Planning and Scheduling to Support EOS Science Data Processing

Dan Marinelli
NASA/Goddard Space Flight Center
Greenbelt, MD 20771
dan.marinelli@gsfc.nasa.gov

Abstract

The architecture that has been designed to provide planning and scheduling support for EOS science data processing is described. An operational concept is presented showing how the software interacts to support the production of a very large number of data products which are archived and disseminated to a user community.

Overview

The EOSDIS Core System (ECS) is being developed to produce 260 standard data products which represent information-creation from raw instrument data. Daily, ECS will process 480 gigabytes of this raw instrument data and produce 1.6 terabytes of value-added information. This information is mostly geophysical parameters derived from radiances measured from instruments onboard orbiting spacecraft. Instrument teams provide the knowledge which is coded into science processing software which then executes within the ECS framework to produce information. Several attempts at producing what the instrument teams, or ITs, would call correct information will be needed as the characteristics of the operating instrument become understood and as scientific learning occurs. There is also the desire to share information among instruments because of their complementary nature. Because the ECS will be instantiated at multiple sites, there will be intersite data dependencies that must be managed.

The ECS (EOSDIS Core System) architecture to support execution of science processing software is primarily composed of two subsystems, Planning and Data Processing. These two subsystems are a part of the larger ECS which also provides long term data storage needs, ingest of data from external entities, an interoperability infrastructure, system-wide distributed search and access services, and a graphical user interface for querying and ordering the data. The Planning subsystem provides a human interface to planning the execution of jobs and to planning resource usage. The Data Processing subsystem is responsible for scheduling the jobs on the available resources and keeping operations staff updated on progress. This paper gives an overview of the architecture which is being built to support these functions.

The term "planning", in the domain of ECS, refers to laying out what activities should occur over some timeline so that operations can be efficiently planned and so that the performance of the system can be measured against expectations. Activities related to planning are implemented in the Planning subsystem. "Scheduling" refers to carrying out the plan given a set of resources and dynamic events such as data arrival. Scheduling activities are implemented in the Data Processing subsystem. The two subsystems represent a data driven system in that the plan is a prediction of events that will occur. Processes will only be scheduled once all required data is available. Processes will only execute once they are scheduled and all required physical resources are available.

ECS Architecture

ECS is composed of nine subsystems which interface to meet the demands of ingesting, processing, and distributing EOS data. Two of these will be described here, Planning and Data Processing, but the interfaces to the other subsystems are necessary to accomplish the goals of planning and scheduling. The other subsystems are: Interoperability, Communications, Management, Data Server, Client, Data Management, and Ingest.

The division of the Planning and Data Processing subsystems was influenced by the following factors:

- Follows Client/Server Architecture - Planning acts as the Client, and Data Processing acts as the Server. Planning is responsible for developing a production plan and informing Data Processing which activities must be executed as listed in the production plan.

- Increases Fault Tolerance of the Planning and Data Processing Subsystems - By separating the Planning services from the Data Processing services, an increase in the fault tolerance of the Planning and Data Processing subsystem software is achieved. The failure of the Planning subsystem will not cause an immediate breakdown in production processing. Data Processing will be able to continue with production processing until the processing queues are depleted. Also, a failure of the Data Processing software or hardware will not cause the Planning subsystem to fail. Although the flow of information between Planning and Data Processing will be interrupted, this should not affect the overall production plan.

• Evolvability of the roles and responsibilities of Planning as ECS becomes operational - The separation of Planning services from Data Processing services allows for different configurations of Planning services and Data Processing services. For example, there may be some special event which would result in the Planning services of a DAAC creating a production plan for itself as well as other DAACs. By separating Planning from Data Processing this becomes an easier problem to resolve.

Planning Subsystem Architecture

The Planning subsystem has three main responsibilities: 1) defining the goals for data production, 2) preparing plans for production, and 3) coordinating the production from those plans.

