order. All data saved in the system can be straightforwardly retrieved, exported, and migrated.

Using TSDSS's interactive data visualization tool, a user can conveniently choose any combination and mathematical computation of interesting telemetry points from a large range of time periods (life cycle of mission ground data and mission operations testing), and display a graphical and statistical view of the data. With this graphical user interface (GUI), the data queried graphs can be exported and saved in multiple formats. This GUI is especially useful in trending data analysis, debugging anomalies, and advanced data analysis. At the request of the user, mission-specific instrument performance assessment reports can be generated with a simple click of a button on the GUI.

From instrument level to observatory level, the TSDSS has been operating supporting functional and performance tests and refining system calibration algorithms and coefficients, in sync with the Aquarius/SAC-D spacecraft. At the time of this reporting, it was prepared and set up to perform anomaly investigation for mission operations preceding the Aquarius/SAC-D spacecraft launch on June 10, 2011.

This work was done by Lakesha Bates and Liang Hong of Goddard Space Flight Center. Further information is contained in a TSP (see page 1). GSC-16035-1

CropEx Web-Based Agricultural Monitoring and Decision Support

Changes in crop health are monitored over time.

Stennis Space Center, Mississippi

CropEx is a Web-based agricultural Decision Support System (DSS) that monitors changes in crop health over time. It is designed to be used by a wide range of both public and private organizations, including individual producers and regional government offices with a vested interest in tracking vegetation health. The database and data management system automatically retrieve and ingest data for the area of interest. Another stores results of the processing and supports the DSS. The processing engine will allow server-side analysis of imagery with support for image sub-setting and a set of core raster operations for image classification, creation of vegetation indices, and change detection.

The system includes the Web-based (CropEx) interface, data ingestion system, server-side processing engine, and a database processing engine. It contains a Web-based interface that has multi-tiered security profiles for multiple users. The interface provides the ability to identify areas of interest to specific users, user profiles, and methods of processing and data types for selected or created areas of interest. A compilation of programs is used to ingest available data into the system, classify that data, profile that data for quality, and make data available for the processing engine immediately upon the data's availability to the system (near real time).

The processing engine consists of methods and algorithms used to process the data in a real-time fashion without copying, storing, or moving the raw data. The engine makes results available to the database processing engine for storage and further manipulation. The database processing engine ingests data from the image processing engine, distills those results into numerical indices, and stores each index for an area of interest. This process happens each time new data is ingested and processed for the area of interest, and upon subsequent database entries, the database processing engine qualifies each value for each area of interest and conducts a logical processing of results indicating when and where thresholds are exceeded. Reports are provided at regular, operator-determined intervals that include variances from thresholds and links to view raw data for verification, if necessary.

The technology and method of development allow the code base to easily be modified for varied use in the real-time and near-real-time processing environments. In addition, the final product will be demonstrated as a means for rapid draft assessment of imagery.

This work was done by Craig Harvey and Joel Lawhead of Stennis Space Center.

Inquiries concerning rights for its commercial use should be addressed to:

NVision Solutions, Inc.
13131 Hwy 603
Stennis Technology Park, Suite 301
Bay St. Louis, MS 39520
Phone No.: (228) 242-0015
Refer to SSC-00312.

High-Performance Data Analysis Tools for Sun-Earth Connection Missions

Applications include medical image analysis, hyperspectral imaging, wildlife tracking, and sensor data processing.

Goddard Space Flight Center, Greenbelt, Maryland

The data analysis tool of choice for many Sun-Earth Connection missions is the Interactive Data Language (IDL) by ITT VIS. The increasing amount of data produced by these missions and the increasing complexity of image processing algorithms requires access to higher computing power. Parallel computing is a cost-effective way to increase the speed of computation, but algorithms oftentimes have to be modified to take advantage of parallel systems. Enhancing IDL to work on clusters gives scientists access to increased performance in a familiar programming environment. The goal of this project was to enable IDL applications to benefit from both computing clusters as well as graphics processing units (GPUs) for accelerating data analysis tasks.

