ARCHITECTURES FOR INDEPENDENT TEST DATA REVIEW ON NPOESS VIIRS

Carl H. Fischer [*] [1], *Michael Denning* [2], *Kristin E. Clark* [3], *Bruce Guenther*[4]

[1] Ball Aerospace and Technologies Corporation, 2875 Presidential Drive, Fairborn, OH 45324

[2] Integrity Applications Incorporated, 5160 Parkstone Drive, Suite 230, Chantilly, VA 20151

[3] Massachusetts Institute of Technology Lincoln Laboratory, 244 Wood Street, Lexington, MA 02420

[4] NPOESS Integrated Program Office, 8455 Colesville Road, Suite 1050, Silver Spring, MD 20910

1. INTRODUCTION

To ensure timely and accurate execution of thermal vacuum testing on the first VIIRS flight unit, the Integrated Program Office (IPO) requested near real-time, independent government assessment of test execution. Several organizations shared this responsibility, including: The Aerospace Corporation, NASA, MIT Lincoln Laboratory, and others. In the previous phases of the test program, these organizations performed test data analysis on a two week time-line. To deliver independent assessments of test progress in time for consent to proceed meetings would require a dramatic reduction in response time. To achieve the required response times, changes were needed in ways the program distributed and reviewed data.

Two major tasks were essential to the success of this effort. First, the time necessary to distribute daily test data to analysts around the country would need to be less than 24 hours. This was no small task, as VIIRS generates data at a considerable rate[1, 2]. Typical daily data sets were on the order of 100 GB. To facilitate this, the Data Products Division at the Integrated Program Office designed and executed a unique, adaptable, effective, and low-cost data distribution operation. Second, a suite of tools for rapid inspection of commonly reviewed statistics was developed and shared with the community to optimize the efficiency of government analysts. The VIIRS Data Analysis and Decision Support cluster (DADS) was developed at Lincoln Laboratory to fill this need. This paper will describe the system and programmatic architectures necessary to deliver these results, highlighting key design decisions that could be carried forward for use on other instruments and programs.

2. METHOD

The on-site government team data processing architecture consisted of components from two teams: The IPO and MIT Lincoln Laboratory. IPO took responsibility for capturing test data from the contractor and distributing it to the distributed team of scientists responsible for its analysis. Lincoln Laboratory developed a system to allow on-site science team members to browse several different levels of summarized instrument data in real-time through a simple web interface.

Four key attributes of the DADS system were critical to its success. First, we developed an easy to use web based interface that reduced training time for users. Second, the system pre-computes commonly used statistics so they can be displayed to the user on interactive time scales. Third, Structured Query Language (SQL) queries to extract data for the plotting engine are built from administratively defined configuration tables, thereby enabling rapid configuration of new data visualizations as they are requested. Finally, the reporting engine is based on re-usable, user-defined templates that can be quickly run against sensor data as it arrives. These attributes encouraged use of the system by scientists from several organizations, and substantially reduced the level of effort necessary to accomplish the most frequently occurring test data review tasks.

The DADS system operates by computing a standard set of statistics from every data file as they are delivered to the file server. The results of these data analysis are cataloged in a MySQL® database. The database delivers these statistics on demand to a web interface that allows users to browse data and build test data reports. Figure 1 illustrates the data flow in the DADS system from the archive of raw sensor data files to the end user's web interface. Each major component will be described in more detail in this section.

2.1. Data Capture and Distribution

The VIIRS contractor, Raytheon Space and Airborne Systems, El Segundo, CA, directed test data from the high-bay to a Raytheon owned and operated data server in near-real time. Raytheon provided the customer with access to test data on this server via a read-only workstation. For security reasons, an air gap existed between this read-only workstation and government equipment. In place of the air gap, the DPD formed an on-site "data clerk team" responsible for many aspects of data capture for the duration of the 100+ day thermal

[*] This work was performed while Dr. Fischer was at MIT Lincoln Laboratory

vacuum test.

The data clerk's primary responsibility was to transfer test data via removable external hard disk from the read-only workstation to an on-site IPO-provided test data server. The data clerk typically transferred several gigabytes of data at thirty-minute intervals. Once data were present on the on-site server, dubbed "SantaNOSA," it became available to analysts and a variety of software agents.

