Infrastructures and services for remote sensing data production management across multiple satellite data centers

Noname manuscript No.
(will be inserted by the editor)

Infrastructures and services for remote sensing data production management across multiple satellite data centers

Jie Zhang · Jining Yan · Yan Ma · Dong Xu · Pengfei Li · Wei Jie

Received: date / Accepted: date

Abstract With the number of satellite sensors and date centers being increased continuously, it is becoming a trend to manage and process massive remote sensing data from multiple distributed sources. However, the combination of multiple satellite data centers for massive remote sensing (RS) data collaborative processing still faces many challenges. In order to reduce the huge amounts of data migration and improve the efficiency of multi-datacenter collaborative process, this paper presents the infrastructures and services of the data management as well as workflow management for massive remote sensing data production. A dynamic data scheduling strategy was employed to reduce the duplication of data request and data processing. And by combining the remote sensing spatial metadata repositories and Gfarm grid file system, the unified management of the raw data, intermediate products and final products were achieved in the co-processing. In addition, multi-level task order repositories and workflow templates were used to construct the production workflow automatically. With the help of specific heuristic scheduling rules, the production tasks were executed quickly. Ultimately, the Multi-datacenter Collaborative Process System (MDCPS) were implemented for large-scale remote sensing data production based on the effective management of data and workflow. As a consequence, the performance of MDCPS in experiments envi-

Jie Zhang, Jining Yan, Yan Ma and Dong Xu
Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, P. R. China.

Jie Zhang, Jining Yan and Dong Xu
University of Chinese Academy of Sciences, Beijing, P.R. China

Pengfei Li
Institute of Information Sciences and Engineering, Yanshan University, Hebei, P. R. China

Wei Jie
School of Computing and Engineering, University of West London, U.K.

Corresponding author: Yan Ma, mayan@radi.ac.cn

ronment showed that those strategies could significantly enhance the efficiency of co-processing across multiple data centers.

Keywords multi-datacenter infrastructure · remote sensing data processing · distributed computing · big data computing · data management · workflow management

1 Introduction

In recent decades, with the rapid development of earth observing technology, many countries and regions have generally established various satellite platform and satellite data centers with the space observing capacity of the multi-spectral [1], multi-angle [2], multi-temporal [3] and multi-spatial resolution [4]. And these different types of satellite platforms have generated and will continue to produce vast amounts of remote sensing data. These data will be used to meet the requirements of specialized information extraction and analysis. For example, the United States Earth Observing System Data and Information System (EOSDIS) currently has 12 satellite data centers [5], and its archive data [6] is about 7,000 unique datasets, and the total data amount is over 7.5 PB. China RS Satellite Ground Station has four receiving stations which receive China Brazil Earth Resources Satellite (CBERS), HuanJing (HJ) satellites, Land Satellite (Landsat), SPOT satellites and other data types. The amount of their daily data is about 996 GB, and of their total annual data is about 354TB [7].

Due to a number of factors' limits, such as revisit period, coverage limits, spectral channels etc., it is difficult for single-source satellite data to meet the needs of large-scale integrated RS application. For example, the worldwide production of the 1KM land surface temperature (LST) product needs 4 types of RS data from Advanced Very High Resolution Radiometer (AVHRR), Moderate Resolution Imaging Spectroradiometer (MODIS), FengYun-3 (FY-3) and Advanced Along-Track Scanning Radiometer (AATSR) [8]. Therefore, the current multiple satellite data centers agencies should be combined together to provide the services of multi-source RS data. And there is no doubt that carrying out large-scale RS data processing and analysis to meet the demand of different end users has become a popular development trend [9, 10].

However, there are many problems to integrate multiple satellite remote sensing data centers and build a distributed data processing system. Firstly, a comprehensive remote sensing application requires massive RS data. Due to the multi-source data distributed in different data centers, large-scale data migration could be generated easily. It will not only cause a high load of satellite data centers, but also affect the efficiency of multi-datacenter collaborative process because of the low data transmission efficiency. Secondly, there are different data levels during the collaborative process, including the raw data, intermediate products and final products. The dependencies between data and products are more complex. On one hand, different products may need the same input data, if these data cannot be effectively managed, it would result in the duplication of some data processing, thus reduce the overall processing efficiency of the system. On the other hand, RS data processing is very complex, including pre-processing, post-processing and other complex processes. For example, there are many differences between the processing tasks with different types of data. Additionally, the integrated remote sensing applications often requires the system to automatically complete the bulk of data processing tasks, which can automatically manage the complex process flow and carrying out the relevant processing in a distributed scenarios. In summary, the multi-level RS data management and complex processes flow management are two key issues to build an efficient RS data collaborative processing system based on multi-datacenter infrastructure.

In this paper, we present the design and implementation of a massive remote sensing data production system based on the multiple satellite data centers infrastructure, the Multi-datacenter Cooperative Process System (MDCPS). In order to solve the duplication problem of data request and data processing, we adopt spatial metadata repositories and distributed grid file system to build a distributed, dynamic remote sensing data caching system. We build the remote sensing image processing repositories and multi-level task orders repositories for decomposition and manage complex processing flow, and composed some processing workflow templates and heuristic scheduling rules to automatically match and schedule the specific complex processing. Finally, we provide a use case of remote sensing production on several data centers, to show the feasibility of MDCPS in processing multi-source, massive, distributed remote sensing data.

The rest of this paper is organized as follows: Section 2 describes the related work of distributed remote sensing data processing. Section 3 presents the MDCPS environment and its infrastructures. Section 4 presents the design and implementation of data management and workflow management in MDCPS, Section 5 takes a specific remote sensing production process as an example to evaluate the performance of data management and task scheduling in MDCPS. Section 6 gives some further discussions to evaluate the MDCPS. Section 7 describes the conclusions and future work prospects.

2 Related Work

Spatial big data processing usually requires significant computational capabilities. Several studies have attempted to apply parallel computing, distributed computing and cloud computing to speed up the calculation process [32,36, 34,35]. In the area of RS data distributed processing, lots of grid-based distributed systems were built [11], such as Grid Processing on Demand (G-POD) [12], DataGrid [13], InterGrid [14], MedioGrid [15], Earth Observing System Data and Information System (EOSDIS) [16]. In these distributed data processing systems, data management and workflow management are two important components. For the distributed remote sensing data management, data transmission [17] is often carried out by means of grid middleware, such as

GridFTP [18]. In the aspect of data access and integration, multi-source data are often exchanged in accordance with certain standards or the corresponding conversion, for example, the data format of Committee on Earth Observation Satellites, and the data standard of Open Geospatial Consortium (OGC) and Geographic information/Geomatics ISO/TC 211 [19,20]. In the area of replicas management, Globus Tookits middleware are often used for data replication and distribution [21]. Gfarm Grid File System [22], as a distributed file system, designed to support the data-intensive calculation based on wide area network (WAN), combines each local file system to be a global virtual file system through the metadata server MDS and distributed I/O nodes, and improves the read and write bandwidth for distributed replicas. L. Wang, et al. [23] have designed and implemented a distributed multi-datacenter system G-Hadoop, which applied MapReduce framework [24,25] to the distributed clusters. In the area of workflow management, lots of scientific workflow systems have made an important progress, with a certain ability in tasks monitoring and control, scheduling policy management, and workflow fault tolerance [26]. But those work management systems are only designed and developed for the specific computing scenarios in their researches, and lack the common skills of abstract describing. So it is very difficult to meet the needs of different users when they require the deep customizations of scheduling policy, fault tolerance, etc.

The remote sensing data processing system based on multi-datacenter infrastructure is a solution to process massive, multi-source and distributed remote sensing data, and the current research in this field is under developing. The system based on multi-datacenter infrastructure can improve the efficiencies of the acquisition, organization and processing of distributed data [27]. Like grid-based distributed processing systems, data management and task scheduling are also two major challenges among the massive spatial data processing under this infrastructure. Most relative studies are focused on the algorithm of task scheduling [28,33,40] so far. For examples, W. Song, et al. [29] proposed the task scheduling mechanism and framework for spatial information processing and geocomputation across multiple satellite data centers; W. Zhang, et al [8,30] proposed a workflow scheduling method based on nearby data calculation, and designed a kind of image processing infrastructure based on multi-satellite center. However, few studies focus on the aspects of data management across multiple data centers, particularly on the data management of distributed collaborative processing.

3 Infrastructures Overview

3.1 Target environment

As a research result of the 863 program, MDCPS is designed to produce the large-scale and global coverage RS data production based on multi-datacenter infrastructure. It combines China Centre for Resources Satellite Data and Application (CRESDA), National Ocean Satellite Application Center (NSOAS), National Satellite Meteorological Center (NSMC), Computer Network Information Center (CNIC), Twenty First Century Aerospace Technology Co. Ltd (21AT), and Institute of Remote Sensing and Digital Earth (RADI). More than 60 kinds of RS data types, which sourced from the Aqua satellite, Terra satellite, Landsat series,CBERS, ZiYuan (ZY) satellite series, HaiYang (HY) satellite series, FengYun (FY) satellite series, BeiJing-1 (BJ-1) satellite series, National Oceanic and Atmospheric Administration (NOAA) satellite series, Multi-functional Transport Satellites (MTSAT) series, etc., could be processed in this environment. The amount of RS data could be over 1PB. It aims to provide a safe, reliable and efficient environment to support the applications of massive remote sensing production. Currently, MDCPS has the production capacity of more than 40 kinds of RS products, and its products are summarized in Table 1.

