Large-Scale Data Analysis on Cloud Systems

by Fabrizio Marozzo, Domenico Talia and Paolo Trunfio

The massive amount of digital data currently being produced by industry, commerce and research is an invaluable source of knowledge for business and science, but its management requires scalable storage and computing facilities. In this scenario, efficient data analysis tools are vital. Cloud systems can be effectively exploited for this purpose as they provide scalable storage and processing services, together with software platforms for developing and running data analysis environments. We present a framework that enables the execution of large-scale parameter sweeping data mining applications on top of computing and storage services.

The past two decades have been characterized by an exponential growth of digital data production in many fields of human activity, from science to enterprise. In the biological, medical, astronomic and earth science fields, for example, very large data sets are produced daily from the observation or simulation of complex phenomena. Unfortunately, massive data sets are hard to understand, and models and patterns hidden within them cannot be identified by humans directly, but must be analyzed by computers using knowledge discovery in database (KDD) processes and data mining techniques.

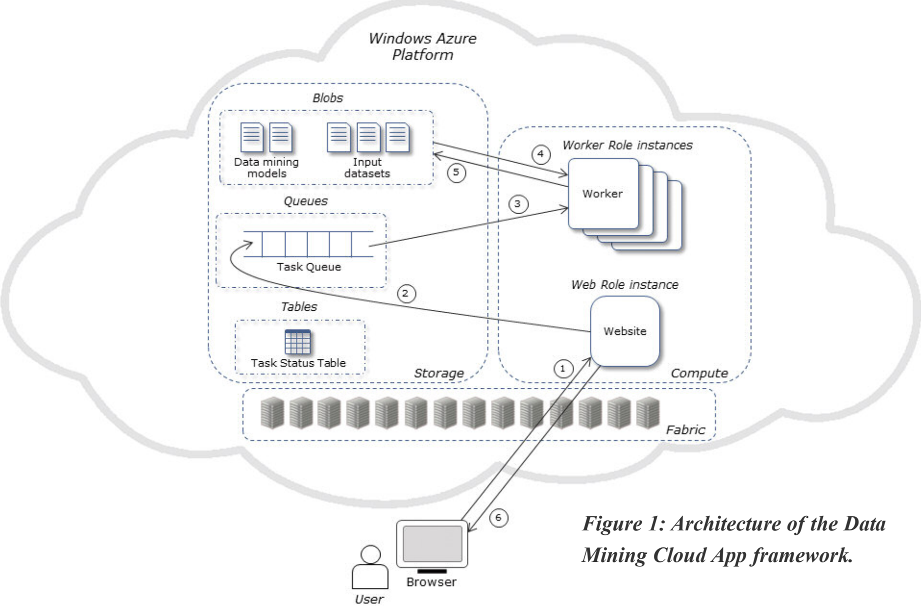


Figure 1: Architecture of the Data Mining Cloud App framework.

Data analysis applications often need to run a data mining task several times, using different parametric values before getting significant results. For this reason, parameter sweeping is widely used in data mining applications to explore the effects of using different values of the parameters on the results of data analysis. This is a time consuming process when a single computer is used to mine massive data sets since it can require very long execution times.

Cloud systems can be effectively employed to handle this class of application since they provide scalable storage and processing services, as well as software platforms for developing and running data analysis environments on top of such services.

We have worked on this topic by developing Data Mining Cloud App, a software framework that enables the execution of large-scale parameter sweeping data analysis applications on top of Cloud computing and storage services. The framework has been implemented using Windows Azure and has been used to run large-scale parameter sweeping data mining applications on a Microsoft Cloud data centre.

Figure 1 shows the architecture of the Data Mining Cloud App framework, as it is implemented on Windows Azure. The framework includes the following components:

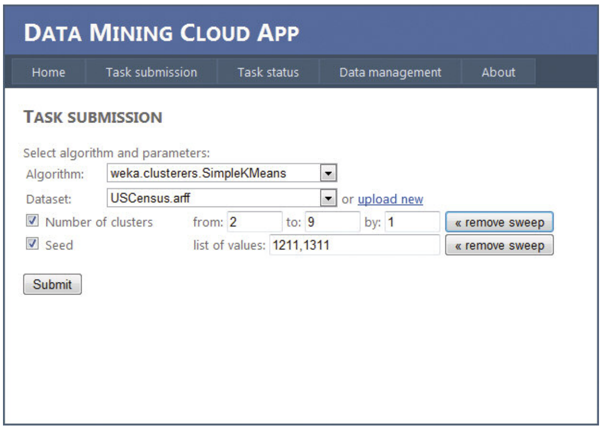
- a set of binary and text data containers (Azure blobs) used to store data to be mined (input datasets) and the results of data mining tasks (data mining models)
- a task queue that contains the data mining tasks to be executed
- a task status table that keeps information about the status of all tasks
- a pool of k workers, where k is the number of virtual servers available, in charge of executing the data mining tasks submitted by the users
- a website that allows users to submit, monitor the execution, and access the results of data mining tasks.

The website includes three main sections: i) task submission that allows users to submit data mining tasks; ii) task status that is used to monitor the status of submitted tasks and to access results; iii) data management that allows users to manage input data and results.

Figure 2 shows a screenshot of the task submission section of the website, taken during the execution of a parameter sweeping data mining application. An application can be configured by selecting the algorithm to be executed, the dataset to be analyzed, and the relevant parameters for the algorithm. For parameter sweeping applications, the system submits to the Cloud a number of independent tasks that are executed concurrently on a set of virtual servers.

