Workflow Coordinated Resources Allocation for Big Data Analytics in the Cloud

Niki Sfika, Konstantinos Manos, Aigli Korfiati, Christos Alexakos, Spiridon Likothanassis

Workflow coordinated Resources allocation for Big Data Analytics in the cloud

N. Sfika, K. Manos, A. Korfiati, C. Alexakos and S. Likothanassis

Pattern Recognition Laboratory, Dept. of Computer Engineering and Informatics, University of Patras, Rio, 26500, Greece
{sfika,manosk,korfiati,alexakos,likothanassis}@ceid.upatras.gr

Abstract. Cloud computing consists of a set of new technologies that permit the dynamic allocation of computational resources (storage, CPU, memory) when performing high demanding data analysis. In the modern world of information data, cloud computing can provide valuable solutions for the Big Data Analytics domain. The correct allocation of resources in a Big Data analysis problem can both increase performance and decrease cost. This article proposes a system architecture for allocating computational resources according to the problem demands in a cloud infrastructure. Workflows are utilized in order to coordinate the execution of complex data analysis pipelines.

Keywords: cloud computing, workflows, resources allocation, big data analytics

1 Introduction

Cloud computing involves a set of new technologies that permit the dynamic allocation of computational resources (storage, cpu, memory). This is extremely useful, especially when performing high demanding data analysis. In the modern world of information, there are a lot of sources (web, open data databases, IoT) that lead to the collection of huge amounts of data [1]. Big Data Analytics is the business and informatics domain, responsible for processing such large capacity of data. Problems in this domain require, in most cases, a lot of computational resources, which can be provided by the cloud on demand [2]. Moreover, Big Data Analytics solutions are based on complex pipelines where several tasks must be performed by different systems in a specific order. Some tasks may also require parallelization and/or distribution on computer systems clusters in order to be executed. On the other hand, the notion of workflows is not new for the representation of complex jobs. Workflows are considered as powerful tools for enabling the composition and execution of complex analysis jobs in distributed environments [3]. Initially, workflows were used to represent in a formal way the business processes, but in the last years they have been widely used for the orchestration of the execution of complex processes in the field of integration of information systems [4]. Now, workflows are able to provide a reliable

solution that encompasses all the steps of Big Data Analytics, from data access and filtering to data mining and processing for extracting knowledge [5].

Cloud computing is a relatively new and prominent set of technologies in the area of ICT and it is anticipated to play a key role in the modern information systems. In cloud computing, all available computational resources (such as networks, storage, applications, servers, etc.) are provided via web as utilities. The main difference of cloud computing against traditional approaches can be summarized to the on-demand access to the resources pool and the flexible and adaptable resources provision management by the cloud provider. In cloud computing, users can increase or decrease the capacity and consume as many resources as they need for their applications. Furthermore, the technologies composing cloud computing can be combined in order to provide highly scalable and adaptable applications owing to the fact that new applications and features can be deployed significantly faster. Another worth mentioning characteristic of cloud computing is that all the provided resources are hosted on remote server infrastructures, allowing them to be accessed remotely as long as there is Internet connection and users and systems can collaborate simultaneously on these resources [6].

Big Data Analytics is an emerging field and it has been applied to various areas such as Business Intelligence and Analytics (BI&A), scientific problem solving and bioinformatics. Especially in the field of BI&A there are several proposed applications and case studies on a) ecommerce and market intelligence, b) e-government and politics, c) science and technology, d) smart health and well-being, and e) security and public safety [7]. Additionally, in the field of bioinformatics, the large amount of biological data and information that has been gathered or generated the last decade is situated in different databases and thus, Big Data Analytics is considered to be able to give both a solution for the information integration as well in its analysis [8]. Variety, velocity, scaling, complexity, interpretation and security problems with Big Data raise challenges at all phases of the data analysis pipeline [9]. Especially, in problems of data mining the analysis pipeline is complex including tasks for data collection and integration from different sources, data preprocess and cleaning and finally data analysis using various techniques as machine learning, statistics or heuristics methods [10].

