High-Throughput Geocomputational Workflows in a Grid Environment

Jia Liu, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences

Yong Xue and Dominic Palmer-Brown, London Metropolitan University

Ziqiang Chen and Xingwei He, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences

A grid-computing platform facilitates geocomputational workflow composition to process big geosciences data while fully using idle resources to accelerate processing speed. An experiment with aerosol optical depth retrieval from satellite data shows a 25 percent improvement in runtime over a single high-performance computer.

Technological advancements and their global dissemination are often predicated on the integration of traditionally separate fields, such as geoscience and computer science, to obtain fresh approaches for solving complex problems, such as efficiently processing data about a highly integrated Earth system, which comprises subsystems that cover interlinked aspects of the Earth's hydrosphere, atmosphere, and geological composition.[1] Geographers and geoscientists have assembled massive amounts of digital information with spatial attributes, which—when combined with the extreme complexity of open geospatial problems—has motivated geocomputation. Geocomputation is a discipline that exploits computational advances to solve a variety of problems in integrating and analyzing Earth system data. Geocomputational workflows, particularly those in the retrieval of quantitative remote-sensing data, consist of several subworkflows that contain data dependencies and are both data and computing intensive.[2,3]

Grid computing, already an attractive environment for developing and running large-scale applications in domains other than geoscience, is a potential solution for processing these workflows, which are characterized by volumes of spatiotemporal data. The grid environment provides standardized access to a pool of heterogeneous and distributed resources, creating the illusion of a powerful computer that can break down the data-processing bottleneck characteristic of large-scale remote-sensing applications.

Despite grid computing's potential use in these applications, little work has focused on adapting it to this context. To address that need, we developed the Remote Sensing Information Service Grid Node (RSSN)—a high-throughput geocomputational grid-computing environment based on the HTCondor (formerly Condor; http://research.cs.wisc.edu/htcondor/description.html) system—which increases an individual computer's processing power by

- accelerating and facilitating the retrieval of aerosol optical depth (AOD) data (which measures the extent to which atmospheric particles extinguish solar radiation) through a GUI that lets users compose, submit, and execute workflows;
- fully exploiting idle computing resources; and
- using workflow-optimized scheduling and execution.

To validate RSSN's feasibility, we retrieved a year's worth of AOD data to evaluate the workflow composition, workflow task-execution performance, and time-series dataset generation for AOD data retrieval and processing. We chose AOD retrieval because it is both a computing- and data-intensive application.

We also compared RSSN's performance with that of a single high-performance computer, which scientists typically use daily in the retrieval of remote-sensing image data. Our results show that overall runtimes decreased 25 percent over runtimes with the high-performance computer. These results imply that RSSN can significantly facilitate and accelerate AOD retrieval from satellite data and could be a promising solution for other problems related to high-throughput geocomputation, such as retrieving the temperature of land surfaces and calculating the albedo (surface reflectivity measure) and leaf-area index.

COMPUTING IN THE GRID ENVIRONMENT

Geocomputational workflow in the grid environment has many challenges. The main one is that these workflows, particularly those in quantitative remote-sensing applications, typically require data with varying time steps and resolution. For example, the same application might require a 10-year AOD dataset at 1-km resolution from the Moderate Resolution Imaging Spectrometer (MODIS) satellite sensor's data—29 terabytes (Tbytes) of original data—as well as a 30-year AOD dataset at 0.1-degree resolution from the National Oceanic and Atmospheric Administration's (NOAA's) Advanced Very High Resolution Radiometer (AVHRR) data—100 Tbytes of original data.[4] Not only does the volume differ between datasets, but each dataset involves disparate processing time. Thus, efficient data management must not only address throughput but also select the appropriate computing mode.

THE GRID PROVIDES ACCESS TO HETEROGENEOUS AND DISTRIBUTED RESOURCES TO BREAK DOWN THE DATA-PROCESSING BOTTLENECK.

This challenging mix of data and computational intensity is at the root of other issues, such as model organization, accelerating distributed processing, workflow-related problems, and resource scheduling. Progress in solving all these issues is apparent, but open problems remain.

Model organization

Efficiently and automatically organizing and executing numerous preprocessing and inverse models is essential to handling the mix of computational intensity and big data within an application. To enable the calculation of myriad geophysical parameters including the aerosol content for each observation—oxygen, carbon dioxide, particle matter, and so on—the MODIS Adaptive Processing System generates nearly 2.5 Tbytes of land, atmospheric, and oceanic geophysical parameters daily on a combination of supercomputers and commodity Intel Pentium processors.[5]

Accelerating data acquisition and distribution

Complexities associated with the combination of data volume and variety and computational intensity can significantly delay data acquisition and distribution. Several research groups have proposed solutions that use grid computing to mitigate these delays. Taries.net, for example, is a model that uses a distributed system built on grid computing's basic principles to process images from remote-sensing observations.[6]

The GiSHEO platform (on-demand grid services for higher education and training in Earth observation) uses grid and Web services technologies to process remote-sensing data for training quantitative data–retrieval models for Earth observation.[7] GiSHEO consists of a *processing-services* component, which comprises the machine interface (visible as a Web service) and workload management system, as well as *data-management*, *workflow- engine*, *user-interface*, and *e-learning* components.

