Special Task Scheduling and Control of Cluster Parallel Computing for High-Performance Ground Processing System*

Wanjun Zhang[1,2], Dingsheng Liu[1], Guoqing Li[1], and Wenyi Zhang[1]

[1] Key laboratory, China Remote Sensing Satellite Ground Station, Chinese Academy of Sciences, No 45 Bei San Huan Xi Road, Beijing, 100086, China
{zhangwanj, dsliu, gqli, wyzhang}@ne.rsgs.ac.cn
[2] Graduate University of the Chinese Academy of Sciences

Abstract. This paper mainly discusses the special problems and solutions for multi-task and data flow control in cluster parallel computing system which dedicated used for High-performance Remote Sensing Satellite Ground Pre-processing System (GHIPS). After giving the overview of the GHIPS, the structure and function of Operation and Mission Subsystem (OMS) shall be formulated. The more detail discussion shall be focused on the organization and processing mechanism based on workflow of the task procedure as well as the task scheduling strategies. Based on our experiences more flexible and reasonable solutions will be given.

1 Introduction and System Background

Satellite ground processing system is used to process satellite remote sensing data and produce standard image products for application. Since the huge of downlink data from satellite and the complex computation of algorithms, cluster parallel computing appeared as a promising way [1] [2]. With the development of spacecraft technology more and more huge data are acquired from satellite and the requirements on the performance of ground processing system are increasing dramatically.

The High-performance Remote Sensing Satellite Ground Pre-processing System (GHIPS), which is the first general ground pre-processing system based on cluster parallel computing, was successfully developed by our team and can be regarded as a multi-purpose, multi-user and multi-product processing system. In particular, the system was integrated into parallel and high-performance computational environment to provide powerful processing ability. Now it has been successfully operated as an operational processing system for BEIJING-1, which has been successfully launched on 27 Oct. 2005. Most notably, the design and architecture of GHIPS are not only support BEIJING-1, but also can apply to other satellites data with minor modifications.

Each task in GHIPS will be processed by a serial of steps, including cloud evaluation, systemic radiometric and geometric correction, and so on. It was appeared to be crucial for the whole system how to organize such steps to form reasonable processing workflows

* The work was supported by the National Key Science and Technology Research Program, Ministry of Science and Technology, China.

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part III, LNCS 3993, pp. 17 – 23, 2006.

and how to control the running of such workflows with multi-task scheduling in the cluster parallel computing ground processing system. In GHIPS, such functions was implemented and integrated as the Operation and Mission Subsystem (OMS). This paper will describe the structure and characteristics of OMS, and then give some architecture analysis and implementation introduction. Further issues will focus on special problems related to task scheduling and data flow control in parallel environment, and then some preliminary solutions as well as its test results shall be given.

2 Functions and Structure of OMS

2.1 OMS Functions

OMS serves as the task scheduling and central control module. Nearly all of the other subsystems of GHIPS, including Data Archive Subsystem, Radiation Calibration, Geometric Calibration, Value-added Processing, etc, will interact with it.

OMS is divided into two parts: one is called OMS Master (OMS-M) running at cluster side to schedule and control the tasks, and the other is used to interact with users, which could be called OMS Client running at client workstation and provide simply operational interfaces for users to manipulate the clusters.

The detail mission of the OMS is including:

1. User Interface Management

OMS provides a friendly interactive user interface, which can be used for high speed downlink raw data separation, cloud evaluation, geometric correction with GCPs, image fusion, processing parameter setting, system configuration, and processing progress controlling etc. It can encapsulate all the parameter and control information to Object Description Language (ODL) [3] scripts, which can be parsed and executed by server-side daemon service of OMS-M. The status of each processing step and result can also be monitored in the interface.

2. Task Scheduling and Process Control for Clusters:

It is one of the core components of the OMS. When a task is submitted, OMS will schedule and allocate resources in clusters to process. The task processing flow can be controlled by the operators' commands coming from interfaces. Operators can cancel, hold or suspend the tasks processing of tasks when necessary. So the flexible organization of tasks with diverse algorithms, data flow and scale and reliable control of processes in the high-performance system is important issues to be considered.

3. User Authority Management:

There are three level user groups with different rights in OMS, including roots, engineers, and operators. OMS will ensure the security and right for each level user. OMS should manage all the user accounts for operation systems, database, NAS, and each subsystem of GHIPS. It has provided a single-sign-on mechanism to manage all the user profiles.

