Abeja VGML: Geometallurgical Applications on Cloud Services

Antonio Barberán[5]*, Alfredo López[5], Carsten Friedrich[5], Felipe Navarro[1,2,4], Carlos González[1, 2,4], Nelson Morales[1,3,4] and Álvaro Egaña[1,2,4]

1. *Mining Departmant, Universidad de Chile*
2. *Advanced Laboratory for Geostatistical Supercomputing (ALGES), Universidad de Chile,*
3. *Delphos Mine Planning Laboratory, Universidad de Chile*
4. *Advanced Mining Technology Center-Universidad de Chile*
5. *CSIRO Mineral Resources, Australia*

ABSTRACT

In recent years, cloud computing has been adopted in many industrial and academic environments, as a way to improve the efficiency of the individual execution times by sharing resources on servers with higher compute and storage capacity. The platform used in this work follows the Virtual Laboratory systems of CSIRO, supported by NeCTAR which is a research cloud computing system based on OpenStack which offer up to 32,000 cores. It can be accessed through a user friendly scientific workflow platform providing geoscientists with an integrated environment that exploits e-Research tools and cloud computing technologies.

Using the Virtual Geophysical Laboratory (VGL) as the main concept, the objective is to provide a system that allows researchers to collaborate and execute processes concerning geometallurgical applications on a web environment, taking advantage of the cloud infrastructure provided by NeCTAR. This system is called Abeja Virtual Geometallurgy Laboratory (Abeja VGML). Abeja was built to support task management, which includes the setup of virtual machines on demand, remote execution of geometallurgical analysis scripts, fault tolerance features and task workflow support.

The system consists of a main server, which is configured as an entry point to receive all task requests. A front-end works as a web application, allowing authenticated users to choose the hardware resources, script parameters, files upload/download files, task execution control, creation of pipelines to run chained geometallurgical processes and to receive real-time notifications and callbacks through a dedicated channel. As a result, the platform has been used as a collaborative tool allowing both CSIRO and University of Chile researchers to develop cloud computing ready applications, allowing to solve complex task workflows through a user friendly and modern web based interface.

INTRODUCTION

Developing a distributed execution environment involves many non-trivial aspects: efficient memory management, parallel/distributed computing, hardware usage, task and big data handling and collaborative development among others. Several software solutions allow software developers to make use of cluster or grid computers to distribute workload and make use of available computational resources [Celery (2016), Gridgain (2016), IPython Parallel (May 2016), Pandas (2016)]. On the other hand, common researchers should not focus on programming strategies but they could benefit from running large numerical models, executing several scenarios in parallel or organizing large calculations in task workflows.

Geometallurgy is one of the disciplines were executing a number of related processes is currently very expensive in terms of time processing and hardware requirements. Solving numerical models involve sequences of interdependent steps (pre-processing, evaluation, reduction, extraction) and generally they are characterized by unknown parameters inferred from input data, which adds undesirable uncertainty to the final models [Deutch, C. V. (2013)]. The latter issue is commonly tackled using conditional simulations, where several scenarios are generated with different setup parameters to address the uncertainty on the estimation of those parameters, leading to heavy computational workloads.

In this paper we present Abeja Virtual Geometallurgy Laboratory (Abeja VGML) a novel platform that exploits eResearch tools and Cloud computing technology in this particular field. The project starts from the collaboration between CSIRO International Research and the University of Chile from the idea of transforming the current CSIRO VGL (Virtual Geophysics Laboratory) [VGL (2016)] into an environment that allows executing on-line processes applied to the Geometallurgy. VGL is a collaboration between CSIRO, Geoscience Australia (GA), and the National Computational infrastructure (NCI) and has been funded by the Federal Government's Education Investment Funds through NeCTAR [Nectar (2016)].

Abeja VGML is equipped with an interface that allows researchers to create and execute chained processes concerning to Geometallurgy, on a web environment, taking the advantage of the cloud infrastructure provided by NeCTAR. The underlying architecture that allows the remote execution will be described in detail in the next section, followed by a case study to illustrate the operation of the entire system.

