AUTOMATED ANALYSIS OF REMOTELY SENSED IMAGES USING THE UNICORE WORKFLOW MANAGEMENT SYSTEM

Shahbaz Memon[1,2]*, Gabriele Cavallaro*[1]*, Björn Hagemeier*[1]*, Morris Riedel*[1,2]*, Helmut Neukirchen*[2]

[1] Jülich Supercomputing Centre, Forschungszentrum Jülich, Jülich, Germany
[2] School of Engineering and Natural Sciences, University of Iceland, Reykjavik, Iceland

ABSTRACT

The progress of remote sensing technologies leads to increased supply of high-resolution image data. However, solutions for processing large volumes of data are lagging behind: desktop computers cannot cope anymore with the requirements of macro-scale remote sensing applications; therefore, parallel methods running in High-Performance Computing (HPC) environments are essential. Managing an HPC processing pipeline is non-trivial for a scientist, especially when the computing environment is heterogeneous and the set of tasks has complex dependencies. This paper proposes an end-to-end scientific workflow approach based on the UNICORE workflow management system for automating the full chain of Support Vector Machine (SVM)-based classification of remotely sensed images. The high-level nature of UNICORE workflows allows to deal with heterogeneity of HPC computing environments and offers powerful workflow operations such as needed for parameter sweeps. As a result, the remote sensing workflow of SVM-based classification becomes re-usable across different computing environments, thus increasing usability and reducing efforts for a scientist.

Index Terms— Remote Sensing, Support Vector Machine (SVM), High-Performance Computing (HPC), Scientific Workflows, UNICORE.

1. INTRODUCTION

Due to the advancement of the latest-generation remote sensing instruments a wealth of information, such as spatial, multi-temporal, physical parameters are generated almost on a continuous basis and with an increasing rate at global scale. This sheer volume and variety of sensed data leads to a necessary re-definition of the challenges within the entire lifecycle of remote sensing data [1]. Trends in parallel High-Performance Computing (HPC) architectures are in continuous expansion to attempt the growing demand of domain-specific applications for handling computationally intensive problems. In the context of large scale remote sensing applications, where the interpretation of the data is not straightforward and near real time answers are required, HPC and Cloud Computing can bear the chance to overcome the limitations of serial algorithms. Data analysis is a life cycle of multiple phases, and the classification of images is just one of the tasks in this realm. To have error free data analysis, it is imperative to have the tasks managed in a pre-defined manner and –in accordance with application requirements– executed on distributed HPC resources. In the simplest case, all the phases are to be deployed on one HPC resource (e.g., cluster), but if they are distributed across different clusters with different access mechanisms, then it becomes tedious for users to manage all the processes. Moreover, the data access also plays a vital role, since the data sets or resultant output requires interaction with data repositories through heterogeneous data management and file transfer interfaces and protocols. Besides the execution and data management requirements, the data analysis pipeline has a set of tasks, that can be either sequential, concurrent or iterative, thus forming a workflow.

To realize this scenario, we use tools and techniques from the area of scientific Workflow Management Systems (WMS). Several WMS have been implemented to support physical and data modelling applications, but there are less cases known where machine learning scenarios, such as classification methods, have been catered. In our previous work [2], we proposed to automate only the cross-validation phase of the Support Vector Machine (SVM) [3], which are one of the most used classifiers for analyzing and extracting information from remote sensing data; in this earlier work, we use a standards-based parameter sweep model and the HPC middleware UNICORE [4] for the cross-validation.

This paper goes one step further by extending the workflow to the model generation (i.e., training) and prediction (i.e., classification) phases by using UNICORE's more advanced workflow management capabilities and it's graphical client, the UNICORE Rich Client (URC) [5]. We introduce an end-to-end workflow design, implementation, execution, and monitoring through the URC's visual interface. URC helps in reducing tremendous amounts of time, development effort, and makespan to analyze remote sensing data not only for the classification phase but also within pre-processing steps such as feature extraction.

