

Sandro Fiore
Giovanni Aloisio *Editors*

Grid and Cloud Database Management

Chapter 14

Grid Technologies for Satellite Data Processing and Management Within International Disaster Monitoring Projects

Nataliia Kussul, Andrii Shelestov, and Sergii Skakun

Abstract This chapter describes the use of Grid technologies for satellite data processing and management within international disaster monitoring projects carried out by the Space Research Institute NASU-NSAU, Ukraine (SRI NASU-NSAU). This includes the integration of the Ukrainian and Russian satellite monitoring systems at the data level, and the development of the InterGrid infrastructure that integrates several regional and national Grid systems. A problem of Grid and Sensor Web integration is discussed with several solutions and case-studies given. This study also focuses on workflow automation and management in Grid environment, and provides an example of workflow automation for generating flood maps from images acquired by the Advanced Synthetic Aperture Radar (ASAR) instrument aboard the Envisat satellite.

14.1 Introduction

Nowadays, satellite monitoring systems are widely used for the solution of complex applied problems such as climate change monitoring, rational land use, environmental, and natural disasters monitoring. To provide solutions to these problems not only on a regional scale but also on a global scale, a “system of systems” approach that is already being implemented within the Global Earth Observation System of Systems¹ (GEOSS) and Global Monitoring for Environment and Security²

¹<http://www.earthobservations.org>.

²<http://www.gmes.info>.

N. Kussul (✉) · A. Shelestov · S. Skakun

Space Research Institute NASU-NSAU, Glushkov Prospekt 40, building 4/1, Kyiv 03680, Ukraine

e-mail: inform@ikd.kiev.ua; andrii.shelestov@gmail.com; serhiy.skakun@ikd.kiev.ua

(GMES) is required. This approach envisages the integrated use of satellite data and corresponding products and services, and integration of existing regional and international satellite monitoring systems.

In this chapter, existing approaches and solutions to satellite monitoring systems integration with an emphasis on practical issues in this area are discussed. The following levels of system integration are considered: *data integration level* and *task management level*. Two examples of system integration that use these approaches are discussed in detail. The first one refers to the integration of the Ukrainian (SRI NASU-NSAU) and Russian (Space Research Institute RAN, IKI RAN) systems at the data level. The second example refers to the development of an InterGrid infrastructure that integrates several regional and national Grid systems: the Ukrainian Academician Grid (with satellite data processing Grid segment, UASpaceGrid) and the Center for Earth Observation and Digital Earth of the Chinese Academy of Sciences (CEODE-CAS) Grid segment.

Different practical issues regarding the integration of the emerging Sensor Web technology with Grids are discussed in the study. We show how the Sensor Web can benefit from using Grids and vice versa. A flood application example is given to demonstrate the benefits of such integration.

A problem of workflow automation and management in Grid environment is reviewed in this chapter, and a practical example of workflow automation of Envisat/Advanced Synthetic Aperture Radar (ASAR) data processing to support flood mapping is given.

14.2 Levels of Integration: Main Problems and Possible Solutions

At present, there is a strong trend for globalization of monitoring systems with a purpose of solving complex problems on global and regional scale. Earth observation (EO) data from space are naturally distributed over many organizations involved in data acquisition, processing, and delivery of dedicated applied services. The GEOSS system is aimed at working with and building upon existing national, regional, and international systems to provide comprehensive, coordinated Earth observations from thousands of instruments worldwide, transforming the data collected into vital information for society. Therefore, a considerable need exists to support integration of existing systems for solving applied domain problems on a global and coordinated basis.

With the regard to satellite monitoring systems, integration can be done at the following levels: data integration level and task management level. The data integration approach aims to provide an infrastructure for sharing data and products. Such an infrastructure allows data integration where different entities provide various kinds of data to enable joint problem solving (Fig. 14.1). The integration at data integration level could be done using common standards for EO data exchange, user interfaces, application programming interfaces (APIs), and data and metadata catalogues.

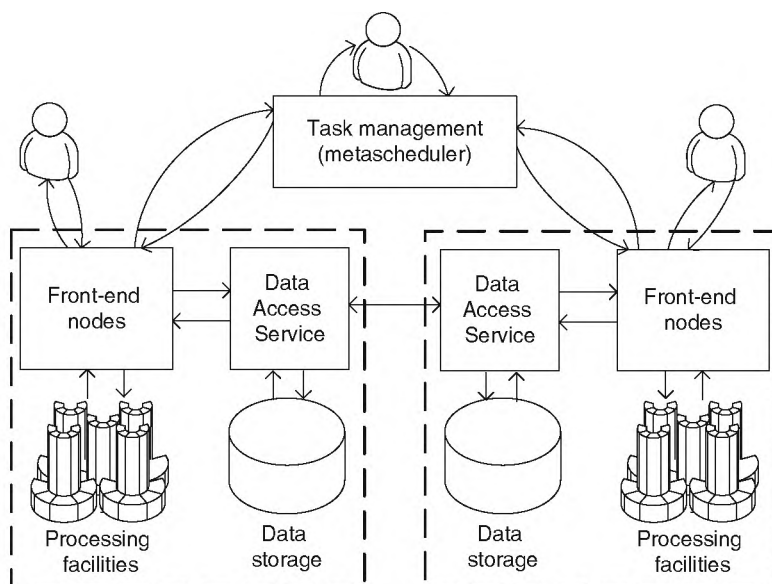


Fig. 14.1 The integration of monitoring systems at data level and at task management level

The task management level approach aims at running applications on distributed computing resources provided by different entities (Fig. 14.1). Since many of the existing satellite monitoring systems heavily rely on the use of Grid technologies, appropriate approaches and technologies should be evaluated and developed to enable the Grid system integration (we define it as *InterGrid*). In such a case, the following problems should be tackled: the use of shared computational infrastructure, development of algorithms for efficient jobs submission and scheduling, load monitoring enabling, and security policy enforcement.

14.2.1 Data Integration Level

At present, the most appropriate standards for data integration are the Open Geospatial Consortium³ (OGC) standards. The following set of standards could be used to address data visualization issues: Web Map Service (WMS), Style Layer Descriptors (SLD), and Web Map Context (WMC). The OGC Web Feature Service (WFS) and Web Coverage Service (WCS) standards provide a uniform way to data delivery. To provide interoperability at the level of catalogues, a Catalogue Service for Web (CSW) standard can be used.

Since the data are usually stored at geographically distributed sites, there are issues regarding optimization of different visualization schemes. In general, there

³<http://www.opengeospatial.org>.

are two possible ways to do visualization of distributed data: a centralized visualization scheme and a distributed visualization scheme. Advantages and shortcomings of each of the schemes and experimental results are discussed in detail in [1].

14.2.2 Task Management Level

In this section, the main issues and possible solutions to Grid systems integration are given. The main prerequisite of such integration is enabling certificates trust. This can be done, for example, through the EGEE infrastructure that at present brings together the resources of different organizations from more than 50 countries. Other problems that should be addressed within the integration are as follows: data transfer, high-level access to geospatial data, development of common catalogues, enabling jobs submission and monitoring, and information exchange.

14.2.2.1 Security Issues

To enable security trust between different parties in the Grid system, a Public Key Infrastructure (PKI) is traditionally applied. X.509 is the most widely used format which is supported by most of the existing software.

To get access to resources of the Grid system, a user should make a request to a Certificate Authority (CA) which is always a known third party. The CA validates the information about the user and then signs the user certificate by the CA's private key. The certificate can thus be used to authenticate the user to grant access to the system. To provide a single sign on and delegation capabilities, the user can use the certificate and his private key to create a *proxy certificate*. This certificate is signed not by CA but rather the user himself. The proxy certificate contains information about the user's identity and a special time stamp after which the certificate will no longer be accepted.

To enable Grid system integration with different middleware installed and security mechanisms and policies used, the following solutions were tested:

1. To create our own CAs and to enable the trust between them
2. To obtain certificates from a well-known CA, for example, the European Policy Management Authority for Grid Authentication⁴ (EUGridPMA)
3. To use a combined approach in which some of the Grid nodes accept only certificates from the local CA and others accept certificates from a well-known third party CAs.

Within the integration of the UASpaceGrid and the CEODE-CAS Grid, the second and the third approaches were verified. In such a case, the UASpaceGrid accepted

⁴<http://www.eugridpma.org>.

the certificates issued by the local CA that was established using the TinyCA, and certificates issued by the UGRID CA.⁵

It is worth mentioning that Globus Toolkit v.4⁶ and gLite v.3⁷ middleware implement the same standard for the certificates, but different standards for describing the certificate policies. That is why it is necessary to use two different standards for describing the CA's identity in a policy description file.