GEOMETALLURGICAL APPLICATION ON CLOUD SERVICES

Architecture Overview

Abeja VGML is composed by a) a front end, b) a back end and c) the cloud computing. A *main server*, which contains the front end and the back end, receive input data from users and manage the execution of processes. The *cloud* is composed by several amount of virtual machines, preconfigured on-demand to receive and process all the requirements sent by the *main server*.

a. **Front end.** Developed using a combination of programming languages and technologies: HTML5, JavaScript and CSS. It works as a web application, allowing authenticated user to choose the hardware resources, script parameters, file upload/download, creation and configuration of pipelines to run chained geometallurgical processes and to receive real-time notifications from the *back end*.
b. **Back end.** Composed by a Python application server, handles the user requirements and allocate the hardware resources using a specific NeCTAR REST API (novaclient [Novaclient (2016), Openstack (2016)]). When the Virtual Machine (in the cloud) is up and running, the *back end* schedules processes form the users through the Giulietta framework [14]. During the execution and until it is finished, the process will send different notifications and callbacks, which are listened and routed by the *back end* to the *front end*. Finally, after the last notification, the *back end* releases the hardware resources using the API again.
c. **Cloud**. Consist on virtual or real machines that can be configured and used on-demand. In this work, the *cloud* corresponds to NeCTAR cloud computing, which provides a certain amount of virtual machines. This *cloud* is responsible for receiving and executing the processes that are sent by the *back end* on the main server. We use a pre-configured image that contains software requirements for task execution and the same version of Giulietta framework installed on the *back end*.

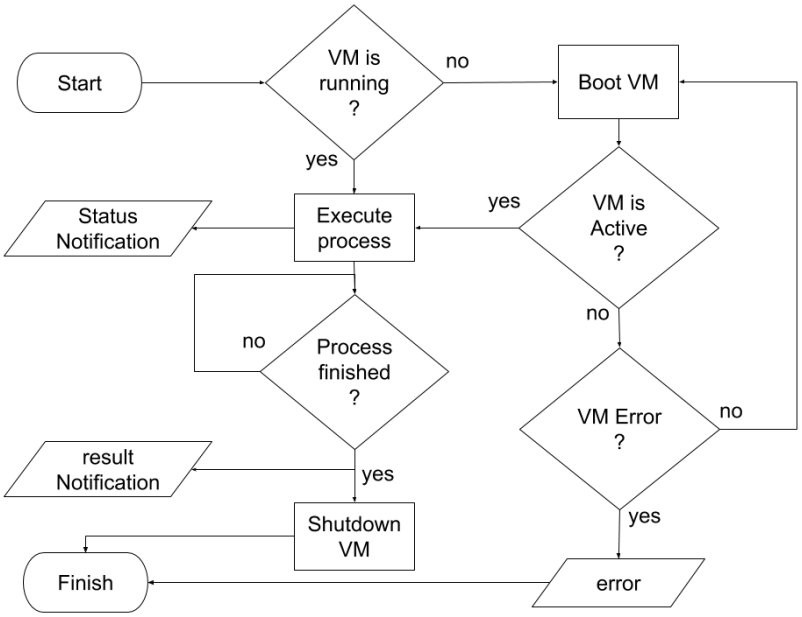

Figure 1 Geometallurgical execution flow on cloud services (NeCTAR)

Process execution

In order to execute applications in the described system, it is necessary to create a pipeline of execution. The users can create one or more processes in each pipeline. Those processes can be executed sequentially or in a distributed way, depending on the user's choice. As a general rule, each process receives a list of the input data (called *data collection*) and as output the process must also return a data collection. Those

data collections can be variable in quantity on each process stages. After every executed stage, data collections can be viewed depending on their datatype (tables, text files, images, graphs, among others). Abeja VGML provides tools for upload/download files and simple visualization tools to access each data types through the *front end* (using a web browser).

After receiving the starting order from the *front end*, the *back end* will extract from the pipeline each process at the time for execution. The *back end* prepares and configures properly a virtual machine (VM), and after is up and running, it will send the process for execution. Input data (as *data collection*) is handled seamlessly using a distributed file system, and it is available on each stage. In case that the process has finished the VM will be dismissed. The described execution flow is showed in Figure 1.