To support defining production goals, the system uses PGE (Product Generation Executive) profiles. PGEs contain executables and scripts that perform desired transformations on the input science data to produce the output science data. The function of the system is to execute PGEs repetitively with new data as it arrives. The profile for a PGE gives its resource usage requirements to the planning subsystem. Production rules state under which conditions the PGE should execute, e.g., should run each time data comes in, or only a fourth of the time, should it use alternate inputs if the primary doesn't arrive, etc. The human interaction comes in the form of production requests to actually execute the PGEs. These production requests are submitted to a candidate plan which becomes the statement of the operator's goal of what the system should accomplish. The candidate plan is generated 'on top of', or using, a resource plan that describes the availability of the hardware resources. Hardware resources are modeled as CPUs, disks, networks, and platforms which are combinations of CPUs and disks. Resources can be allocated to non-processing activities such as maintenance, test, or training. The description of the resources which exist in the system are provided from a configuration managed database in the Management Subsystem.

The ability to generate candidate plans is important to operations for several reasons. Disconnects in data that may be required to execute some PGEs can be spotted and point to corrective action. Efficiencies can be gained by examining the expected completion times of different plans which gives operations a better feel for what works and what doesn't. Staffing projections can be supported in cases where PGEs require a man-in-the-loop activity for verifying results. Since the ECS executes at several sites called DAACs (Distributed Active Archive Centers), each requiring data from other DAACs, it is important to share the candidate plans by publishing them. Once published, they can be extracted by other Planning subsystems which execute at related DAACs or at the SMC (System Monitoring and Coordination) for a system-wide analysis of the interdependent plans. The planning horizon can go to 30 days.

Once the candidate plans are evaluated, one is chosen for activation. Activation is the process of merging the new plan into the system. Each production request for a PGE causes many DPRs (Data Processing Requests) to be planned and later executed. The previously active plan contains DPRs that match those in the new plan, so these are reconciled and then the new DPRs are added to the list of things to do. As time passes, DPRs execute, and various other events occur in the Data Processing subsystem. The plan does not imply that a DPR must execute at a specific time of day, rather it says what the goal would be if other DPRs executed and the system performed as expected up until that time. The Data Processing subsystem's goal is to execute the DPRs upon receipt of all inputs, and to use it's resources efficiently while doing so. The reality is that DPRs won't all execute to the specifics of their PGE resource profiles and there will be system outages. Should the plan deviate greatly from the expected, or should resources become unused to any large degree, operations staff can replan. A replan is simply the generation and activation of another candidate plan. Another reality is that there may be high priority processing requests that don't get planned, especially in the early mission phases, but also in periods of critical spacecraft activity such as special maneuvers. Requests may be submitted without planning directly into the system and executed. PDPS also supports the concept of 'on-demand' requests. Some resource bandwidth can be set aside in anticipation of these requests during the planning period.

The Planning subsystem is comprised of various utility applications and servers. At its core is the PDPS Database which serves as the central repository of all information used by both the Planning and Data Processing subsystems. A high-level software architecture is shown in Figure 1.

