Development of climate data storage and processing model

I G Okladnikov[1,2,3], E P Gordov[1,2,3,4] and A G Titov[1,2,3]

[1]Instutite of Monitoring of Climatic and Ecological Systems of the Siberian Branch of the Russian Academy of Sciences, 10/3, Akademichesky pr., Tomsk 634055, Russia
[2]Institute of Computational Technologies of the Siberian Branch of the Russian Academy of Sciences, Tomsk Branch, 10/4, Akademichesky pr., Tomsk, Russia
[3]Instutite of Atmospheric Optics of the Siberian Branch of the Russian Academy of Sciences, 1, V.E. Zuev sq., Tomsk 634055, Russia
[4]Tomsk State University, 36, Lenina str., Tomsk 634050, Russia

Abstract. We present a storage and processing model for climate datasets elaborated in the framework of a virtual research environment (VRE) for climate and environmental monitoring and analysis of the impact of climate change on the socio-economic processes on local and regional scales. The model is based on a «shared nothing» distributed computing architecture and assumes using a computing network where each computing node is independent and self-sufficient. Each node holds a dedicated software for the processing and visualization of geospatial data providing programming interfaces to communicate with the other nodes. The nodes are interconnected by a local network or the Internet and exchange data and control instructions via SSH connections and web services. Geospatial data is represented by collections of netCDF files stored in a hierarchy of directories in the framework of a file system. To speed up data reading and processing, three approaches are proposed: a pre-calculation of intermediate products, a distribution of data across multiple storage systems (with or without redundancy), and caching and reuse of the previously obtained products. For a fast search and retrieval of the required data, according to the data storage and processing model, a metadata database is developed. It contains descriptions of the space-time features of the datasets available for processing, their locations, as well as descriptions and run options of the software components for data analysis and visualization. The model and the metadata database together will provide a reliable technological basis for development of a high-performance virtual research environment for climatic and environmental monitoring.

Introduction

The present state of the art of modern environment monitoring tools and Earth system models is characterized by a continuous growth of the produced geospatial data volume. These data are used for monitoring and analysis of the climate change impact on the environment [1]. The data produced by various scientific and commercial institutions within the framework of hundreds of scientific projects usually have significant differences in the sets of physical variables, file formats, as well as different syntax and semantics, hampering the use of a common terminology. Due to the increasing volume of modeling and observational (especially, satellite) data, reaching several petabytes for a single dataset, there arise substantial computational limitations on their processing. Researchers regularly encounter those while working with large geospatial datasets in such scientific fields as meteorology, physics [2], and in biological and ecological studies [3]. The significant increase in the variety, volume, and speed of generation of meteorological and climatic data makes it impossible to collect and process them on a single computational node utilizing the traditional approaches [4]. A solution of the problems outlined is using a dedicated hardware and software infrastructure based on some new approaches to storing, retrieving, and processing of climatic geospatial data.

We present a model of storing and processing large climatic datasets, elaborated in the framework of development of a virtual research environment for the climate and environmental monitoring and analysis of the impact of climate change on the socio-economic processes on local and regional scales. This environment integrates known and new sets of climatic data, software implementations of classic and new methods of statistical analysis of large data sets. It provides an opportunity to scientists and decision-makers to use different geographically distributed spatially referenced data, processing resources, and services through a web browser by integrating distributed systems that store, process and provide information via a Geoportal.

