Development of a Real-Time Parallel UHF SAR Image Processor

Matthew Alexander, Michael Vai, Thomas Emberley, Stephen Mooney, Joseph Rizzari
{maa, mvai, temberley, smooney, jrizzari}@ll.mit.edu
MIT Lincoln Laboratory, 244 Wood St., Lexington, MA 02420-9108

Introduction

MIT Lincoln Laboratory (MIT/LL) has developed a UHF SAR system for target detection by performing image formation, image processing, and change detection. This system is comprised of a variety of components including radar antennas, receivers, analog to digital converters, a FPGA-based front-end processor, and a programmable parallel signal processor. This abstract will focus on the design and development of the programmable parallel signal processor, which consists of 128 processing cores in 16 computing nodes. This processor, which real-time software has been developed with exceptional productivity, is one of the most capable airborne real-time processors developed at MIT/LL.

The UHF SAR algorithm is comprised of a number of steps that include: image formation, backlobe cancellation, focusing, target specific filtering, registration, and change detection. Each of these processing steps is implemented in the programmable parallel signal processor using an open architecture development approach. This approach allows for changes in underlying hardware and/or software without having to modify the software application itself.

Open architecture middleware software libraries were used in order to produce a real-time implementation of the UHF SAR algorithm within an aggressive development schedule. MIT/LL's Parallel Vector Library (PVL) enabled the software development team to produce a large-scale software application in a short amount of time without compromising open architecture objectives. Starting with a Matlab implementation of approximately 4500 lines of code, the development team produced 52K lines of application code in approximately 1250 staff-days (about 20 lines of code per day). The team also reused an estimated 27K lines of middleware code. Middleware libraries were the enabling technology that rendered feasible the development of an application of this complexity in a short amount of time.

The processing chain has very high computation throughput, memory, I/O, and storage requirements with respect to an airborne processor. The final system that performs image formation and image processing has a total of 16 computing nodes summing up to:

- 1.536 TFLOPS of throughput
- 512 gigabytes of memory
- 10 Tbytes of storage
- 32 Gbit/sec I/O

The signal processor development effort included rigorous integration and testing. After unit tests, the processor was first integrated with other sub-systems in a systems integration lab (SIL). Once the initial integration was complete, the UHF SAR system was installed on a Twin Otter aircraft acting as a surrogate airborne platform for engineering tests. This allowed initial radar testing while the deployable platform was still undergoing modifications and improvements. Finally, the end-to-end SAR UHF system was installed and tested on a Dash-8 aircraft.

UHF SAR Real-Time Processing

The software development team was responsible for the delivery of a real-time software application for the processing and exploiting of UHF SAR data. The UHF SAR application is a parallelized software component written in C++ that implements image formation and target specific processing as described in the introduction section.

An open architecture approach, outlined in Fig. 1, was used to facilitate processor scaling and portability. Open standard interfaces allow for changes in hardware and software without having to change the application. This is important not only for the sustainability and longevity of the system, but also for providing scalability during various integration and testing phases. For example, a scaled down system with only four computing nodes was used in an early engineering test to accommodate the more restrictive SWaP (size, weight, and power) constraints of the Twin Otter surrogate platform. After checking out the operations, the system was readily scaled up to all 16 computing nodes for final testing.

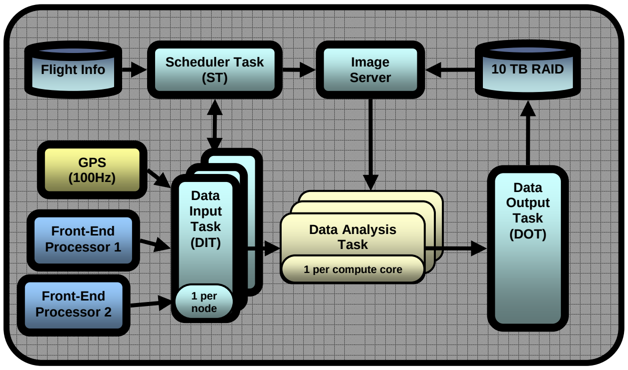

Figure 1. UHF SAR Open Architecture

The principal components of the open architecture infrastructure are data input tasks (DIT), data analysis tasks (DAT), and a data output task (DOT). The DIT is responsible for receiving data from the front end processor and GPS sub-systems, and distributing them to multiple cores for data analysis. The DAT is responsible

This work is sponsored by the Army under Air Force Contract FA8721-05-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

for executing the UHF SAR algorithm. Finally, the DOT aggregates results from each of the analysis tasks and writes them to files.