We keep the following goals when developing MDCPS:

- Effective management of RS data: In the collaborative process, we try to effectively manage raw data, intermediate products and final products. We try to use the data dependencies and data caches, and try to avoid the large-scale data migration by reducing the duplication of data processing;
- Automated processing platform: Aiming at improving the efficiency of co-processing, we try to implement the following process automation: match the complex process workflow, task decomposition, workflow organization, and workflow scheduling.

3.2 MDCPS infrastructures overview

MDCPS adopts the centralized system framework for massive RS data production. It consists of a master datacenter and several different data centers in geographic distribution. Master data center (MDC) is mainly composed of data management system (DMS) and business processing system (BPS). DMS manages raw data, intermediate products, and final products in co-processing, and provides the service of data query, scheduling and data discovery. BPS is responsible for the overall mission receiving, workflow organization, and task scheduling. Each data center consists of two subsystems: one is its own data service system with the responsibility of providing raw data services, the other one is the task execution proxy system (TEPS) with the responsibility for pre-processing raw data. The MDC will decompose global task, schedule each sub-task to the data centers proxy system over the WAN, post process together after merging the intermediate processing results in MDC, and ultimately complete the processing tasks. The system architecture diagram is shown as Figure 1.

The software architecture of MDCPS is shown as Figure 2, including application interface layer, business logic layer, software architecture layer and resource layer. In resource layer, the distributed resources underlying MDCPS include RS data resources, algorithms resources and computing resources over

Table 1 The RS products in MCCPS, their spatial and temporal characteristics

Product	Product	Spatial	Temporal
ID	Name	Resolution	Resolution
ADR	Aerodynamic roughness	1km	5d
AOD	Aerosol optical depth	1km	1d
AOD	Aerosol optical depth	30m	10d
ARVI	Atmospherically resistant vegetation index	1km	5d
ARVI	Atmospherically resistant vegetation index	30m	10d
BRDF	Bidirectional reflectance distribution function	1km	
CLI	Cloud index	1km	5d
DLR	Downward longwave radiation	5km	3h
DSR	Downward shortwave radiation	5km	1d
ET	Terrestrial evapotranspiration	25km	5d
EVI	Enhanced vegetation index	1km	5d
EVI	Enhanced vegetation index	30m	10d
FPAR	Fraction of photosynthetically active radiation	1km	5d
FPAR	Fraction of photosynthetically active radiation	30m	10d
FVC	Fractional vegetation cover	30m	10d
FVC	Fractional vegetation cover	1km	5d
HAI	Hydroxy abnormal index	30m	365d
LAI	Leaf area index	1km	5d
LAI	Leaf area index	30m	10d
LHF	Latent heat flux	1km	1d
LSA	Land Surface Albedo	30m	16d
LSA	Land surface albedo	1km	5d
LSE	Land surface emissivity	1km	1d
LST	Land surface temperature	1km	1d
LST	Land surface temperature	5km	1d
LST	Land surface temperature	300m	4d
NDVI	Normalized vegetation index	1km	5d
NDVI	Normalized vegetation index	30m	10d
NDWI	Normalized difference water index	1km	1d
NPP	Net primary productive force	1km	5d
NPP	Net primary productive force	300m	10d
NRD	Net radiation data	300m	4d
PAR	Photosynthetically active radiation	5km	1d
PRE	Precipitation	10km	1d
SAI	Suicide abnormal index	30m	365d
SBI	Soil brightness index	1km	
SBI	Soil brightness index	30m	
SHF	Sensible heat flux	1km	1d
SID	Sea ice distribution	1km	10d
SMI	Soil moisture index	1km	1d
SWE	Snow water equivalent	25km	5d
TCWV	Total column water vapour	1km	1d

data centers. In the software services layer, we adopts MyProxy and Globus Simple Certificate Authority (CA) as its security certification middleware between the data centers. And we use Gfarm grid file system to manage the distributed data replicas, use GridFTP to supply distributed data transmission services, use Ganglia to monitor the TEPS on distributed satellite data centers and get the information of performance. At the same time, the Kepler scientific workflow system is chosen as our processing workflow engine and we adopt MySQL as the backend database to complete the persistent of all data.

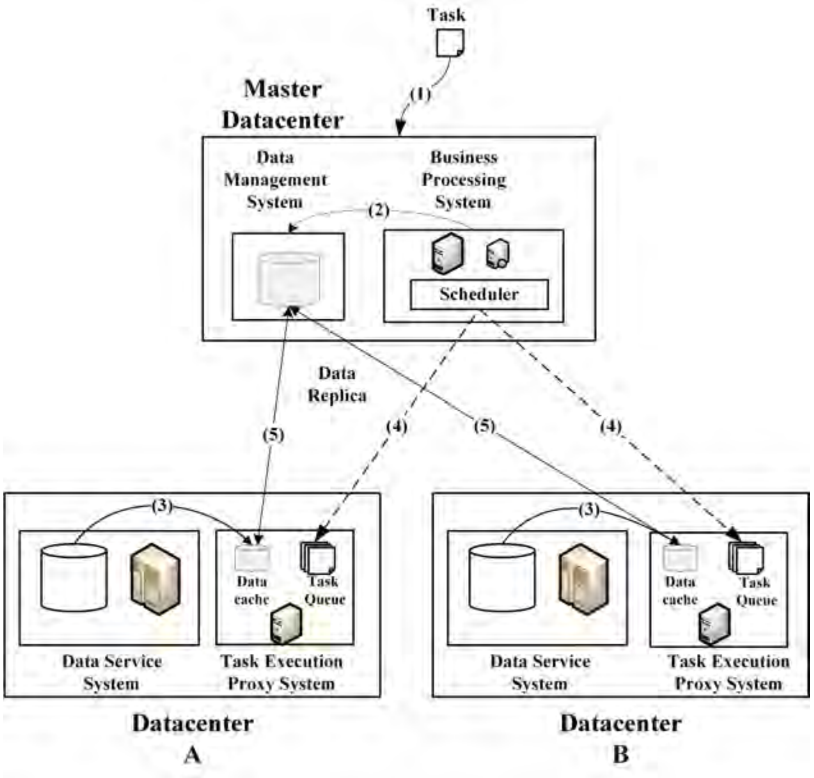

Fig. 1 The System Architecture Diagram of MDCPS

In business logic layer, MDCPS has the daemon module, data management module, workflow management module, task scheduling module, order management module, spatial metadata management module, computing resource management module, algorithm management module, log management module and other functional modules. These modules are used to manage all the distributed data replicas and organize tasks workflow automatically, scheduling and execution. In the application interface layer, the web portal of MDCPS supplies a friendly interface for users to submit their needs for processing massive RS data.

4 Design and Implementation

4.1 Data management

Global and large coverage of RS products often require massive input datasets, and the distributed computing, multi-datacenter collaborative process has a huge amounts of data transfer, including raw data, intermediate products and final products. And as a multi-user remote sensing data processing platform,

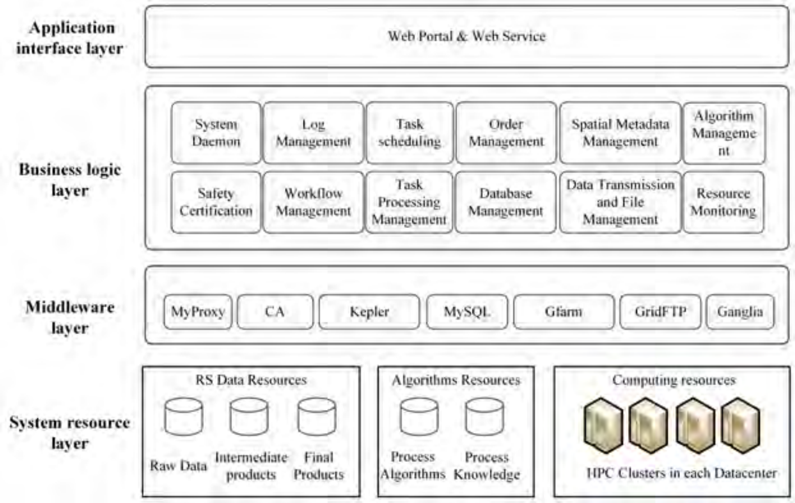

Fig. 2 The Software Architecture Diagram of MDCPS

there may be a lot of the same processing requests, the same spatial and temporal scales of input datasets, or the same remote sensing products. All these mentioned above can cause a lot of repeat transmission and processing. To reduce these unnecessary duplication transmission and production, MDCPS needs to achieve a unified management of raw data, intermediate products and final products in the process of co-processing. It would not only need to manage the metadata of distributed data, achieve reliable distributed file operations, but also need to manage the complex relationships between RS data. If these issues are addressed, MDCPS would be able to use effective data scheduling strategy based on the spatial relationships, the distribution of relations, affiliations, dependencies between RS data. It could maximize reuse the cached data and reduce large scale data migration in collaborative process.