Thanks to Jülich Supercomputing Centre (JSC) for funding. This work was partly supported by NordForsk as part of the Nordic Center of Excellence (NCoE) eSTICC (eScience Tools for Investigating Climate Change at High Northern Latitudes). Correspondence: m.memon@fz-juelich.de

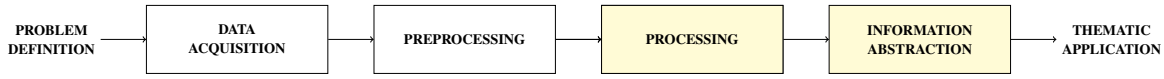

Fig. 1. Stages of a general remote sensing data processing flow. The blocks highlighted in yellow are the considered steps.

2. SCIENTIFIC CASE STUDY

The entire lifecycle of remote sensing data consists of a multi-step pipeline (see Fig. 1) that includes several data-driven methods between the acquisition to the application phase: preprocessing (e.g., geometric and atmospheric corrections), processing (e.g., feature extraction) and information abstraction (e.g., classification and clustering).

Classification is one of the essential techniques used for abstracting information and can serve a wide variety of thematic applications, such as the separation of different types of land-cover classes in order to understand urban development, mapping, impacts of natural disasters, crop monitoring, tracking, risk management, etc. Despite the efficiency and complexity of the selected classification algorithm, the aforementioned variety and high dimensionality of remote sensing data (e.g., World-View-3 satellite sensor with 0.31m spatial resolution, AISA Dual airborne sensor with 500 bands) can lead to computational and statistical challenges related to the processing scalability and the effectiveness in extracting knowledge. Problems arise for instance when the classifiers require fast and highly scalable implementation in real-time applications (e.g., earthquake scenarios or glacial surges). Traditional desktop approaches (e.g. MATLAB, R, SAS) have several limitations due to the fact that big remote sensing data cannot be stored or processed by algorithms designed for single shared-memory machines. As a consequence, algorithms become able to exploit parallel environments such as HPC clusters, grids, or clouds which provide a tremendous computation capacity and outstanding scalability. However, the heterogeneity of world of parallel environments complicates the user experience. Therefore, a comprehensive solution abridging the gap in knowledge in using the computational resources is required.

Classification algorithms can greatly benefit from an integrated framework in which spatial and spectral information are both included into the analysis. In this study, a two-step approach to classification is considered where the features and the predictive model are computed by two separate algorithms (see the highlighted processing and information abstraction blocks in Fig. 1) The selected algorithms have been developed for running on many-core systems, which enables them to handle large scale datasets [6] and overcome limitations of serial algorithms. In the first step, Morphological Attribute Profiles (APs) compute features that characterize the spatial information within a given image [7, 8]. The APs result from a sequential application of attribute filters based on component trees which can be computed a the shared- and distributed-memory hybrid algorithm proposed by some of the authors [9]. This implementation outperforms traditional serial algorithms whose performances are strongly affected by the size and the quantization of the data. In the second step, the APs serve as the sequential input to the SVMs [3], which are adopted as a classifier. Our implementation of an SVM is based on the parallel implementation πSvM [10] which we improved [6] to make more efficient use of the Message Passing Interface (MPI) for parallel processing. This improved PiSvM [11] offers significant speed-ups for the cross validation, training and testing steps while maintaining the same accuracy as achieved when performing the classification with serial algorithms.

3. WORKFLOW MANAGEMENT

Data analysis of remotely sensed images is composed of multiple phases which typically require separate applications to be executed. Automating the data analysis pipeline in a parallel scientific computing environment without automated workflow management may be difficult to realize, e.g., job submission may be different in every environment. To solve this problem, we use a scientific Workflow Management System to help the user with composing and executing the data analysis steps in a seamless manner.

Remote sensing users intending to manage complex data processing through machine learning tools (such as classification using SVMs) on remote and distributed environments face a couple of issues while accessing massively parallel HPC platforms. As a first step to solve the underlying data analysis challenges, the following requirements have been identified: R1) workflow composition: create and combine tasks which are dependent on each other; R2) task enactment: remote execution of the tasks as they are defined during composition; R3) data access: facilitating data access for each task from different sources and pushing results to data sinks; R4) manage remote execution: capability of monitoring, holding or resuming running workflow tasks; and R5) parametric tasks in order to iterate over n-dimensional values used as input parameters.