Another effort to accelerate data distribution is the Namibia SensorWeb Pilot Project, an international multidisciplinary initiative to create a testbed for evaluating and prototyping key technologies suchas SensorWebs, grids, and computational clouds, to enable the rapid data product acquisition and distribution to support flood monitoring.[8] The system provides access to real- time data about rainfall estimates and forecasts of flood potentials, and can rapidly generate flood maps. Computational and storage services are enabled through an infrastructure that relies on both grid and cloud computing.

HTCondor is open source soft- ware developed by the Center for High Throughput Computing at the University of Wisconsin–Madison to support high-throughput computing on large collections of computing resources with distributed ownership. One research group used HTCondor to support the validation of a data-placement strat- egy in applications with big data and intensive computation, such Montage, which generates science-grade mosaics of the sky.[9] The goal is to demonstrate that, by combining the functionality of the data-replication service for data placement and the Pegasus system for workflow management, data-intensive workflows can execute faster with asynchronous data placement than with on-demand data staging by the workflow-management system. Pegasus relies on HTCondor's DAGMan workflow engine to launch tasks and maintain intertask dependencies.

Another effort used HTCondor to establish a system for processing Earth observation images from remote sensors that integrated components such as the Virtual Data Toolkit and the Globus Toolkit. Integration enabled structural biology researchers to securely share large volumes of data and computational workflows,[10] which proved effective in rapidly processing, distributing, and sharing massive numbers of remote-sensing images.[11]

Another approach to solving delays in remote-sensing data acquisition and distribution is the grid-enabled paral- lel algorithm of geometric correction (GPGC), which computes an irregular local output area. The area allows the system to change the parallel method's frequent and fine-grained communication mode to a delayed but concentrated communication-exchange mode.[12] By enabling geometric correction and minimizing communication or synchronizationduring time-consuming resampling, GPGC effectively supports ChinaGrid, a project sponsored by the China Ministry of Education to provide high-performance services in a grid computing environment.

SCIENTIFIC WORKFLOW TECHNOLOGY ENABLES THE COMPOSITION AND EXECUTION OF COMPLEX ANALYSIS ON DISTRIBUTED RESOURCES.

Streamlining scientific workflow

Not all applications require an expert understanding of remote-sensing data, and demand is growing for the ability to immediately retrieve simple and easily understood information from remotely sensed data that has already undergone complex processing and analysis.

To meet this demand, researchers have attempted to apply workflow composition and management technology in a grid environment. Scientific workflow technology has become essential in many applications, enabling the composition and execution of complex analysis on distributed resources.

Grid computing with workflow technology has four main advantages:[13]

- it provides a composition function for grid applications;

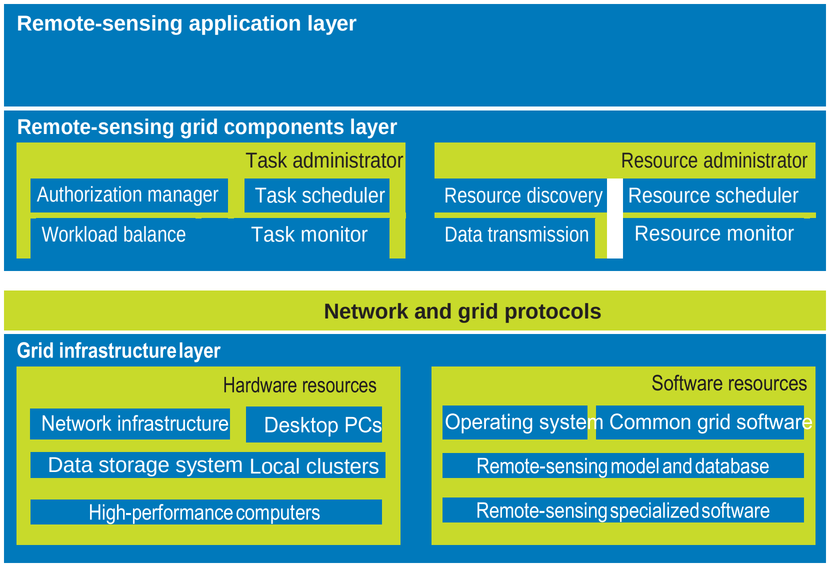

FIGURE 1. RSSN's three-layer architecture. The layers ensure that remote-sensing information is communicated within components in the simplest form and as rapidly as possible. The network and grid protocols are middleware services to support a common set of applications in a distributed network environment.

- it uses local resources, thereby increasing throughput and reducing implementation cost;
- it provides users with special-purpose processing and task solving across multiple management areas; and
- it promotes interorganizational cooperation.

The technology life cycle includes workflow composition and representation, the creation of data models, the mapping of modeling concepts into an executable representation, and execution-model creation. Although many business workflow–management systems exist, they lack features and characteristics that are essential in scientific applications. Special dynamic workflow management for quantitative remote sensing is still nascent.

Efficient resource scheduling

Scheduling is a key issue in applications with big data and high computational demands. Most grid scheduling algorithms are based on heuristic scheduling, which usually takes computing-capability parameters—the number of CPU cores and CPU clock speed, for example—as the workload vector. Data transfer is largely ignored. With additional considerations such as workflow model, scheduling criteria and process, and resource and task model, grid scheduling becomes even more challenging and complicated.