When the user has selected a pipeline for sequential execution, the *back end* will continue operating over the same VM before dismissing it. Figure 2 shows the execution of a sequential pipeline. The *back end* creates a single instance of the process and configures the input data. Each stage of process execution is coordinated by the *back end* using properly message system and their *data collection* is shared through the distributed file system. The processes may return or not a *data collection*, for instance could only create graphs for analysis.

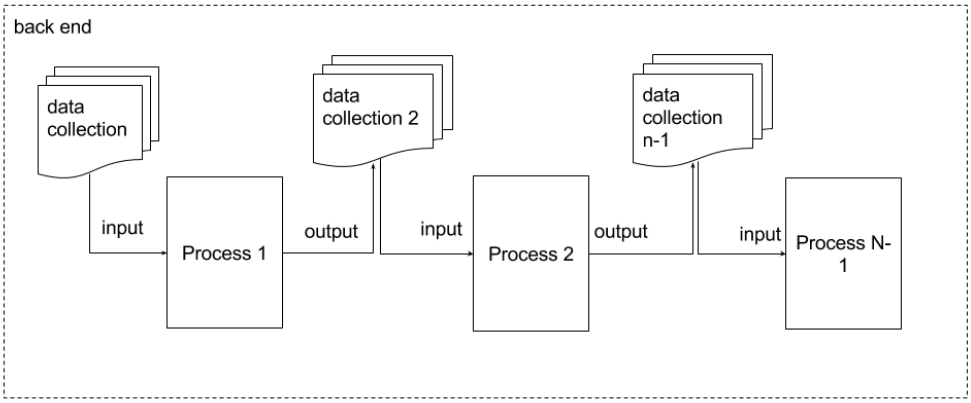

Figure 2 Pipeline sequential execution: back end uses the output data from a process as input data to the next one.

Pipeline distributed execution, is handled by the *back end* with a simple strategy: separates the data and the configuration parameters for the script into a defined number of instances. Each instance will be executed on a dedicated VM or any available cores on already running VM. Figure 3 shows the execution flow for a distributed pipeline. After every script instance is finished, the *back end* can run a final script to do post-process into the output data (as any reduce operation). Finally, after receiving the results notifications (end of processes) the *back end* dismisses the VM asynchronously, freeing resources for other executions.

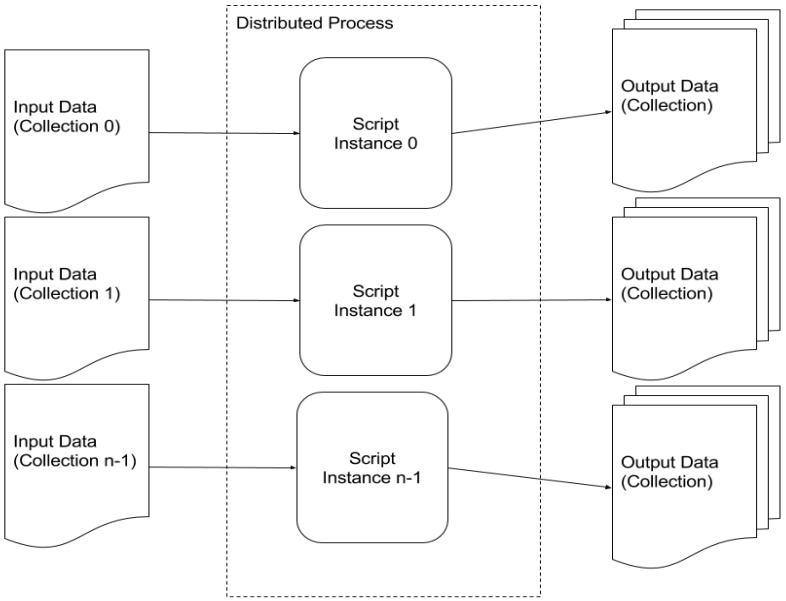

Figure 3 Pipeline distributed execution: back end coordinate the execution of scripts using each data in the collection.