1. Climate datasets storage and processing model

1.1. General concept

Our approach to storing and processing large spatial data is based on a distributed computing architecture called “shared nothing” [5]. In this approach, each computing (or working) node is independent and self-sufficient, without shared memory and storage space. The advantages of this architecture are: high reliability of computer network (a node failure does not affect the whole network functioning), high data processing speed (multi-node parallel data processing), and simple network scalability (the new nodes are easily integrated into the existing infrastructure). Each working node holds a dedicated software for processing and visualizing spatial data (“computing core”), which provides an application programming interface (API) for the interconnecting working nodes. The nodes are linked by data networks. They exchange data and control instructions via a secure connection or web processing service (WPS) (see Figure 1). The data processing tasks (“jobs”) are distributed among the computing cores by a special task manager (“Job Manager”), located on the master node. The common functions of the task manager are:

- to get data processing and visualization tasks from clients;
- to take into account the locations of datasets to be processed (“data awareness”) and to divide the jobs into individual tasks for the computing cores;
- to pass individual tasks to the working nodes corresponding to the datasets to be processed;
- to collect the individual results of processing from the working nodes; and
- to aggregate the individual results into a final result and provide it to the clients.

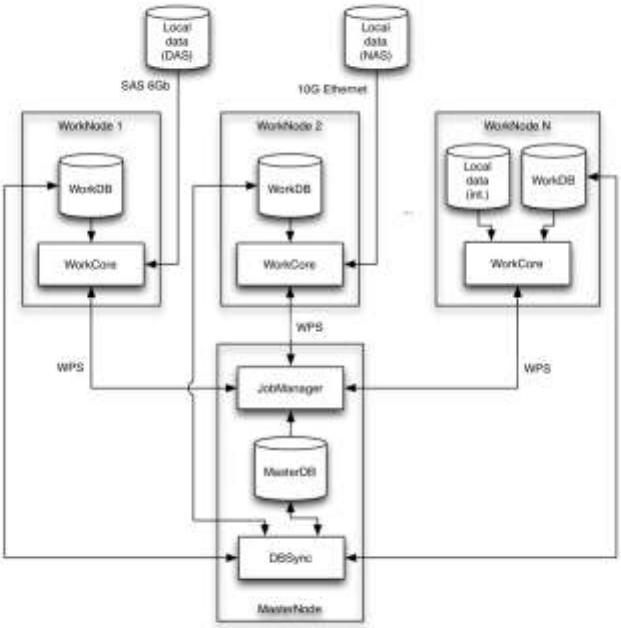

Figure 1. Computing network architecture.

1.2. Data

Currently, two major approaches to storing geospatial data are used: geospatial databases and file collections. The database approach utilizes relational and non-relational, local or distributed spatial databases such as Apache HBase (https://hbase.apache.org/), Esri Geodatabase (http://www.esri.com/products/arcgis-capabilities/data-management), Paradigm4 SciDB (http://www.paradigm4.com/), etc. This approach requires inserting all data into a spatial database before their actual use, which is a quite disk space- and time-consuming operation. The file collections approach relies on storing data as a collection of data files in file system directories. Usually, self-describing formats are used for storing geospatial data. It was shown [6] that the retrieval of data chunks larger than 40 Mb from a spatial database is less effective than from a simple collection of data files. On the other hand, in most cases the file-based approach provides fast data extraction without redundant preprocessing. However, this approach requires the development of an additional software layer to provide API adapters for storing and processing distributed file collections. In our case, the simplicity and flexibility of the file-based data storage approach prevailed over the other one.

Network Common Data Form (netCDF) was chosen as the major file format for most geospatial data in our data archive. This format is formally acknowledged by scientific institutions (including UCAR - University Corporation for Atmospheric Research) and OGC (Open Geospatial Consortium) as a standards' candidate for storing geospatial data and stimulating data exchange in binary form. It is perfectly suited for storing geospatial data and supported by a wide range of commercial and open source software.

The data are stored on data storage systems as collections of netCDF files and arranged in a strict hierarchy of directories:

- /<data root directory>/
- <data collection name>/
- <spatial domain resolution>/
- <time domain resolution>/
- <files and directories with data>

Here, <*data root directory*> is the root location of data collections, <*data collection name*> is the name of a directory containing a single data collection, <*spatial domain resolution*> is the name of a directory containing data with the same horizontal resolution, <*time domain resolution*> is the name of a directory containing data with the same time step. All data files (sometimes grouped in subdirectories) are located deeper in the hierarchy. The names of files and subdirectories are not regulated and determined by the specifics of a particular data set. Each data file contains one or more multi-dimensional arrays of meteorological parameters.