In documenting a study of the relationship between asynchronous data placement and scheduling,[14] the authors suggested that combining data scheduling and computation is an effective solution for performance problems in data-intensive grid computing.

Another group that studied data placement and scheduling in a grid environment, proposed placing data before computation execution. They also proposed a method to combine data placement and workflow management,[9] but their method applies only to the lightweight data replicator service and workflow mechanism in Pegasus (http://pegasus.isi.edu).

A dedicated data scheduler, Stork,[15] considers data placement as the highest-priority operation, efficiently queuing, scheduling, and monitoring data-transmission services. Experiments show that Stork enhanced the data-transmission service's efficiency and fault tolerance and reduced the dependence on user interaction in a complex data-transmission application. One disadvantage, however, is that Stork does not support the Windows OS.

RSSN: HIGH THROUGHPUT AND EFFICIENT SCHEDULING

RSSN aims to address the specific problems of applying grid computing solely to acquire and distribute remote-sensing data, such as the need for faster throughput and more efficient scheduling that uses idle computer resources for data-intensive computing applications. We developed RSSN using HTCondor running on a Windows system. RSSN's computing nodes are commodity PCs used in daily scientific work.

Architecture and task processing

Figure 1 shows the RSSN architecture. At the bottom is the *grid infrastructure* layer, which includes the software and hardware entities. The *remote-sensing grid components* layer includes task and resource monitors, the task scheduler, resource discovery, and data transmission—all to support the *remote-sensing application* layer at the top. The application layer packages the lower-layer functions and supports the sharing and servicing of remote-sensing information. The grid middleware is HTCondor, which serves as the local resources manager to construct RSSN.

We designed RSSN so that components within each layer can share characteristics and thus can build on any lower-layer capabilities and behaviors.

Figure 2 shows the task and processing flow in RSSN:

- Users compose workflows through the grid workflow

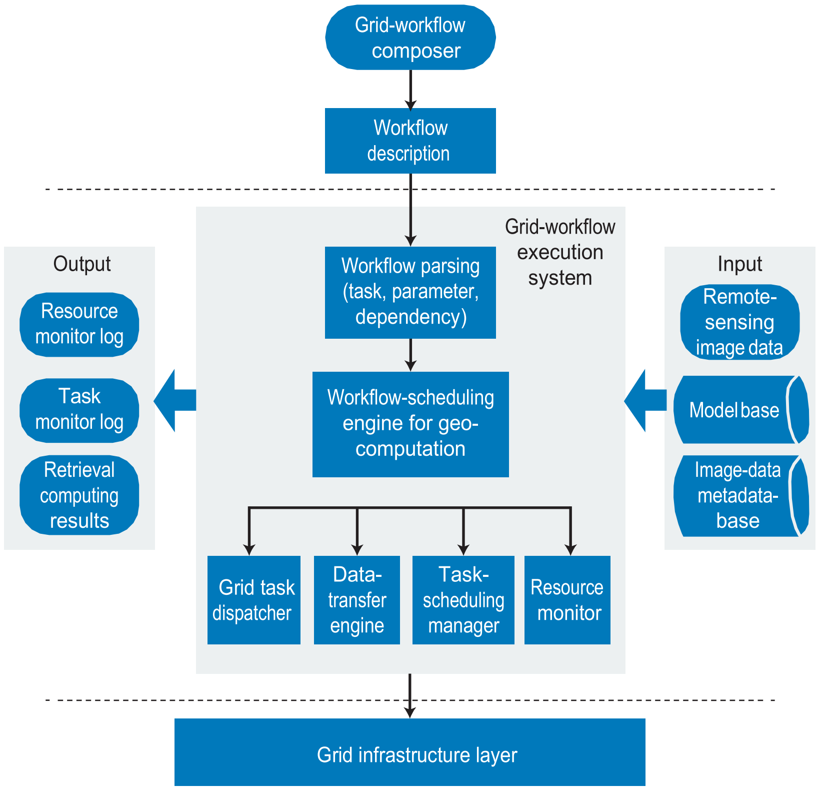

FIGURE 2. Task and processing flow in RSSN. Through the GUI (above upper dashed line), users compose workflows and submit them for execution. Scheduling is handled by the grid-task dispatcher, data-transfer engine, task-scheduling manager, and resource monitor. The workflow execution system feeds into the grid infrastructure layer, which powers its functions.

