P1.31

DESIGN AND FUNCTIONALITY OF THE NOAA MODIS NEAR-REAL-TIME SCHEDULER

Paul Haggerty*: Science and Technology Corporation, Suitland, Maryland
Kristina Sprietzer: Science and Technology Corporation, Suitland, Maryland
Gene Legg: National Oceanic and Atmospheric Administration, Suitland, Maryland

1. INTRODUCTION

The NOAA MODIS NRT processing system was designed under the Computer Sciences Corporation (CSC) Central Satellite Data Processing (CSDP) contract in order to provide proof of concept that near-real-time processing of the large volume MODIS (Moderate Resolution Imaging Spectroradiometer) data stream could be accomplished, and to provide risk reduction experiments for upcoming NPP and NPOES processing systems.

The NOAA MODIS NRT Scheduler system (NANOOK) is a complex fusion of perl scripts, C and Fortran Programs, and a collection of simple database tables, which are responsible for the ingestion, processing, and distribution of MODIS data in a Near Real Time environment.

The associated poster will describe in a graphical format the inputs, outputs, and intermediary connections that bind the system together, while this paper describes in more detail three of the four subsystems: Ancillary Data Handler, Raw Satellite Data Ingestion, and Product Generation. A separate poster will handle the Data Distribution subsystem.

2. Ancillary Data Handler Subsystem

Input Data is provided by two NASA facilities. The raw satellite data is provided via a link to the EOS Data Operations System (EDOS) and the ancillary data used for preliminary calculations and calibration is provided via the Goddard Distributed Active Archive Center (DAAC).

The ancillary data are picked up by a dedicated subsystem that detects the existence of new files and moves them to the ancillary holding area. In the case of One Degree Global Data Assimilation System, Reynolds Sea Surface Temperature, and Total Ozone Mapping Spectrometer (TOMS) Column Ozone files the system also inserts a directive into the main processing system to run the conversion programs that transform the raw ancillary data into the Hierarchal Data Format (HDF) files that some of the MODIS science processing code requires.

3. Raw Satellite Data Ingest Subsystem

The EDOS group provides raw MODIS data at a rate of six gigabytes every 100 minutes for Aqua and three gigabytes every 50 minutes for Terra, resulting in nearly periodic rate of over 7 gigabyte of data to be processed every hour or 172 gigabytes of data per day. In addition to the MODIS data, the EDOS also transmits approximately 15 gigabytes of data per day of other instrument data, which is handed off to other processing groups.

Two perl scripts and a C program form another dedicated subsystem consisting of a sentinel script which watches for newly arrived data, the conversion program that repackages the raw rate buffered data (RBD) into the five minute Level 0 granules that are required by the NASA science processing code, and a distribution script which sends the data to its storage area and inserts commands into the main processing system to begin processing of the science data. Each day the system detects, converts, and stages 576 of these granules for processing through Level 1A, Level 1B, Geolocation, and seven Level 2 products.

4. Product Generation Subsystem

The Product Generation subsystem is by far the largest and most complex in the entire suite. It consists of two primary database tables and five primary control scripts.

4.1 CPU Database Table

In order to prevent over saturation of the available resources, NANOOK borrowed a concept from NASA's scheduling system. A processing directory was created to represent each physical CPU, and a corresponding line in the CPU database keeps track of the status of each processing directory. When a granule needs to be run, a query is made to this table to determine if any resources are currently free for use. If so, the entry is marked in the database to prevent that directory from being used again until the current processing finishes. NANOOK currently is operating with 64 concurrent processing directories.

4.2 STASIS Database Table

The STASIS database table is the partner to the CPU Database. While the CPU Database maintains a list of what resources are available, STASIS keeps track of what data needs to be processed. This database

* Corresponding author address: Paul Haggerty, STC, NOAA/NESDIS/IPD Federal Center FB-4, 4401
Suitland Road, Suitland, MD 20746;
e-mail: Paul.Haggerty@noaa.gov

holds a set of records consisting of the name of the Program Generation Executable (PGE) that need to be run and datestamp of the granule that needs to be processed with it. This database empties as the processing system removes entries for processing, and fills again with new entries as data is ingested into the system, and as intermediary PGEs spawn requests for higher level products.