The tool suite developed in this project enables scientists now to solve demanding data analysis problems in IDL that previously required specialized software, and it allows them to be solved orders of magnitude faster than on conventional PCs. The tool suite consists of three components: (1) TaskDL, a software tool that simplifies the creation and management of task farms, collections of tasks that can be processed independently and require only small amounts of data communication; (2) mpiDL, a tool that allows IDL developers to use the Message Passing Interface (MPI) inside IDL for problems that require large amounts of data to be exchanged among multiple processors; and (3) GPULib, a tool that simplifies the use of GPUs as mathematical coprocessors from within IDL.

mpiDL is unique in its support for the full MPI standard and its support of a broad range of MPI implementations. GPULib is unique in enabling users to take advantage of an inexpensive piece of hardware, possibly already installed in their computer, and achieve orders of magnitude faster execution time for numerically complex algorithms. TaskDL enables the simple setup and management of task farms on compute clusters.

The products developed in this project have the potential to interact, so one can build a cluster of PCs, each equipped with a GPU, and use mpiDL to communicate between the nodes and GPULib to accelerate the computations on each node.

This work was done by Peter Messmer of Tech-X Corporation for Goddard Space Flight Center. Further information is contained in a TSP (see page 1). GSC-15749-1

Experiment in Onboard Synthetic Aperture Radar Data Processing

The algorithm runs in a parallel/multicore environment, and integrates radiation hardening by software (RHBS) self-protection strategies.

Goddard Space Flight Center, Greenbelt, Maryland

Single event upsets (SEUs) are a threat to any computing system running on hardware that has not been physically radiation hardened. In addition to mandating the use of performance-limited, hardened heritage equipment, prior techniques for dealing with the SEU problem often involved hardware-based error detection and correction (EDAC). With limited computing resources, software-based EDAC, or any more elaborate recovery methods, were often not feasible. Synthetic aperture radars (SARs), when operated in the space environment, are interesting due to their relevance to NASAs objectives, but problematic in the sense of producing prodigious amounts of "raw" data. Prior implementations of the SAR data processing algorithm have been too slow, too computationally intensive, and require too much application memory for onboard execution to be a realistic option when using the type of heritage processing technology described above.

This standard C-language implementation of SAR data processing is distributed over many cores of a Tilera Multicore Processor, and employs novel Radiation Hardening by Software (RHBS) techniques designed to protect the component processes (one per core) and their shared application memory from the sort of SEUs expected in the space environment. The source code includes calls to Tilera APIs, and a specialized Tilera compiler is required to produce a Tilera executable. The compiled application reads input data describing the position and orientation of a radar platform, as well as its radar-burst data, over time and writes out processed data in a form that is useful for analysis of the radar observations.

The application is capable of recovering from some types of SEU-induced interference with component processes and/or corruption of the shared application memory, and also writes out performance statistics designed to assist in evaluating the effectiveness of the novel RHBS techniques employed. These performance data are useful in identifying, time-stamping, and (indirectly) geo-locating SEU incidents along with the application's responses.

The tileSAR software distributes the problem of processing SAR data over an "engine" made up of a number of cooperating parallel processes (one per core). This engine is replicated three times within the Tilera processor; always one process per core, and all engines running in parallel. Each engine also includes an additional scrubbing process (core), and there is one final triple-voting process external to the engines. When distributing the SAR algorithm among the processes of each engine, the usual single-stringed implementation (each sub-task executed in sequence) is replaced with an implementation where independent operations are carried out concurrently by independent processes. Every opportunity for concurrency within this algorithm is exploited, as this dramatically reduces execution time. The result of each engine's processing is a series of output records. The processes that make up each engine share a single working set of data, collectively called the engine's "workspace." The state of each workspace at each synchronization point is expected to be identical to that of the other engines, and reflects the state of progress the engine has made through its execution of the algorithm. The combined effect of scrubbing and triple-voting enables certain types of workspace corruption to be detected and corrected such that processing may continue without interruption or error.

This work was done by Matthew Holland of Goddard Space Flight Center. Further information is contained in a TSP (see page 1). GSC-15757-1