Based on the analysis of the above requirements, we took the steps (essentially, an abstract workflow) depicted in Fig. 1 and implemented it as automated scientific workflow using the UNICORE [12] workflow management system. UNICORE is based on multi-tier architecture with servers and clients: the server layer offers a set of web service interfaces to manage remote workflow job submissions and data access on a variety of HPC resource management systems, for instance Torque, SLURM and Load Leveler; the client layer

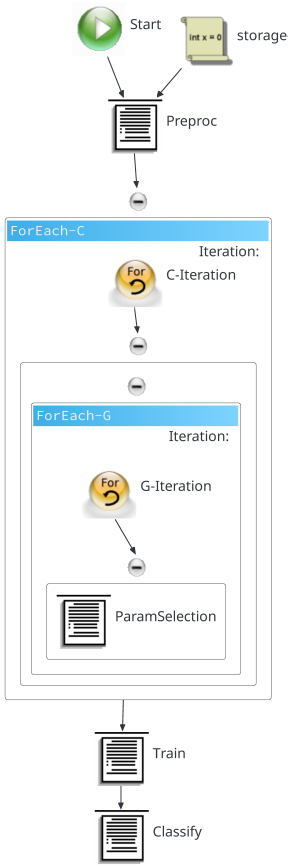

Fig. 2. The classification workflow in the UNICORE Rich Client.

consist of a GUI called the UNICORE Rich Client (URC) and a command line interface.

The purpose of the workflow that we created is the classification of remotely sensed images through using SVMs based on the steps in Fig. 1. In the course of classification, there are multiple steps: pre-processing, cross-validation, model generation and prediction phases. These steps are implemented using the URC through it's visual workflow composer. Fig. 2 shows the classification workflow that we designed and implemented using the URC visual workbench. The details of each step of the workflow are as follows:

1. Preprocessing: It is implemented as a bash script job. Preprocessing uses data already pre-processed through morphological Attribute Profiles (AP) [8]. This step mainly creates a global workflow directory. For each workflow instance, a separate directory is maintained wherein the output of each task of the workflow that successfully finishes gets stored and manages the overall result of the classification.

2. ParamSelection: This step provides the cross validation phase in which the main task is to identify the optimal model parameters C and G that produce the most accurate results. ParamSelection is a parametric step enclosed in a nested iteration in order to achieve a two dimensional For-Each loop. The For-Each loops are implemented through the value-sets feature of the URC. It contains a set of discrete values, each of which will be passed to an individual job by the UNICORE workflow management system. The output of this steps generates the best combination of C and G parameters and stores them in the global workflow directory. ParamSelection use heavily compute and data resources, as each job runs in parallel and use separate input dataset. Details on this selection of the C and G parameters can be found in our earlier work [2].

3. Train: This step take cares of generating or training the model based on best C and G parameters produced in the previous step. The output of this step is the model file. The Train step is also very compute-intensive and requires a parallel computing execution environment.

4. Classify: Classification is the final phase and takes two inputs: the model file generated by the Train step and an unseen image dataset. The main task of this job is to classify the unseen data set. At the end, it produces a unit vector as a text file with classification markers. Also this step takes a considerable amount of computing resources.

Since the workflow management is provided by the UNICORE middleware, it is mandatory to have the UNICORE workflow components deployed. In our case, the UNICORE workflow management services are hosted on the cluster JURECA [13]. Each of the above workflow steps needs a separate executable from the PiSvM application suite which need to be available on the resource executing that step.

In order to run the data analysis workflow on UNICORE, the user has to compose the workflow on URC to obtain the workflow shown in Fig. 1. In addition, the internals of each step specified in the workflow needs to be implemented as a shell script. As part of creating a workflow, the user also specifies for a step what data sets need to be fetched and where the results will be stored, and on which compute resource a step runs on. Once the workflow composition and the underlying steps are configured, the user can simply hit a start button in the URC to submit it to the workflow management service. The URC will take care of the management and monitoring of each step's submission and execution.

The whole data analysis workflow can be exported as reproducable workflow by the URC export mechanism. With this feature, the URC exports a workflow template – in a machine-readable format, which can be easily reused by any other user who runs an URC instance. The only variations that need to be adjusted will be the compute resource selections, which may differ as a new user may not have access to the same computing cluster where the initial results were

produced. The data sets are publicly downloadable for the use case presented in this work; thus, no change to the data sources are required for reproducing the analysis performed by our workflow.