The data are stored on working nodes or on connected data storage systems (Local data) according to a "data awareness" principle. Data awareness is some knowledge about all of the data available within each working node and where those data are located [7]. According to this information, the datasets are processed locally, on the nodes containing them. The metadata describing geospatial datasets are stored in a distributed database on the master node (MasterDB) and on working nodes (WorkDB). The metadata include a description of the spatio-temporal characteristics of each dataset, a list of meteorological parameters and physical locations of datasets on data storage systems. The metadata regularly or on demand are synchronized (DBSync) among all nodes in order to maintain up-to-date information about datasets available for processing and their locations.

To speedup the data reading and processing, three approaches are used: a preliminary calculation and storage of the frequently used intermediate products, a distribution of the data across multiple storage systems (with or without redundancy), and caching and reuse of the previously obtained results. The first approach involves storing not only the geospatial datasets, but also the pre-calculated intermediate products obtained on their basis. First of all, these products include the results of spatial and/or temporal averaging: data with coarser spatial resolution, and averages per day, month, and year. In addition, some independent preliminary intermediate products, whose calculation takes a considerable time, can be obtained for the future use. The intermediate datasets will also be available as netCDF files, and their metadata will be stored in the database together with the information on the original datasets. The second approach involves a distribution of the datasets among the working nodes (possibly with redundancy), thereby providing increased performance due to parallel data reading and processing on multiple nodes. The third approach lies in a short-term storage of the intermediate and a long-term storage of the final results obtained by the users with automatic inclusion of their metadata into the database. When a task for the calculation of a cached dataset is passed to a working node, this result is not calculated again but extracted from the cache.

1.3. Metadata database
To describe the geospatial datasets and their processing routines and provide effective VRE functioning, a dedicated metadata database (MDDB) is required. Currently, there is a lack of such database in the Earth sciences. We suggest a comprehensive description of the climatic geospatial datasets and processing routines in a single database, which is characterized by the following features (Figure 2). The database contains structured spatial and temporal characteristics of the available geospatial datasets, their locations, and run options of the software components for data analysis. The structure contains "Dataset" and "Dataset collection" levels. Dataset is a set of data which is a) given on a single temporal and spatial grid, b) covers the same time range and c) obtained under the same simulation or observation conditions (if applicable). Spatial datasets according to the chosen data storage model are mostly represented by collections of netCDF files grouped by the spatio-temporal features and placed in the hierarchy of directories on the data storage systems. Each netCDF file includes one or more variables containing values of meteorological parameters given on a spatio-temporal domain. The file names in the dataset are unified. It is necessary to distinguish the term "parameter" (meteorological parameter) and "variable". A meteorological parameter is the name of some meteorological characteristic: temperature, pressure, humidity, etc. A variable is the unique name of a multidimensional array in a netCDF file. The names of meteorological parameters are standardized. In contrast, the names of variables in different datasets may be different and usually depend on the preferences of the institution that produced them. Along with data, the netCDF files

contain horizontal, vertical, and time domain grids. A dataset collection is a collection of datasets created by an organization within a specific project, but specified on different spatial and/or temporal grids, or for different scenarios. A collection may consist of one dataset. MDDB contains two major parts: 1) a description of climate datasets; and 2) a description of data processing software components (computing modules).