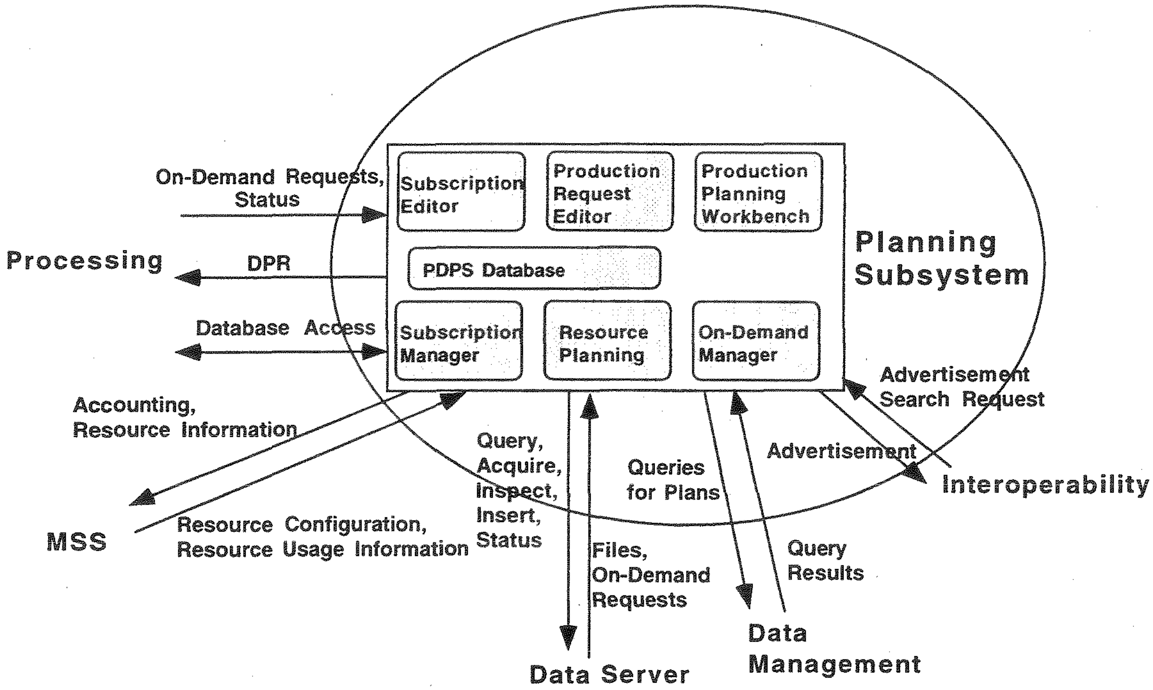

Figure 1 - Planning Subsystem High Level Software Architecture

The utilities, applications that execute in support of human interaction with the system, are:

- Production Request Editor - allows the operator to submit production requests for the data to be produced,
- Production Planning Workbench - used to prepare a plan for the production at a site and forecast the start and completion times of the activities in the plan,
- Planning Subscription Editor - provides the capability of reviewing subscriptions to data that are required for production, and,
- Resource Planning Workbench - allows operations to allocate and plan the use of the hardware resources.

The servers, which execute continuously so that work keeps flowing, are:

- Subscription Manager - waits for and then acts upon subscription notifications for data that arrives into the ECS,
- On-Demand manager - manages the receipt of on-demand requests, i.e., those that aren't planned for except in the general sense of reserving a set of resources for their execution, and,
- PDPS Database Server - provides secure, fault tolerant, concurrent access to system data.

The hardware configuration for PDPS at a large site is shown in Figure 2. The allocation of software to hardware is shown in Figure 3. The drivers for the Planning half of this hardware selection are:

- the PDPS database for storing persistent data
- backup and recovery for the PDPS database, and,
- growth.