14.2.2.2 Enabling Data Transfer Between Grid Platforms

GridFTP is recognized as a standard protocol for transferring data between Grid resources [2]. GridFTP is an extension of the standard File Transfer Protocol (FTP) with the following additional capabilities:

- Integration with the Grid Security Infrastructure (GSI) [3] enabling the support of various security mechanisms
- Improved performance of data transfer using parallel streams to minimize bottlenecks
- Multicasting by doing one-source-to-many-destinations transfers

The Globus Toolkit 4 also provides the OGSI-compliant Reliable Transfer Service (RFT) to enable reliable transfer of data between the GridFTP servers. In this context, reliability means that problems arisen during the transfer are managed automatically to some extent defined by the user.

Some difficulties in using GridFTP exist in networks with a complex architecture. The bunch of these problems originates from the use of the Network Address Translation (NAT) mechanism. To overcome these problems, the appropriate configurations to the network routers and GridFTP servers should be made.

The gLite 3 middleware provides two GridFTP servers with different authorization mechanisms:

1. The GridFTP server with the Virtual Organization Membership System (VOMS) [4] authorization
2. GridFTP server with the Grid Mapfile authorization mechanism

These two servers can work simultaneously under the condition they will use different TCP ports. To transfer files between gLite and GT platforms, both versions of GridFTP servers can be applied. But the server with the VOMS authorization requires all clients to be authorized using the VOMS server. In such a case, this may pose some limitations. In contrast, the GridFTP server with the Grid Mapfile authorization mechanism does not pose such a limitation, and thus can be used with any other authorization system.

⁵<https://ca.ugrid.org>.

⁶<http://www.globus.org/toolkit/>.

⁷<http://glite.web.cern.ch/>.

To test file transfers between different platforms used at the UASpaceGrid and the CEODE-CAS Grid, the GridFTP version with the Grid Mapfile authorization was used. File transfers were successfully completed in both directions between two Grids with configured client and server roles.

14.2.2.3 Enabling Access to Geospatial Data

In a Grid system that is used for satellite data processing, corresponding services should be developed to enable access to geospatial data. In such a case, the data may be of different nature, and different formats may be used for storing them.

Two solutions can be used to enable a high-level access to geospatial data in Grids: the Web Services Resource Framework (WSRF) services or the Open Grid Services Architecture–Database Access and Integration⁸ (OGSA–DAI) container. Each of these two approaches has its own advantages and shortcomings. A basic functionality for the WSRF-based services can be easily implemented, packed, and deployed using proper software tools, but enabling advanced functionality such as security delegation, third-party transfers, and indexing becomes much more complicated. The difficulties also arise if the WSRF-based services are to be integrated with other data-oriented software. A basic architecture for enabling access to geospatial data in Grids via the WSRF-based services is shown in Fig. 14.2.

The OGSA–DAI framework provides uniform interfaces to heterogeneous data. This framework allows the creation of high-level interfaces to data abstraction layer hiding the details of data formats and representation schemas. Most of the problems such as delegation, reliable file transfer, and data flow between different sources are handled automatically in the OGSA–DAI framework. The OGSA–DAI containers are easily extendable and embeddable. But comparing to the WSRF basic functionality, the implementation of an OGSA–DAI extension is much more complicated. Moreover, the OGSA–DAI framework requires a preliminary deployment of additional software components. A basic architecture for enabling access to geospatial data in Grids via the OGSA–DAI container is shown in Fig. 14.2.

14.2.2.4 Job Submission and Monitoring

Different approaches were evaluated to enable job submission and monitoring in the InterGrid composed of Grid systems that use different middleware. In particular:

1. To use a Grid portal that supports job submission mechanism for different middleware (Fig. 14.3). The GridSphere and P-GRADE are among possible solutions.

⁸<http://www.ogsadai.org.uk>.

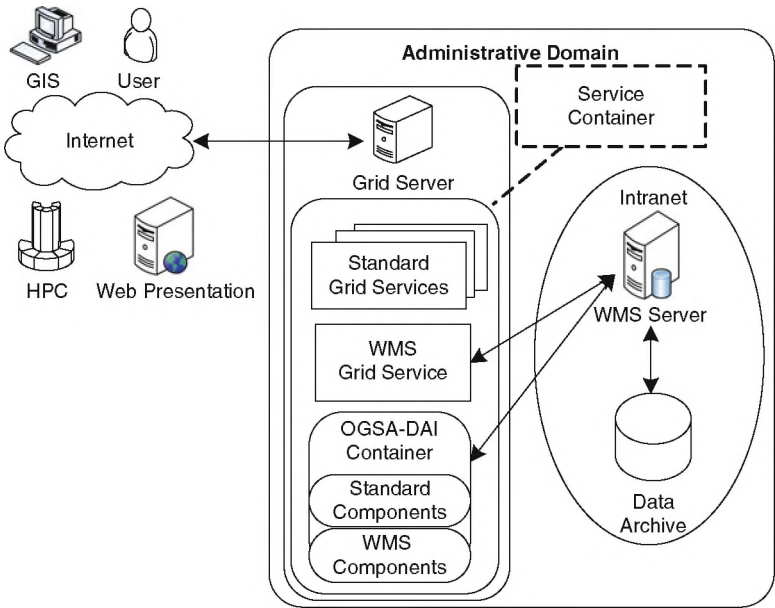


Fig. 14.2 High-level access to geospatial data via the WSRF-based services and the OGSA-DAI container

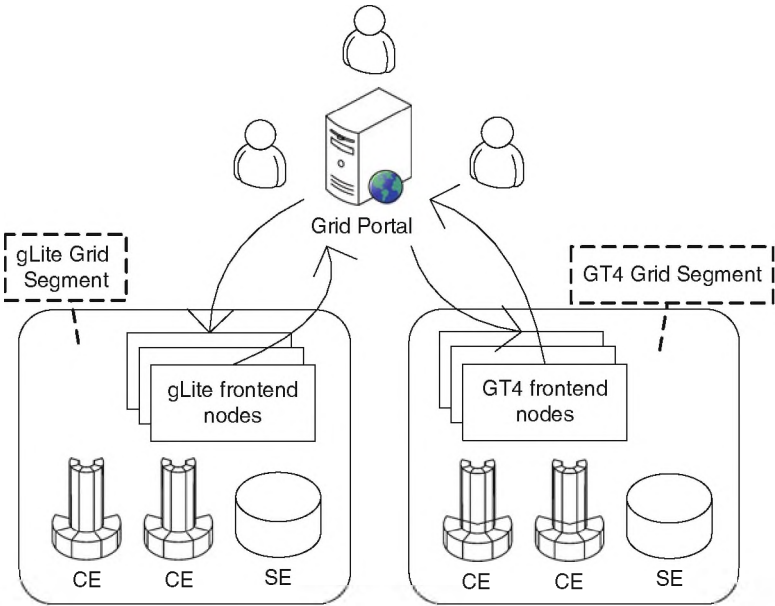


Fig. 14.3 Portal approach to Grid system integration

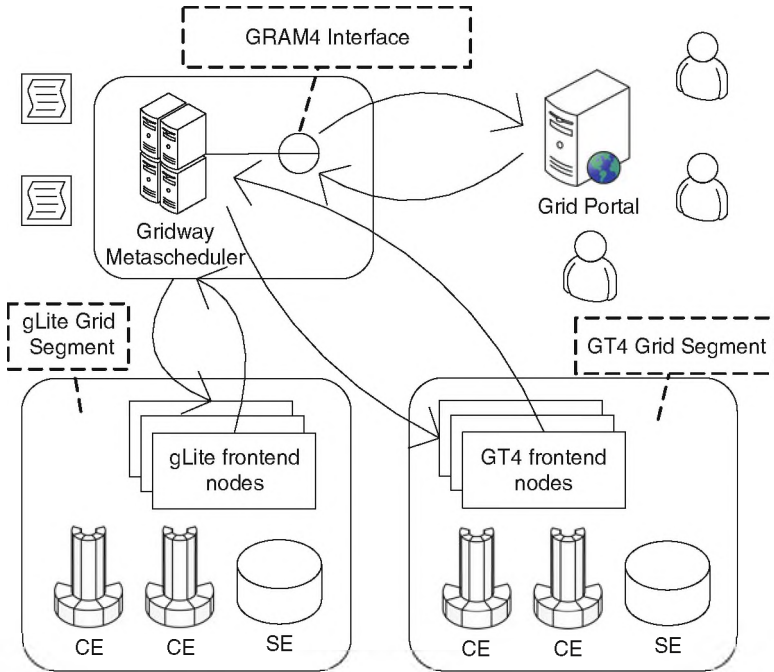


Fig. 14.4 Metascheduler approach

2. To develop a high-level Grid scheduler – *metascheduler* – that will support different middleware by providing standard interfaces (Fig. 14.4).