1.3.1. Description of climate datasets. The climate datasets available for processing are described by metadata stored in several interconnected tables. Each dataset is represented by a record in the table dataset which links together a unique set of dataset features. The dataset features are given in the following tables: collection, dataset_root_directory, scenario, resolution, time_step, as well as time_span, filetype. The table collection contains short names of data collections (e.g., "ERA40", "MERRA") and URLs of the corresponding internet sites with a detailed description. The related table collection_tr contains multi-lingual information about the collections: long names (e.g., "ERA-40 Reanalysis", "MERRA Reanalysis"), short descriptions (e.g.: "ECMWF 40 Year Re-analysis", " NASA/GMAO Modern-Era Retrospective Analysis for Research and Applications"), and names of organizations of origin (e.g., "ECMWF, UK", "NASA/GMAO, USA"). Table dataset_root_directory contains root paths to datasets. Usually these paths point to dedicated directories on data storage systems relative to the file system root directory on a computing node. For example, "/mnt/storage1/data" or "/nfs/megaraid1/data". The table scenario contains names of data acquisition conditions or modeling scenarios and the first part of a path to the data files relative to the dataset root directory. For example, climate projection modeling scenarios, such as "RCP45" or "RCP85". The table resolution contains a list of horizontal spatial resolutions and the second part of the path to data files. The resolutions can be given directly, in some units: "0.67x9.67 deg.", "30x30 m.", or by using topical notation: "T62", "T106". The table time_step contains a list of short notations of temporal grid steps and the third part of the path to data files. Typical notations of temporal grid steps for meteorological data are: "3h" (3 hours), "6h" (6 hours), "1d" (1 day), "1m" (1 month), "1y" (1 year). Related table time_step_tr contains multi-lingual full names of temporal grid steps (e.g., "6 hours"). Being concatenated, all the path fragments corresponding to the selected collection, scenario, horizontal resolution and temporal grid resolution form a full path to data files: / <dataset_root_directory.rootpath> / <scenario.subpath0> / <resolution.subpath1> / <time_step.subpath2> /. The table time_span contains short and long names of a temporal span covered by one NetCDF file, i.e., length of a time grid in one file. For example, "6h" and "6-hours span", "1d" and "1-day span", "1m" and "1-month span". In the latter case it means that one data file holds data obtained for a month-long period. Thus, data for one year will take 12 files, and for 50 years 600 files. Usually, a time-stamp is included in the data file names. The table filetype contains types of data files in the datasets. For example, such well-known types as "netcdf", "grid", or "shapefile".

Each meteorological parameter in a dataset given on the whole spatio-temporal domain of this dataset and one vertical level, and stored in a single variable with the same name in all files of the dataset is treated as a "distributed multidimensional data array". Each data array is represented by a record in the table *data* which links dataset (*data*.dataset_id), variable (*data*.variable_id), vertical level (*data*.levels_id), as well as data file name template (*data*.file_id) and a variable containing a list of vertical levels (*data*.level_variable_id). The table *variable* contains names of variables storing various meteorological parameters (e.g., "air"). It is linked with a table *parameter* containing IDs and a table *parameter_tr* containing multi-lingual standard names of meteoparameters (e.g., "Air temperature"). Also it is linked with a table *units* containing IDs and a table *units_tr* containing multi-lingual names of meteoparameter units (e.g., "K" – Kelvin degrees). The table *levels* contains lists of vertical levels at which meteoparameters are given (e.g., ":300:400:500:600:700:850:925:1000:" or ":msl (Mean sea level):"). Related table *levels_tr* contains multi-lingual names of corresponding vertical levels (e.g., "Mean sea level"). Also, the table *levels* is linked with a table *level_type* containing types of vertical levels. The table *level_type* is linked with a table *level_type_tr* containing corresponding multi-lingual names of types of vertical levels (e.g., "Isobaric level") and tables *units*/*units_tr* containing multi-lingual names of units (e.g., "hPa"). The table *file* contains data file name templates and time ranges covered by a corresponding collection of data files. For example, a template "%year%/macc_tmp_%year%%mm%.nc" uses two placeholders: "%year%" corresponding to a four-digit year and "%mm%" corresponding to a two-digit month number (e.g., 1990/macc_tmp_199005.nc). The data file name template can include many different placeholders or not include them at all. On the basis of the time range required and a file name template the dedicated data-access software modules prepare a list of names of data files. Then required variables are read from these files. The table *level_variable* contains names of auxiliary variables storing lists of vertical levels in NetCDF-files (e.g., "levels") and is used by service software modules for searching and reading data arrays from files.