In the current article, we are focusing on the problems of scaling and complexity in the Big Data analysis. The proposed architecture adds a level of calculation of the required computational resources based on the problem input and their reservation in a cloud infrastructure. The proposed system manages the creation/deletion of the Virtual Machines (VMs) that will finally execute the analysis and ensures that the minimum required resources in terms of performance and cost will be used. Moreover, the system employs workflows in order to face the challenges of the complexity of the Big Data analysis pipeline. Users can define with a visual tool the flow of the analysis tasks easily, decreasing the complexity for developing complex programs code.

The proposed approach is presented in detail in the rest of the paper. In Section 2 some significant technologies and related work in the area of cloud computing, workflows and Big Data Analytics are presented. Section 3 describes the basic functional components of the proposed architecture and Section 4 presents in details the utiliza-

tion of workflows for a) resources allocation and b) Big Data Analytics processes execution. Finally, Section 5 concludes the paper.

2 State of the Art Technologies

2.1 Cloud computing and resources allocation

As mentioned above, Cloud Computing is a set of emerging technologies giving end-users access to all types of computational resources and the ability to consume them according to their demands. The efficient management of the actual hardware resources (storage, memory, processors and bandwidth) on virtualization environments concludes to a more efficient cost model on the cloud [11].

Cloud Computing is a general term denoting various levels of resources provision as services to the end-users. In an attempt to categorize the services provided by the cloud, the National Institute of Standards and Technology (NIST) of USA proposed three main delivery models [12]:

1. **Infrastructure-as-a-Service (IaaS)** model offers on demand virtual machines with their own operating system, storage and network,
2. **Platform-as-a-Service (PaaS)** model offers functionalities on the computational resources as Application Programming Interfaces (APIs) for new applications development ,
3. **Software-as-a-Service (SaaS)** model offers applications as a service. Users of SaaS applications are able to utilize their functionalities through a front-end web portal without additional hardware requirements in their local device.

The full installation of a cloud infrastructure requires the installation of software for managing the IaaS infrastructure. This software is often referred as cloud computing platform and its primary scope is the management of the virtualization infrastructure of the bare-metal servers. Some of the most commonly used open-source platforms are OpenNebula, Nimbus, Openstack and Cloudstack. Nevertheless, there are commercial IaaS vendors, such as Amazon WS (Amazon.com, Inc., WA, US), Google Cloud (Google Inc., CA, US), Rackspace (Rackspace Inc., TX, US) and Azure (Microsoft Corporation, WA, US).

In the proposed approach, the cloud platform used in order to provide cloud services infrastructure was Apache Cloudstack. Cloudstack is an open source cloud computing platform that controls and manages computational resources, storage and network to deploy a private or a public IaaS cloud [13]. It provides a flexible cloud orchestration for the developer. For the virtualization environment, our infrastructure is based on a Xen Hypervisor delivered from the open-source XenServer (Citrix Systems, Inc., FL, US). XenServer is a virtualization platform which practically includes a hypervisor providing the appropriate isolation between the running virtual machines (VMs) [14]. Cloudstack offers a web graphical user interface (GUI) for both administrators and end-users to manage the cloud. Using the GUI, users can manage all provisioned virtual machines and networks. Another important functionality is Cloud-

stack's HTTP API which can be used for the execution of any required operation by third-party applications. Following the basic architecture of most cloud computing platforms, CloudStack consists of two major components: a) the resources to be managed and b) the management server which must be informed about all these resources.

The management server orchestrates and allocates the declared resources in the cloud infrastructure. It allocates virtual machines to hosts and controls the assignment of storage and IP addresses to each virtual machine instance. It provides a series of options through the GUI and the API interfaces such as private/public networks, templates and snapshots management.