In MDCPS, we used the strategies of "Spatial Metadata Management & Distributed File Cache Management" to realize the dynamic data management of multi-datacenter collaborative process. Firstly, for the management of metadata and data relationships aspect, we established three basic metadata repositories in MDC to manage the spatial metadata information of the raw data, intermediate products and final products. And then we established a public repository encoding geographic coordinates for the unity of all data spatial relationships. In addition, knowledge repositories of final products were established in MDC for input data parsing for RS products. We also added a series of relationships knowledge between products and implemented the management of products' relations. For distributed data file management, we realized a unified management in the data catalog access, data transmission and data cleaning. In the catalog management, we builded specific cache directories in MDC and others datacenters. The cache directories in MDC, as products cache catalog, it would cache intermediate products and final products. These directories in TEPS across each data center are used to store the raw data downloaded from the data center. All data files' information about distribution and request will be registered to spatial metadata repositories. Therefore, we achieved unified management of the cached data file's metadata. For the capacity monitoring and cleanup of data cache catalogs, we adopted Ganglia system client to monitor the capacity of cache directory in a near real-time. If exceeding limited quotas, we would choose specific data by querying spatial metadata repositories. In data transmission, we deployed GridFTP to implement a safe, efficient and stable data transmission service. MDCPS data management system structure is depicted in Figure 3.

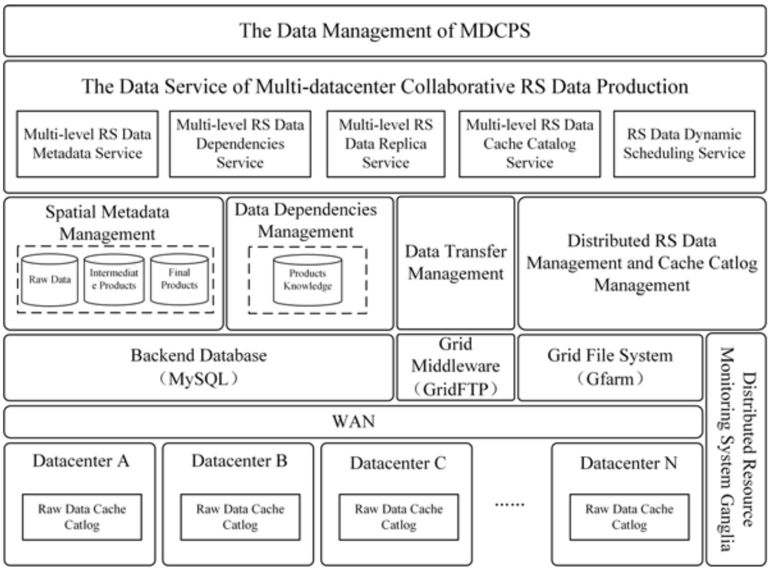
The Data Management of MDCPS
The Data Service of Multi-datacenter Collaborative RS Data Production
Multi-level RS Data Metadata Service
Multi-level RS Data Dependencies Service
Multi-level RS Data Replica Service
Multi-level RS Data Cache Catalog Service
RS Data Dynamic Scheduling Service
Spatial Metadata Management
Raw Data
Intermediat e Products
Final Products
Data Dependencies Management
Products Knowledge
Data Transfer Management
Distributed RS Data Management and Cache Catlog Management
Backend Database (MySQL)
Grid Middleware (GridFTP)
Grid File System (Gfarm)
Distributed Resource Monitoring System Ganglia
WAN
Datacenter A
Raw Data Cache Catlog
Datacenter B
Raw Data Cache Catlog
Datacenter C
Raw Data Cache Catlog
......
Datacenter N
Raw Data Cache Catlog

Fig. 3 The data management system of MDCPS

4.1.1 Spatial metadata management for co-processing

A normal production processes in RS data processing system includes: accept request, parsing the input raw data, raw data query, data download, preprocessing, post processing, products registration to the repositories, and final products feedback. Usually, there are two typical scenarios to meet the user's products need in a processing system.

– Carry out the whole process, and finally feedback processed products;
– Direct feedback these products, because all the needed products have been produced and archived.

To achieve the goal of greatest reuse of the resource, reduce the large-scale data migration and repetitive processing, and in addition to the above two production scenarios, MDCPS also has the dynamic production scenario based on the cached raw data and intermediate products. In the production process of multi-datacenter co-processing, it not only requires the management of the final products, but also requires the management of the intermediate data, such as the downloaded raw data and the intermediate products after preprocessing (standard products) and other auxiliary products. If realized, we could avoid the duplication of raw data download and intermediate products production.

In order to achieve the dynamic management, we builded a spatial metadata repository to manage the metadata of raw data, standard products and final products. The metadata information include data ID, file name, created time, data type, data size and cloud cover. In order to unify management of data spatial relations, we established a common grid of geographical coordinates. In addition, we also registered some information about data replica, such as request time, request frequency, replica distribution. These metadata information will be used for data scheduling and cleanup. For the management of data dependencies, we builded knowledge repositories to resolve the final products rely on the raw data, intermediate products and other auxiliary products. Finally, the dynamic management of data in multi-datacenter collaboration processing will be shown in Figure 4 as follows:

(1) Receive request of production;
(2) Inquire the final products repositories and confirm whether there are corresponding products. If yes, direct return, otherwise, continue;
(3) Analyze the dependency of the intermediate products and raw data;
(4) Inquiry products repositories, if the intermediate dependency products have been archived, calculate spatial relationships and decide the uncovered area, recursive products records and check the missing products;
(5) Keep on query until all the missing products data have been identified and make sure the corresponding data plan;
(6) Query the raw data cache repository to determine whether you need the data downloads;
(7) Query distribution information of data replicas, select an appropriate replica.
(8) Determine the final data plan dynamically.

4.1.2 Distributed file management

In the management of distributed RS data file, we integrated the Gfarm grid file system, GridFTP, Ganglia monitoring system and our spatial metadata repositories. It supplied the services of distributed data management, data transmission, file operations and catalogs monitoring. The efficiency and consistency of distributed file operations could be guaranteed in this environment. Figure 5 shows the system deployment.

Firstly, we builded a cache catalog in each data center for storing data replica files. A new created file in this catalog would be uploaded to Gfarm

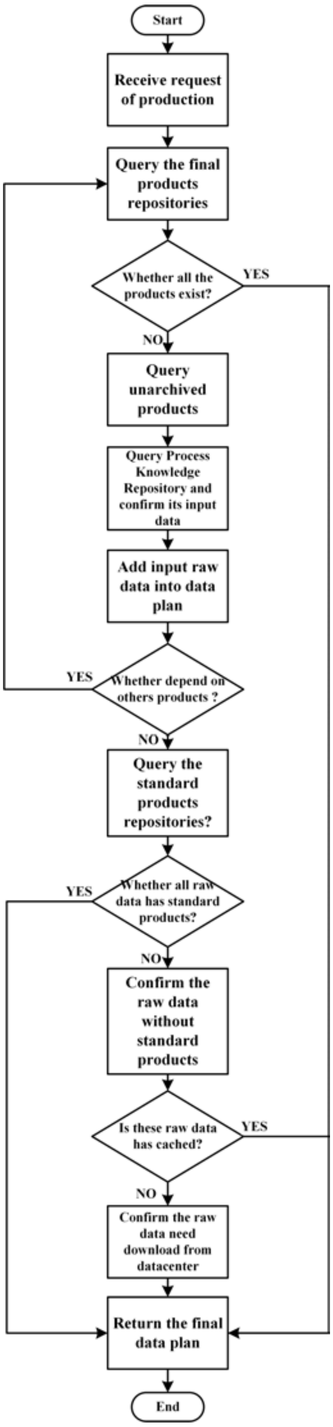
Start
Receive request of production
Query the final products repositories
Whether all the products exist?
YES
NO
Query unarchived products
Query Process Knowledge Repository and confirm its input data
Add input raw data into data plan
Whether depend on others products ?
YES
NO
Query the standard products repositories?
Whether all raw data has standard products?
YES
NO
Confirm the raw data without standard products
Is these raw data has cached?
YES
NO
Confirm the raw data need download from datacenter
Return the final data plan
End

Fig. 4 The RS data dynamic scheduling strategy in MDCPS

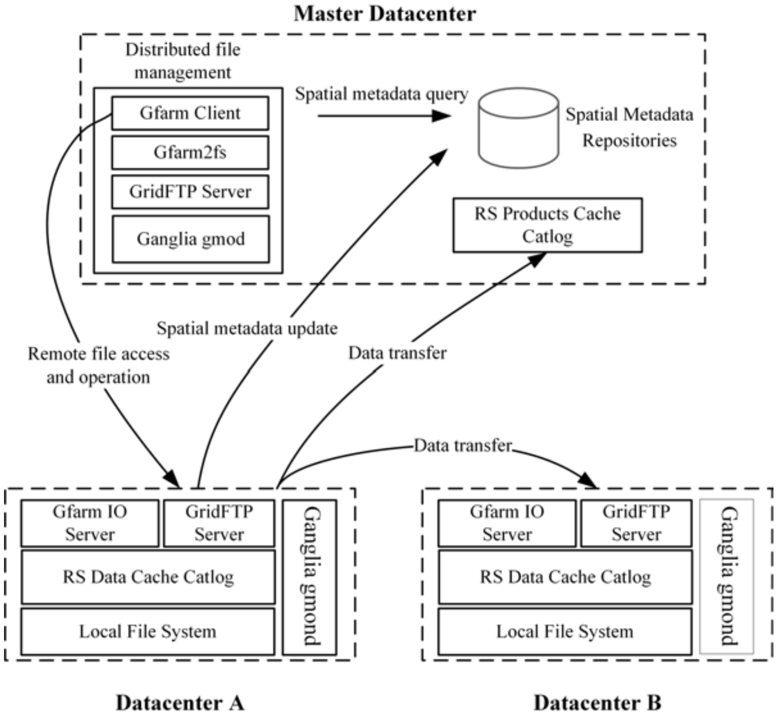

Fig. 5 MDCPS distributed file management

file system. In addition, we builded a public cache catalog in the master center of MDCPS and used Gfarm2fs to mount Gfarm system to this cache catalog. Then, all replicas of RS data in distributed Gfarm system could be displayed in this catalog. Just like as local files, we could use gfexport command to export data easily. Since the metadata in Gfarm backend database contains limited information and lack spatial metadata information, we saved the data distribution information in our spatial metadata repository which could instead the service of replicas matedata in Gfarm. To ensure the consistency of distributed remote sensing data files and its' metadata, we will update the corresponding information when it alters. The MDC is responsible for the global control of data transmission between data centers, which requires the third-party control of data transmission. In addition, large-scale data transmission requires a secure and reliable data transfer service. So we adopt the GridFTP as MDCPS's data transmission middleware.