CASE STUDY

The problem of modeling geometallurgical variables is well known as very demanding in time processing and hardware requirements [11]. An example of the workflow for solving numerical models involve sequences of interdependent steps (pre-processing, evaluation, reduction, extraction) as showed in Figure 4. Starting from the original data, different pre-processing and cleaning must be performed in order to continue with the geostatistical modeling to mine planning strategies. Each of those steps has its own difficulties. In this paper we show the execution of the initial stages of the mentioned workflow, with data cleaning and exploratory data analysis tools to make a proof-of-concept with distribution and parallel execution, in order to prepare the whole system to further steps in the geometallurgical workflow.

In most data mining applications, the datasets often contain ambiguous, misleading, missing or duplicate data, which if not treated can spread to the next stages of the process and could lead to poor quality results or misinterpreted conclusions. Preprocessing data tackles these problems and takes an important role in workflows mining [Kotsiantis (2006), Han, J. (2011), Shi, G. (2013), García, S. (2015)]. In geoscience therefore, it is essential to have a good set of tools that ensure a correct preprocessing stage [Shi, G. (2013)]. The proposed case study is focused on three preprocessing tasks created on the described platform, each of which can be used in applications involving geometallurgical data.

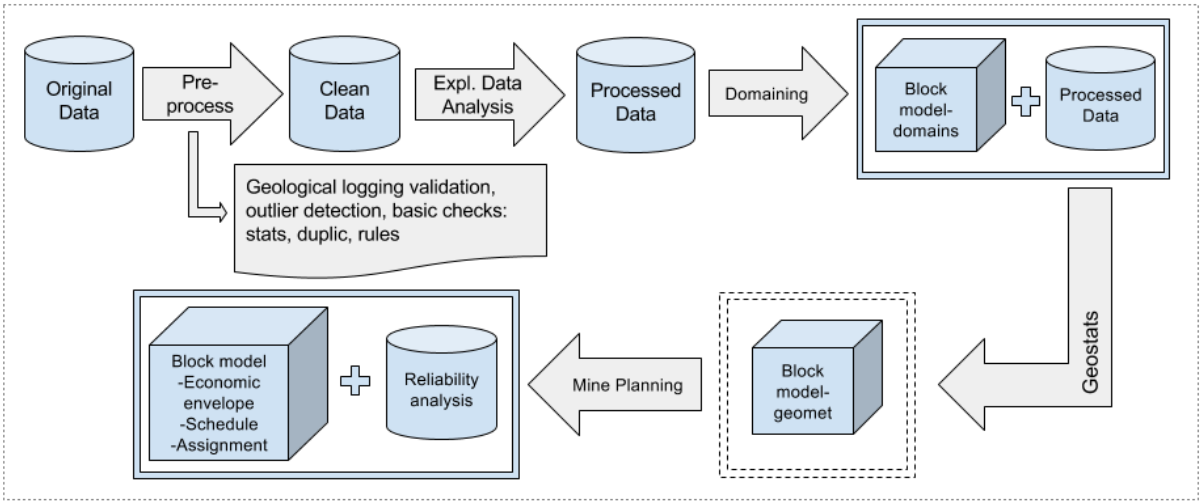

Figure 4 Example of a geometallurgical workflow.

Input data. We consider datasets consisting of collections of values, which can be either numbers (quantitative case) or strings (categorical case), resulting from the observation of several regionalized variables related to spatially distributed rock properties [Matheron, G. (1971)]. As it is standard in data mining applications, these datasets are organized in the form of a data table where the columns correspond to variables and rows correspond to the observations of these variables. Additional information is included as an array of metadata indicating various attributes associated with each variable: source (space coordinates, chemical, hyperspectral, geometallurgical, geological, etc.), type (numeric, categorical, etc.), unit of measurement (percentage, parts per million, etc.) and range of admissible values, among others. In this context, we assume that data-tables contains variables associated to north, east, depth and hole identifier, that determines physical spatial locations and sampling units.