4. Others:

Many other aspects are involved in OMS. For example, it will monitor the status of all hardware and devices in the system, generate statistical reports of tasks for analysis, and manage log events for analysis etc.

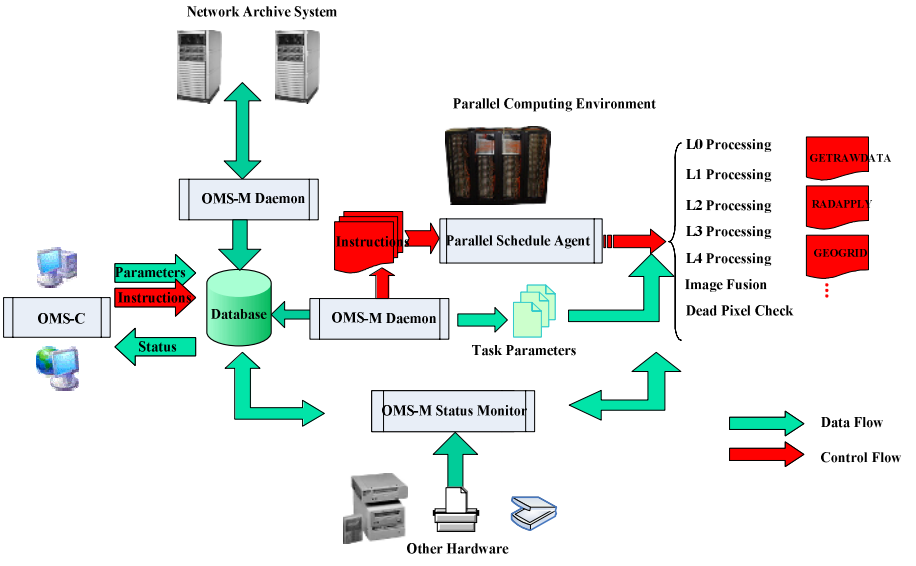

Fig. 1. Structure of Operation and Mission Subsystem

2.2 OMS Structure

The structure and organization of OMS can be showed in figure-1 based on the above discussion.

The cluster-based parallel computation environment provides near real-time processing of massive downlink data. The product generating function can provides standard data products as well as some value-added products. Meanwhile, the NAS data archive and the comprehensive user interface plus the above modules form this multi-platform system. And besides, varieties of I/O hardware link to the system for data transmission and storage, such as scanners and super-DLT tape library. All these factors make the management related to task schedule and data flow control much complex and crucial in high-performance environment.

3 Crucial Problems and Solutions

3.1 Management of Processing Flow

The flexible organization of multi-task with diverse algorithms, data flow and scale in the high-performance system is the first important issue to be considered. When a processing flow is performed, it should deal with the different satellite parameters, select appropriate radiation calibration algorithms and modes, and organize the processing sequence. Workflow [4] related concepts have been applied to the design and implementation for the processing of tasks in OMS.

As shown in Figure 1, a serial of process activities and prototype steps, such as GETRAWDATA, RADAPPLY, GEOGRID, and RESAMPLE… etc., which can be

reused in many typical processing flows, and each flow has its distinctive parameters, I/O interfaces, special algorithms and many other aspects. For example, the standard Level-3 and Level-4 products generating can be achieved with different serials. Object Define Language (ODL) describes the processing flow using object-orient design pattern with defined schema. Each ODL file contains the process definition and activity, transition information, participant declaration, and organization model.

The following script is typical ODL containing task parameter and control information example.