Secondly, with the ongoing production, the cache catalog in each datacenter will keep caching lots of RS data and when it will be out of quota, the data transmission and production will terminate. Therefore, it is necessary for MDCPS to adjust and manage cache catalog capacity automatically during the production. In MDCPS, the capacity monitor was used, whose specific programs of data replacement and clean-up are as follows: Ganglia monitoring system was adopted to monitor disk usage. When a datacenter's usage exceeds the quota threshold, the monitor daemon in MDC will throw a warning. Then, MDCPS will use Gfarm client to statistic the data amount in that cache catalog, and the MDC will decide whether to clear the cache catalog according the feedback. If need to clean up, the system will query the metadata information of data replicas, filter some data based on the Least Recently Used (LRU) algorithm and delete these data replicas in corresponding cache catalog by Gfarm client. The sequence of data monitoring and cleanup operations are shown in Figure 6.

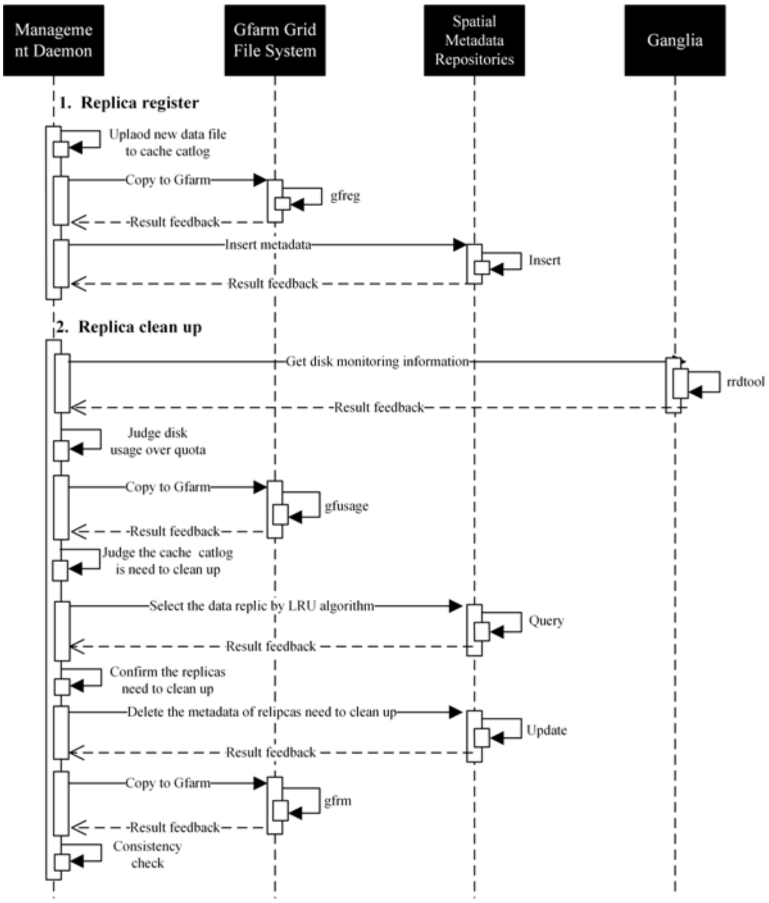

Fig. 6 Distributed file management sequence in MDCPS

4.2 Workflow management

Workflow management is a key module of the distributed processing system, which determines the reliability and stability of the system. The processing tasks of large-scale RS data based on the multi-datacenter infrastructure is complex, and it is difficult to organize, manage and schedule. The difficulties can be classified into two aspects:

- Complex distributed remote sensing data processing: Remote sensing data processing involves pre-processing and post-processing generally, and the processing methods are very different between different kinds of data, what's more, due to the different distribution of data and computing, the input and output data parameters of different processing flow are quite complex, and it is difficult to automatically match and organize a suitable workflow in a distributed environment;
- Complex scheduling of multi-datacenter scenarios: Multi-datacenter collaborative processing system combines multiple data centers to build a system of systems. The single scheduling policy is difficult to meet this hierarchical scheduling scenarios. In addition, the task scheduling in MDC involves a variety of dynamic resources across distributed datacenters, including data resources, computing resources and algorithm resources. An optimal scheduling model needs a comprehensive consideration for overall factors and it is a difficulty so far.

So as to solve the problems above, MDCPS builds an entire workflow system which connects the MDC and other datacenters. In MDC, we constructed a process flow repository and multi-level task orders repositories for matching and decomposition of complex processing tasks. We integrated Kepler workflow system as the workflow execution engine, and built Kepler workflow template repository for RS data production. The workflow template could help achieve the automatic construction of concrete workflow. Considering the features of hierarchical architecture, two-level scheduling strategy was adopted. MDC will schedule tasks to each datacenter by heuristic scheduling repository. TEPS over each datacenter batch the sub-processing tasks by Torque PBS. The monitoring service of task order could effectively monitor workflow status and provide support for workflow fault-tolerant. The workflow management architecture of MDCPS is depicted in Figure 7.

4.2.1 Workflow construction

In order to solve the complex issues of distributed remote sensing data processing, we constructed a processing repository, multi-level orders repository and workflow template repository to decompose, decouple and map the complex processing task. To begin with, we distinguished the different process flow by unified naming, and established a unique processing depending on its corresponding RS data types. Then, we divided it into several sub-processes

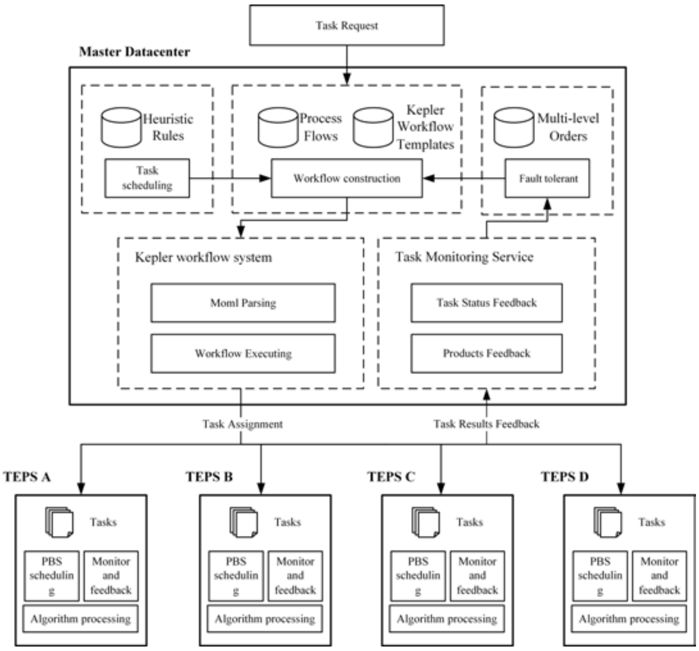

Fig. 7 The workflow management architecture of MDCPS

following the pre-processing, post-processing steps of each process flow. Finally, we store the hierarchical relations between different levels of processing flow in the process repository. For example, CP represents a top level processing of common RS products. CP1 and CP2 are two subclasses of CP in the second level. CP1 represents the processing flow of Landsat data and the CP2 represents the processing flow of MODIS data. CP1 processing contains five sub-processes in the third level: data preparation (DP), geometric normalization (GN), radiation normalization (RN), standard product uploading (SPU), common product production (CPP) and common product register (CPR). In addition to the former five sub-processes, CP2 also has a data swath (DS) sub-processing. The hierarchical relationship between different CP processing is shown as Figure 8 (a), and it will be stored in the processing repository. When a task was submitted, MDCPS will automatically match the processing repository, divide the task into several different levels of orders according to the corresponding processing, and store them to the task orders repository. We added the L1, L2, L3 prefix in the front of different levels of task orders. L1 represents the top level order of product production, L2 represents the second level order of processing for certain RS data type, L3 represents the third level order of sub-processing belonging to L2 order. In addition, we divide L3 order based on its input data's distribution in which data center and increase the number suffix to distinguish. For example, L3GN1, L3GN2, L3GN3 represent the GN process is conducted in different data centers. And all these make up a multi-level order for CP, which is shown in Figure 8 (b). The existence of a multi-stage task execution sequence order constitutes an abstract workflow. Finally, we got an original abstract workflow of processing task without any resource parameters.