The rapid (2-day) sensor performance results were handled by a distributed team of scientists and engineers. Raw sensor data was distributed to this team via one of three methods. On-site sensor scientists had access to data over a secure virtual local area network (VLAN) and could commence their analysis work immediately. Off-site scientists received daily overnight shipments of USB hard disks containing the previous days' complete test data set. Data were also automatically transferred to IPO headquarters via dedicated DS3 (Digital Signal 3) line. Once at IPO headquarters, data were made available to analysts around the country via the "CasaNOSA" website, IPO's official test data repository serving sensors scientists and the future cal/val teams.

An network file system (NFS) share on SantaNOSA provided the DADS cluster with access to sensor test data as it appeared on the government archive. Details of the data processing on DADS will be described in the following sections.

2.2. Data Ingest

New data on the CasaNOSA file share are discovered and queued by a set of bash and perl scripts that run periodically on the lead DADS compute node. The processing queue is maintained as a table in the MySQL database from which worker nodes claim and process tasks. In addition to the raw sensor data files, the satellite vendor delivers ancillary information such as test equipment parameters in text format such as comma or tab separated values. Perl scripts parse and load these data into tables that can be efficiently joined to the appropriate sensor data statistics tables. This critical capability allowed the DADS system to deliver plots of sensor response versus test equipment parameters like monochromator wavelength or blackbody temperature upon request of the GOST representative.

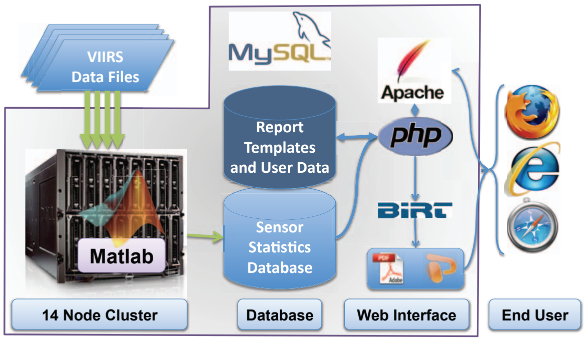


Fig. 1. High level data flow in the VIIRS Data Analysis and Decision Support compute cluster

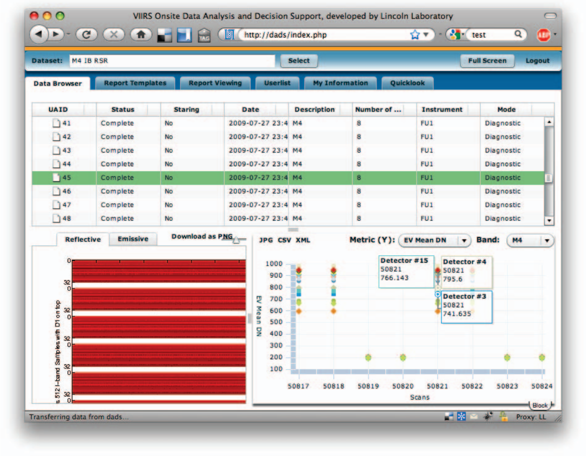
Once ingested and queued, an array of fourteen worker nodes (Dell Power Edge M600 blade processors) begin to process the files. Each node has 8 GB of ram and two quad-core Intel Xeon processors. The nodes are managed using Rocks cluster management software[3, 4]. Two Matlab threads run on each node for a total of 28 simultaneous processing tasks. All 28 nodes perform the same data reduction task on every file in the queue. A simple recipe ensures no two nodes process the same file.

2.3. Data Reduction

VIIRS test data sets often contain hundreds to thousands of data files, varying in size from 10-200 MB. Each data file contains data from a stable set of test conditions. Most test data analysis routines begin by reducing the volume of test data to a manageable size – often one or two metrics are necessary per detector from each data file. The method for calculating these metrics varies widely based on the specific parameter under test. For example, to assess the quality of relative spectral response testing, an analyst would typically plot the mean difference between shutter-open and shutter-closed data versus monochromator wavelength. On the other hand, to assess emissive band linearity, the analyst might review plots of sensor digital number response divided by blackbody radiance versus blackbody radiance. To facilitate the wide array of potential user requests, an extensible processing system was developed in Matlab to handle the known processing requirements as well as the unexpected needs that arose during test.