Server-side OMS parser can be seen as workflow engine. It will load and translate the scripts into binary codes for clusters. It can get the transition information dynamically and organize the processing activity sequence with relevant modules and parameters. Thus tasks with new algorithms and parameters will be easily organized by modifying the item value of the script, not changing programs. The Process flow can be arranged at design time expressed by ODL definition script. Because the system is in high-performance and parallel environment, there are some processing activities should run by parallel mode. OMS can dynamically determine the mode of steps with correspondent algorithms and implementation when perform parse the process definition. For example, when doing resample, OMS can schedule the operation in one node with serial mode, and it can also be performed in all nodes with Parallel Remote Sensing Image Processing File System (PIPFS) [5].

```
OBJECT              = CONTROL
       JOBID    = DMC4_00000858
       COMMAND  = "S"
       LEVEL_INPUT = 0
       LEVEL_OUTPUT = 4
       NEWJOB  = "Y"
END_OBJECT        = CONTROL
OBJECT                        =  RADAPPLY
    LEVEL                      = 1
       L1_SKIP              = "N"
       PARALLEL            = "Y"
       PRODFORMAT           = 1
       L1_BITSPERPIXEL     = 8
       STRRADPARM           = 11
       RADCAL_DIR           = ""
       DISPOSE_DEADPIXEL = "N"
END_OBJECT           = RADAPPLY
OBJECT                              = GEOGRID
      LEVEL               = 2
       STRGEOPARM         = 21
       GEOCALFILE         = ""
       FRAME_TYPE         = 1
       GRIDSIZE           = {32, 32}
       RESAMPLE           = "CC"
       PROJCODE           = "+proj=tmerc +k
                                 =1 +x_0=500000 +y_0=0 "
       SPHEROID           = "krass"
       PIXELSIZE          = {32.00, 32.00}
END_OBJECT           = GEOGRID
```

Operators could modify the processing status after the task is scheduled and executed. The commands include cancel, hold, suspend or restart a specified task processing. The dynamical transformation of the tasks status can be shown in figure 2.

In order to response the command and status converting, OMS-M sets checkpoints on each activity of the flow. When the activity is finished, it will check the execution logs, the current status and the user commands. Then the process management mechanisms of Linux operation system will be called by OMS to control the status change and the results could be passed to operators immediately. In addition, the interface of OMS enables operators to control the flow of both processing and data automatically or manually.

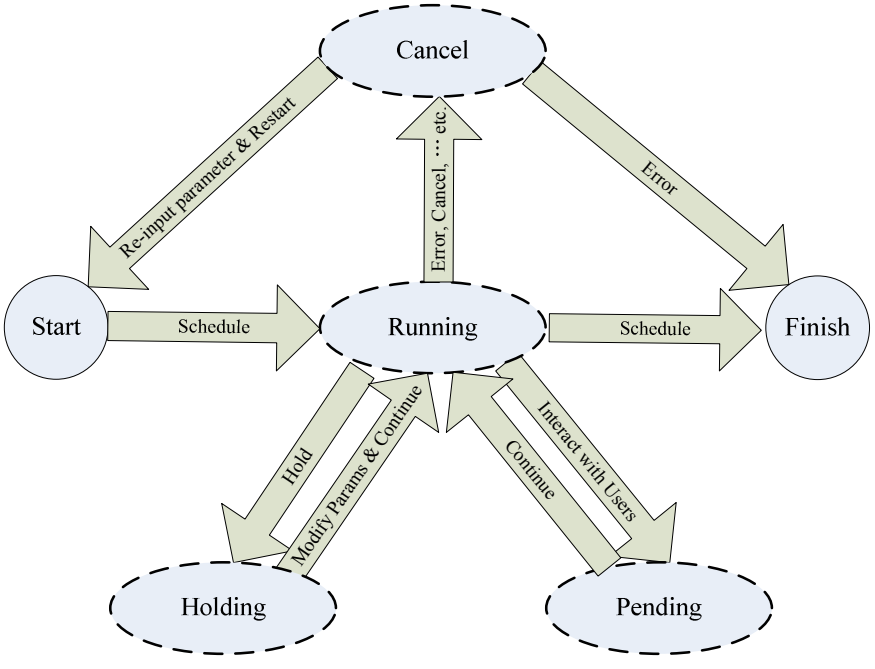

Fig. 2 Tasks Status Transformation in OMS

3.2 Schedule Strategies

The efficient control of the processes and data flow in clusters is also essential for the system, which can be seen from the figure 1. Although there are many perfect task management tools in Linux Cluster platform, OMS cannot use them directly. The main problem here is how to control the process between clusters and user interface clients. As mentioned in the last chapter, OMS has two parts: OMS-M and OMS-C. The OMS-C created preliminary ODL files which include task instructions and parameters based on user input, and the ODL file is swapped in a schedule pool shared by the OMS-M. The OMS-M can be viewed as an agent which transfers ODL instructions and parameters to cluster tasks and then those tasks can be performed by task

management tools. As a response, the agent will adjust the parallel resource to meet the needs of OMS. All the progress and status of performance will return to OMS-M, and it can pass the message to schedule pool, by which the OMS-C will acquire the run time information. The parallel intra-mechanism of such agents in clusters is similar with common tools and also can be modified from the latest update of such tools, which is separated from the OMS. With the mechanism, OMS will not be affected by the hardware update of clusters.