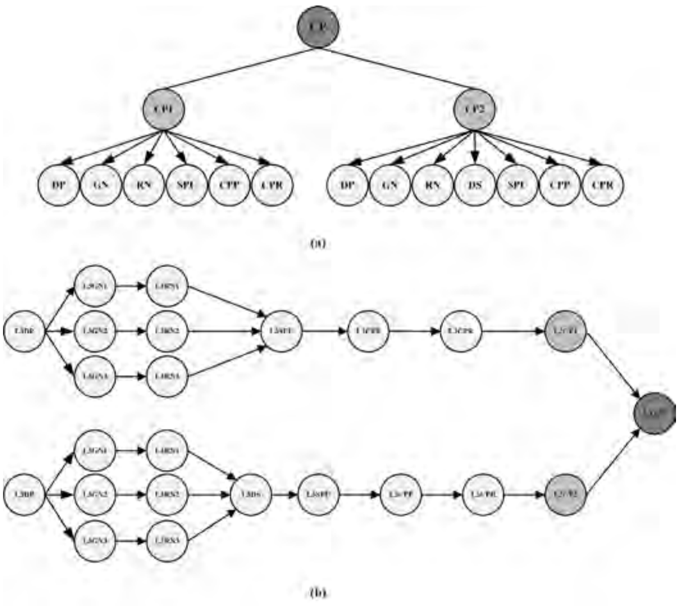

Fig. 8 The processing tree and multi-level task orders

Kepler workflow system comes from Ptolemy system [38]. It is an actor oriented open source scientific workflow system. It enables scientists to easily design and efficient execution of local or distributed workflows. We choose Kepler as the workflow system of MDCPS, because its web and grid services actors allow scientists to utilize computational resources on the net in a distributed scientific workflow [37]. We developed some user-defined actors for job submission and status monitoring, and also customized servals workflow template corresponding RS data processing based Kepler's Modeling markup language (Moml). Based on these template, abstract workflow will get specific data, algorithms, computing resources after workflow scheduling, and become a concrete workflow automatically. Workflow organizational process is shown in Figure 9.

4.2.2 Task scheduling

Multi-datacenter collaborative process is a process of data-intensive computing. MDCPS system task scheduling strategy should consider not only the performance of distributed computing resources, but also consider large-scale data migration. We investigated the correlation scheduling algorithm, finally

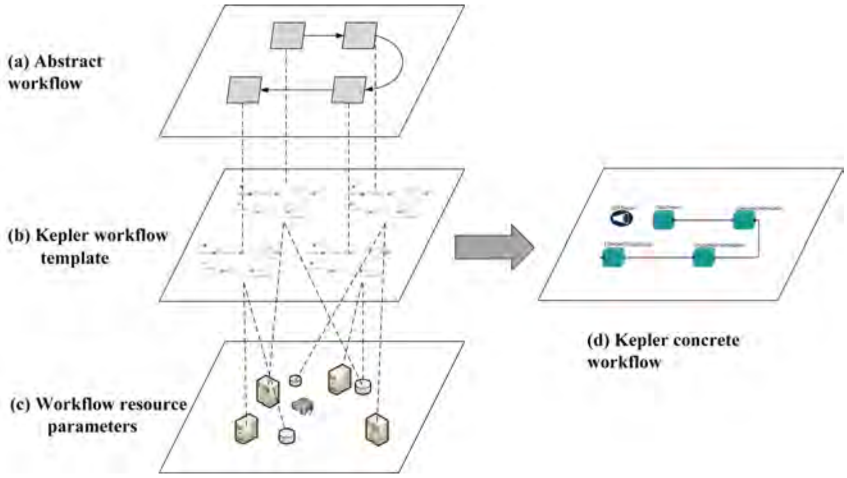
(a) Abstract workflow
(b) Kepler workflow template
(c) Workflow resource parameters
(d) Kepler concrete workflow

Fig. 9 Workflow organizational progress in MDCPS

applied Min-Min algorithm [39] to our system scheduling strategy. The difference between Min-Min scheduling and our scheduling is that we choose the computing resource (data center) instead of computing tasks in this step. Based on the Best-effort scheduling, MDCPS focus on to the execution time constraints to build the objective function. The target of scheduling is to achieve a minimum execution time for L3 task orders, and scheduling objective function is shown as Equation 1:

$$ECT(t,r) = max\{EAT(t,r), FAT(t,r)\} + EET(t,r) \tag{1}$$

In above equation, t represents a L3 task order, the resource r represents a distributed datacenter. $ECT(t,r)$ (Estimated Completion Time) represents the estimated time by which task t will complete execution at resource r. $EAT(t,r)$ (Estimated Availability Time) represents the time at which the resource r is available to execute task t. $FAT(t,r)$ (File Available Time) represents the earliest time that all the required RS data files of the task t will be available at the datacenter r. $EET(t,r)$ (Estimated Execution Time) represents the amount of time the datacenter r will take to execute the task t, from the time the task starts to execute. We computed ECTs of each task on all available datacenters and obtained the MCT (Minimum Estimated Completion Time) for each task. We assigned the task on the datacenter to complete it at earliest time. The basic steps of this scheduling are listed in Algorithm 1 as following.

In the data scheduling and computing resource scheduling stage, the time estimation methods are shown as follows:

Algorithm 1 Scheduling algorithms based Min-Min

```
while  ∃ task ∈ U is not scheduled do
    priorTask ← get an unscheduled ready task whose priority is highest.
    DOSHEDULE(priorTask)
end while
procedure doShedule($t$)                          ▷ Select the optimal data center
    while task is unscheduled do
        for all $r\Xi$ availDatacenters do
            compute $ECT(t, r)$
        end for
        R ← $min\{ECT(t, r)\}$              ▷ get a datacenter with minimum $ECT(t, r)$
        schedule $t$ on R
    end while
end procedure
```

In the stage of RS data scheduling, the RS data that have been cached in MDCPS spatial metadata repository will be the first choice to allocate. For non-cached data resources, the system sends download request to the corresponding data center according to the data type. We will estimated each datacenter's $FAT(t, r)$ according to amount of the data request.

In the stage of computing resource scheduling, MDCPS will dynamically monitor system performance information for each datacenter by deploying Ganglia system, The information such as CPU, I/O, memory, network, load, will update in near real-time to the computing resource repository. We are able to predict the capacity based on these monitor information of datacenter in the scheduling stage, and get the $EET(t, r)$. In addition, we can predict availability time by querying each data center's PBS task queue and task order running status in task order repository, and estimated the $EAT(t, r)$.

On this basis, we built a heuristic scheduling rules repository on the basic principle of "Near Data Calculation" [31]. It contains some heuristic rules for scheduling, including some empirical parameters for compute FAT, performance indicators of weight parameters for EET and EAT, performance metrics thresholds for resource scheduling, etc. MDCPS could dispatch the processing tasks automatically appropriate data centers based on the heuristic scheduling rules.

Taking the L3GN order processing for example to explain the scheduling method in MDCPS. Under the assumed conditions that MDCPS system has five data centers (DC1∼DC5), DC2 caches 60% of the required RS data and DC3 caches 20%. The remaining 20% of the required RS data will be required and downloaded from some datacenters. When L3GN order is submitted to MDCPS scheduling system, the detailed scheduling processes are as follows:

(1) Firstly, MDCPS determines which input data are already cached in the datacenter based on the data management system, and which data should be requested and downloaded at this time. In this example, the system determined 80% of the data were already cached on DC2 and DC3, 20% of the data should be requested and downloaded;

(2) Then, according to ”near data computing” heuristic rule, the processing tasks to the data centers who have already cached data are assigned preferentially. So the GN processing tasks will be scheduled on DC2 and DC3;
(3) Next, MDCPS selects a suitable data center to download, and assigns the GN processing tasks of the 20% non-cached data by our scheduling algorithm based on Min-Min, In this example, MDCPS firstly determined that only DC2, DC3, DC5 could provide the services of download for non-cached data based on the data service system. Then, MDCPS calculated the *ECT* of DC2, DC3 and DC5 to process the 20% non-cached data. The methods of time estimation are as previously described. Finally, a datacenter with the minimum *ECT* should be selected to execute tasks. Here, we assume DC5 was the final selection. The dynamic scheduling process is shown as Figure 10;
(4) Finally, L3GN order is split into three sub-orders L3GN1, L3GN2 and L3GN3, and the processing task of GN will be scheduled to DC2, DC3, and DC5.