The host, being a single machine, provides its resources, such as memory, CPU, storage and network, in order for the virtual machines hosted on it to run. A hypervisor software (XenServer in our implementation) is installed in the host for managing the guest VMs and the virtual networks. If more capacity is needed for the guest VMs, a supplementary host can be added. What is important is that CloudStack has the ability to detect the amount of all the aforementioned resources automatically. Finally, the actual allocation of the resources on the hardware infrastructure is information completely unsighted to the end-users.

2.2 Workflow for services orchestration

A workflow is an orchestrated sequence of tasks describing the execution of a specific scientific or business process, and the exchange of information between them. Each task consists of a set of defined actions executed in a specified order. After a workflow starts operating, work and data pass through each step from start to finish until the process is complete. Usually, the creation and operation of a workflow is assigned to a Workflow Management System (WFMS) that handles the invocation and coordination of its various components [15]. Workflow applications have steadily increased in complexity and variation leading to the need of more flexible workflow systems able to handle and organize a large set of often incompatible resources [16].

Workflows present a technology to achieve this supporting automation in the definition and execution of complex and integrated tasks orchestrating the invocation of web services (WSs) provided by different systems. The above approach, combining broadly accepted WSs and workflows, has mandated WS description in terms of workflow relevant standards. These include among others [17], the Web Services Business Process Execution Language (WS-BPEL) that follows the Business Process Model and Notation (BPMN) [18], and the Web Services Choreography Description Language (WS-CDL) [19]

The workflows depict processes that are being executed and coordinated by Workflow Management Systems. The last decade, workflows are widely used in the interoperability of information systems for integrating complex business processes. Thus, some of the well-known software vendors have introduced Workflow Management Systems (WMS), such as IBM BPM, Microsoft Windows Workflow Foundation, and SAP Business Workflow. Also, open-source projects have followed this trend, such as Apache ODE, jBPM, YAWL and Taverna. The later was used in the proposed approach because of its simplicity in installation and administration.

Taverna [20] is an open source and domain-independent Workflow Management System. It provides the infrastructure needed to design and run workflows, monitor the execution process, and manage the handling and coordination of a variety of different workflow services, while pipelining the data processing. Due to its flexibility, it can interact with a large variety of resources and tools, and can be used to create workflows that apply to a wide range of scientific fields. Among other things Taverna has been designed to offer automated communication and invocation of several components coming from different service providers, efficient handling of data exchange between different types of applications and easy data conversion in order to avoid compatibility issues between services. It also offers a better insight into the operating procedure through a detailed view of each component's interactions and results.

2.3 Big Data Analytics in the cloud

Applications for Big Data analytics are tightly connected to the process workload characteristics, such as scale, scope and nature. These characteristics provide the principal requirements to the future hardware and software [21]. Thus, the on demand provision of both hardware and software as a service from cloud providers makes cloud computing sufficient to be used as infrastructure for Big Data Analytics applications.

Scalable and distributed management of both process tasks and stored data for large amount of data has been a challenge for the research community for more than three decades. Today, the researchers focus on the design and development of systems that serve both intensive and ad-hoc analysis workloads [22]. There are approaches in the direction of automatic scaling of cloud-based applications. Vaquero et al. [23] proposed a holistic approach for cloud applications scalability in both IaaS and PaaS models. Their approach is based on the overall management of collections of interrelated and context-dependent VMs by using policies and rules. Ming and Mao [24] presented an approach for scheduling the execution of data analysis problems. In their approach the basic computing elements are VMs that are characterized by sizes and costs. The jobs are depicted as workflows and the final goal is to ensure all jobs are finished at minimum financial cost within their deadlines. User requirements are considered as deadlines. Another approach that incorporates the Handoop infrastructure for computer clustering and distributed analysis of Big Data is Starfish. Starfish is a self-tuning system for Big Data analytics, which automatically adapts to user needs and system workloads in a way transparent to the user [25].

