

Framework of cloud web services for processing remote sensing data

Alexey Buchnev¹, Pavel Kim¹, Valery Pyatkin^{1,}, Fedor Pyatkin², and Evgeny Rusin¹*

¹The Institute of Computational Mathematics and Mathematical Geophysics SB RAS,
Academician Lavrentyev Ave., 6, Novosibirsk, 630090, Russia

²Siberian Centre Scientific Research Centre of Space Hydrometeorology ‘Planeta’,
Sovetskaya St., 30, Novosibirsk, 630099, Russia

Abstract. We consider a distributed network of cloud web services for processing satellite data, which provides data processing facilities for Earth remote sensing within SaaS model. In fact, this is a set of web services that implement the functional modules of the PlanetaMonitoring remote sensing data processing system.

Introduction

The aim of this work is to develop and implement a model (prototype) of a framework for distributed network of cloud web services for processing satellite data, which provides data processing facilities for Earth remote sensing within the SaaS model. In fact, this is a set of web services that implement the functional modules of the PlanetaMonitoring system for thematic processing of remote sensing data [1].

1. PlanetaMonitoring software

PlanetaMonitoring software was developed as a joint effort of the Institute of Computational Mathematics and Mathematical Geophysics (ICM&MG) SB RAS and the State Research Centre ‘Planeta’. It provides a functionally complete set of operations for processing remote sensing data and implements a number of software technologies for processing multispectral satellite information in the optical, infrared, and microwave ranges. Here are some of these software technologies: satellite image filtering; radiometric and geometric correction; geo-referencing; transformation into cartographic projections and the construction of mosaics from individual images; detection of lineaments and ring structures; recognition and classification of environmental objects (cluster analysis and supervised classification); determination of objects’ spatial movements using multi-temporal satellite images. To use the above-listed parts of PlanetaMonitoring in the cloud environment, we implemented each of them as autonomous (i.e. non-interactive) Windows. This article proposes an approach to the organization of a distributed cloud environment capable of ensuring reliable and efficient execution of these Windows applications when processing remote sensing data.

* Corresponding author: pvp@oii.sgcc.ru

2. Cloud computing

At present, the use of Internet technology for the operational integration of information and computing resources in solving the problems of processing remote sensing data is becoming relevant. The paradigm of cloud computing gives such an opportunity [2]. The main idea of cloud computing is distributed data processing and storage technologies, where all necessary resources are provided to the user as an Internet service. There are the following kinds of cloud computing technologies, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and many other. In ICM&MG SB RAS, we develop the prototypes for web services that use software technologies (autonomous Windows applications) of the PlanetaMonitoring software package; which solves a new problem of providing remote sensing data processing facilities within the cloud SaaS model [3]. Service prototypes are implemented on the Windows platform and consist of the following two components:

- *computational component*, created on the basis of the a previously developed corresponding Windows application (autonomous mode); and
- *web interface*, based on the free and open-source Apache web server.

The main ('master') cloud web server for satellite data processing is implemented on the basis of ICM&MG SB RAS. A number of circumstances impose such a solution. First of all, a so-called cloud environment is formed on the main server, providing remote sensing data processing services as part of the SaaS cloud model [4]. It is important that there is the possibility to organize high-performance scalable computing at the Siberian Supercomputer Center (SSCC) SB RAS, which is a part of ICM&MG SB RAS [5].

3. High performance scalable computing

The developed framework assumes the use of high-performance SSCC resources for time-consuming calculations. The high-performance computing subsystem is implemented on the basis of the SSCCIP (Siberian Scientific Computing Center – Image Processing) software developed by the authors for integrating remote high-performance computers into the processing and analysis of satellite data [6]. SSCCIP consists of four components:

- *client component* - the operator's workplace in the Windows environment (it provides a GUI for configuring a request for data processing, tracking its execution results, and visualizing them);
- *server component* - Unix-application running on a supercomputer (it provides the processing of the client component request and its dispatching);
- *communication component* - network exchange between client and server components (which is implemented on the basis of the secure SSH protocol);
- *computational component* – on the basis of the high-performance image processing library ParImProLib [7] developed by the authors. The library provides calculations on the NKS-30T + GPU SSCC hybrid cluster: inter-node exchange is provided by the MPI interface; and calculations on the GPU (currently in development), by Nvidia CUDA technology. In the future, it is planned to expand the library with the function of processing on the Intel Xeon Phi CPU to enable it to perform calculations on the new SSCC cluster NKS-1P.