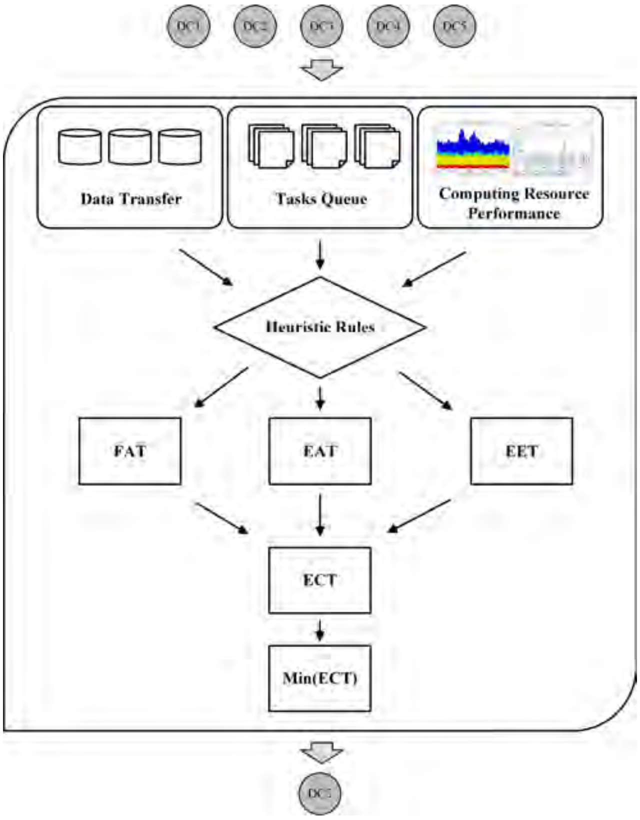

Fig. 10 The procedure of workflow dynamic scheduling in MDCPS

4.2.3 Workflow fault-tolerant

To ensure the reliability of the workflow, the fault-tolerance policies should consider the following aspects:

- Fault-tolerant based on retry: the construction phase in the workflow, if workflow can't be built correctly caused by the resource temporarily unavailable, the system will retry build after a certain time interval. The reasons of this kind of fault-tolerant mainly include: unsuccessful webservice call caused by network congestion, insufficient disk space while waiting for cleanup, excessive tasks in the data center, data center overload and data resource is in the ready state;
- Fault-tolerant based on checkpoint recovery: In order to ensure a fast reboot of the complex process flow, and to avoid duplication of data transmission and computing, each sub-workflow's condition will be monitored , and the parameters and conditions information will updated to the task orders libraries at the completion of the sub-task. As soon as the workflow errors occur in the implementation phase, the system will automatically check the recently completed state and rebuild the unfinished task, then resume operation;
- Timeout-based exit strategy: The system will set a threshold, the longest wait time of workflow and PBS job, to avoid long-term occupation of computing resources due to the abnormal operation of workflow, if it goes over the threshold ,the workflow or algorithm will stop automatically, and the resources occupied will be recovered.

5 Experiments

In order to verify the validity of distributed remote sensing data management and complex workflow management in MDCPS, we conducted the following experiments of performance comparison on several specific data production.

In this paper, we constructed our experiment MDCPS environment with four distributed datacenters: CRESDA, NSOAS, CNIC and RADI. RADI is the MDC, consisted of two compute nodes: one is for workflow management and the other one is for post-processing. The former which is a blade server is configured with 8 cores Intel(R) Xeon(R) E5-2603 (1.80GHz) and 32 GB memory. The latter is configured with 24 cores Intel(R) Xeon(R) E5645 CPU (2.40GHz) and 62 GB memory. All TEPS of CRESDA, NSOAS and CNIC are configured with 16 cores Intel(R) Xeon(R) E5-2640 CPU (2.00GHz) and 32 GB memory. The operating system is CentOS 6.5 and the Java version is 1.8.05.

The groups of experiments produced two kinds of RS product, 1KM Normalized Vegetation Index (NDVI) product and 1KM Net Primary Productive force (NPP) product, during the 180th-185th day of 2014, Zhangye, Gansu Province, China (36° E-43° E, 95° N-103° N), which is shown in Figure 11. The NDVI experiment requires 11.5GB of MODIS and FY3 RS data, and the

NPP needs 168.7 GB of MODIS, MST2 and FY3 RS data. The results of 1KM NDVI and 1KM NPP products produced by MDCPS are shown as Figure 12 and Figure 13.

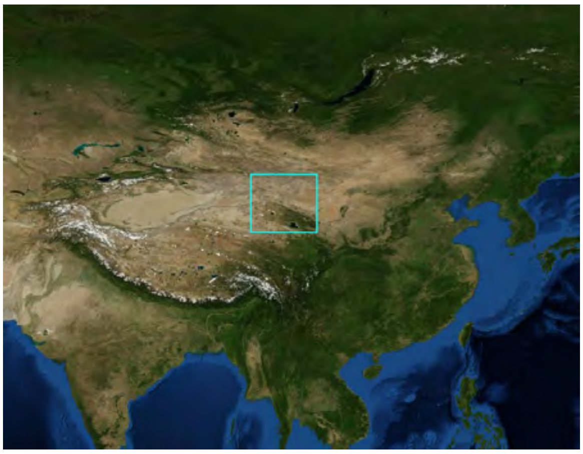

Fig. 11 Zhangye experiments area

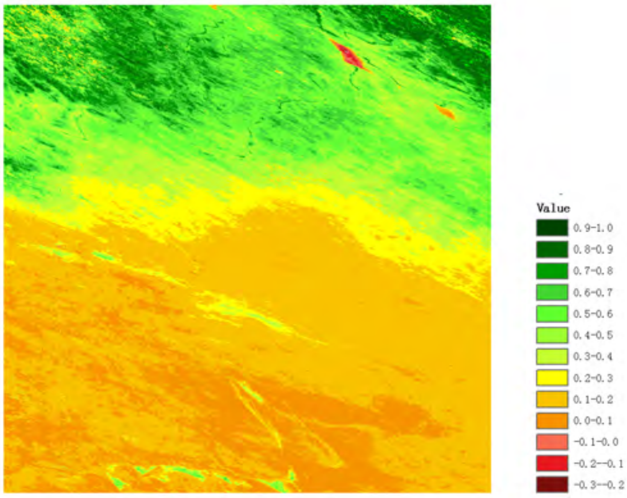

Fig. 12 1KM NDVI Product produced by MDCPS

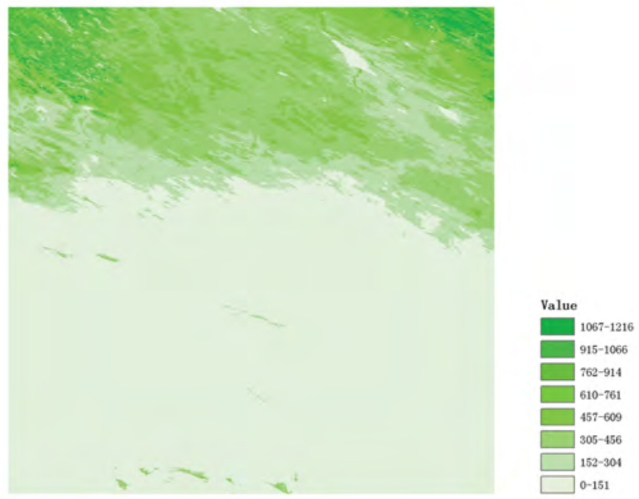

Fig. 13 1KM NPP product produced by MDCPS

5.1 Related experiments on dynamic data management

MDCPS realized the dynamic management of the raw data, intermediate products, and the final products, and reduced the large-scale data transmission and repetition. In order to verify the validity of reducing data transmission and data processing in MDCPS, we adopted the following comparative experiments to validate the effect of multi-level RS data cache management. In MDCPS, the production of NDVI and NPP contains seven processing stages: DP, GN, RN, DS, SPU, CPP and CPR, of which, DP and SPU are two stages of data transmission, GN, RN, CPP and DS are four stages of RS algorithmic processing. Generally, distributed massive RS data processing is a data-intensive computing, data transmission and RS algorithmic processing are very time-consuming.

Firstly, we carried out a normal production of NDVI and NPP in MDCPS. This normal production wouldn't reuse any cached data and product, and it is a typical scenario in other distributed data processing system over WAN. The time-consume statistics for each stage in normal production of NDVI and NPP is shown in Figure 14. It is easy to find out that the runtime of data transmission occupies larger proportion, the proportion of NDVI production process is about 29.2% and the NPP production process is up to 56.3%. The runtime of algorithm processing also accounted for a considerable proportion, the proportion of NDVI production process is about 69.5% and the NPP production process is 43.3%.

Secondly, we carried out other three typical scenarios based on the same production experiment in MDCPS system. These productions reused different

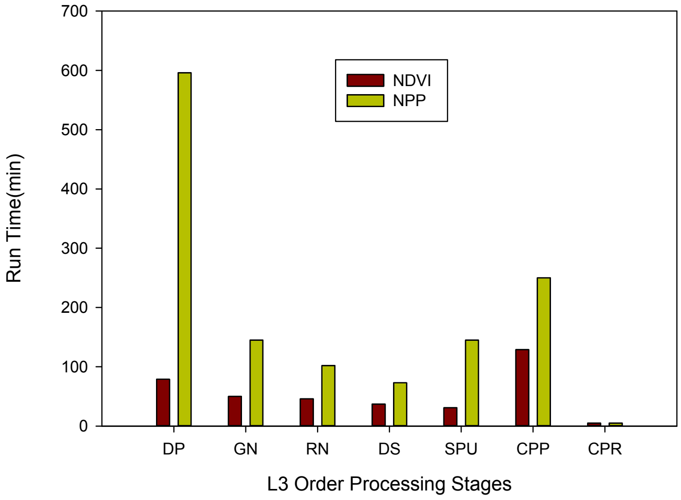

Fig. 14 Runtime of each stage of NDVI and NPP normal production

levels of data cache in MDCPS system, including: raw data cached (80% hit), intermediate products cached (80% hit) and final products cached (100% hit). Compared to the normal production, the processed statistical results are shown in Figure 15 and 16. By comparison, we can see that the dynamic management of the final products could maximize production efficiency with little time to feedback final products. By comparing the time-consume of cache raw data and cache intermediate products production scenarios, we could found that the raw data cache can only reduce the time-consume preparation in DP stage, because it can avoid the repeated request of the same data from satellite data center. But the data transfer (mainly at the stage of SPU) and algorithmic processing are still time-consuming. The intermediate products caching is better than the raw data caching to enhance the efficiency of production. Its effect is obvious both in stage of data transmission and algorithm processing. The total time-consume of intermediate products cached (80%hit) is generally a quarter of that in normal process (NDVI process was 32% and NPP process was about 18%).