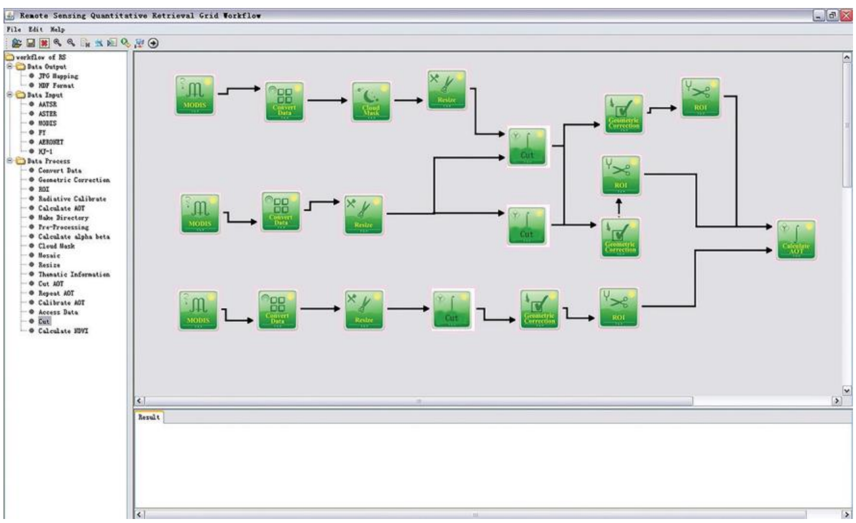

FIGURE 3. Workflow composition in RSSN. The user has composed a workflow for AOD retrieval through the GUI by dragging icons from a list displayed to the left of the composition area. The icons represent data type, data and processing models, and corresponding algorithms. RSSN converts the graphical workflow to an XML file, which it uses to communicate with the webserver about the users' workflow information.

composer GUI, selecting and defining models and data types.
- Users submit the composed workflows and RSSN's workflow parsing service extracts task, data parameters, and depen- dency information on the basis of the model base and image-data metadatabase.
- RSSN generates executable workflow by parsing results and executable model programs. The workflow-scheduling engine determines task scheduling and binds the task with resources.
- The grid task dispatcher and data transfer components dispatch tasks and remote-sensing image data to grid-computing resources.

Workflow composition

RSSN's GUI facilitates the composition of remote-sensing workflows by allowing users to fully employ CPU resources that typically remain idle on scientific computers for daily work.[16] The main aspects of workflow composition are data structure, model management, the actual composition, and its parsing.

Workflow composition and parsing. RSSN uses the Apache Tomcat (http://tomcat.apache.org) webserver, and a Java-programmed Web application. Figure 3 shows the GUI, which is displaying an AOD retrieval workflow.

Although the workflow composer runs on the client computer, RSSN generates a socket connection, which it uses to communicate the workflow, converted to an XML workflow description file, to the webserver. The workflow parse component analyzes the XML file to obtain task information, parameters, and dependencies and generates executable programs according with HTCondor rules. Once the task monitor receives the XML file, the parsing component submits the analysis results to the HTCondor pool.

Data structure and model management. At present, RSSN processes raster data and uses the Oracle relational database to manage it, storing image data in the file directory and managing the data path and other metadata information in the database. RSSN uses the directed acyclic graph data structure, which includes two lists.[16] The *nodes* list saves the remote-sensing algorithm's quantitative information, such as the source data's spatial resolution and latitude and longitude ranges. The nodes list also includes user-specified parameters that guide the tasks' parallelization. The *relationship* list notes dependencies among algorithms.

The Oracle relational database manager manages model and algorithm metadata and information such as the executable algorithms path—all of which are registered in the data- base. Database tables are divided into model tables and relevant algorithm tables, which include the Algorithm_Info, Algorithm_Semantics, Algorithm_Inputs, and Algorithm_Outputstables.

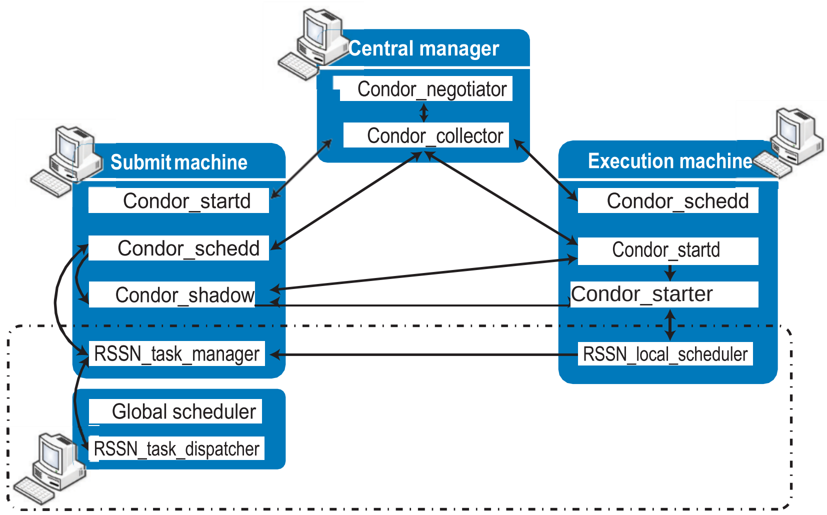

FIGURE 4. Workflow scheduling and execution mechanism extended from HTCondor. Elements in the dashed box are specific to RSSN.

Workflow scheduling and execution

Figure 4 shows RSSN's workflow scheduling and execution mechanism, which is an extension of HTCondor's approach. RSSN uses HTCondor's Classified Advertisements (ClassAds) mechanism to match machines and tasks.

Subtask creation and matching. Workflow scheduling starts when the *global scheduler* accesses data nodes to request the data list. It then analyzes the workflow script and data list and divides the entire user task into subtask packages. Each subtask package is described by ClassAds; HTCondor uses the description to match tasks with available machines. During remote-sensing data transmission, which can occur at any time, RSSN records the network bandwidth between computing nodes and the data server, as well as the task execution success rate, idle time, and other aspects of computing node status. It then summarizes the recorded information and registers it as additional attribute data in HTCondor's task scheduling configuration file, in essence expanding ClassAds attributes.