The Grid portal is an integrated platform to end users that enables access to Grid services and resources via a standard Web browser. The Grid portal solution is easy to deploy and maintain, but it does not provide APIs and scheduling capabilities.

On the contrary, a metascheduler interacts with low-level schedulers used in different Grid systems enabling system interoperability. The metascheduler approach is much more difficult to maintain comparing to the portal; however, it provides necessary APIs with advanced scheduling and load-balancing capabilities. At present, the most comprehensive implementation for the metascheduler is a GridWay system. The GridWay metascheduler is compatibility with both Globus and gLite middleware. Beginning from Globus Toolkit v4.0.5, GridWay becomes a standard part of its distribution. The GridWay system provides comprehensive documentation for both users and developers that is an important point for implementing new features.

A combination of these two approaches will provide advanced capabilities for the implementation of interfaces to get access to the resources of the Grid system, while a Grid portal will provide a suitable user interface.

To integrate the resources of the UASpaceGrid and the CEODE-CAS Grid, a GridSphere-based Grid-portal was deployed.⁹ The portal allows the submission and monitoring of jobs on the computing resources of the Grid systems and provides access to the data available at the storage elements of both systems.

14.3 Implementation Issues: Lessons Learned

In this section, two real-world examples of system integration at the data level and task management level are given. The first example describes the integration of the Ukrainian satellite monitoring system operated at the SRI NASU-NSAU and the Russian satellite monitoring system operated at the IKI RAN at the data level. The second example refers to the development of the InterGrid infrastructure that integrates several regional and national Grid systems: the Ukrainian Academician Grid and Chinese CEODE-CAS Grid.

14.3.1 *Integration of Satellite Monitoring Systems at Data Level*

Figure 14.5 shows the overall architecture for integrating the satellite monitoring systems at data level. The satellite data and corresponding products, modeling data, and in situ observations are provided in a distributive way by applying the OGC standards. In particular, the SRI NSAU-NSAU provides OGC/WMS-compliant interfaces to the following data sets:

- Meteorological forecasts derived from the Weather Research and Forecast (WRF) numerical weather prediction (NWP) model [5]
- In situ observations from a network of weather stations in Ukraine
- Earth land parameters such as temperature, vegetation indices, and soil moisture derived from NASA's Moderate resolution Imaging Spectro-radiometer (MODIS) instrument onboard Terra and Aqua satellites

The IKI RAN provides OGC/WMS-compliant interfaces to the following satellite-derived products:

- Land parameters that are primarily used for agriculture applications
- Fire risk and burnt area estimation for disaster monitoring applications

The products provided by the IKI RAN cover both Russia and Ukraine countries. Coupling these products with modeling data and in situ observations provided by the SRI NASU-NSAU allows information of a new quality to be acquired in almost near-real time. Such integration would never be possible without the use of standardize OGC interfaces. The proposed approach is used for the solution of applied problems of agriculture resources monitoring and crop yield prediction.

⁹<http://gridportal.ikd.kiev.ua:8080/gridsphere>.

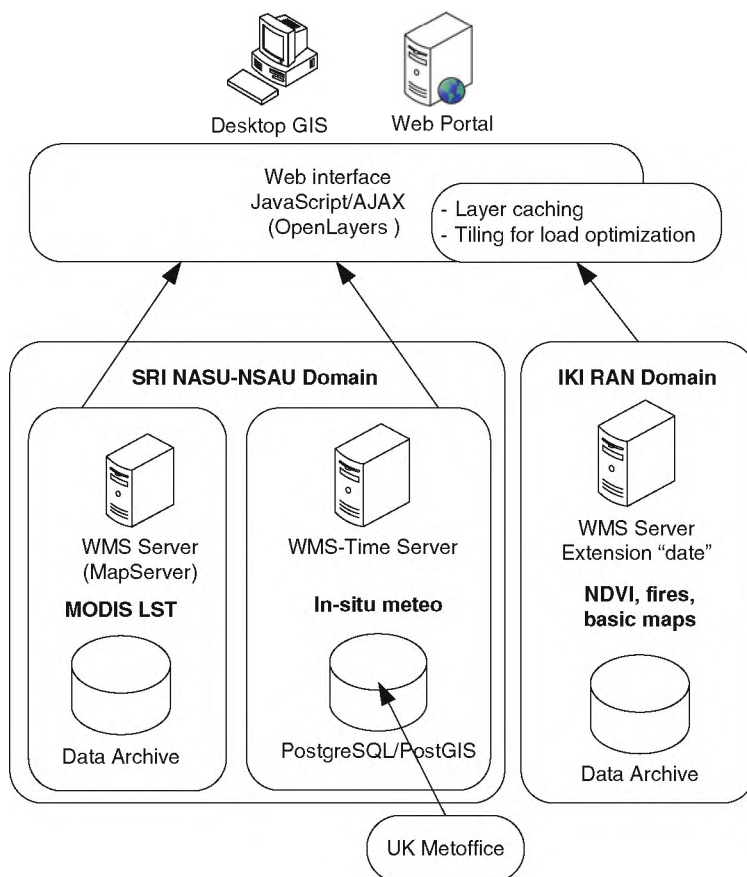


Fig. 14.5 Architecture of satellite monitoring systems integration

To provide a user interface that will enable integration of data coming from multiple sources an open-source OpenLayers¹⁰ framework is used. OpenLayers is a thick client software based on JavaScript/AJAX technology and fully operational on a client side. Main OpenLayers features also include:

- Support of several WMS servers
- Support of different OGC standards (WMS, WFS)
- Caching and tiling support to optimize visualization
- Support of both raster and vector data

The data and satellite-based products provided by the SRI NASU-NSAU and IKI RAN are available at <http://land.ikd.kiev.ua>. Figure 14.6 shows a screenshot of OpenLayers interface in which data from multiple sources are being integrated.

¹⁰<http://www.openlayers.org>.

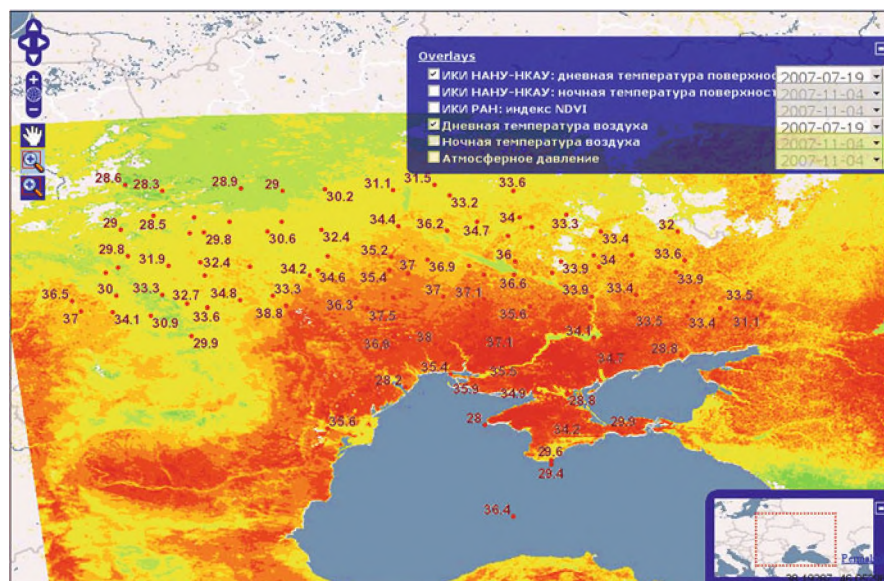


Fig. 14.6 OpenLayers interface to heterogeneous data integration from multiple distributed sources

14.3.2 The InterGrid Testbed Development

The second case study refers to the development of the InterGrid aimed at solving applications of environment and natural disasters monitoring. The InterGrid integrates the Ukrainian Academician Grid with a satellite data processing Grid segment UASpaceGrid and the CEODE-CAS Grid. This InterGrid is considered as a testbed for the Wide Area Grid (WAG) – a project initiated within the CEOS Working Group on Information Systems and Services¹¹ (WGISS).

An important application that is being solved within the InterGrid environment is flood monitoring and prediction. This task requires the adaptation and tuning of existing meteorological, hydrological and hydraulic models for corresponding territories [5], and the use of heterogeneous data stored at multiple sites. The following data sets are used within the flood application:

- NWP modelling data provided within the UASpaceGrid
- *Satellite data*: synthetic aperture radar (SAR) imagery acquired by ESA's Envisat/ASAR and ERS-2/SAR satellites, optical imagery acquired by Terra, Aqua and EO-1 satellites
- Products derived from optical and microwave satellite data such as surface temperature, vegetation indices, soil moisture, and precipitation

¹¹<http://www.ceos.org/wgiss>.