The DADS processing architecture performs four fundamental steps on each data file. First, the worker node claims a file from the processing queue table in the central database. This critical step ensures that compute nodes work on different data files. Next, the compute node extracts and loads the data for the file. Extracted raw data files are stored on a network share, allowing other analysts to skip the time consuming extraction process. Once the file is loaded, the compute node runs through a set of Matlab processing scripts on the file. The required parameters and results from the scripts are documented with examples, allowing other teams to develop custom statistics for future expansion. These scripts populate statistics tables on the central database server with simple metrics such as: mean, standard deviation, max, min, and counts of saturated samples Statistics are stored with the file from which they originated, the date and time as reported by the sensor, and the absolute scan number. Once the processing scripts finish, the node marks the file complete in the processing queue table and repeats the recipe with another file in the queue.

The end result of this step is a comprehensive table of relevant statistics for each file processed. These statistics are or-



VIIRS Onsite Data Analysis and Decision Support, developed by Lincoln Laboratory

http://dads/index.php

Dataset: M4 IB RSR

UAID	Status	Staring	Date	Description	Number of ...	Instrument	Mode
41	Complete	No	2009-07-27 23:4	M4	8	FU1	Diagnostic
42	Complete	No	2009-07-27 23:4	M4	8	FU1	Diagnostic
43	Complete	No	2009-07-27 23:4	M4	8	FU1	Diagnostic
44	Complete	No	2009-07-27 23:4	M4	8	FU1	Diagnostic
45	Complete	No	2009-07-27 23:4	M4	8	FU1	Diagnostic
46	Complete	No	2009-07-27 23:4	M4	8	FU1	Diagnostic
47	Complete	No	2009-07-27 23:4	M4	8	FU1	Diagnostic
48	Complete	No	2009-07-27 23:4	M4	8	FU1	Diagnostic

Metric (Y): EV Mean DN

Band: M4

Detector #15
50821
766.143

Detector #4
50821
795.6

Detector #3
50821
741.635

Transferring data from dads...

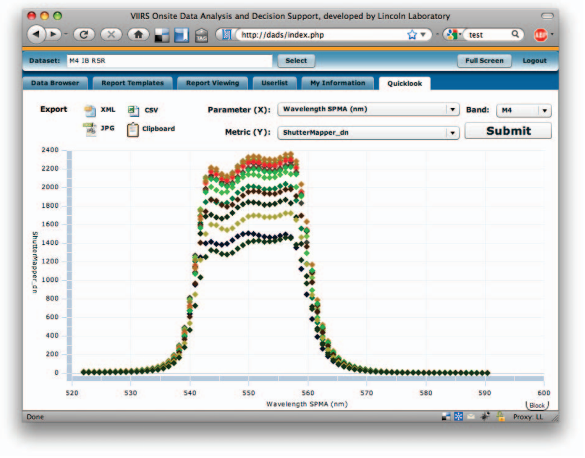
Fig. 2. Screenshot of the DADS user interface for browsing sensor data files. The table up top lists all data files associated with the selected test, while the two panels at the bottom show data from the selected file. Clicking a new file refreshed the bottom panels. Drop-down menus on the bottom panels allow the user to select plots and figures that are relevant to the specific test.

ganized by a relational database for efficient access, and represent a data reduction of several orders of magnitude. Multiple instances of this process can run simultaneously across a compute cluster without conflicts because the processing queue ensures only one instance works on each file.

2.4. User Interface

We developed a custom web interface to display the contents of our statistics database to the end users. Since the government onsite science team was comprised of users from various organizations, it was important that we select a user interface that was compatible with a variety of different operating systems and security postures. Adobe Flash was selected for its extensive library of visualization components, rapid development cycle, and the wide adoption of Flash Player on end users' web browsers. The interface went through four major revisions in the short three month Agile Scrum development cycle[5]. Each revision added functionality and improved the user experience.

Using the interface, scientists and engineers can browse a table of data files and retrieve plots and images with a single click and near instantaneous updates (Figure 2), generate plots of test parameters versus sensor response from hundreds of data files in just a few seconds (Figure 3), design report templates to tabulate and plot relevant information from a set of data files, run request interim reports from existing report templates, annotate those reports based on their interpretation of the results, and finalize annotated reports as PDF or Power Point format to be included in the program archive.



VIIRS Onsite Data Analysis and Decision Support, developed by Lincoln Laboratory

http://dads/index.php

Dataset: M4 IB RSR

Parameter (X): Wavelength SPMA (nm)

Band: M4

Metric (Y): ShutterMapper_dn

Done

Fig. 3. Screenshot of the DADS user interface for plotting data sets. Each x coordinate represents a single data file of tens of megabytes. All detectors are shown for the selected band.