The *RSSN task manager* submits the subtask packages to the HTCondor pool. If there is a match, the task manager sends the task packages to the matched machine for execution. Once the executing machine receives the task packages, the RSSN task manager starts the local task scheduler to process the task package. During the local scheduler's working cycles, the RSSN task manager monitors the nodes' workloads and other status aspects while periodically checking the job and machine lists for potential new matches.

The cycle-scheduling time span should be based on the expected data-transfer time. For example, in our AOD retrieval experiment, we found that the average file size of a subtask package is about 200 Mbytes—about a 20-second data-transfer—so sched- uling time should not be less than 20 seconds.

Subtask scheduling. When the local scheduler receives the sub- task packages, it queues them as first-come-first-served and generates two job lists: one for each package's data transmission task and one for the computing task.

In general, there is no dependency between input data to the subtask packages and the intermediate results from each computational step. Thus, while the computing task in the previous subtask package is running, the RSSN task manager schedules data transmission for the current package synchronously. The result is improved CPU and network bandwidth use and a shorter overall task-execution time.

Submitting results. As soon as the subtask running on the computing node completes, the RSSN task manager sends the result to the machine that submitted the workflow composition. The *task monitor* running on the user's machine collects the subtask package information; the result might need to be organized together automatically if necessary.

Rescheduling failed tasks. The local scheduler also monitors the entire scheduling and execution process. If any part of the process fails, the scheduler will record the package number and error message, discard the corrupt intermediate data, and send the log file to the user. The local task manager can reschedule the failed task package.

TO IMPROVE CPU AND BANDWIDTH USE, CURRENT-PACKAGE DATA TRANSMISSION OCCURS SYNCHRONOUSLY WITH PREVIOUS-PACKAGE TASK EXECUTION.

Parallel scheduling and execution

Remote-sensing application workflows generally have subworkflows that could be scheduled and executed in parallel in a coarse-grained pattern. RSSN implements this approach by adding an agent layer between the webservers and computing pool. The workflow-parsing component analyzes XML files and generates executable programs for each subworkflow, which it submits to *agents*—computers that handle subworkflows in the HTCondor pool. The agents gather the submitted subworkflow tasks after tasks they complete.

The main idea is to collapse the preprocessing stage and reduce the overhead from the I/O of one submission machine by adding agents that work in parallel as submission machines.

Fault-tolerance mechanism

At present, RSSN supports fault tolerance by relying on HTCondor's middleware, which provides a process checkpoint and a mechanism to migrate failed processes by assigning a unique global ID for each computing task, and by setting a time threshold for task suspension because of an unexpected computing error. When the task exceeds the threshold, the RSSN task manager will reschedule the corresponding subtasks.

CASE STUDY: AOD RETRIEVAL

AOD is a significant parameter in remote-sensing data because it reflects aerosol optic properties, which provide insights into many scientific concerns, such as aerosol radiative forcing (the difference in sunlight absorbed and energy released back into the atmosphere), cloud microphysics, and atmospheric correction of satellite images. AOD retrieval over a long operational period involves big data and complicated processing, so retrieving data with high precision and resolution remains difficult and time-consuming.

Retrieving AOD from a satellite, such as MODIS, eliminates the need to preprocess data, but requires organizing many workflows. To date, research in AOD retrieval has focused more on exploring algorithms and less on exploring how to organize and reuse geocomputational workflows in a way that would accelerate computing and fully use available computing resources.

To examine how RSSN supports workflow organization, we retrieved a year of MODIS satellite AOD data from over China and evaluated how RSSN facilitated workflow organization from three perspectives: workflow composition, task-execution performance and time-series dataset generation.

Workflow composition

We used the Synergic Retrieval of Aerosol Property MODIS (SRAP-MODIS) algorithm[17] to retrieve AOD data and RSSN's GUI to compose the workflow shown in Figure 3. We selected models, defined the data time and data type, chose supporting algorithms, and added dependencies between models. We saved the workflow as an XML file and submitted it to the webserver for parsing and execution in the HTCondor computing pool.

Execution performance

We used data from January 2008 (while the satellite was over China), which we acquired from the National Aeronautics and Space Administration's Distributed Active Archive Center, to produce AOD at 1-km resolution. We processed the data on a single computer, on a personal high-performance computer (PHPC), and on RSSN. Figure 5 shows the results for each day.

The single PC took from 43.5 to 62.5 hours to process daily AOD data, with an average time of 50 hours. The PHPC with no modification to the programs provided by scientific researchers took from 25.9 to 38.2 hours, with an average of 33 hours. RSSN with optimizing scheduling and execu- tion took only 4.3 to 7.6 hours, with an average of 6.4 hours.

We were also interested in testing performance with a coarse-grained pattern of parallel subworkflows, so we selected several sample days and performed the improved AOD retrieval pattern. Figure 6 shows the results, which isolate three stages: preprocessing, creating the image-data mosaic and partitioning it, and inverting the data. For the four samples of daily AOD retrieval, the preprocessing stage with coarse-grained parallel subworkflows (left bars) reduces the original runtime (right bars) by 20.81, 39.74, 51.54, and 59.41 percent.