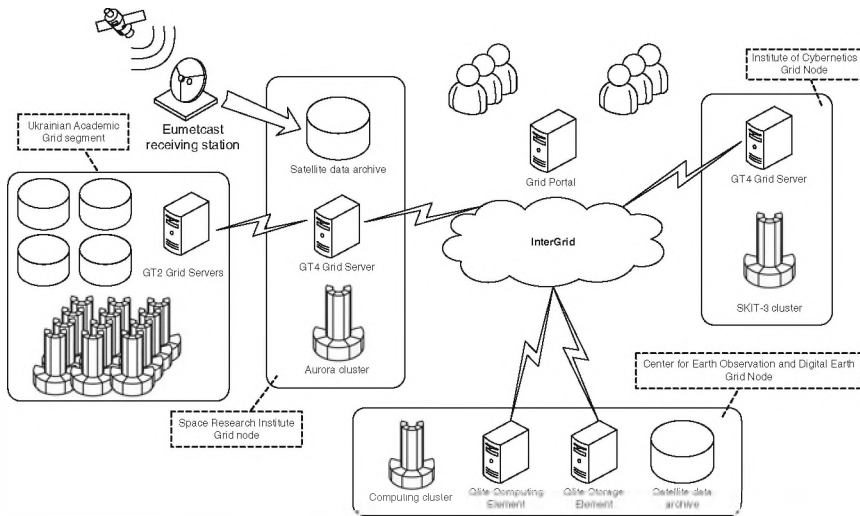


Fig. 14.7 Architecture of InterGrid

- In situ observations from weather stations
- Topographical data such as digital elevation model (DEM)

The process of model adaptation can be viewed as a complex workflow and requires the solution of optimization problems (so-called parametric study) [5]. The processing of satellite data and generation of corresponding products is also a complex workflow and requires intensive computations [6, 7]. All these factors lead to the need of using computing and storage resources of different organizations and their integration into a common InterGrid infrastructure. Figure 14.7 shows the architecture of the proposed InterGrid.

Currently, the InterGrid infrastructure integrates the resources of several geographically distributed organisations, in particular:

- SRI NASU-NSAU (Ukraine) with deployed computing and storage nodes based on the Globus Toolkit 4 and gLite 3 middleware, access to geospatial data and a Grid portal
- Institute of Cybernetics of NASU (IC NASU, Ukraine) with deployed computing and storage nodes based on Globus Toolkit 4 middleware and access to computing resources (SCIT-1/2/3 clusters,¹² more than 650 processors)
- CEODE-CAS (China) with deployed computing nodes based on gLite 3 middleware and access to geospatial data (approximately 16 processors)

In all cases, the Grid Resource Allocation and Management (GRAM) service [8] is used to execute jobs on the Grid resources.

It is also worth mentioning that satellite data are distributed over the Grid environment. For example, the Envisat/ASAR data (that are used within the flood

¹²<http://icybcluster.org.ua>.

application) are stored on the ESA's rolling archive and routinely downloaded for the Ukrainian territory. Then, they are stored at the SRI NASU-NSAU archive that is accessible via the Grid. MODIS data from Terra and Aqua satellites that are used in flood and agriculture applications are routinely downloaded from the USGS archives and stored at the SRI NASU-NSAU and IC NASU.

The GridFTP protocol was chosen to provide data transfer between the Grid systems. Access to the resources of the InterGrid is organized via a high-level Grid portal that has been deployed using a GridSphere framework.¹³ Through the portal, a user can access the required data and submit jobs to the computing resources of the InterGrid. The portal also provides facilities to monitor the resources state such as CPU load and memory usage. The workflow of the data processing steps in the InterGrid is managed by a Karajan engine.¹⁴

14.4 Integration of Grid and Sensor Web

Decision makers in an emergency response situation (e.g., floods, droughts) need rapid access to the existing data, the ability to request and process data specific to the emergency, and tools to rapidly integrate the various information services into a basis for decisions. The flood prediction and monitoring scenario presented here is being implemented within the GEOSS Architecture Implementation Pilot¹⁵ (AIP). It uses precipitation data from the Global Forecasting System (GFS) model and NASA's Tropical Rainfall Measuring Mission¹⁶ (TRMM) to identify the potential flood areas. Once the areas have been identified, we can request satellite imagery for the specific territory for flood assessment. These data can be both optical (like EO-1, MODIS, SPOT) and microwave (Envisat, ERS-2, ALOS, RADARSAT-1/2).

This scenario is implemented using the Sensor Web [9, 10] and Grid [6, 7, 11, 12] technologies. The integration of sensor networks with Grid computing brings out dual benefits [13]:

- Sensor networks can off-load heavy processing activities to the Grid.
- Grid-based sensor applications can provide advance services for smart-sensing by deploying scenario-specific operators at runtime.

14.4.1 Sensor Web Paradigm

Sensor Web is an emerging paradigm and technology stack for integration of heterogeneous sensors into a common informational infrastructure. The basic

¹³<http://www.gridsphere.org>.

¹⁴<http://www.gridworkflow.org/snips/gridworkflow/space/Karajan>.

¹⁵<http://www.ogcnetwork.net/AIpilot>.

¹⁶<http://tmm.gsfc.nasa.gov>.

functionality required from such infrastructure is remote data access with filtering capabilities, sensors discovery, and triggering of events by sensors conditions.

Sensor Web is governed by the set of standards developed by OGC [14]. At present, the following standards are available and approved by consortium:

- OGC Observations and Measurements¹⁷ – common terms and definition for Sensor Web domain
- Sensor Model Language¹⁸ – XML-based language for describing different kinds of sensors
- Transducer Model Language¹⁹ – XML-based language for describing the response characteristics of a transducer
- Sensor Observations Service²⁰ (SOS) – an interface for providing remote access to sensors data
- Sensor Planning Service²¹ (SPS) – an interface for submitting tasks to sensors

There are also standards drafts that are available from the Sensor Web working group but not yet approved as official OpenGIS standards:

- Sensor Alert Service – service for triggering different kinds of events basing of sensors data
- Web Notification Services – notification framework for sensor events

The Sensor Web paradigm assumes that sensors could belong to different organizations with different access policies or, in a broader sense, to different administrative domains. However, existing standards does not provide any means for enforcing data access policies leaving it to underlying technologies. One possible way for handling informational security issues in Sensor Web is presented in the next sections.

14.4.2 *Sensor Web Flood Use Case*

One of the most challenging problems for Sensor Web technology implementation is global ecological monitoring in the framework of GEOSS. In this section, we consider the problem of flood monitoring using satellite remote sensing data, in situ data, and results of simulations.

Flood monitoring requires the integrated analysis of data from multiple heterogeneous sources such as remote sensing satellites and in situ observations. Flood prediction is adding the complexity of physical simulation to the task. Figure 14.8 shows the Sensor Web architecture for this case-study. It presents the integrated use

¹⁷<http://www.opengeospatial.org/standards/om>.

¹⁸<http://www.opengeospatial.org/standards/sensorml>.

¹⁹<http://www.opengeospatial.org/standards/tml>.

²⁰<http://www.opengeospatial.org/standards/sos>.

²¹<http://www.opengeospatial.org/standards/sps>.