Integration. Is a process to bring both, information from the meta-data, and the values of the data to a consistent and standard format. The process reads the input file, incorporates the metadata information, make some changes in the variable names and add columns if required and reorders the columns of the file. The output of this process is a csv that stores the processed data.

Cleaning. Identify or remove erroneous observations that potentially distort the subsequent analysis. The process reads a csv input file, encoded (with Nan, say) missing and non-admissible values, identify and remove duplicated observations and filter-out useless observations according to given configurations of Nan (ex: at any spatial variable, at all the geochemical, etc.). The output is a csv file storing the cleaned data, a csv file with a summary of the cleaning operation results and several images including histograms highlighting removed and/or marked data.

Imputation. Consists on a set of tools for visualization and imputation of missing data. After reading the input csv file, missing values are fill-in by a predictive mean matching imputation method [Buuren, S., & Groothuis-Oudshoorn, K. (2011)]. The output is one or several completed datasets (csv files) and charts (image files), showing observed, missing and imputed values.

Figures 5 and 6 were obtained after the execution of the pipeline with the three preprocessing tasks, applied to an input data set consisting of observations from 9 geometallurgical variables. Figure 5 shows a scatter matrix of East, North, Depth and Fe, with colored tags according to observed (blue circles) and missing (yellow crosses or vertical ticks) Fe values [Templ, M.(2011)]. It can be noticed that the missing Fe values are uniformly distributed in the spatial domain defined by East, North and Depth variables. Figure 6 illustrates multiple imputation results [Buuren, S., & Groothuis-Oudshoorn, K. (2011)], with 5 imputed datasets. Each panel corresponds to a strip plot of a given variable where observations are colored according to observed/inputted (blue/red) values and shown across the original (value 0) and imputed data sets. Observe that the distribution of observed and imputed values is similar, as it is expected from the method.

The execution of this pipeline in Abeja VGML allowed the user to run the desired task on different scenarios (data sets), having to configure only the parameters, leading to a semi-automated execution. At the end of the pipeline execution, the user receives the proper end-notification and the system is ready to perform the exploratory data analysis.

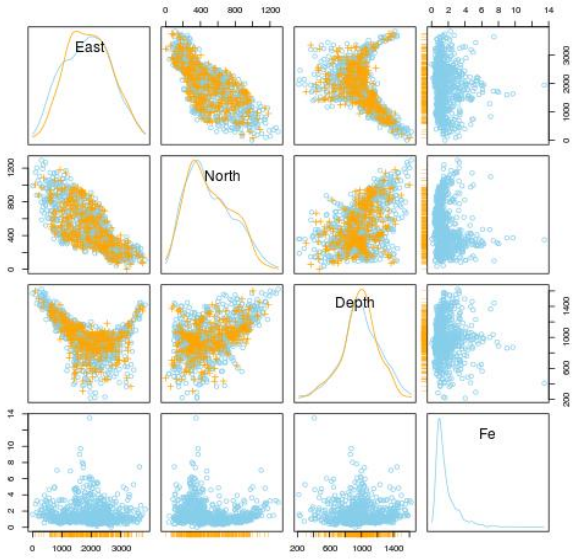

Figure 5 Scatter matrix. Observations with observed and missing Fe values are marked with blue circles and yellow crosses/ticks respectively.

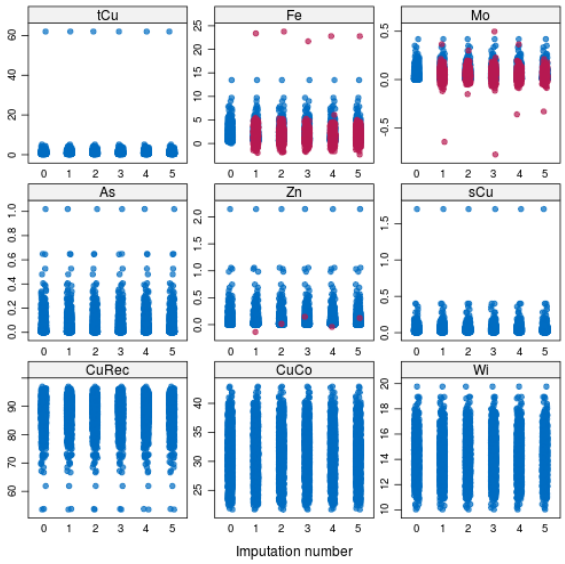

Figure 6 Strip plot of observed (blue) and imputed (red) data.