The mosaic and partition stages also took less time with a 42.27, 40.14,

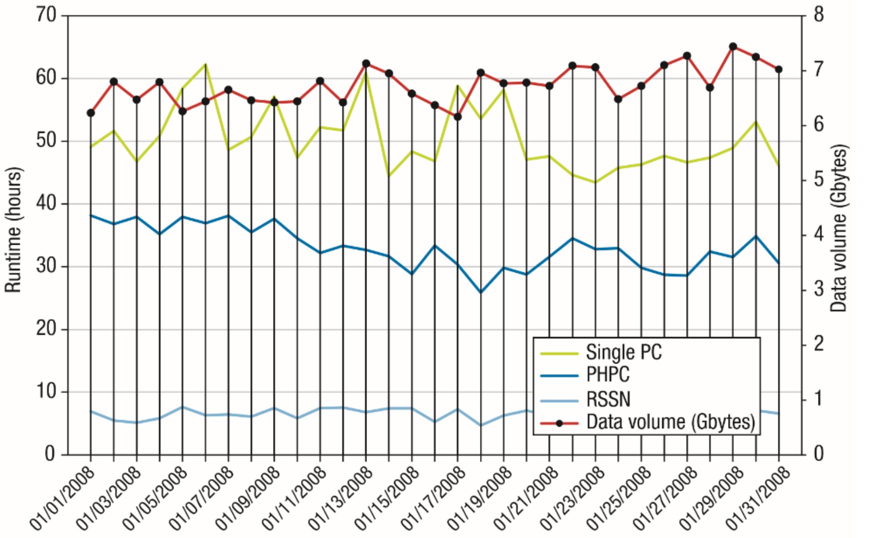

FIGURE 5. Time to process the Synergic Retrieval of Aerosol Property (SRAP)-MODIS algorithm in different computing environments during January 2008. The single PC is a computer with an Intel Core i5-3450 CPU running at 3.1 GHz with four cores and 4 Gbytes of memory. PHPC represents the Sugon PHPC200, a personal high-performance computer equipped with two dual-route Intel 5600 multicore computing modules.

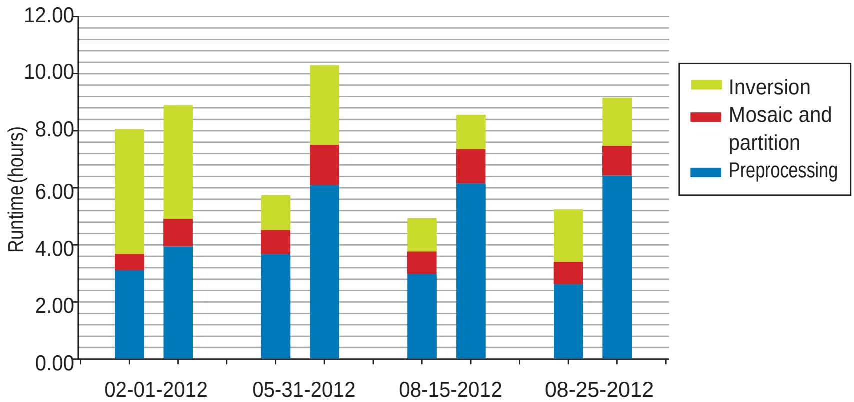

FIGURE 6. Sample results of AOD retrieval with (left bars in each pair) and without (right bars in each pair) a coarse-grained pattern of subworkflows running in parallel. The length of all three stages—preprocessing, creating the mosaic and partitioning the data, and inverting the data to solve the equations—is the total runtime in each case, which is consistently and often dramatically lower with parallel execution.

TABLE 1. Average monthly runtime, data volume, and task number for AOD retrieval data from September 2011 to August 2012.

Month	Preprocessing runtime (hrs)	Mosaic, partitioning, and inversion runtime (hrs)	Total runtime (hrs)	Volume (Gbytes)	Number of tasks
9-2011	3.78	1.67	5.45	518	47.67
10-2011	3.64	2.61	6.25	526	46.29
11-2011	3.32	4.23	7.55	426	38.93
12-2011	3.72	2.92	6.64	388	35.03
01-2012	3.50	2.50	6.00	409	36.96
02-2012	4.18	2.46	6.64	454	43.72
03-2012	4.35	3.57	7.92	530	47.84
04-2012	4.07	3.51	7.58	520	47.70
05-2012	4.09	3.29	7.38	548	48.39
06-2012	4.40	3.98	8.38	552	50.64
07-2012	4.64	2.68	7.32	553	49.00
08-2012	4.31	1.85	6.16	542	47.48

34.17, and 23.81 percent improvement over the original runtime. The retrieval stages show no apparent improvements. The significant reductions in the preprocessing and mosaic and par- tition stages resulted in a severe drop in total runtime.

Dataset generation and analysis

We used RSSN along with the SRAP-MODIS algorithm to retrieve a year of AOD data. Table 1 gives the average monthly preprocessing runtime, retrieval runtime, total runtime, data volume, and task number. Figure 7 shows results for one AOD parameter, and Figure 8 shows the runtime of daily AOD retrieval. In keeping with the chosen retrieval workflow, task execution takes place in two parallel stages:

- The RSSN task manager submits preprocessing tasks, such as cutting, resizing, and geometric to nodes in the HTCondor pool. Each computing node uses the same program to process its designated image data.
- The machine that submitted the task gathers the results, generates new retrieval tasks, and submits them to the HTCondor pool.