The first problem which is solved with the help of the MDDB is the determination of the full path to data files and NetCDF file names containing the required meteorological parameter from a selected climate dataset and given on a selected spatio-temporal domain. Besides, there are interrelated tables in the MDDB containing multi-lingual descriptions of collections (*collection_tr*), resolutions of horizontal (*resolution_tr*) and temporal (*time_step_tr*) grids, vertical levels (*levels_tr*, *level_type_tr*), meteoparameters (*parameter_tr*) and units (*units_tr*). Together they describe all datasets in a human-readable form and are used in drop-down menu of the graphical user interface. The second problem which is solved with the help of the MDDB is providing a graphical user interface with relevant interrelated information describing the spatial datasets available for analysis. The advantage of using the MDDB is a rapid actualization and delivery to the user the information concerning available datasets just after their integration into the VRE.

1.3.2. Description of data processing software components (software modules) and their run options. The dedicated software procedures ("computing modules" or "processing modules") are used for climate datasets processing. "Module" is a functionally finished fragment of a computing core implementing a number of methods and designed to solve one elementary task. Each module has at least one input dataset (argument) and at least one output dataset (result). To control the module functionality, some run options can be passed to it as a set of different constants. Any module's result can be an argument for another one. When several modules are run consequently and results are passed from the output of one module to the input of another one a "computing pipeline" is formed. Run and execution control of the computing pipeline is performed by the computing core of the VRE. Some module options can be set or selected from a list of valid values using a graphical user interface (GUI). The following option types are valid: 1) Boolean; 2) string; 3) number. Each option type has a corresponding GUI element: 1) Boolean – checkbox; 2) string – dropdown list; 3) number – textbox. A set of certain options determines the only way of data processing by a particular computing pipeline. That way of processing is called hereinafter "a process". The processes based on the same computing pipeline differing only in run options are combined in a "class of processes". A process is described in an XML "task-file". Only one task-file corresponds to each process. If a task-file describes a processing pipeline, intermediate datasets and particular run options, while the descriptions of datasets to be processed and spatio-temporal domain are omitted it becomes a "task-template". The omitted descriptions and options are inserted into a task-template after the user made the corresponding selections in the GUI. The prepared task-file is passed to the computing core for further processing.

Each class of processes is associated with a class type which is used to establish a correspondence between the data arrays and the applicable processes. The table *class_type* contains the names and descriptions of class types. It is linked to the table *handled_data* where correspondence between data arrays and class types are set. The table *process_class* contains IDs of classes of processes, and *process_class_tr* , their multi-lingual names. A class of processes combines the processes which differ only in run options. The table *option* contains the IDs of all available options of all processes represented in the MDDB, and related table *option_tr* , their multi-lingual names. The options of one type related to the same process are combined into groups. The table *options_group* contains the IDs of groups of options, and a related table *options_group_tr*, their multi-language names which can be used as captions of GUI elements. Another related table, *group_type*, contains types of groups of options. The type of a group defines the type of the GUI element used for displaying the options in the group. For example, "checkbox" corresponds to a check box element. The options and groups are linked in the table *grouped_options*. The groups of options and classes of processes are linked in the table *class_options*. Thus, all possible run options linked to a class of processes define all possible processes of this class. Selection of a class of processes and a certain set of run options by the user defines a specific process. All valid combinations of run options for each class that the user can select are specified in the table *selected_options*. This table is linked to the table *options_set* containing IDs of valid combinations. Each process is represented by a record in the table *process* linking a process class (*process*.process_class_id) and a combination of run options (*process*.options_set_id). The related table *task_template*_files contains the file names of template-tasks corresponding to each process. These template-tasks are used by VRE's middleware to prepare file-tasks for the computing core.