The system was architected based on Parallel Remote Sensing Image Processing File System (PIPFS), which can support operating massive data parallelized in clusters environment. Thus OMS has two categories parallel schedule. One is task-level and it mainly exerts for none computation-sensitive operations, such as get raw data, put product, etc. The other is algorithm-level. It will be used for the computation-sensitive algorithms and other massive data processing based on MPI and PIPFS, such as resample, rotation, and calibration. The parallel schedule strategy can be defined at design time, or changed at running time with modified the flag of processing parameter in ODL. These two strategies ensure the real-time capability of processing multi-task with massive data.

There are many strategies involved in task management. The topological architecture of the clusters is designed as multi NFS nodes to distribute the data efficiently, which is the key issue for massive remote sensing image processing. All the computing resources are allocated for tasks with different priority based on a FIFO queue, and the queue can support to schedule and process a large number of tasks simultaneously. The parallel schedule agent on clusters will assign the top priority task to the node with maximum idle resources. As a result, load-balance can be achieved dynamically.

Database is widely used in OMS, including OMS-C and OMS-M side. GHIPS is architected on heterogeneous platforms, such as the Microsoft Windows operations system of OMS-C, the clusters on Redhat Linux, and Network Archive System. At the same time there are many hardware and devices, including DVD-recorders, tapes, printer and scanner, to be controlled. Database is a good solution to communicate and exchange information between platforms, and monitor the status of devices. OMS has special services based on Oracle9i to serve as proxy and stub among the heterogeneous platforms for interchanging status. High speed database access engines which include Microsoft ADO [6] and Oracle OCCI [7] are used in the implementation of the services.

4 Results and Future Work

Through the successful integrating these solutions in GHIPS, a good performance was obtained. OMS can generate nearly fifteen processing flows and new processing will be easily added to the system. There are three radiation and geometric calibration modes managed by OMS, and each mode has different algorithms. End-users can select more than 20 projections to generate products. In the test-bed, OMS can stably schedule about 100 tasks with 30MB/s through-out capability simultaneously. At the same time, OMS can deal with the MS and infrared data of CBERS-1 by changing satellite parameters in OMS.

Based on the current work, the next step about OMS will focus on the flexible design and runtime control for processing using graphic tools. The standard of Ground Station Processing Markup Language, which is based on XML and consistent with WFMC schema [8] of Workflow Management Coalition, is putting forward. Many other relevant design toolkits are developing, including computational resource organization chart, processing flow designer studio, and automation parser. Our efforts will provide a state-of-the-art design for management of schedule and control in ground processing system based on parallel and cluster computation environment.

References

1. Moon-Gyu Kim, Sung-Og Park, Ji Hyeon, Sung-Og Park etc, Development of Satellite Image Ground Receiving and Processing System for High Resolution Satellites, http://www.gisdevelopment.net/aars/acrs/2002/vhr/103.pdf
2. Chao-Tung Yang, Chih-Li Chang, Using a Beowulf Cluster for a Remote Sensing Application, in 22nd Asian Conference on Remote Sensing, Nov.2001
3. NASA, Object Description Language Specification and Usage, http://pds.jpl.nasa.gov/documents/sr/Chapter12.pdf
4. Van der Aalst, W., M., P. , Barros, A., P., ter Hofstede, etc, Advanced Workflow Patterns, in Conference on Cooperative Information Systems, pp. 18–29, 2000.
5. Zhu Yaofei, Li Guoqing, The research and experimentation of parallel file system in remote sensing image parallel processing system, Master's thesis, China remote sensing satellite ground station.
6. Shi Jun, Ge Jun, Programming ADO, Tsinghua University Press, 2001
7. Oracle, Oracle C++ Call Interface Programmer's Guide, http://otn.oracle.com/
8. Workflow Management Coalition, http://www.wfmc.org/