CONCLUSIONS AND FUTURE WORK

A platform for executing sequential and distributed processes in a cloud ambient, called Abeja VGML, was presented in this paper. The case study was based on automatic pipelined preprocessing steps for geometallurgical input data, to demonstrate general usage capabilities. That exercise showed the flexibility of Abeja VGML to configure a set of automated task, where the user initially setup the input

parameters for each stage and then run the whole process in an unattended way. In the pipeline execution, the platform automatically saves checkpoints allowing the user to reconfigure and re-execute processes in case of errors, resuming from the last successful checkpoint. Giulietta callbacks notifications provide an essential feature in order to achieve a rich user interface during execution. Finally, we show the capability to generate processed results in the form of data files collection and plots (scatter matrix, histograms and strip plots) for further validation.

The simplicity in the usability of the system, gives a new opportunity to semi-automate tasks that traditionally requires a lot of attention and can be very tedious. Abeja VGML helps users to focus on process creating and results analysis instead of details about back end or cloud complexities. With this system, the user can run several scenarios with different parameters each, in order to compare and reduce the uncertainty associated to the geometallurgical modeling.

Nectar was used as a cloud infrastructure; however, Abeja VGML can be adapted to current available cloud services that provide configurable API (E.g., Microsoft Azure, Amazon Web Services and Google Cloud, among others). Distributed and sequential pipeline execution is not exclusive to geometallurgical field; it could be used on applications in other mining areas like resources evaluation or mine planning but also, on other fields demanding high computational resources (E.g., natural sciences, data mining, financial data analysis and telecommunications industry), leading Abeja VGML as a platform with a promising transferable product to industry.

ACKNOWLEDGES

The authors would like to thank the industrial supporters of ALGES and DELPHOS laboratories, as well as the support from the Advanced Mining Technology Center (AMTC) and the whole CSIRO Team.

REFERENCES

Celery (2016), *Distributed task queue*. http://www.celeryproject.org

Gridgain (2016), *In-memory-data-fabric*. http://www.gridgain.com

Nectar (2016), *Research cloud and virtual laboratory*. https://www.nectar.org.au

Novaclient (2016), *A client for the openstack nova API.* http://docs.openstack.org/developer/python-novaclient/api.html

Openstack (2016), *Open source software for creating private and public clouds*. https://www.openstack.org

VGL (2016), *Virtual geophysics laboratory*. https://www.nectar.org.au/virtual-geophysics-laboratory

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised learning. *International Journal of Computer Science*, 1(2), 111-117.

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco.

Shi, G. (2013). *Data mining and knowledge discovery for geoscientist*. Elsevier.

García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining*. New York: Springer.

Deutch, C. V. (2013). Geostatistical modelling of geometallurgical variables-problems and solutions. Melbourne, In *The Second AusIMM International geometallurgy Conference*.

IPython Parallel (May 2016), *Using IPython for parallel computing*. https://ipython.org/ipython-doc/3/parallel/parallel_intro.html

Pandas (2016), *Python data analysis library*. http://pandas.pydata.org/

Navarro, F., González, C., Peredo, Ó., Morales, G., Egaña, Á., Ortiz, J. M. (2014). A Flexible Strategy for Distributed and Parallel Execution of a Monolithic Large-Scale Sequential Application. In *High Performance Computing* (pp. 54-67). Springer Berlin Heidelberg.

Matheron, G. (1971). *The theory of regionalized variables and its applications* (Vol. 5, p. 211pp). École national supérieure des mines.

Templ, M., Kowarik, A., Filzmoser, P., & Alfons, A. (2011). A computational and methodological framework for visualization and imputation of missing values: the R package VIM.

Buuren, S., & Groothuis-Oudshoorn, K. (2011). Mice: Multivariate imputation by chained equations in R. *Journal of statistical software*, *45*(3).