “The Next Million Years”. We will attempt something much more modest - looking only one year into the future.

Consider the 5 most powerful computers in the world (TOP500 Supercomputers List, <http://www.top500.org>):

- Earth Simulator, Yokohama, NEC, Rmax: 35.86 Tflops
- ASCI Q, Los Alamos, HP Alphaserver SC, Rmax: 7.72 Tflops (there are two such systems at Los Alamos)
- ASCI WHITE, Livermore, IBM SP Power3, Rmax: 7.22 Tflops
- MRC Linux Cluster, Livermore, Linus Networks/Quadrics, Rmax: 5.69 Tflops

What’s wrong with this picture from the perspective of 1994, when the ASCI program was initiated? As the humorous American baseball player Yogi Berra quipped, “It’s hard to make predictions, especially about the future.” Yogi Berra’s statement definitely applies to what is currently the fifth most powerful computer in the world. In 1994 it would have been considered laughable that a cluster of *PCs* running *Linux* could challenge the Department of Energy’s Accelerated Strategic Computing Initiative.

We will make our modest prediction for 2004 in the full version of this paper.

## **Features and Benefits of Array’s Scalable GSP Technology**

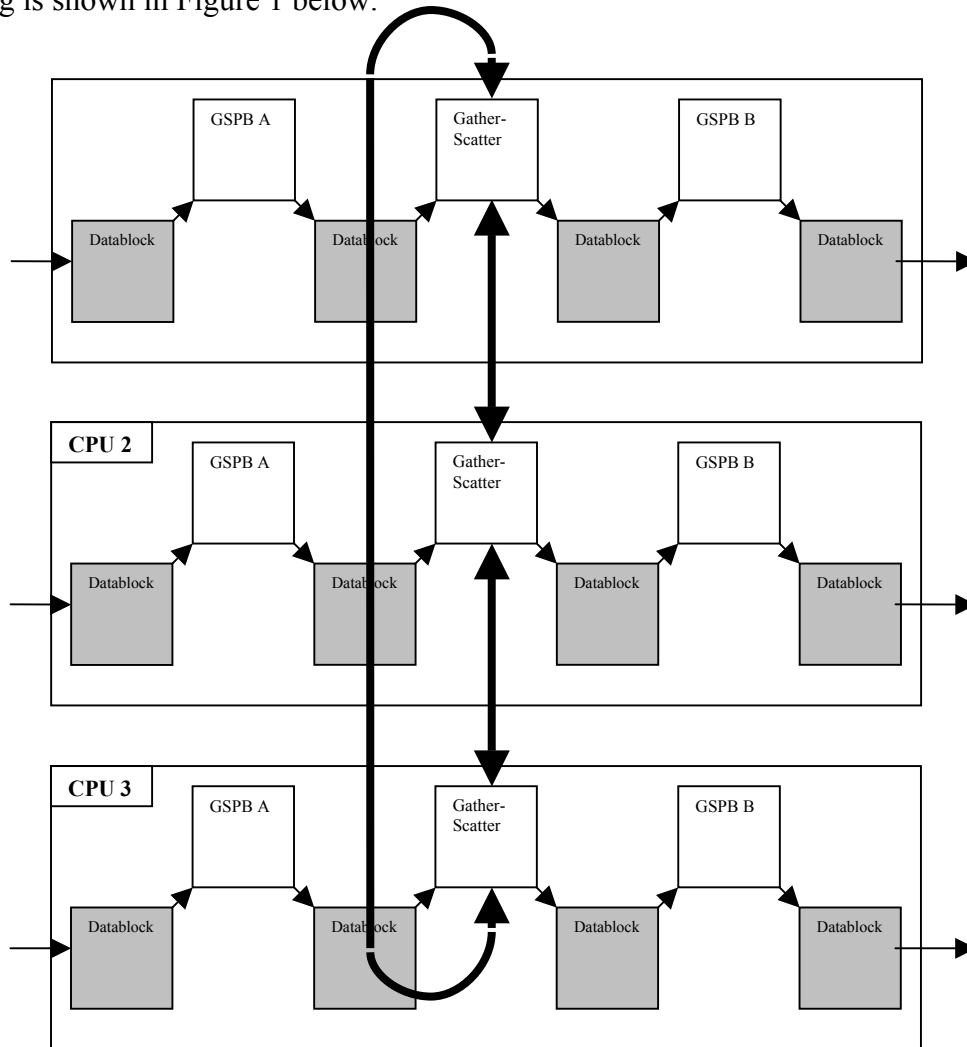
### ***Object-Oriented Auto-Scaling Generic Signal Processor Software***

The GSP is designed following an object-oriented approach using the C++ programming language. Object oriented design techniques allow the software to be flexible and extensible. The most important aspect of extensibility is the support for user-written signal processing modules. All signal processing operations in the GSP framework are implemented as individual GSPBs. All signal processing modules are classes derived from the GSPB base class, allowing the GSP engine to treat all GSPBs identically (polymorphism ensures that the correct methods of the derived classes are called).

### ***Automatic Load Balancing***

The GSPB framework supports the parallelization of both non-adaptive and adaptive processing such as FIR (Finite Impulse Response) and IIR filters. Suppose we need to construct a stream of  $n$  GSPBs, each of which needs a data input size that is a multiple of a different prime number,  $p_1, p_2, \dots, p_n$ . The only way to satisfy each of these constraints is to use an input block size that is a multiple of  $(p_1 * p_2 * \dots * p_n)$ . Such a buffer may require excessive memory. The solution to this problem is to insert a *GatherScatter*, between the GSPBs to convert the data from one block size to another. The re-blocking operation is performed automatically and is transparent to the GSPBs themselves.

For illustration, a processing stream with three GSPBs, running on three processors and featuring reblocking is shown in Figure 1 below.



**Figure 1. A Stream Including Re-Blocking**

### Future Directions

We have presented a software architecture that addresses a central problem in computer science.

Moore's law predicts that computing performance increases by a factor of 1.84 yearly, using statistics from <http://www.top500.org>. Based on Moore's law, Ray Kurzweil estimates that supercomputers will achieve one human brain capacity by 2010, and personal computers will do so by around 2020, on the assumption that the human brain computes at 20,000 TFlops. It is generally recognized that a major paradigm shift in the semiconductor basis of computing will be required in order for a personal computer to have the computing power of a human brain.

Since we are unlikely to witness a paradigm shift in the semiconductor basis for Moore's law within the next 10 years, the technology proposed in this paper can be the basis for a practical engineering tool for the next 10 years. Whereas the traditional semiconductor chip will approach some fundamental physical limits by the year 2017, alternate technologies such as DNA computing will come to the forefront of computing.

Brian Markle, Chief Technology Officer, Array Systems Computing Inc.  
[www.array.ca](http://www.array.ca), [bmarkle@array.ca](mailto:bmarkle@array.ca)

ENDS