For the experiment the dataset used is a processed hyperspectral data that is made up of 1417617 pixels (with spatial resolution 20 m) and 30 features [6]. In this experiment the workflow was deployed and executed on JURECA [14], on which we have used 1 compute node with 24 cores for each of the workflow jobs. Preprocessing took a fraction of a second, whereas the ParamSelection step ran 25 times, among which the average processing of the cross-validation phase took 14 minutes. The Train step finished in a minute, and Classify completed in 1.5 minutes. The execution times have no difference with the manual configuration if the scripts are well prepared for JURECA, otherwise the execution and monitoring may have a significant time and usability overhead.

4. CONCLUSIONS

Scientific Workflow Managament Systems (WMS) allow to abstract away underlying resource management and application details thus leading to re-usable, understandable, portable and maintainable workflows. For remote sensing image classification using SVMs, so far only the parameter sweep needed of the cross-validation step was automated using a WMS [2]. In this paper, we automate the full chain of classification using SVMs, including training that follows the cross-validation and the classification itself. We developed a re-uable end-to-end workflow based on the UNICORE workflow management system that allows to execute the workflow in an automated and portable way across heterogeneous computing systems. By using the graphical UNICORE Rich Client, the productivity of scientists gets increased. Our experience is that the UNICORE middleware reduces the human efforts in terms of time needed for manual work and understanding the technology adopted. Furthermore, the workflow can be easily adapted to operate on different parallel computing environments as long as they use other, but standards-based middleware. As a future work, we intend to study the impact of our approach in supporting large scale analysis of remotely sensed data using multiple nodes and also analyse deep learning methods for classification.

5. REFERENCES

[1] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu, "Big Data for Remote Sensing: Challenges and Opportunities," *Proc. of the IEEE*, vol. 104, no. 11, pp. 2207–2219, 2016.

[2] M. Shahbaz, G. Cavallaro, M. Riedel, and H. Neukirchen, "Facilitating Efficient Data Analysis of Remotely Sensed Images Using Standards-Based Parameter Sweep Models," in *Proc. IGARSS*, 2017.

[3] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20(3), pp. 273–297, 1995.

[4] M. Shahbaz Memon, A. Shiraz Memon, M. Riedel, B. Schuller, D. Mallmann, B. Tweddell, A. Streit, S. van de Berghe, D. Snelling, V. Li, M. Marzolla, and P. Andreetto, "Enhanced resource management capabilities using standardized job management and data access interfaces within UNICORE Grids," in *Int. Conf. on Parallel and Distributed Systems*. 2007, IEEE.

[5] B. Demuth, B. Schuller, S. Holl, J. Daivandy, A. Giesler, V. Huber, and S. Sild, "The unicore rich client: Facilitating the automated execution of scientific workflows," *2013 IEEE 9th Int. Conf. on e-Science*, vol. 0, pp. 238–245, 2010.

[6] G. Cavallaro, M. Riedel, M. Richerzhagen, J.A. Benediktsson, and A. Plaza, "On understanding big data impacts in remotely sensed image classification using support vector machine methods," *IEEE J. Sel. Topics Appl. Earth Observ.*, vol. 8, no. 10, pp. 4634–4646, 2015.

[7] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Morphological Attribute Profiles for the Analysis of Very High Resolution Images," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 10, pp. 3747–3762, 2010.

[8] G. Cavallaro, N. Falco, M. Dalla Mura, and J. A. Benediktsson, "Automatic attribute profiles," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1859–1872, 2017.

[9] M. Götz, G. Cavallaro, T. Géraud, M. Book, and M. Riedel, "Parallel Computation of Component Trees on Distributed Memory Machines," *IEEE Trans. Parallel Distrib. Syst.*, 2017, press.

[10] D. Brugger, "πSvM," website, 2014, `http://pisvm.sourceforge.net`.

[11] M. Richerzhagen, "piSvM," website, 2015, `https://github.com/mricherzhagen/pisvm`.

[12] A. Streit et al., "Unicore 6 recent and future advancements," *Annals of Telecommunications*, vol. 65, no. 11-12, pp. 757–762, 2010.

[13] D. Krause and P. Thörnig, "JURECA: General-purpose supercomputer at Jülich Supercomputing Centre," *Journal of large-scale research facilities JLSRF*, vol. 2, March 2016.

[14] Jülich Supercomputing Centre, "JURECA," website, 2016, `http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JURECA/JURECA_node.html`.