Note that the SSCCIP system and the ParImProLib library are also built in the form of a framework: their source code is organized so that operations common to a certain type of processing algorithms are separated from the operations specific to particular algorithms of this type. In [6], it was shown that this approach allows maintaining high performance of the program code while significantly simplifying the process of adding new processing algorithms to the system. To use SSCCIP as a subsystem of a cloud web service, its client component will be enhanced with the ability to perform autonomous processing with an

interface identical to the interface provided by PlanetaMonitoring. Thus, the framework being developed will isolate the web server from the special characteristics of a particular computer.

4. Roshydromet Core Ground Segment (CGS) Ground system of receiving, processing, archiving and distribution of satellite data at State Research Center ‘Planeta’

The cloud environment implemented on the main server (either totally or partially) is transferred to the CGS servers using the “mirroring” technology [8]. The Ground system of receiving, processing, archiving and distribution of satellite data includes three regional centers that are the part of the State Research Center ‘Planeta’: European (Obninsk – Moscow – Dolgoprudny), Siberian (Novosibirsk) and Far Eastern (Khabarovsk) ones. In addition, the CGS includes a network of fixed and mobile autonomous satellite data reception points (about 70) in Russia, Antarctica, and on marine vehicles under the scientific guidance of the State Research Centre ‘Planeta’. The first backup mirror will be implemented on the server of the Siberian Center (SC) of the Scientific Research Centre ‘Planeta’. It is important that this Scientific Research Centre supports the web server, which presents the operational satellite data arriving in real-time mode (<ftp1.rcpod.ru>).

5. Computational components of a distributed cloud web services system

Virtually every computational component of a web service is an autonomous version of the corresponding software technology of the PlanetaMonitoring system. This article presents a cloud prototype consisting of two web services created on the basis of previously developed Windows applications: LINECOIL (lineaments and ring structures’ detection on satellite images) and KMEAN (clustering by the K-means method). To access the corresponding web service, one need to transfer a text file with processing parameters to the cloud environment.

Note that the technology of detecting lineaments and ring structures has already been implemented in the form of a cloud web service and is available to the Internet users [3]. At present, the testing of a web-based cloud clustering of remote sensing data is being completed. The main clustering algorithm in our system is the known K-means algorithm [9]. The algorithm is based on an iterative procedure for assigning feature vectors to clusters by the criterion of the minimum vector distance to the center of the cluster. It is considered optimal to divide the input vectors into clusters, at which the intraclass spread cannot be reduced when transferring any vector from one cluster to another. The K-means algorithm provides the minimum (in the general case of the local) of the following error function:

$$E = \sum_{i=1}^K \sum_{j=1}^N \gamma_{ij} \rho(x_j, m_i), \gamma_{ij} = \begin{cases} 1, & x_j \in C_i \\ 0, & x_j \notin C_i \end{cases}$$

Here N is the number of feature vectors, x_j is the feature vector number j , K is the number of clusters, C_i is the cluster with number i , m_i is the center of the cluster number i , $\rho(x, y)$ is the selected metric (distance) between the vectors x and y .

The results of clustering can be controlled using the following parameters [10]: the number of clusters to be allocated; the number of iterations of the algorithm; metric type (distance) between vectors (Euclidean, Chebyshev, Mahalanobis, Manhattan), the selected metric defines the shape of the resulting clusters; the method of selecting the initial cluster centers (one out of three, two of them are determined on the basis of the data set statistical characteristics and one is based on a random sampling); and computing accuracy. In addition,

it is possible to save statistical data of the obtained cluster map (cluster volumes, vectors of mean values in clusters, standard deviations, etc.) in a text file.