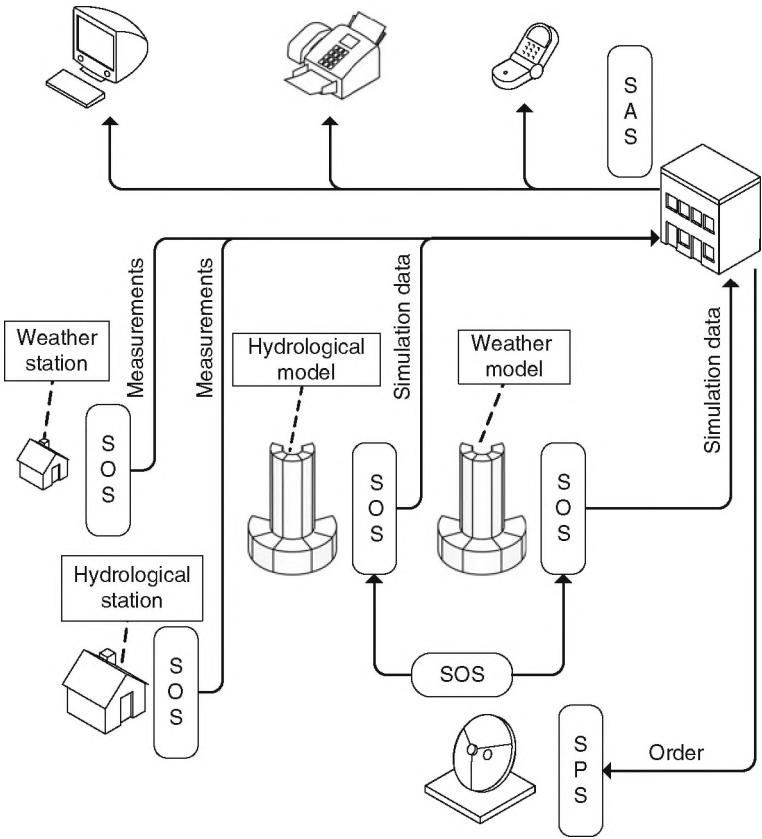


Fig. 14.8 The sensor web architecture of the flooding test case

of different OpenGIS[®] specifications for the Sensor Web. The data from multiple sources (numerical models, remote sensing, in situ observations) are accessed through the Sensor Observation Service (SOS). An aggregator site is running the Sensor Alert Service to notify interested organization about potential flood event using different communication means. The aggregator site is also sending orders to satellite receiving facilities using the SPS service to acquire new satellite imagery.

14.4.3 Sensor Web SOS Gridification

The Sensor Web services such as SOS, SPS, and SAS can benefit from the integration with the Grid platform like Globus Toolkit. Many Sensor Web features can take advantage of the Grid services, namely:

- Sensor discovery could be performed through the combination of the Index Service and Trigger Service.

- High-level access to XML description of the sensors and services could be made through queries to the Index Service.
- Grid platform provides a convenient way for the implementation of notifications and event triggering using corresponding platform components [15].
- The RFT service [2] provides reliable data transfer for large volumes of data.
- The GSI infrastructure provides enforcement of data and services access policies in a very flexible way allowing implementation of desired security policy.

To exploit these benefits, an SOS testbed service using Globus Toolkit as a platform has been developed. Currently, this service works as a proxy translating and redirecting user requests to the standard HTTP SOS server. The current version uses client-side libraries for interacting with the SOS server provided by the 52North in their OX-Framework. The next version will also include in-service implementation of the SOS server functionality.

The Grid service implementing SOS provides an interface specified in the SOS reference document. The key difference between the standard interfaces and the Grid-based implementations of the SOS lies in the encoding of service requests. The standard implementation uses custom serialization for the requests and responses, and the Grid-based implementation uses the Simple Object Access Protocol (SOAP) encoding.

To get advantage of Globus features, the SOS service should export service capabilities and sensor descriptions as WSRF resource property [16]. Traditionally, the implementation of such a property requires the translation between XML Schema and Java code. However, the XML Schema of the SOS service and related standards, in particular GML [15], is a very complex one, and there are no available program tools able to generate Java classes from it. This problem was solved by storing service capabilities and sensor description data as the Document Object Model (DOM) element object and using a custom serialization for this class provided by the Axis framework that is used by the Globus Toolkit. Within this approach, particular elements of the XML document cannot be accessed in an object-oriented style. However, the SOS Grid service is acting as a proxy between the user and the SOS implementation, so it does not have to modify the XML document directly. With resource properties defined in this way, they can be accessed by using a standard Globus Toolkit API or command line utilities.

14.5 Grid Workflow Management for Satellite Data Processing Within UN-SPIDER Program

One of the most important problems associated with satellite data processing for disaster management is a timely delivery of information to end users. To enable such capabilities, an appropriate infrastructure is required to allow for rapid and efficient access to processing and delivery of geospatial information that is further used for damage assessment and risk management. In this section, the use of Grid

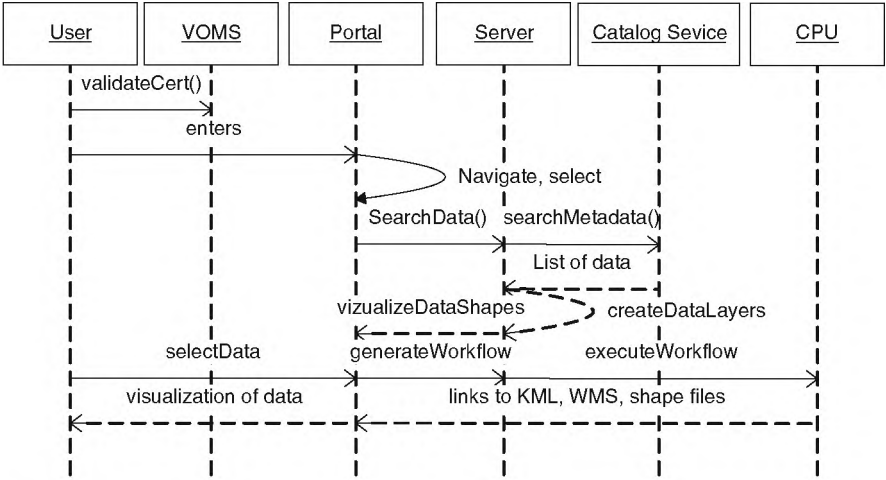


Fig. 14.9 UML sequence diagram

technologies for automated acquisition, processing and visualization of satellite SAR, and optical data for rapid flood mapping is presented. The developed services are used within the United Nations Platform for Space-based Information for Disaster Management and Emergency Response²² (UN-SPIDER) Regional Support Office (RSO) in Ukraine that was established in February 2010.

14.5.1 Overall Architecture

Within the infrastructure, an automated workflow of satellite SAR data acquisition, processing and visualization, and corresponding geospatial services for flood mapping from satellite SAR imagery were developed. The data are automatically downloaded from the ESA rolling archives where satellite images are available within 2–4 h after their acquisition. Both programming and graphical interfaces were developed to enable search, discovery, and acquisition of data. Through the portal, a user can perform a search for the SAR image file based on geographical region and a time range. A list of available SAR imagery is returned and the user can select a file to generate a flood map. The file is transferred to the resources of the Grid system at the SRI NASU-NSAU, and a workflow is automatically generated and executed on the resources of the Grid infrastructure. The corresponding UML sequence diagram is shown in Fig. 14.9.

To enable execution of the workflow in the Grid system, a set of services has been implemented (Fig. 14.10). We followed the approach used in the Earth System Grid [17]. The four major components of the system are as follows:

²²<http://www.un-spider.org>.

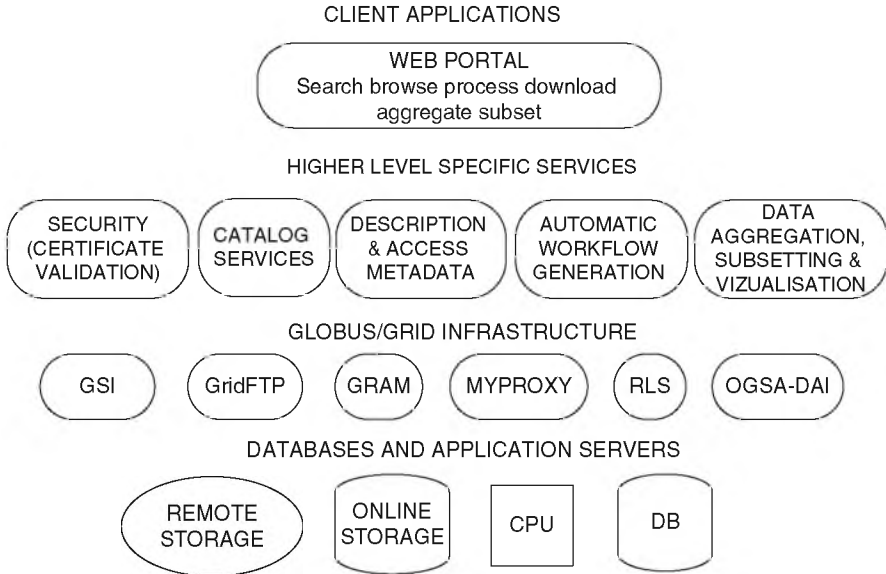


Fig. 14.10 System architecture

1. *Client applications.* Web portal is a main entry point, and provides interfaces to communicate with system services.
2. *High-level services.* This level includes security subsystem, catalogue services, metadata services (description and access), automatic workflow generation services, and data aggregation, subsetting and visualization services. These services are connected to the Grid services at the lower level.
3. *Grid services.* These services provide access to the shared resources of the Grid system, access to credentials, file transfer, job submission, and management.
4. *Database and application services.* This level provides physical data and computational resources of the system.