As Figure 8 shows, preprocessing runtime is relatively stable, from 1.65 to 7.81 hours, with an average of 4.00 hours. Runtime for the retrieval stage is from 0.59 to 18.39 hours, with an average of 2.95 hours. The input retrieval data volume is fixed, and runtime two depends primarily on the number of valid pixels, which can vary widely. For example, the valid pixel percentage on 31 March 2012, was 39.49 percent, whereas on 21 October 2011 it was 16.58 percent. The runtime of model SRAP_AOD Retrieval for these two dates is 5.19 and 1.47 hours, respectively. The convergence of iterative processing becomes a retrieval bottleneck.

Grid computing is emerging as a common production environment in scientific research, but work is needed to reap benefits for geocomputational applications that involve the retrieval data from remote sensors. RSSN is a step toward accelerating data acquisition and distribution and facilitating workflow organization. We plan to enhance RSSN by designing and implementing an algorithm to schedule data-intensive workflows and optimize data storage and management.

ACKNOWLEDGMENTS

We thank the Center for High-Throughput Computing at the University of Wisconsin–Madison for the open source HTCondor software used in our research.

This work was supported in part by the Ministry of Science and Technology of China under grant 2013AA122801, the National Natural Science Foundation of China (NSFC) under grant 41271371, and by the CAS-RADI Innovation project under grant Y3SG0300CX.

REFERENCES

1. C.W. Yang, Y. Xu, and D. Nebert, "Redefining the Possibility of Digital Earth and Geosciences with Spatial Cloud Computing," *Int'l J. Digital Earth*, vol. 6, no. 4, 2013, pp. 297–312.
2. Y. Xue et al., "Quantitative Retrieval of Geophysical Parameters using Satellite Data," *Computer*, vol. 41, no. 4, 2008, pp. 33–40.

3. Y. Xue et al., "Workload and Task Management of Grid-enabled Quantitative Aerosol Retrieval from Remotely-Sensed Data," *Future Generation Computer Systems*, vol. 26, no. 4, 2010, pp. 590–598.
4. M.F. Goodchild et al., "Next-Generation Digital Earth," *Proc. Nat'l Academy of Sciences (PNAS)*, vol. 109, 2012; www.pnas.org/content/109/28/11088.full.pdf.
5. E. Masuoka et al., "Evolution of the MODIS Science Data Processing System," *Proc. IEEE Int'l Geoscience and Remote Sensing Symp.*, (IGARSS 01) 2001, pp. 1454–1457.
6. Z.F. Shen et al., "Distributed Computing Model for Processing Remotely Sensed Images Based on Grid Computing," *Information Sciences*, vol. 177, no. 2, 2007, pp. 504–518.
7. D. Petcu et al., "Experiences in Build- ing a Grid-Based Platform to Serve Earth Observation Training Activities," *Computer Standards & Interfaces*, vol. 34, no. 6, 2012, pp. 493–508.
8. N. Kussul et al., "Interoperable Infrastructure for Flood Monitoring: SensorWeb, Grid and Cloud," *IEEE J. Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 6, 2012, pp. 1740–1745.
9. A. Chervenak et al., "Data Placement for Scientific Applications in Distrib- uted Environments," *Proc. 2007 8th IEEE/ACM Int'l Conf. Grid Computing* (Grid 07), 2007, pp. 146–153.
10. I. Stokes-Rees et al., 2012. "An Integrated Science Portal for Collaborative Compute and Data Intensive Protein Structure Studies," *Proc. IEEE 8th Int'l Conf. on E-Science* (E-Science 12), 2012, pp. 1–8.
11. L. Zhong et al., "The Design and Implementation of a Remote Sensing

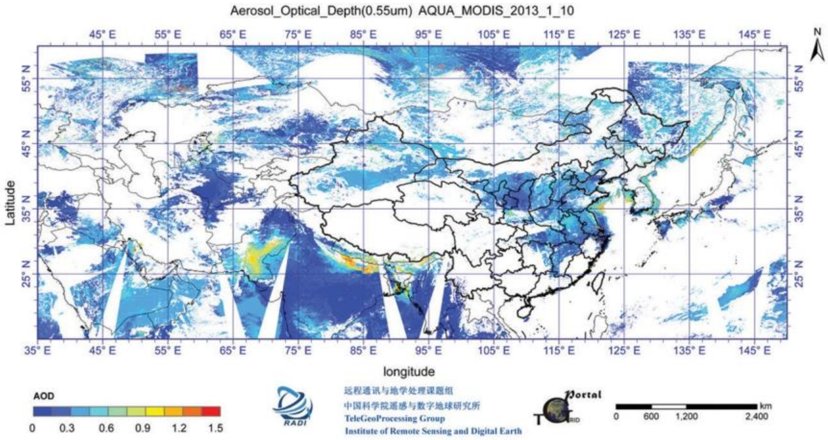

FIGURE 7. A sample AOD retrieval result. Images such as these are typical in AOD data, which is why daily retrieval can take many hours to process. This image is in response to the request to retrieve an image for a single parameter, the AOD at 0.55 μm channel for the AQUA MODIS sensor.