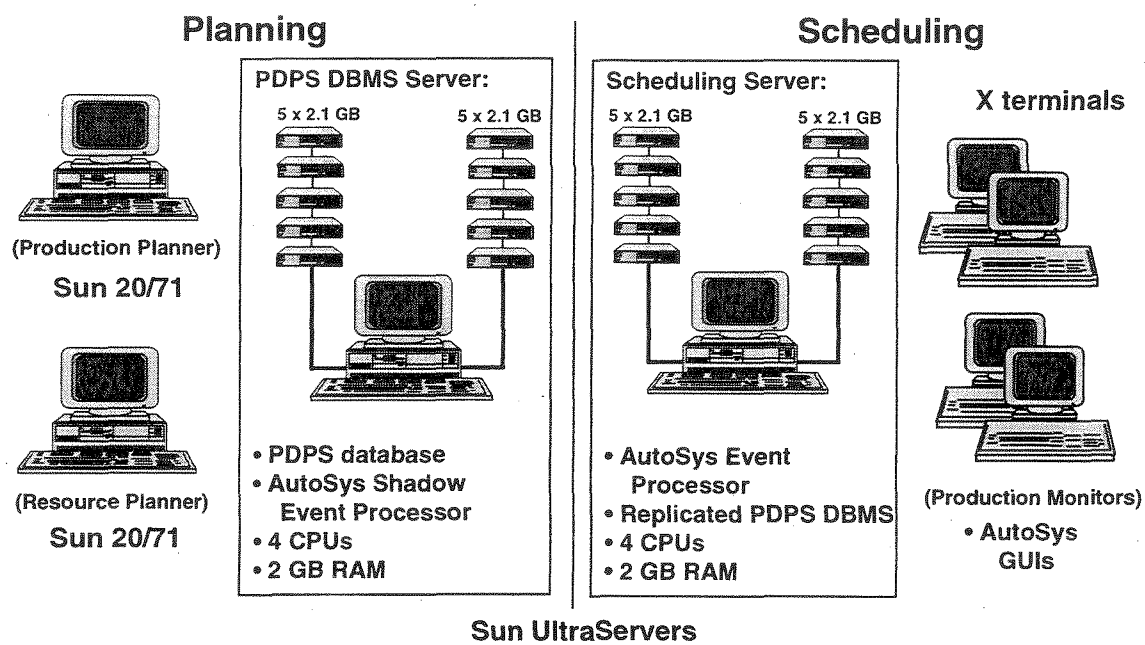

Figure 2 - PDPS Hardware Configuration (sans Science Processors)