6. Web interface of a distributed cloud web services system

The development of “private” cloud services has also such an argument as the possibility of ensuring the safekeeping of intellectual property for the creation and use of proprietary software algorithms in the know-how mode. Thus, it turns out to be justified to invest some effort into the cloud technologies development, which is incommensurable in scale to the capabilities of software corporations. When designing their software content, the numerical expansion of the web services range (developed within the framework of a single cloud technology) should take into account qualitative features in the structure of user requests, in particular - their async origin. Large time expenditures for servicing a particular online service require tracking its current state, and the decision to interact with the user on this task successfully combines the balance between the computational loads on the process itself and its auxiliary software. Another vital problem of cloud development, the solution of which increases its effective potential, is the processing of multiple concurrent same-type requests. It can be most possible without serious consequences when “mirroring” service processes with the help of an available network, i.e. duplicating infrastructure using file system directory hierarchy repositories that fit into the hypertext space of possible web services browsers. The “mirroring” operation is provided by the resource allocation manager for specific service requests. The architecture of embedding the object-oriented cloud of web services into the distributed CGS network determines the structural complexity of the cloud resource allocation manager. “Replaceable” computational components rely on a unified network interface and provide for the possibility to implement computational components on various architectural platforms of heterogeneous networks, combining both multiprocessor SSCC architectures and CGS distributed networks. Considering the heterogeneity and vastness of the market of user browsers used on the Internet, the web interface in the prototype version is implemented using the basic means of the hypertext language HTML. The service part is made on Apache 2.2, while taking into account the natural heterogeneity of the underlying infrastructure, other platforms are being also developed in the IIS Windows architecture. Informational data streams are provided via FTP protocol. Thus, the user has the ability to search for “hidden” in the “cloud” of objects, both in archival data and in individual samples of images. The cloud storage paradigm is tracked here.

Fig. 1-4 show an example of a session with a K-means cluster web service prototype: Fig. 1 is a screenshot of the configuration web interface form setting up the detection of 15 clusters; Fig. 2 is the the source image of the flood in the Tyumen region (satellite Sentinel-2, resolution 10 m, 05/11/2017); Fig. 3 is the result of clustering using the Euclidean metric; Fig. 4, the Mahalanobis metric.

Nn	Пояснения	Параметр	Значение
1	Имя файла в формате BMP, содержащего данные для обработки с размерностью векторов измерений от 1 до 4.	Input file	S2_170511_0637_10m_band843.bmp
2	Имя файла в формате BMP, в котором будет сохранен результат обработки.	Output file	S2_170511_0637_10m_band843_res.br
3	Максимально допустимое число кластеров.	Kk	15
4	Тип метрики для определения расстояния между векторами (Евклида/Махаланобиса/Манхэттена/Чебышева).	Metric_Type	1
5	Способ выбора начальных центров кластеров.	Center_Type	1
6	Способ изменения центров кластеров.	Dynamic_Centers	1
7	Радиус шара для кластеризуемых векторов.	Distance_M	-1
8	Количество итераций.	Iterations	50
9	Точность вычислений.	Delta	0.0000001
10	Нужно/не надо формировать файл статистик.	Statistics file	1

Запустить программу

Fig. 1.

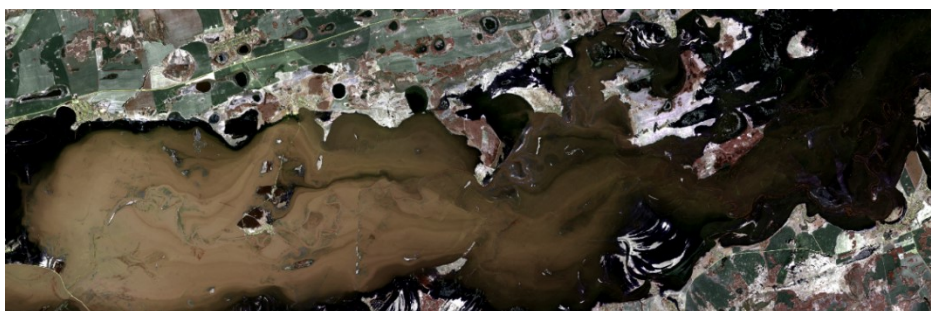


Fig. 2.