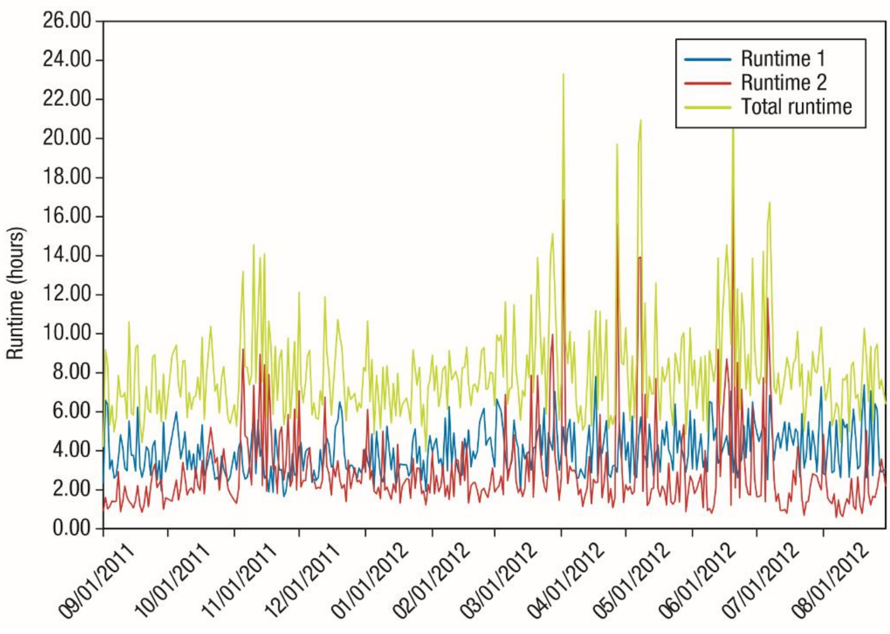

FIGURE 8. Runtime of AOD retrieval from RSSN running SRAP-MODIS algorithm. Runtime 1 represents the time to preprocess submitted tasks; runtime 2 reflects the gathering of results and generation of new retrieval tasks, which is done in parallel with runtime 1; and total runtime is the time between the user's request submission and the end of the entire retrieval process.

About The Authors

JIA LIU is a postgraduate student in cartography and geographic information systems at the Institute of Remote Sensing and Digital Earth at the Chinese Academy of Sciences (RADI-CAS), Beijing. Her research interests include high-performance computing technologies in remote-sensing applications, with an emphasis on grid computing and general-purpose computing on GPUs. Liu received a BSc in remote sensing science and technology from Wuhan University. She is a student member of IEEE. Contact her at libraspring@163.com.

YONG XUE is a professor of computation at London Metropolitan University. His research interests include geocomputation, aerosol optical depth retrieval from remotely sensed data, thermal inertia modeling, and heat exchange calculation for the boundary layer. Xue received a PhD in remote sensing and geographical information systems from the University of Dundee. He is a chartered physicist, a Senior Member of IEEE, a member of the UK Institute of Physics, and an editor of the *International Journal of Remote Sensing*. Contact him at y.xue@londonmet.ac.uk.

DOMINIC PALMER-BROWN is dean of Life Sciences and Computing at London Metropolitan University. His research interests include virtual learning environments, intelligent systems, and neural networks. Palmer-Brown received a PhD in neural networks from Nottingham University. Contact him at d.palmer-brown@londonmet.ac.uk.

ZIQIANG CHEN is a doctoral student in signal processing at RADI-CAS. His research interests include grid computing and workflow management and scheduling. Chen received an MSc in electronics and communications engineering from RADI-CAS. He is a student member of IEEE. Contact him at chenziqiang613@163.com.

XINGWEI HE is a doctoral student in quantitative remote sensing at RADI-CAS. Her research interests include remote-sensing image processing and aerosol optical depth retrieval. He received an MSc in electronics and communications engineering from RADI-CAS. She is a student member of IEEE. Contact her at hexingweiph@163.com.

Image Processing System Based on Grid Middleware," *Proc. Geoinformatics and Joint Conf. GIS and Built Environment: Advanced Spatial Data Models and Analyses*, vol. 7146, 2008, pp. 71462C-1–71462C-8.

12. H.F. Zhou et al., "GPGC: a Grid-Enabled Parallel Algorithm of Geometric Correction for Remote-Sensing Applications," *Concurrency and Computation: Practice and Experience*, vol. 18, no. 14, 2006, pp. 1775–1785.
13. D.P. Spooner et al., 2005. "Performance-Aware Workflow Management for Grid Computing," *The Computer J.*, vol. 48, no. 3, pp. 347–357.
14. K. Ranganathan and I. Foster, "Simulation Studies of Computation and Data Scheduling Algorithms for Data Grids," *J. Grid Computing*, vol. 1, no. 1, 2003, pp. 53–62.
15. T. Kosar and M. Livny, "Stork: Making Data Placement a First Class Citizen in the Grid," *Proc. 24th IEEE Int'l Conf. Distributed Computing Systems* (ICDCS 04), 2004, pp. 342–349.
16. Y. Xue et al., "A High-Throughput Geocomputing System for Remote Sensing Quantitative Retrieval and a Case Study," *Int'l J. Applied Earth Observation and Geoinformation*, vol. 13, no. 6, 2011, pp. 902–911.
17. Y. Xue and A.P. Cracknell, "Operational Bi-Angle Approach to Retrieve the Earth Surface Albedo from AVHRR Data in the Visible Band," *Int'l J. Remote Sensing*, vol. 16, no. 3, 1995, pp. 417–429.