The user can monitor the status of each single task through the task status section of the website, as shown in Figure 3. For each task, the current status (submitted, running, done or failed) and status update time are shown. Moreover, for each task that has completed its execution, the system enables two links: the first (Stat) gives access to a file containing some statistics about the amount of resources consumed by the task; the second (Result) visualizes the task result.

We evaluated the performance of the Data Mining Cloud App through the execution of a set of long-running

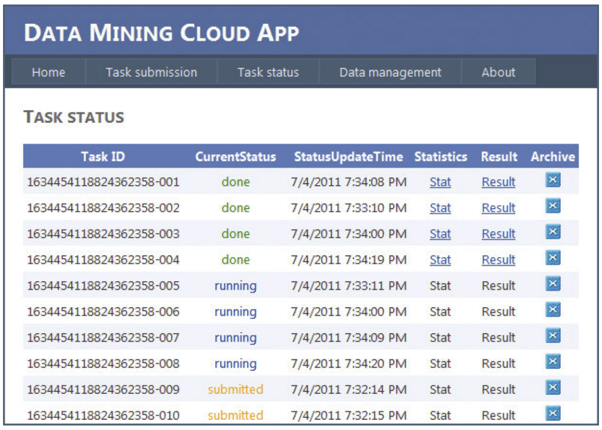


TASK SUBMISSION

Select algorithm and parameters:
Algorithm: weka.clusterers.SimpleKMeans
Dataset: USCensus.arff or upload new
☑ Number of clusters from: 2 to: 9 by: 1 « remove sweep
☑ Seed list of values: 1211,1311 « remove sweep

Submit

Figure 2: Data Mining Cloud App website: A screenshot from the task submission section.



TASK STATUS

Task ID	CurrentStatus	StatusUpdateTime	Statistics	Result	Archive
1634454118824362358-001	done	7/4/2011 7:34:08 PM	Stat	Result	☒
1634454118824362358-002	done	7/4/2011 7:33:10 PM	Stat	Result	☒
1634454118824362358-003	done	7/4/2011 7:34:00 PM	Stat	Result	☒
1634454118824362358-004	done	7/4/2011 7:34:19 PM	Stat	Result	☒
1634454118824362358-005	running	7/4/2011 7:33:11 PM	Stat	Result	☒
1634454118824362358-006	running	7/4/2011 7:34:00 PM	Stat	Result	☒
1634454118824362358-007	running	7/4/2011 7:34:09 PM	Stat	Result	☒
1634454118824362358-008	running	7/4/2011 7:34:20 PM	Stat	Result	☒
1634454118824362358-009	submitted	7/4/2011 7:32:14 PM	Stat	Result	☒
1634454118824362358-010	submitted	7/4/2011 7:32:15 PM	Stat	Result	☒

Figure 3: Data Mining Cloud App website: A screenshot from the task status section.

parameter sweeping data mining applications on a pool of virtual servers hosted by a Microsoft Cloud data center. The experiments demonstrated the effectiveness of the Data Mining Cloud App framework, as well as the scalability that can be achieved through the parallel execution of parameter sweeping data mining applications on a pool of virtual servers. For example, the classification of a large dataset (290,000 records) on a single virtual server required more than 41 hours, whereas it was completed in less than three hours on 16 virtual servers. This corresponds to an execution speedup equal to 14.

Other than supporting users in designing and running parameter sweeping data mining applications on large data sets, we intend to exploit Cloud computing platforms for running knowledge discovery processes designed as a combination of several data analysis steps to be run in parallel on Cloud computing elements. To achieve this goal, we are currently extending the Data Mining Cloud App framework to also support workflow-based KDD applications, in which complex data analysis applications are specified as graphs that link together data sources, data mining algorithms, and visualization tools.

Links:
http://www.microsoft.com/windowsazure
http://grid.deis.unical.it

Please contact:
Domenico Talia
ICAR-CNR and
DEIS, University of Calabria, Italy
Tel: +39 0984 494726
E-mail: talia@deis.unical.it

Fabrizio Marozzo and Paolo Trunfio
DEIS, University of Calabria, Italy
E-mail: fmarozzo@deis.unical.it,
trunfio@deis.unical.it

Big Software Data Analysis

by Mircea Lungu, Oscar Nierstrasz and Niko Schwarz

In today's highly networked world, any researcher can study massive amounts of source code even on inexpensive off-the-shelf hardware. This leads to opportunities for new analyses and tools. The analysis of big software data can confirm the existence of conjectured phenomena, expose patterns in the way a technology is used, and drive programming language research.

The amount and variety of available external information associated with evolving software systems is staggering: data sources include bug reports, mailing list archives, issue trackers, dynamic traces, navigation information extracted from the IDE, and meta-annotations from the versioning system. All these sources of information have a time dimension, which is tracked in versioning control systems.

Software systems, however, do not exist in isolation but co-exist in larger contexts known as software ecosystems. A software ecosystem is a group of software systems that is developed and co-evolves together in the same environment. The usual environments in which ecosystems exist are organizations (companies, research centres, universities) or communities (open source communities, programming language communities). The systems within an ecosystem usually co-evolve, depend on each other, have intersecting sets of developers as authors, and use similar technologies and libraries. Analyzing an entire ecosystem entails dealing with orders of magnitude more data than analyzing a single system. As a result, analysis techniques that work for the individual system no longer apply.

Recently, we have seen the emergence of a new type of large repository of information associated with software systems which can be orders of magnitude larger than an ecosystem: the super-repository. Super-repositories are repositories of project repositories. The existence of super-repositories provides us with an even larger source of information to analyze, exceeding ecosystems again by orders of magnitude.