4.3 STAGER Script

The STAGER script is the entry point into the MODIS product generation subsystem. The script is called once per minute to check to see if any data is awaiting processing. STAGER queries the STASIS database for a list of the PGEs that need to be run, and then queries the CPU database once for each entry, and if an available CPU directory slot is returned, calls the RUN_SCRIPTS script to activate processing of that product. All remaining entries are then returned to STASIS to await the next iteration.

4.4 RUN_SCRIPTS Script

The RUN_SCRIPTS script controls the processing of an individual PGE on an individual granule of data. First a Program Control File (PCF) is generated using the PCF_GENERATOR script. Then the COPYANC script is called to load all the required data (specified in the PCF file) into the processing directory. Then the actual PGE science code is called to create the indicated product. Lastly the CPU_CLEANUP script is called to clean up the directory and to reset the CPU database so that the directory can be used for the next product.

4.5 PCF_GENERATOR Script

The NASA science code (PGEs) was constructed to use a Product Configuration File (PCF). This file contains a variety of entries mapping Logical Unit Numbers (LUNs) to actual files located on the production server. This system allows the science code to be programmed using the pre-designated LUNs instead of filenames. Since each granule will use hundreds of different input files, output files, ancillary files, and configuration files, the use of LUNs frees the science code from having to be recompiled to change hardcoded names, or from having excessively long command line argument strings.

The PCF_GENERTOR script uses a predefined template file, an updatable PCF database which stores version dependant information, and some general logic to generate a valid PCF file based on the granule of data and the PGE to be run.

4.6 COPYANC Script

Each PGE requires a different set of lower level MODIS data and various Ancillary data files for proper functioning. COPYANC is the script that makes sure that all the appropriate files are available. The COPYANC script examines the pre-generated PCF file and searches the appropriate directories for filenames stored within it.

For MODIS data, the COPYANC script verifies that the actual requested file is present, and if it is, then copies the file to the processing directory. If the file is not available, this indicates an unexpected error has occurred. The error handling system is then activated and the situation is resolved.

For ancillary data, the situation is a bit different. The NASA processing system was not designed with Near Real Time processing in mind, and several days would elapse between receipt of data and processing. As a result many of the ancillary files that the science code expects are global analyses over the course of an entire day. This means that today's MODIS data needs to be processed using an analysis of today's ozone data, which will not be available until tomorrow at the earliest. This design causes a minimum of a day delay that is unacceptable in NRT processing. The solution adopted by NANOOK is to use the most recent data available. This solution is based on the premise that the global ozone distribution of today is not appreciably different than the distribution from yesterday's analysis, and any algorithm that uses such a first guess system will still converge on a similar final figure. This premise has been tested by comparing NOAA generated products to NASA generated products. No appreciable difference has been noted.

4.7 CPU_CLEANUP Script

The CPU_CLEANUP script is the final step in processing. This script is responsible for moving the newly created products and logs to their holding areas, activating the product distribution system, cleaning up stray files that might remain, inserting requests for higher level PGEs based on the newly created ones, and finally instructing the CPU database to mark the directory free for use.

5. Conclusions

The NOAA MODIS NRT Scheduler is a complex synthesis of scripts, database tables, holding directories, and network connections all working to keep the flow of MODIS data moving in near real time. This paper has attempted to outline only the highest-level components that make up the system. In addition to what has been outlined here, there are several dozen additional database tables and scripts that take care of Attitude and Ephemeris generation, system health monitoring, error detection and correction, and obsolete product purging. While vital to the proper processing of MODIS data, these additional components fall outside the scope of this poster.

A copy of this poster session is available for download from:

http://www.osdpd.noaa.gov/MODIS/AMS