14.5.2 Workflow of Flood Extent Extraction from Satellite SAR Imagery

A neural network approach to SAR image segmentation and classification was developed [6]. The workflow of data processing is as follows (Fig. 14.11):

1. *Data calibration.* Transformation of pixel values (in digital numbers) to backscatter coefficient (in dB).
2. *Orthorectification and geocoding.* This step is intended for a geometrical and radiometric correction associated with the SAR imaging technology, and to provide a precise georeferencing of data.

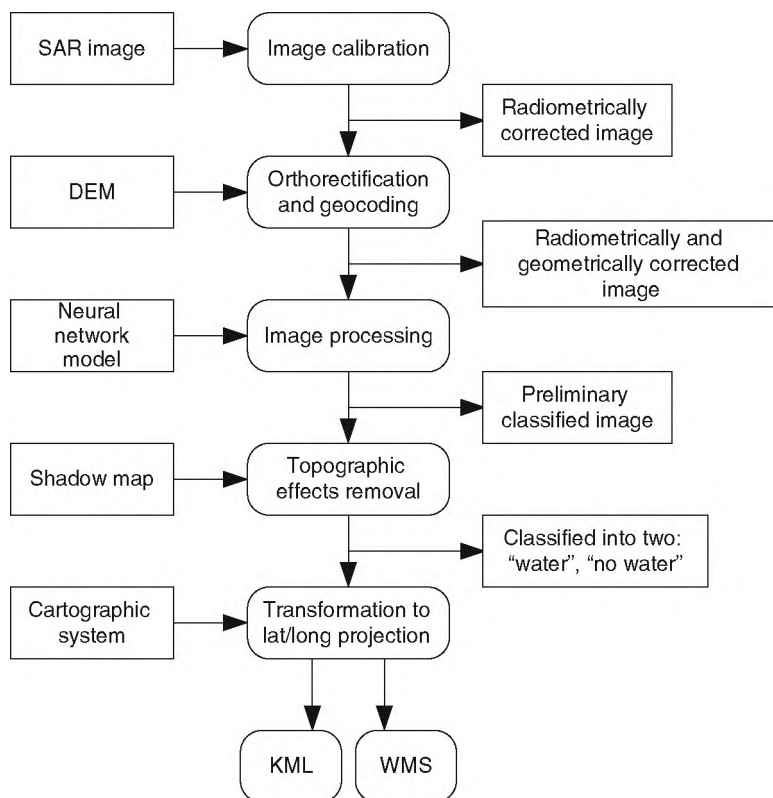


Fig. 14.11 Workflow of flood extent extraction from SAR satellite imagery

3. *Image processing*. Segmentation and classification of the image using a neural network.
4. *Topographic effects removal*. Using digital elevation model (DEM), such effects as shadows are removed from the image. The output of this step is a binary image classified into two classes: “Water” and “No water.”
5. *Transformation to geographic projection*. The image is transformed to the projection for further visualization via Internet using the OGC-compliant standards (KML or WMS) or desktop Geographic Information Systems (GIS) using shape file.

14.5.3 China–Ukrainian Service-Oriented System for Disaster Management

To benefit from data of different nature (e.g., optical and radar) and provide integration of different products in case of emergency, our flood mapping service was integrated with the flood mapping services provided by the CEODE-CAS.

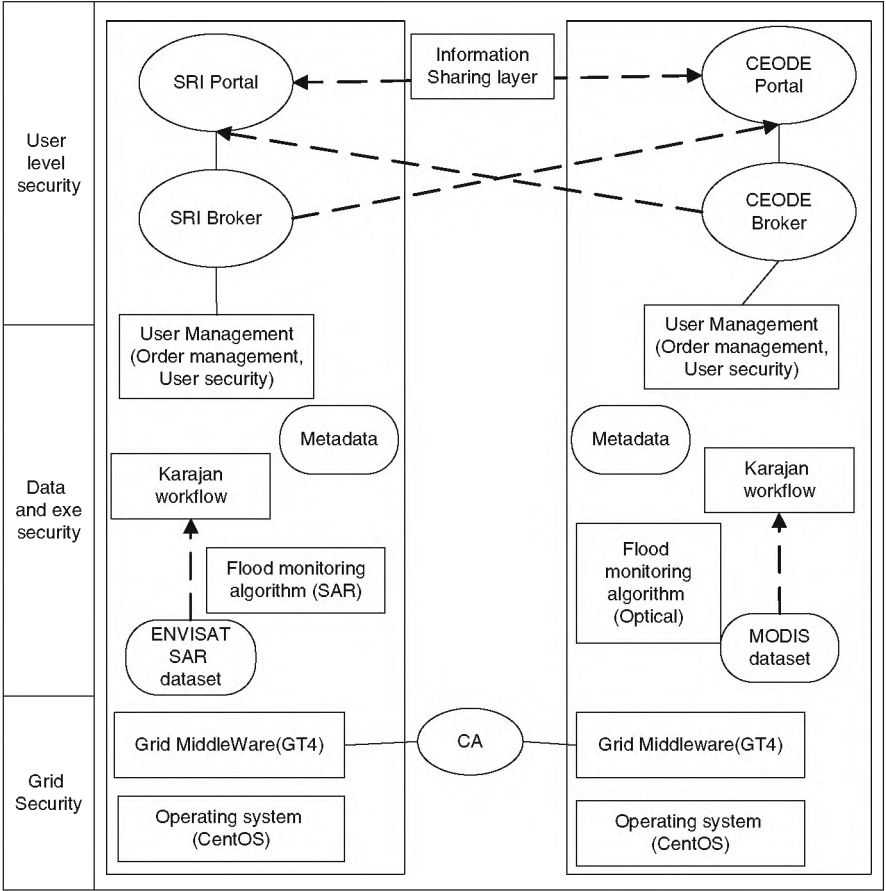


Fig. 14.12 Architecture of China–Ukrainian service-oriented system for disaster management

This service is based on the use of optical data acquired by MODIS instrument onboard Terra and Aqua satellites. Figure 14.12 shows the architecture of the China–Ukrainian service-oriented system for disaster management.

The integration of the Ukrainian and Chinese systems is done at the level of services. The portals of SRI NASU-NSAU and CEODE are operated independently and communicate with corresponding brokers that provide interfaces to the flood mapping services. These brokers process requests from both local and trusted remote sites. For example, to provide a flood mapping product using SAR data, the CEODE portal generates a corresponding search request to the broker at the SRI NASU-NSAU side based on user search parameters. This request is processed by the broker and the search results are displayed at the CEODE portal. The user selects the SAR image file to be processed, and the request is submitted to the SRI NASU-NSAU broker which generates and executes workflow, and delivers the flood maps to the CEODE portal. The same applies to the broker operated at the CEODE

side that provides flood mapping services using optical satellite data. To get access to the portal, the user should have a valid certificate. The SRI NASU-NSAU runs the VOMS server to manage with this issue.

14.6 Experimental Results

14.6.1 Numerical Weather Modeling in Grid

The forecasts of meteorological parameters derived from numerical weather modeling are vital for a number of applications including floods, droughts, and agriculture. Currently, we run the WRF model in operational mode for the territory of Ukraine. The meteorological forecasts for the next 72 h are generated every 6 h with a spatial resolution of 10 km. The size of horizontal grid is 200×200 points with 31 vertical levels. The forecasts derived from the National Centers for Environmental Prediction Global Forecasting System (NCEP GFS) are used as boundary conditions for the WRF model. These data are acquired via Internet through the National Operational Model Archive and Distribution System (NOMADS).

The WRF model workflow to produce forecasts is composed of the following steps [18]: data acquisition, data preprocessing, generation of forecasts using WRF application, data postprocessing, and visualization of the results through a portal.

Experiments were run to evaluate the performance of the WRF model with respect to the number of computing nodes of the Grid system resources. For this purpose, we used a parallel version of the WRF model (version 2.2) with a model domain identical to those used in operational NWP service ($200 \times 200 \times 31$ grid points with a horizontal spatial resolution of 10 km). The model parallelization was implemented using the Message Passing Interface (MPI). We observed almost a linear productivity growth against the increasing number of computation nodes. For example, the use of eight nodes of the SCIT-3 cluster of the Grid infrastructure gave the performance increase 7.09 times (of 8.0 theoretically possible) comparing to a single node. The use of 64 nodes of the SCIT-3 cluster increased the performance 43.6 times. Since a single iteration of the WRF model run corresponds to the forecast of meteorological parameters for the next 1 min, the completion of 4,320 iterations is required for a 3 day forecast. That is, it takes approximately 5.16 h to generate a 3-day forecast on a single node of the SCIT-3 cluster of the Grid infrastructure. In turn, the use of 64 nodes of the SCIT-3 cluster allowed us to reduce the overall computing time to approximately 7.1 min.