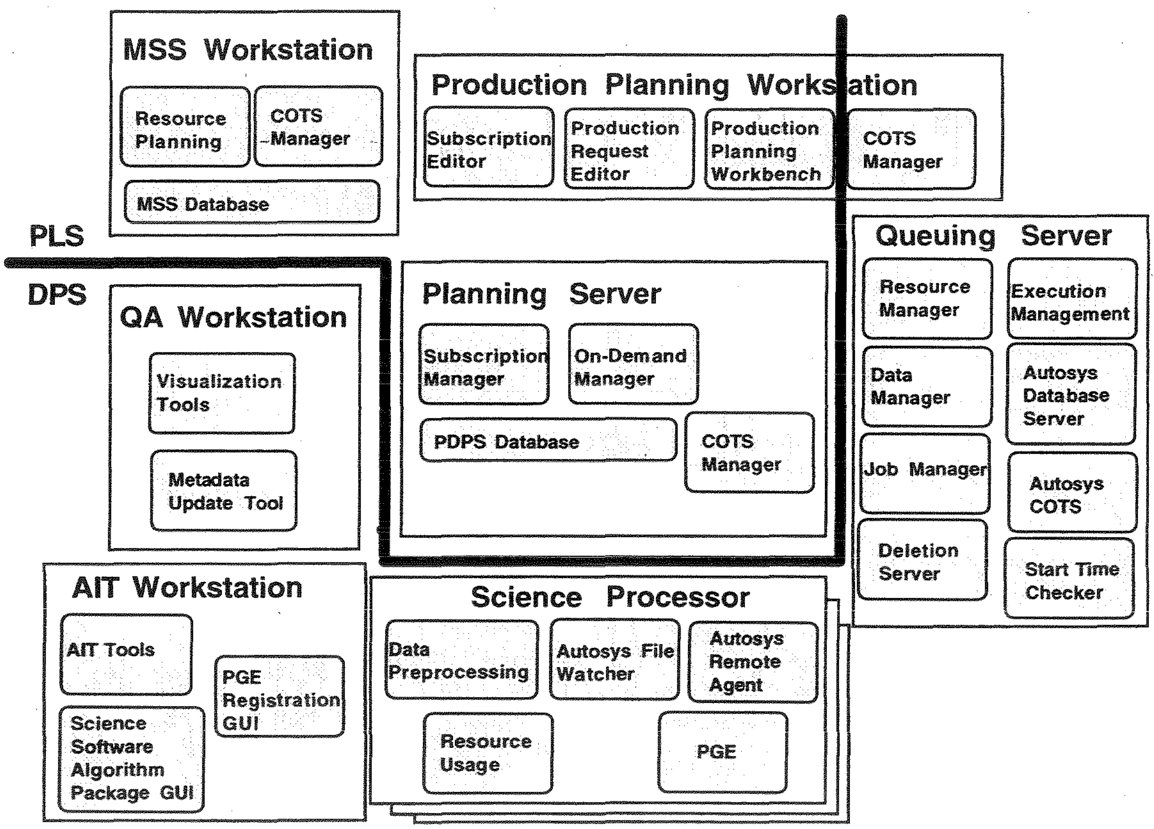

Figure 3 - PDPS Allocation of Software to Hardware

Operations staff must be able to generate a candidate plan in a 'reasonable' amount of time. Since candidate plan generation should occur at most once per day, with once per week the nominal scenario in a steady-state system, two hours is thought to be a reasonable time. For the largest site, there will be about 32,000 jobs per day. The database transactions are the limiting factor in the plan generation. For 30 day plan generation there will be an estimated 9 million DBMS transactions. The 4-CPU Sun UltraServer transactions per second rating is 1360, thus it would take about 110 minutes to generate the 30 day plan.

This configuration provides PDPS database replication to meet the backup/recovery requirement with adequate performance to meet the plan generation times expected. Growth is possible with the addition of CPUs and disk with no software changes. For smaller sites there is a slightly different configuration that meets lower transaction per second requirements.

Data Processing Subsystem Architecture

The Data Processing subsystem must perform two main functions: 1) support science algorithm development, and 2) support execution of science software while making efficient use of the science data processing resources.

Science algorithm development is supported with the Science Data Processing Toolkit, "the Toolkit", and with a set of tools for integration of the algorithm software, "the AIT (Algorithm Integration and Test) environment". The Toolkit was delivered early so that science software development teams could proceed on their development path independently of ECS. It has multiple language bindings and runs on many platforms. The goal is to eliminate redundant development activities among the development teams and to streamline the science software integration process. The library provides interfaces that give access to ECS products, product metadata, ancillary data and processing parameter information. The AIT environment is a set of tools that facilitate the integration of science software into the ECS DAAC environment.

The Data Processing subsystem provides a means of monitoring and managing the queued generation of scientific data products. DAAC Science Data Processing is not a real-time, automatically invoked, single data stream, uniform process such as spacecraft telemetry processing. Science data must be collected, stored, prepared, and "batched" for generation of data products. Processing occurs post (satellite) pass and must be planned on the basis of data quality and availability, and the availability of appropriate system resources.

The planning for the generation of data products is performed by the Planning Subsystem. When the candidate plan is activated, the set of DPRs that is to run in the earlier phases of the plan are passed to Data Processing via the PDPS database. The Autosys COTS product, which performs all the scheduling functionality, determines from the list of DPRs which ones can execute immediately. This is done by checking a data availability flag. The data availability flag is used to signify that the data exists within ECS, not necessarily on the Data Processing local disk resource. Those DPRs for which data are available are metered to the processing resources based on resource availability (CPU, memory, and disk space). The ones which don't have their data are considered by Autosys as 'on-hold'.

Planning has the ability to query the Autosys COTS to determine status of DPRs that have been submitted. As data arrives into ECS, Planning is made aware via its subscription manager and can signal to Data Processing through the PDPS database that pending DPRs can execute. Processing must perform several actions in support of executing the DPR. Each DPR is executed using 4 Autosys jobs which perform the functions of preparation, execution, and cleanup. Each of these 3 functions is wrapped into a larger job so that GUIs can be more easily used to identify activities which simplifies monitoring. Once Data Processing determines that a DPR can execute, i.e., all resources required are available and all data exists within ECS, the Autosys COTS is given the responsibility of executing the jobs. The first job that is 'kicked-off' is the entire job box which is the wrapper for the jobs that do the work which executes the DPR. This results in the kick-off of the preparation job. The responsibilities of the preparation job are to allocate disk resources, load data onto those resources that are needed by the DPR, and load the control information into the Process Control File that is used by the individual PGE which is to execute. Completion of this job results in the kick-off of the execution job. This job executes the PGE. Upon completion, the cleanup job is executed. This job destages data to the Data Server Subsystem for long-term archive, stores production history for the PGE in the Data Server, and deallocates resources. Should the data that has just been created be required by a downstream DPR, that data will be left on the local disk instead of deleted.