Fig. 2 shows a UHF SAR image generated by the programmable parallel processor. The image is a single-pass image; one of the five results for a single current coherent processing interval (CPI), also referred to as a “leg.” Calibration targets can be seen from top to bottom in the center of the image.

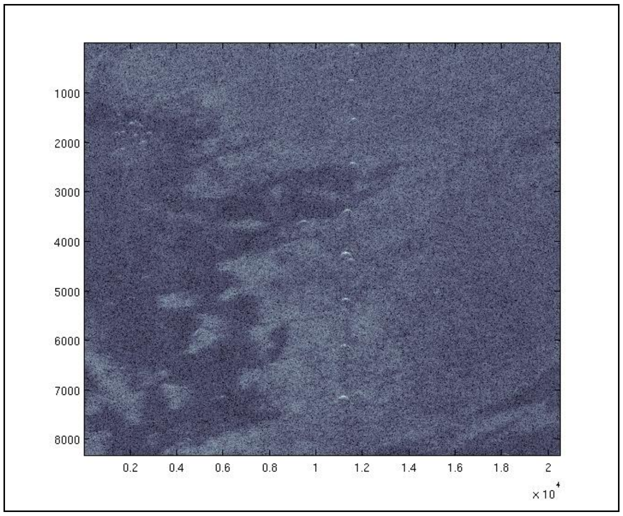

Figure 2. UHF SAR Processing Results

System Performance

To quantify computational throughput requirements, the real-time software team performed a complexity analysis of the UHF SAR algorithm. This consisted of deriving formulas that estimate, as a function of run-time parameters, the total amount of computational throughput, I/O, memory, and storage required by the UHF SAR processing. These formulas enabled the team to evaluate the hardware performance and determine whether or not the signal processing would meet real-time requirements.

Hardware Technology

In order to select a ruggedized hardware platform capable of performing the UHF SAR processing in real-time, the results from the complexity analysis were used to select a real-time hardware platform that fits within the SWaP constraints of the Dash-8 airplane. A cluster of 1U ruggedized Themis processor boxes were chosen for the UHF SAR real-time processing hardware. Each box contains two quad-core Intel Xeon processors operating at 3GHz. They also each contain 32GB of memory, and have dual gigabit Ethernet ports. A 10 terabyte RAID was used for the storage of historical and current imagery. For the final system, the real-time cluster consists of 16 Themis boxes; however, open architecture provides the ability to scale the number of processing units in order to conduct testing on a platform with more restrictive SWaP constraints and/or a relaxed real-time requirement.

Development Methodology

An advanced software engineering methodology, which is summarized in Figure 3, was utilized to expedite the development of the real-time UHF SAR processor. This methodology allowed an algorithm developer to migrate the signal processing code from a single processor (serial with Matlab) all the way to a multiple processor implementation (parallel with C++ code). The development methodology also exploited various hardware architectures such as a single node Intel processor and the target real-time COTS parallel processor (i.e., the Themis processing units). The ultimate objective was to produce parallel C/C++ code that would operate in real-time on the target platform.

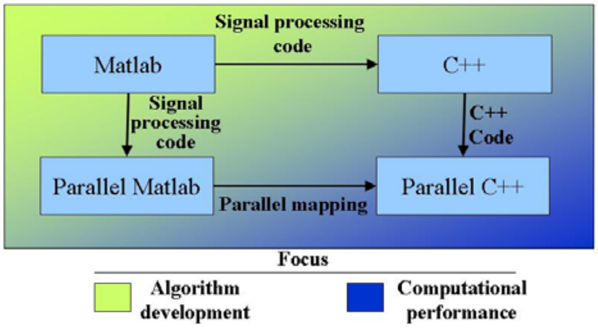

Figure 3. Development Methodology

Summary

Lincoln Laboratory has developed a real-time software application for the processing and exploiting of UHF SAR data. A ruggedized hardware platform was selected using computational throughput requirements to meet with the airborne platform SWaP constraints. Open architecture techniques were used to develop a real-time parallel programmable signal processor. Testing and integration of the programmable parallel signal processor were conducted in three stages: in a systems integration lab, in a surrogate airborne platform, and in the deployable airborne platform. The performance of this real-time processor was demonstrated by producing UHF SAR images within the required real-time latency during airborne operations.