The third problem which is solved with the help of the MDDB is the determination of file names of template-tasks using processing options selected by the user in the graphical user interface. Besides, there are interrelated tables in the MDDB containing multi-lingual names of classes of processes (*process_class_tr*), names of options of processes (*option_tr*), and captions of the GUI elements used for displaying these options (*options_group_tr*). Together they describe spatial data processing options in a human-readable form and are used as data for filling the elements of the graphical user interface. The fourth problem which is solved with the help of the MDDB is providing a graphical user interface with relevant interrelated information describing the available modules for spatial data processing. The advantage of using the MDDB is rapid actualization and delivery to the user of the information concerning the available data processing modules just after their integration into the VRE.

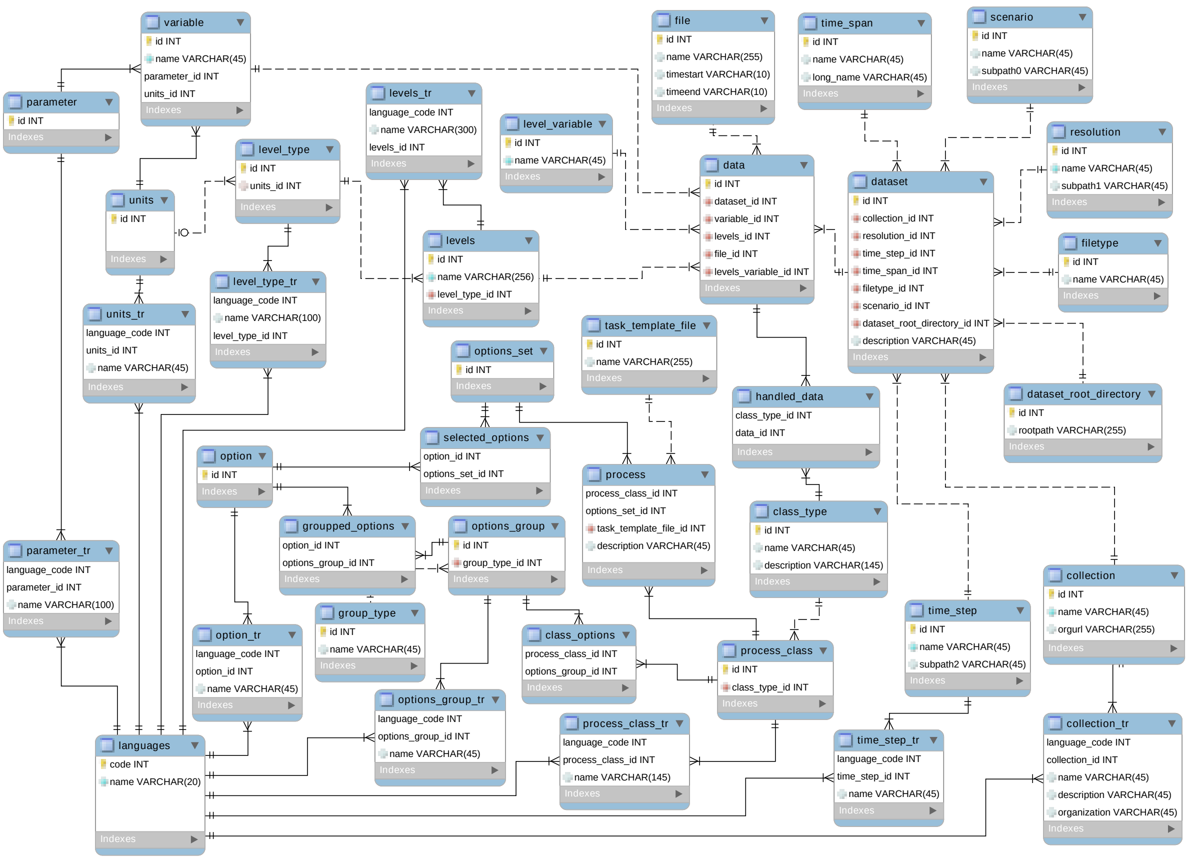

Figure 2. ER-diagram of metadata database describing features of available spatial datasets, data processing computing modules and their inter-relations.