According to the above two experiments, we can conclude that multi-level data caching strategies in MDCPS can reduce data transmission and repetition in varying degrees. It can significantly improve the efficiency of production in multi-datacenter environment.

To test the processing extensions performance of MDCPS for different amount of data, we tested time-consume by processing 11.5 GB, 52 GB, 168.7 GB, and 209.2 GB input data for NDVI production. According to the results are shown in Figure 17, as the amount of process data increasing, it shows that time-consume will non-linear increase. It could draw the conclusion that MDCPS has a certain degree capacity for massive data processing.

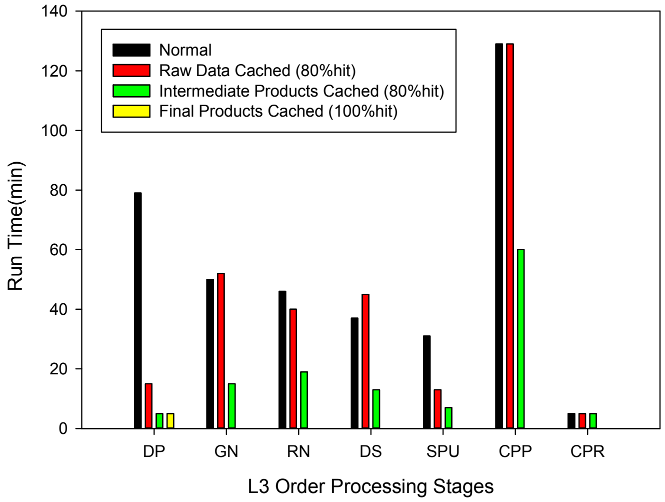

Fig. 15 Runtime of each stage of NDVI with four different scenarios

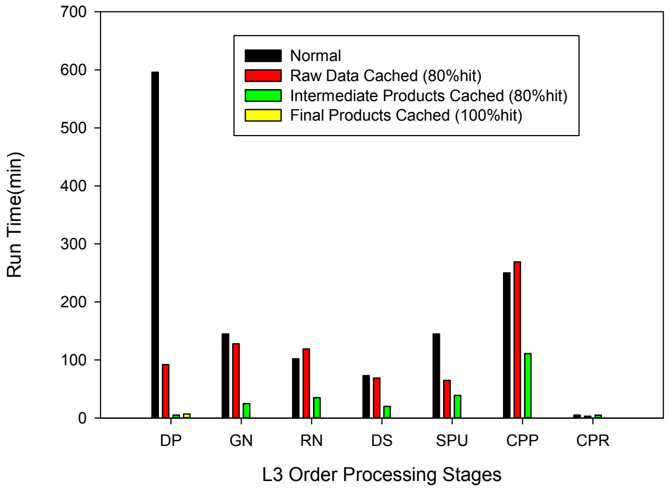

Fig. 16 Runtime of each stage of NPP with four different scenarios

5.2 Related experiments on workflow management

In order to test the relevant performance of MDCPS in workflow management, we used part of the test data for NDVI products to conduct related concurrent expansion experiment, and to test the performance of MDCPS under different scheduling strategies. We tested multi-task concurrent scene under EET, EET + FAT and EET + EAT + FAT three scheduling strategies. The average time-consume is shown in Figure 18. By comparison of time-consume, it could be

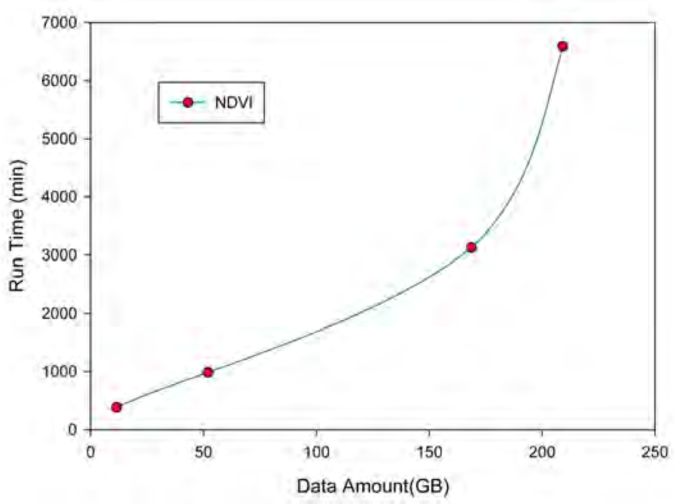

Fig. 17 Runtime of NDVI with the increase of data volume

found that with the increase in the amount of concurrency tasks, task time-consume under the first two scheduling strategies increased rapidly, and that under the last strategy shown a relatively stable growth. The scalability of EET + EAT + FAT is better than the first two scheduling strategies. We can see that the "Near Data Calculation" scheduling strategies adopted in MDCPS can increase the efficiency of remote sensing data production.

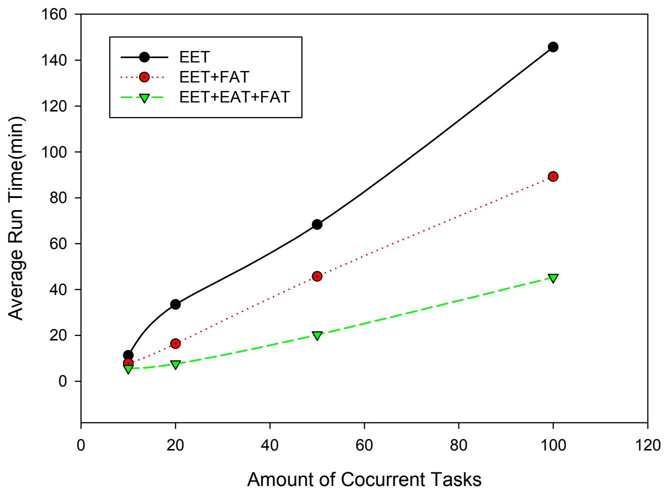

Fig. 18 RunTime of different workflow scheduling scenarios of f NDVI production

6 Discussion

6.1 System Architecture

MDCPS adopts a centralized system architecture to achieve the management of distributed multi-source RS data and production tasks. This centralized system architecture is easy to implement. It not only make good use of the existing centralized grid middleware such as Globus Grid Security Infrastructure (GSI), GridFTP, Ganglia and Gfarm to build, but also has little influence on the current architecture of data centers, just adding a task execution agent system in the satellite datacenters could meet the needs of large-scale production. In addition, the centralized management of workflow can be efficient in global task decomposition and scheduling, and it will significantly improve production efficiency. To avoid the single point of failure in a centralized system, MDCPS could resolve this problem by using the redundant backup of the metadata in back-end databases and distributed replicas in Gfarm.

6.2 System Feasibility

In general, data transmission and processing accounted for a large proportion in a distributed systems for large-scale data processing. As can be seen from the experiment performance, the data management system in MDCPS can reduce the duplication of data transmission and processing by using the distributed RS data cache and dynamic data scheduling strategy. In contrast to a general production system, although the complexity of the data management system will be increased due to the management of cached raw data, cached intermediate products and their dependencies, but the efficiency of large-scale production process can be significantly improved. Thus, the data management system is the key point of a RS data production system based on multi-center architecture.

In the co-processing of multiple satellite data center, each type of RS data is only distributed on several particular data centers. In addition, the tasks execution agent system in each data center has limited computing resources. These reasons result in the workflow scheduling's certain particularity in MDCPS. As can be seen from the performance of the experimental task scheduling, the strategy of "Near Data Calculation" scheduling which MDCPS adopted, optimizes the workflow scheduling model from the perspective of data transmission, task queue status and the performance of computing resources. The model of multi-objective optimization scheduling is applicable to data-intensive computing in the multi-datacenter co-processing.

6.3 System Scalability

MDCPS has achieved the unified management of data resources, computing resources and algorithm resources, which plays an important role in expanding its ability. When the system wants to add new resources, it only needs to deploy the TEPS in datacenter and register its metadata information to master datacenter. It can easily integrate new multi-source remote sensing data, processing algorithms and computing resources. And its extension tools, such as automated deployment scripts, will help users expanding the system on a cluster or cloud computing platform quickly.

7 Conclusion and Future Work

Constructing a remote sensing data processing system based on multiple satellite data centers infrastructure is an effective solution to the problem of massive multi-source remote sensing data processing and analysis. And it is important to support the large scale and global remote sensing application projects. In this systematic project, data management and workflow management are the key issues to build a reliable and efficient distributed processing system.