The server-side processing necessary to deliver data to the flash components was implemented in PHP using an object relational model called plate. Database queries are constructed on-the-fly based on configuration tables that can be edited with an administrative interface. The database can efficiently join information from sensor statistics, test logs, and test equipment settings.

3. RESULTS

The presence of an on-site data clerk team proved to be a low-cost, effective method for VIIRS test data distribution. As opposed to a strictly electronic distribution path, the human data clerk component was always on-hand to obtain and organize relevant test procedure as-run documents, and various other shift logs, test logs, and test reports. It was the data clerk's responsibility to locate data, find documents, manage the data flow process, and serve any requests from the sensor scientist teams. The data clerks served a program librarian role, allowing government sensor scientists to concentrate on data analysis rather than how or where to find the necessary data and documentation.

The off-site "sneaker-net" data distribution method by USB disks was often preferred by analysts over electronic retrieval. Sneaker-net proved to be a low-cost, easy, and reliable method to distribute data. The sneaker-net distribution path allowed scientists to focus on their work; while fully-electronic transfer was always a viable option, sneaker-net eliminated many of technical barriers, lowered total costs, and gave more time to the sensor scientist teams to focus on building a quality VIIRS.

Despite its rather limited collection of automated statistics, the DADS system often delivered the first insight into several potential issues that warranted review of raw data. Screen captures from the DADS system became commonplace in the daily shift reports from scientists serving as the Government On-site Science Team (GOST) representative, and played prominently in the early detection of test issues. We attribute this success to the dramatic reduction in time necessary to inspect data and the corresponding increase in amount of data that was independently reviewed. Simplifying the task of plotting and browsing commonly needed statistics enabled the scientists to review more data and investigate anomalies more closely. A user interface walk-through and examples of analyst reports will be presented.

At the conclusion of thermal vacuum testing, the DADS statistics database contained over 1.2 billion records. Despite its size, the interface remained responsive to user interaction – lags in UI response were dominated mostly by the Flash Player's ability to generate plots, not the database response times.

4. CONCLUSIONS

Though network bandwidth, disk storage, and processing power have dramatically increased over the years, so have the data rates of imaging radiometers such as VIIRS. The task of disseminating, processing, and analyzing test data is by no means trivial. We have described our method for distribution and automated processing of sensor data to support government team quick look assessments of data health. Two examples were described in detail, highlighting instances where human interaction is essential and those where automation can improve efficiency.

In the first example, staffing the project with round-the-clock data clerks proved to be the most efficient use of program resources. The clerks delivered data to scientists in an easy to handle package. They were also able to support scientists in searches for data and documentation in a way not possible to be automated.

In the second example, a high performance Linux cluster and database backed compute solution delivered commonly needed statistics to the on-site scientists through an easy-to-use interface. Planning and flexibility ensured that the most important plotting capabilities were implemented. By reducing the effort necessary to extract, summarize, and plot gigabytes of sensor test data, the compute cluster enabled scientists to review more data. This efficiency led to the early discovery of test anomalies by our independent review team.

This work was sponsored by the National Oceanic and Atmospheric Administration under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and not necessarily endorsed by the United States Government.

5. REFERENCES

[1] C. Welsch, H. Swenson, S. A. Cota, F. DeLuccia, J. M. Haas, C. Schueler, R. M. Durham, J. E. Clement, and P. E. Ardanuy, "VIIRS (Visible Infrared Imager Radiometer Suite): A Next-Generation Operational Environmental Sensor for NPOESS," pp. 7031 – 7037, 1994.

[2] C. Schueler, J.E. Clement, L. Darnton, F. DeLuccia, T. Scalione, and H. Swenson, "VIIRS sensor performance," in *Proc. IEEE IGARSS*, 2003, vol. 1, pp. 369–372 vol.1.

[3] F.D. Sacerdoti, S. Chandra, and K. Bhatia, "Grid systems deployment & management using rocks," in *IEEE International Conference on Cluster Computing*, 2004.

[4] P.M. Papadopoulos, M.J. Katz, and G. Bruno, "Npaci rocks: Tools and techniques for easily deploying manageable linux clusters," *Concurrency and Computation: Practice & Experience*, vol. 15, no. 7, pp. 707–725, 2003.

[5] K. Schwaber and M. Beedle, *Agile software development with Scrum*, Prentice Hall Upper Saddle River, NJ, 2001.