2. Conclusions

The above-presented climate data storage and processing model is designed for a virtual research environment of climatic and environmental monitoring and analysis of the impact of climate change on the socio-economic processes on local and regional scales. It will provide a computing network with reliability and scalability, as well as high computational performance, due to the “shared nothing” basis, netCDF file collections approach, and the use of a metadata database. To date, there is no formalized description of the metadata database for large sets of geospatial meteorological and climatic data in the literature. The database presented in this study is the first attempt to address this crucial data intensive domain problem. The developed metadata database has several advantages. It provides data awareness for the working nodes by minimizing the inter-node data transfer and increasing the processing performance. Also, it facilitates an easy and fast expansion of the existing and new datasets and data processing computing modules into the VRE. This reduces the time required for extending the VRE’s functionality. Finally, it unifies the data processing routines resulting in a stable and solid program code.

The first application of the metadata database demonstrated high efficiency of its use [8]. It provided easy and fast addition of the new data processing computing modules into the information and computation platform “Climate” [9]. It shows that the climate data storage and processing model and the metadata database together will provide a reliable technological basis for the development of a high-performance virtual research environment for climatic and environmental monitoring and analysis of the impact of climate change on the socio-economic processes on local and regional scales.

Acknowledgements
The authors thank the Russian Science Foundation for support of this work under grant №16-19-10257.

References

[1] Lykosov V N, Glazunov A V, Kulyamin D V, Mortikov E V and Stepanenko V M 2012 *Supercomputer Simulations in the Climate System Physics* (Moscow: Moscow State Universoty publishing) p 295

[2] Sandia sees data management challenges spiral *Scientific computing world* 4 August 2009 http://www.scientific-computing.com/news/news_story.php?news_id=922.

[3] Reichman O J, Jones M B, Schildhauer M P Challenges and Opportunities of Open Data in Ecology 2011 *Science* **331** 6018 703–5

[4] Gordov E P, Kabanov M V, Lykosov V N 2006 Information-computational technologies for environmental science: young scientists training *J. Computational Technologies* (Special issue 1) **11** 3–15

[5] Valduriez P 2009 Shared-Nothing Architecture *Encyclopedia of Database Systems* ed L Liu and M Tamer Özsu pp 2638–9

[6] Santokhee A, Blower J, Haines K 2005 Storing and Manipulating Gridded Data In Spatial Databases (Reading: Reading E-science Center, University of Reading) http://go-essp.gfdl.noaa.gov/presentations/06_06_05/Santokhee/Adit_Sank.ppt%20[Read-Only].pdf

[7] Brackett M 2012 *Data Resource Integration: Understanding and Resolving a Disparate Data Resource* (Technics Publications, LLC publishing) p 540

[8] Ryazanova A A, Voropay N N, Okladnikov I G 2016 Application of information and computing web system «Climate» for estimation of aridity of South Siberia *Proc. of Int. Conf. and Early Career Scientists School on Environmental Observations, Modeling and Information Systems ENVIROMIS-2016 (*July 11-16, 2016, Tomsk, Russia) pp 358–62

[9] Okladnikov I G, Gordov E P, Titov A G and Shulgina T M 2015 *Proc. of the XVII Int. Conf. on Data Analytics and Management in Data Intensive Domains* (*October 13–16, 2015, Obninsk, Russia)* (*Electronic Materials*) ed L Kalinichenko and S Starkov pp 76–80