This paper summarized the current status of distributed remote sensing data processing across multiple satellite data centers and analyzed the reasons of low efficiency in co-processing from the perspective of data management. In order to solve the problems of massive data migration, we presented a distributed caching strategy of the raw data, intermediate products and final products. Combined with the Gfarm distributed file system, we implemented a distributed data management system in MDCPS. Aiming at the problems of distributed process task management, we completed the decomposition of complex processing tasks by processing repositories. With help of multi-level orders task repositories and Kepler workflow template, we achieved automated workflow construction. In addition, we designed a two-level distributed scheduling framework for dispatching processing tasks. The NDVI and NPP production experiments showed that the distributed remote sensing data caching and the scheduling strategies of ”Near Data Calculation” could significantly improve the overall efficiency.

In the future, more work will be done to better meet the massive remote sensing data production needs based on MDCPS, including developing the user-defined knowledge repositories of remote sensing data processing, providing a service for users to define their own processing workflow based on Kepler, optimize the knowledge base of RS data production and the heuristic scheduling rules by using the intelligent mobile agent technique, and improving the performance of data distribution strategies to optimize the infrastructure and services of MDCPS.

Acknowledgements Dr. Yan Ma's work is supported by the National High Technology Research and Development Program of China (”863”Program) (No.2013AA12A301). The authors would also like to acknowledge the editors and anonymous reviewers for their valuable comments on the manuscript.

References

1. Holmgren, J., Persson, Å., Söderman, U.: Species identification of individual trees by combining high resolution lidar data with multi-spectral images. International Journal of Remote Sensing **29**(5), 1537–1552 (2008)
2. Hall, F.G., Hilker, T., Coops, N.C., Lyapustin, A., Huemmrich, K.F., Middleton, E., Margolis, H., Drolet, G., Black, T.A.: Multi-angle remote sensing of forest light use efficiency by observing pri variation with canopy shadow fraction. Remote Sensing of Environment **112**(7), 3201–3211 (2008)
3. Lunetta, R.S., Knight, J.F., Ediriwickrema, J., Lyon, J.G., Worthy, L.D.: Land-cover change detection using multi-temporal modis ndvi data. Remote sensing of environment **105**(2), 142–154 (2006)
4. McCabe, M.F., Wood, E.F.: Scale influences on the remote estimation of evapotranspiration using multiple satellite sensors. Remote Sensing of Environment **105**(4), 271–285 (2006)
5. Nasa eosdis web site. `http://www.esdis.eosdis.nasa.gov/`
6. Chi, M., Plaza, A., Benediktsson, J.A., Sun, Z., Shen, J., Zhu, Y.: Big data for remote sensing: Challenges and opportunities. Proceedings of the IEEE (2015)
7. Institute of remote sensing and digital earth,chinese academy of science. `http://english.radi.cas.cn/`
8. Zhang, W., Wang, L., Ma, Y., Liu, D.: Design and implementation of task scheduling strategies for massive remote sensing data processing across multiple data centers. Software: Practice and Experience **44**(7), 873–886 (2014)
9. Bartholomé, E., Belward, A.: Glc2000: a new approach to global land cover mapping from earth observation data. International Journal of Remote Sensing **26**(9), 1959–1977 (2005)
10. Scharlemann, J.P., Benz, D., Hay, S.I., Purse, B.V., Tatem, A.J., Wint, G.W., Rogers, D.J.: Global data for ecology and epidemiology: a novel algorithm for temporal fourier processing modis data. PloS one **3**(1), e1408 (2008)
11. Petcu, D., Gorgan, D., Pop, F., Tudor, D., Zaharie, D.: Satellite image processing on a grid-based platform. International Journal of Computing **7**(2), 51–58 (2014)
12. Cossu, R., Bally, P., Colin, O., Fusco, L.: Esa grid processing on demand for fast access to earth observation data and rapid mapping of flood events. European Geosciences Union General Assembly (2008)
13. Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., Tuecke, S.: The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. Journal of network and computer applications **23**(3), 187–200 (2000)
14. Kussul, N., Shelestov, A., Skakun, S.: Grid approach to satellite monitoring systems integration (2008)
15. Tudor, D.: Mediogrid: a grid-based platform for satellite image processing (2007)
16. Ramapriyan, H.K., Behnke, J., Sofinowski, E., Lowe, D., Esfandiari, M.A.: Evolution of the earth observing system (eos) data and information system (eosdis). In: Standard-Based Data and Information Systems for Earth Observation, pp. 63–92. Springer (2010)
17. Zhang, X., Jiang, J., Zhang, X., Wang, X.: A data transmission algorithm for distributed computing system based on maximum flow. Cluster Computing **18**(3), 1157–1169 (2015)
18. Cafaro, M., Epicoco, I., Quarta, G., Fiore, S., Aloisio, G.: Design and implementation of a grid computing environment for remote sensing. High Performance Computing in Remote Sensing p. 281 (2007)
19. Hoschek, W., Jaen-Martinez, J., Samar, A., Stockinger, H., Stockinger, K.: Data management in an international data grid project. In: Grid ComputingGRID 2000, pp. 77–90. Springer (2000)
20. Di, L.: The development of remote-sensing related standards at fgdc, ogc, and iso tc 211. In: Geoscience and Remote Sensing Symposium, 2003. IGARSS'03. Proceedings. 2003 IEEE International, vol. 1, pp. 643–647. IEEE (2003)
21. Coleşa, A., Ignat, I., Opriş, R.: Providing high data availability in mediogrid. In: Symbolic and Numeric Algorithms for Scientific Computing, 2006. SYNASC'06. Eighth International Symposium on, pp. 296–302. IEEE (2006)

22. Tatebe, O., Hiraga, K., Soda, N.: Gfarm grid file system. New Generation Computing **28**(3), 257–275 (2010)
23. Wang, L., Tao, J., Ranjan, R., Marten, H., Streit, A., Chen, J., Chen, D.: G-hadoop: Mapreduce across distributed data centers for data-intensive computing. Future Generation Computer Systems **29**(3), 739–750 (2013)
24. Shvachko, K., Kuang, H., Radia, S., Chansler, R.: The hadoop distributed file system. In: Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on, pp. 1–10. IEEE (2010)
25. Wang, Y., Liu, Z., Liao, H., Li, C.: Improving the performance of gis polygon overlay computation with mapreduce for spatial big data processing. Cluster Computing **18**(2), 507–516 (2015)
26. El Homsi, A.: Workflow system and method (2006). US Patent 7,065,493
27. Guo, H., Wang, L., Chen, F., Liang, D.: Scientific big data and digital earth. Chinese Science Bulletin **59**(35), 5066–5073 (2014)
28. Yu, J., Buyya, R., Ramamohanarao, K.: Workflow scheduling algorithms for grid computing. In: Metaheuristics for scheduling in distributed computing environments, pp. 173–214. Springer (2008)
29. Song, W., Yue, S., Wang, L., Zhang, W., Liu, D.: Task scheduling of massive spatial data processing across distributed data centers: What's new? In: Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on, pp. 976–981. IEEE (2011)
30. Zhang, W., Wang, L., Liu, D., Song, W., Ma, Y., Liu, P., Chen, D.: Towards building a multi-datacenter infrastructure for massive remote sensing image processing. Concurrency and Computation: Practice and Experience **25**(12), 1798–1812 (2013)
31. Multiple satellite data centre workflow scheduling algorithm on basis of near data calculation principle (2015). URL `https://www.google.com/patents/CN104484230A?cl=en`. CN Patent App. CN 201,410,851,865
32. Wang, L., Lu, K., Liu, P., Ranjan, R., Chen, L.: Ik-svd: dictionary learning for spatial big data via incremental atom update. Computing in Science & Engineering **16**(4), 41–52 (2014)
33. Wang, L., Khan, S.U., Chen, D., Kołodziej, J., Ranjan, R., Xu, C.Z., Zomaya, A.: Energy-aware parallel task scheduling in a cluster. Future Generation Computer Systems **29**(7), 1661–1670 (2013)
34. Wang, L., Ranjan, R., Kolodziej, J., Zomaya, A.Y., Alem, L.: Software tools and techniques for big data computing in healthcare clouds. Future Generation Comp. Syst. **43**, 38–39 (2015)
35. Wang, L., Chen, D., Hu, Y., Ma, Y., Wang, J.: Towards enabling cyberinfrastructure as a service in clouds. Computers & Electrical Engineering **39**(1), 3–14 (2013)
36. Chen, L., Ma, Y., Liu, P., Wei, J., Jie, W., He, J.: A review of parallel computing for large-scale remote sensing image mosaicking. Cluster Computing **18**(2), 517–529 (2015)
37. Jaeger, E., Altintas, I., Zhang, J., Ludäscher, B., Pennington, D., Michener, W.: A scientific workflow approach to distributed geospatial data processing using web services. In: SSDBM, vol. 3, pp. 87–90. Citeseer (2005)
38. Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E.A., Tao, J., Zhao, Y.: Scientific workflow management and the kepler system. Concurrency and Computation: Practice and Experience **18**(10), 1039–1065 (2006)
39. Maheswaran, M., Ali, S., Siegal, H., Hensgen, D., Freund, R.F.: Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems. In: Heterogeneous Computing Workshop, 1999.(HCW'99) Proceedings. Eighth, pp. 30–44. IEEE (1999)
40. Nita, M.C., Pop, F., Voicu, C., Dobre, C., Xhafa, F.: Momth: multi-objective scheduling algorithm of many tasks in hadoop. Cluster Computing **18**(3), 1011–1024 (2015)