14.6.2 Implementation of SOS Service for Meteorological Observations: Database Issues

To provide access to meteorological data, we implemented the Sensor Web SOS service (see Sect. 14.4.1). As a case study, we implemented an SOS service for

retrieving surface temperature measured at the weather stations distributed over Ukraine.

The SOS service output is an XML document in a special scheme specified by the SOS reference document. The standard describes two possible ways for retrieving results, namely “Measurement” and “Observation.” The first form is more suitable to situations when the service returns a small amount of heterogeneous data. The second form is more suitable for a long time-series of homogeneous data. Table 14.1 gives an example of the SOS service output for both cases.

The 52North software was used for the implementation of the SOS service. Since the 52North has a complex relational database scheme, we had to adapt the existing database structure using a number of SQL views and synthetic tables. From 2005 through 2008 there were nearly two million records with observations derived at the weather stations. The PostgreSQL database with the PostGIS spatial extension was used to store the data. Most of the data records were contained within a single table “observations” with indices built over fields with observation time and a station identifier. The tables of such a volume require a special handling; so the index for a time field was clusterized thus reordering data on the disks and reducing the need for I/O operations. The clusterization of the time index reduced a typical query time from 8,000 to 250 ms.

To adapt this database to the requirements of the 52North server, a number of auxiliary tables with reference values related to the SOS service such as phenomena names, sensor names, and region parameters, and a set of views that transform the underlying database structure into the 52North scheme were created. The most important view that binds all the values of synthetic tables together with observation data has the following definition:

```
SELECT observations.`time` AS time_stamp, `procedure`.
procedure_id, feature_of_interest. feature_of_interest_id,
phenomenon.phenomenon_id, offering.offering_id,
`` AS text_value, observations.t AS numeric_value,``
AS mime_type, observations.oid AS observation_id
FROM observations, `procedure`, proc_foi, feature_of_interest,
proc_off, offering_strings offering, foi_off, phenomenon,
proc_phen, phen_off
WHERE `procedure`.procedure_id::text = proc_foi.
procedure_id::text AND proc_foi.feature_of_interest_id::text =
feature_of_interest.feature_of_interest_id AND `procedure`.
procedure_id::text = proc_off.procedure_id::text AND proc_off.
offering_id::text = offering.offering_id::text AND foi_off.
offering_id::text = offering.offering_id::text AND foi_off.
feature_of_interest_id::text = feature_of_interest.
feature_of_interest_id AND proc_phen.procedure_id::text =
`procedure`.procedure_id::text AND proc_phen.phenomenon_id::
text = phenomenon.phenomenon_id::text AND phen_off.
phenomenon_id::text = phenomenon.phenomenon_id::text AND
phen_off.offering_id::text = offering.offering_id::text AND
observations.wmoid::text = feature_of_interest.
feature_of_interest_id;
```

Table 14.1 The two different forms of the SOS service output

Measurement	Observation
<code><om:Measurement gml:id="o255136"></code>	<code><om:result></code>
<code><om:samplingTime></code>	<code>2005-03-14T21:00:00+03,33506,-5@@</code>
<code><TimeInstant xsi:type='gml:TimeInstantType'></code>	<code>2005-03-15T00:00:00+03,33506,-5.2@@</code>
<code><timePosition></code>	<code>2005-03-15T03:00:00+03,33506,-5.5@@</code>
<code>2005-04-14T04:00:00+04</code>	<code>2005-03-15T06:00:00+03,33506,-4.6@@</code>
<code></timePosition></code>	<code>2005-03-15T09:00:00+03,33506,-2.2@@</code>
<code></TimeInstant></code>	<code>2005-03-15T12:00:00+03,33506,1.7@@</code>
<code></om:samplingTime></code>	<code>2005-03-15T15:00:00+03,33506,1.7@@</code>
<code><om:procedure xlink:href=</code>	<code>2005-03-15T18:00:00+03,33506,2.4@@</code>
<code> 'urn:ogc:object:feature:Sensor:WMO:33506'/'></code>	<code>2005-03-15T21:00:00+03,33506,-0.7@@</code>
<code><om:observedProperty xlink:href=</code>	<code>2005-03-16T00:00:00+03,33506,-1.4@@</code>
<code> 'urn:ogc:def:phenomenon:OGC:temperature'/'></code>	<code>2005-03-16T03:00:00+03,33506,-1.1@@</code>
<code><om:featureOfInterest></code>	<code>2005-03-16T06:00:00+03,33506,-1.1@@</code>
<code><sa:Station gml:id='33506'></code>	<code>2005-03-16T09:00:00+03,33506,-1.3@@</code>
<code><name>WMO33506</name></code>	<code>2005-03-16T12:00:00+03,33506,0.5@@</code>
<code><sa:sampledFeature xlink:href='''/'></code>	<code>2005-03-16T15:00:00+03,33506,1.7@@</code>
<code><sa:position></code>	<code>2005-03-16T18:00:00+03,33506,1.5@@</code>
<code><Point></code>	<code></om:result></code>
<code><pos srsName='urn:crs:epsg:4326'></code>	
<code>34.55 49.6</code>	
<code></pos></code>	
<code></Point></code>	
<code></sa:position></code>	
<code></sa:Station></code>	
<code></om:featureOfInterest></code>	
<code><om:result uom='celsius'>10.9</om:result></code>	
<code></om:Measurement></code>	

The 52North's database scheme uses a string as a primary key for auxiliary tables instead of a synthetic numerical one, and is far from being optimal in the sense of performance. It might cause problems in a large-scale SOS-enabled data warehouse. A typical SQL query from the 52North service is quite complex. Here is an example:

```
SELECT observation.time_stamp, observation.text_value,
observation.observation_id, observation.numeric_value,
observation.mime_type, observation.offering_id, phenomenon.
phenomenon_id, phenomenon.phenomenon_description,
phenomenon.unit,phenomenon.valuetype,observation.procedure_id,
feature_of_interest.feature_of_interest_name, feature_of_interest.
feature_of_interest_id, feature_of_interest.feature_type,
SRID(feature_of_interest.geom), AsText(feature_of_interest.geom)
AS geom FROM phenomenon NATURAL INNER JOIN observation NATURAL
INNER JOIN feature_of_interest WHERE (feature_of_interest.
feature_of_interest_id = '33506') AND (observation.phenomenon_id
='urn:ogc:def:phenomenon:OGC:1.0.30:temperature') AND
(observation.procedure_id = 'urn:ogc:object:feature:Sensor:
WMO:33506') AND (observation.time_stamp >= '2006-01-01 02:00:00
+0300'AND observation.time_stamp <= '2006-02-26 01:00:00+0300')
```

An average response time for such a query (assuming a 1-month time period) is about 250 ms with the PostgreSQL server running in a virtual environment on a 4 CPUs server with 8GB of RAM and 5 SCSI 10k rpm disks in RAID5 array. The increase in a query depth results in a linear increase of response time with an estimate of 50 ms per month (see Fig. 14.13).

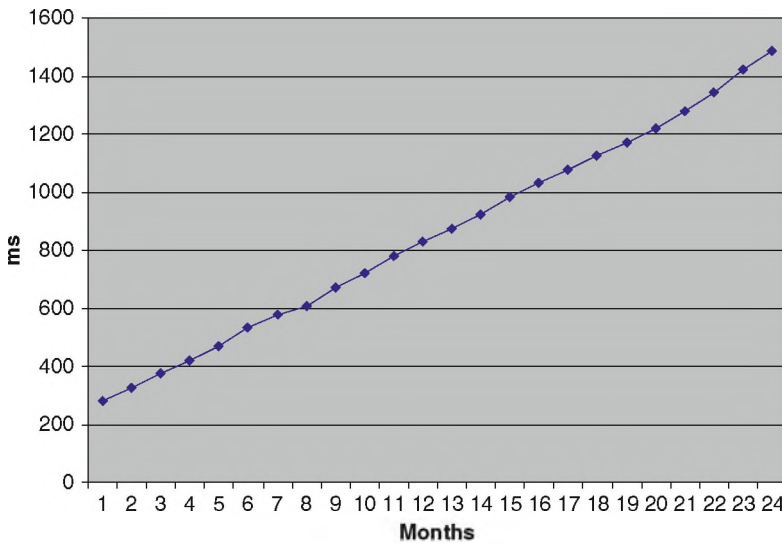


Fig. 14.13 The dependence of query response time against the depth of query

14.6.3 Rapid Flood Mapping from Satellite Imagery in Grid

Within the developed Grid infrastructure, a set of services for rapid flood mapping from satellite imagery is delivered. The use of the Sensor Web services enables automated planning and tasking of satellite (where available) and data delivery, while the Grid services are used for workflow orchestration, data processing, and geospatial services delivery to end users through the portal.