As the jobs are submitted to Data Processing and as they execute, the Autosys GUI can be used to see the progression of execution from 'on-hold' to 'executing' to 'complete'. Color coding is used to indicate problem areas and to give indicators of what is happening.

The high-level software architecture for Data Processing is shown in Figure 4. The AIT portion serves as the 'definition' stage of processing. I.e., the PGEs which are to execute are integrated into the system with this functionality. All information which the PDPS needs to support execution of the PGEs is provided at this stage. CPU, disk, and memory requirements are logged into the PDPS database as well as the structure of the Program Configuration File that is used to execute the science software.

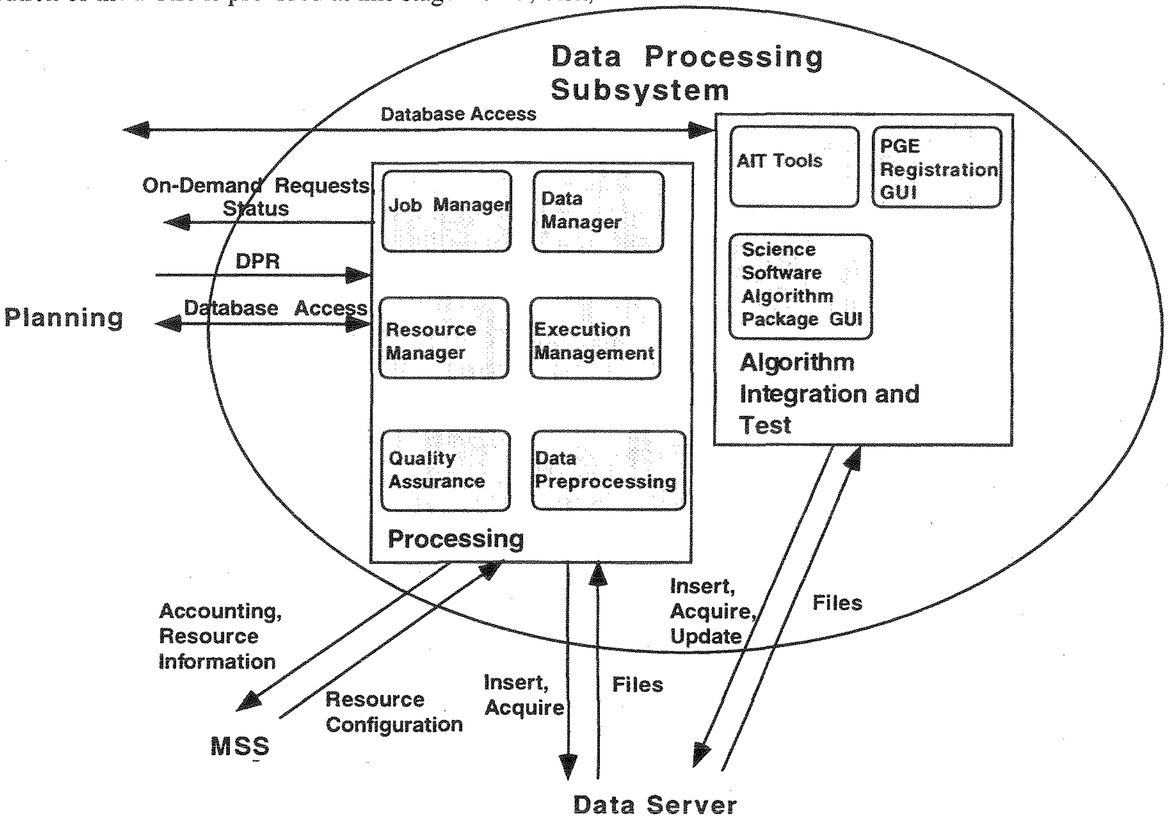

Figure 4 - Data Processing Subsystem High Level Software Architecture

The servers which execute in PDPS are:

- Deletion Server - controls when to delete files that are kept for short periods of time to support processing,
- Job Management Server - provides the interface between Data Processing and the Autosys COTS, encapsulating it,
- StartTimeChecker - continually checks to see if there are any jobs that haven't executed beyond what is considered a reasonable time, and,
- Compression Server - performs compression on data that is to be archived at remote sites.

The remainder of the components, not all of which are discussed in this paper are:

- Resource Management - manages hardware during production,
- Data Management - handles data staging from the Data Server, and data destaging to the Data Server,
- PGE Execution Management - performs pre- and post- PGE execution management functions including generation of the Process Control Files, and bundling of log files,
- Autosys COTS - used to run and monitor jobs on the science processing hardware,
- Quality Assurance Monitor - simple interface allowing DAAC operators to perform QA functions on output data, and,
- Data Preprocessing - a set of PGEs that supports conversion of orbit and attitude data into formats usable by the SDP Toolkit.

The drivers for the Data Processing hardware selection are job throughput and growth. At the largest DAAC, approximately 32,000 jobs per day must flow through the system. The current design will allow for about 190,000 jobs per day and with just the addition of hardware including extra RAM and spinning disk, the design will allow for 560,000 jobs per day.

Summary

The ECS Planning and Data Processing subsystems, with support from the rest of ECS and the science software delivered by the EOS Instrument Teams, are designed to provide the capabilities needed to manage science data processing in a distributed environment. Integration of science processing software is facilitated by the SDP Toolkit and the Algorithm Integration and Test tools. PGEs are integrated and their resource usage profiles are defined to the system. The ability to adjust which data to process is provided with the use of production rules. Resource usage is optimized by the Planning subsystem which enables operations staff to plan routine production, reprocessing, and on-demand activities and to schedule maintenance activities. The Data Processing subsystem is responsible for effectively managing all resources in real-time. It manages the data into and out of the subsystem hardware, and keeps track of whether it's needed in the near future so that network traffic to the Data Server is optimized.

More information on EOSDIS and the ECS can be found at: http://ecsinfo.hitc.com.

Acknowledgments

The work that this paper describes is being done by the people at Hughes Information Technology Systems in Landover, MD. Special thanks to Will Knauss, PDPS Manager, for providing timely information and proofreading of this work.

References

Bopf, M. PDPS Introduction, 706-CD-003-001 Day 1. Presentation for ECS Critical Design Review for Release B. Prepared under Contract NAS5-60000 by Hughes Information Technology Systems. April, 1996.

Fox, S.; Prasad, N.; and Szczur, M. NASA's Earth Observing System Data and Information System (EOSDIS): An Integrated System for Processing, Archiving, and Disseminating High-Volume Earth Science Imagery and Associated Products, 215-TP-001-001. July, 1996.

Release B SDPS Data Processing Subsystem Design Specification for the ECS Project, 305-CD-027-002. Prepared under Contract NAS5-60000 by Hughes Information Technology Systems. March, 1996.

Release B SDPS Planning Subsystem Design Specification for the ECS Project, 305-CD-026-002. Prepared under Contract NAS5-60000 by Hughes Information Technology Systems. March, 1996.

Roth, G. Release B Planning and Scheduling Hardware Selection, 706-CD-003-001 Day 6. Presentation for ECS Critical Design Review for Release B. Prepared under Contract NAS5-60000 by Hughes Information Technology Systems. April, 1996.