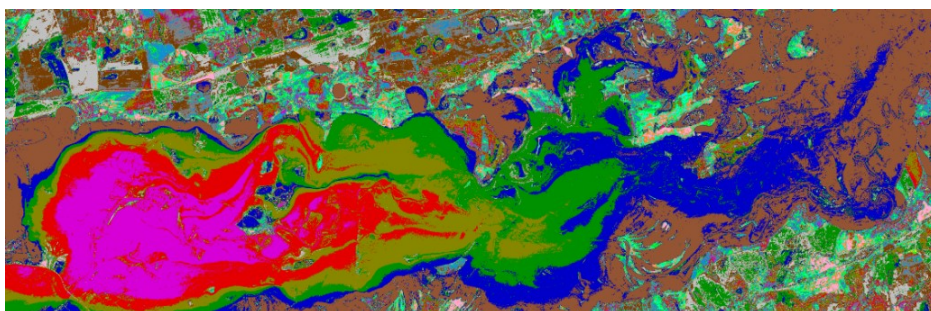


Fig. 3.

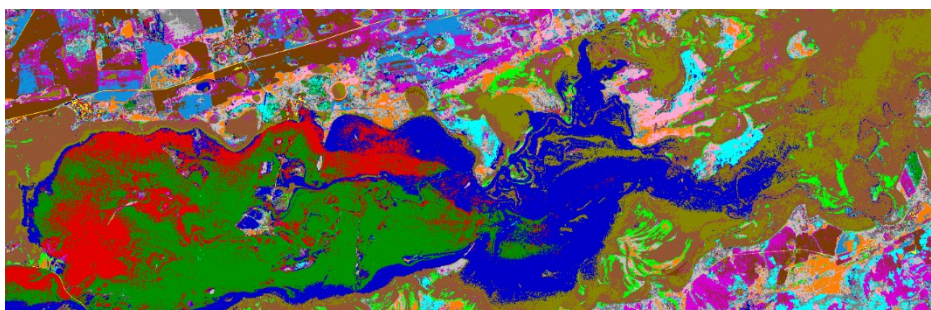


Fig. 4.

Conclusion

Successful experience in implementing a cloud web service prototype for detecting lineaments and ring structures on space images was used to create the cloud service for remote sensing data clustering, which represents the known K-means algorithm. In the future, it is planned to develop the prototypes of cloud services for other programming modules of the PlanetaMonitoring system.

The work was conducted within the framework of the budget project 0315-2016-0003 for ICM&MG SB RAS with the support of Russian Foundation for Basic Research (grant 16-07-00066). Siberian Supercomputer Center SB RAS is gratefully acknowledged for providing supercomputer facilities.

References

1. V.V. Asmus, A.A. Buchnev, V.A. Krovotyntsev, V.P. Pyatkin, G.I. Salov, Optoelectronics, Instrumentation and Data Processing **54(3)** 1–9 (2018)
2. P.A. Kim, P.A. Kalantaev, V.P. Pyatkin, Pattern Recognit. Image Anal. **25(2)**, 220–222 (2015)
3. A.A. Buchnev, P.A. Kim, V.P. Pyatkin, G.I. Salov, Journal of Siberian Federal University. Engineering and Technology **10(6)**, 741–746 (2017)
4. A.A. Buchnev, V.P. Pyatkin, F.V. Pyatkin, Interexpo Geo-Siberia, 8–13 (2017)
5. SSCC ICMMG SB RAS, <http://www.sccc.icmmg.nsc.ru>
6. E.V. Rusin, LNCS **6873**, 467–472 (2011)
7. E.V. Rusin, LNCS **5698**, 344–349 (2009)
8. Mirror website, https://en.wikipedia.org/wiki/Mirror_website
9. A.K. Jain, Pattern Recognit. Lett. **31**, 651–666 (2010)
10. V.V. Asmus, A.A. Buchnev, V.P. Pyatkin, Optoelectronics, Instrumentation and Data Processing **46(2)**, 149–155 (2010)