To benefit from the use of the Grid, a parallel version of the method for flood mapping from satellite SAR imagery has been developed (see Sect. 14.5.2). The parallelization of the image processing was implemented in the following way: an SAR image is split into the uniform parts that are processed on different nodes using the OpenMP Application Program Interface (www.openmp.org). The use of the Grids allowed us to considerably reduce the time required for image processing and service delivery. In particular, it took approximately more than 1.5 h (depending on image size) to execute the whole workflow on a single workstation. The use of Grid computing resources allowed us to reduce the computational time to less than 20 min.

Another case study refers to the use of the Sensor Web for tasking the EO-1 satellite through the SPS service [19]. Through the UN-SPIDER RSO in Ukraine, a request was made from local authorities to acquire satellite images over the Kyiv city area due to a high risk of a flood in spring 2010. The use of the Sensor Web and the Grid ensured a timely delivery of products to end users. In particular, Table 14.2 gives a sequence of events starting from the notification of satellite tasking and ending with generation of final products.

It took less than 12 h after image acquisition to generate geospatial products that were delivered to the Ukrainian Ministry of Emergency Situations, the Council of National Security and Defence, and the Ukrainian Hydrometeorological Centre. The information on river extent that was derived from the EO-1 image was used to calibrate and validate hydrological models to produce various scenarios of water extent for flood risk assessment.

Table 14.2 The timeline of tasking the EO-1 satellite and generating the final geospatial product during the potential flood in Ukraine in spring 2010

Date and time ^a	Event
Mon Apr 12 2010 @ 10:33 PM	Notification on EO-1 tasking through SPS
Tue Apr 13 2010 @ 11:33 AM	Image taken
Tue Apr 13 2010 @ 04:30 PM	Image available at the NASA ftp server and automatically transferred to the Grid system resources
Tue Apr 13 2010 @ 11:30 PM	Generation of geospatial products using Grid computing resources

^aTime is local Ukrainian.

14.6.4 Discussion

Summarizing, we may point out the following benefits of using Grid technologies and Sensor Web for the case studies described in this section. Within the meteorological modeling application, the use of the Grid system resources made it possible to considerably reduce the time required to run the WRF model (up to 43.6 times). It is especially important for the cases when it is necessary to tune the model and adapt it to a specific region and thus to run the model multiple times to find an optimal configuration and parameterization [5]. For the flood application, Grids also allowed us to reduce the overall computing time required for satellite image processing, and made possible the fast response within international programs and initiatives related to disaster management. The Sensor Web standards ensured automated tasking of remote-sensing satellite and a timely delivery of information and corresponding products in case of emergency. Although a successful use case of using the Sensor Web was demonstrated in this section, it is not always the case. Moreover, the case study of the SOS service for surface temperature retrieval from weather stations showed that a lot of database issues still exist that should be properly addressed within the future implementations.

14.7 Conclusions

This chapter was devoted to the description of different approaches to integration of satellite monitoring systems using such technologies as Grid and Sensor Web. We considered integration at the following levels: *data integration level* and *task management level*. Several real-world examples were given to demonstrate such integration. The first example referred to the integration of the Ukrainian satellite monitoring system operated at the SRI NASU-NSAU and the Russian satellite monitoring system operated at the IK RAN at the data level. The second example referred to the development of the InterGrid infrastructure that integrates several regional and national Grid systems: the Ukrainian Academician Grid with a satellite data processing Grid segment UASpaceGrid and the CEODE-CAS Grid. Different issues regarding the integration of the emerging Sensor Web technology with Grids were discussed in the study. We showed how the Sensor Web can benefit from using Grids and vice versa. A flood monitoring and prediction application was used as an example to demonstrate the advantages of integration of these technologies. An important problem of Grid workflow management for satellite data processing was discussed, and automation of the workflow for flood mapping from satellite SAR imagery was described. To benefit from using data from multiple sources, integration of the Ukrainian and Chinese flood mapping services that use radar and optical satellite data was carried out.

References

1. Shelestov, A., Kravchenko, O., Ilin, M.: Distributed visualization systems in remote sensing data processing GRID. *Int. J. Inf. Tech. Knowl.* **2**(1), 76–82 (2008)
2. Allcock, W., Bresnahan, J., Kettimuthu, R., Link, M.: The Globus Striped GridFTP Framework and Server. In: *Proceedings of ACM/IEEE SC 2005 Conf on Supercomputing*. (2005). doi: 10.1109/SC.2005.72
3. Butler, R., Engert, D., Foster, I., Kesselman, C., Tuecke, S., Volmer, J., Welch, V.: A national-scale authentication infrastructure. *IEEE Comp.* **33**(12), 60–66 (2000)
4. Alfieri, R., Cecchini, R., Ciaschini, V., dell'Agnello, L., Frohner, Á., Gianoli, A., Lõrentey, K., Spataro, F.: VOMS, an authorization system for virtual organizations. *Lect. Notes Comp. Sci.* **2970**, 33–40 (2004)
5. Kussul, N., Shelestov, A., Skakun, S., Kravchenko, O.: Data assimilation technique for flood monitoring and prediction. *Int. J. Inf. Theor. Appl.* **15**(1), 76–84 (2008)
6. Kussul, N., Shelestov, A., Skakun, S.: Grid system for flood extent extraction from satellite images. *Earth Sci. Informatics* **1**(3–4), 105–117 (2008)
7. Fusco, L., Cossu, R., Retscher, C.: Open grid services for Envisat and Earth observation applications. In: Plaza, A.J., Chang, C.-I. (eds.) *High Performance Computing in Remote Sensing*, pp 237–280, 1st edn. Taylor & Francis, New York (2007)
8. Feller, M., Foster, I., Martin, S.: GT4 GRAM: A functionality and performance Study. <http://www.globus.org/alliance/publications/papers/TG07-GRAM-comparison-final.pdf> (2007). Accessed 30 Aug 2010
9. Moe, K., Smith, S., Prescott, G., Sherwood, R.: Sensor web technologies for NASA Earth science. In: *Proceedings of 2008 IEEE Aerospace Conference*, pp. 1–7 (2008). doi:10.1109/AERO.2008.4526458
10. Mandl, D., Frye, S.W., Goldberg, M.D., Habib, S., Talabac, S.: Sensor webs: Where they are today and what are the future needs? In: *Proceedings of Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems (DSSNS 2006)*, pp. 65–70 (2006). doi: 10.1109/DSSNS.2006.16
11. Foster, I.: The Grid: A new infrastructure for 21st century science. *Phys. Today* **55**(2), 42–47 (2002)
12. Shelestov, A., Kussul, N., Skakun, S.: Grid technologies in monitoring systems based on satellite data. *J. Automation Inf. Sci.* **38**(3), 69–80 (2006)
13. Chu, X., Kobialka, T., Durnota, B., Buyya, R.: Open sensor web architecture: Core services. In: *Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing (ICISIP)*, pp. 98–103. IEEE, New Jersey (2006)
14. Botts, M., Percivall, G., Reed, C., Davidson, J.: OGC sensor web enablement: Overview and high level architecture (OGC 07–165) Accessible via http://portal.opengeospatial.org/files/?artifact_id=25562 (2007) Accessed 30 Aug 2010
15. Humphrey, M., Wasson, G., Jackson, K., Boverhof, J., Rodriguez, M., Bester, J., Gawor, J., Lang, S., Foster, I., Meder, S., Pickles, S., McKeown, M.: State and events for web services: A comparison of five WS-resource framework and WS-notification implementations. In: *Proceedings of 4th IEEE International Symposium on High Performance Distributed Computing (HPDC-14)*, Research Triangle Park, NC (2005)
16. Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented Systems. In: *IFIP International Conference on Network and Parallel Computing, LNCS, vol. 3779*, pp. 2–13. Springer, Heidelberg (2005)
17. Williams, D.N., et al.: Data management and analysis for the Earth System Grid. *J. Phys. Conf. Ser.* **125**, 012072 (2008). doi: 10.1088/1742-6596/125/1/012072
18. Kussul, N., Shelestov, A., Skakun, S.: Grid and sensor web technologies for environmental monitoring. *Earth Sci. Informatics* **2**(1–2), 37–51 (2009)
19. Mandl, D.: Experimenting with sensor webs using Earth observing 1. *IEEE Aerospace Conference, Big Sky, MT* (2004)