3 System Architecture

The proposed architecture aims to provide users with the flexibility of defining their pipeline analysis through a user-friendly workflow editor - the Taverna Workbench – and to automatically scale the execution of this workflow by calculating the required resources and managing the creation/deletion of the appropriate VMs. The system architecture consists of a series of components and it is based on the IaaS

cloud service model. It is designed in such a way, in order to achieve high performance in terms of time and cost saving. The proposed architecture provides an automated process execution without user's interference, efficient communication between the components and an easily manageable graphical user interface for the users to set a problem's specifications and receive its results. Figure 1 depicts the functional components of the proposed architecture along with their interactions.

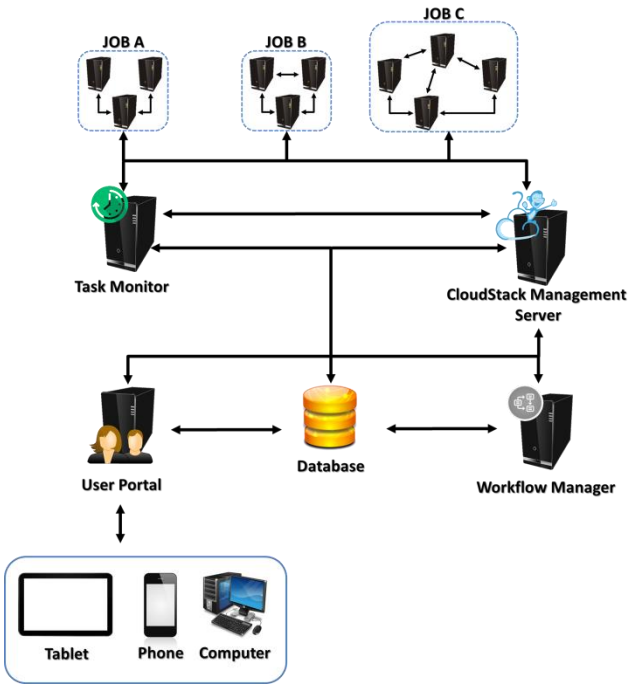

Fig. 1. The proposed system architecture

The **User portal** is a web-based graphical user interface for the communication between the user and the system. It is developed following responsive design techniques and it is accessible from a variety of devices such as smartphones, tablets and PCs. Users can define new jobs by selecting the appropriate workflow from a list of predefined ones created off-line in the Taverna Workbench. The User portal also supports a mechanism for user authentication and permissions. Defined jobs, their parameters and user information and credentials are kept to a central relational Database (MySQL in our approach).

The **Database** is the module of the architecture that holds all the information such as the number of problems, their results and specifications, the number of users and their details etc. It can be accessed by the user portal, the workflow manager and the task monitor in order to store their information.

The **CloudStack management server** is the software module of Cloudstack platform with which the workflow manager and monitor communicate. It receives re-

quests for VM administration tasks through an HTTP API. Its API is used from the other architecture's modules for either VMs creation/deletion or polling information for the running VMS.

The **Workflow manager** is responsible for the execution of the appropriate workflow when a new analysis job appears. It communicates with the database iteratively to receive details about new defined jobs. The actual execution of the workflow is performed on the Taverna Server that is part of the Workflow manager. The monitoring of the workflow's execution is performed by the HTTP API provided by the Taverna Server.

The **Task Monitor** manages the communication between the system and the VMs that are created to execute the actual analysis pipeline. More precisely, it receives a process's progress from the VMs through a series of messages. Upon a message arrival indicating the end of a process, it updates the database by marking this job as completed. The task monitor is based on the Advanced Message Queuing Protocol (AMQP) [26].

4 Workflows for Resources Allocation and Data Analysis

4.1 Resources Allocation Workflow

The Resources Allocation Workflow contains the core workflow that depicts the initial steps that must be made in order to calculate the computational resources needed for the data analysis problem and the instantiation of the virtual infrastructure. The instantiation task contains all the appropriate steps for creating the VM instances and initializing them as depicted in Figure 2.

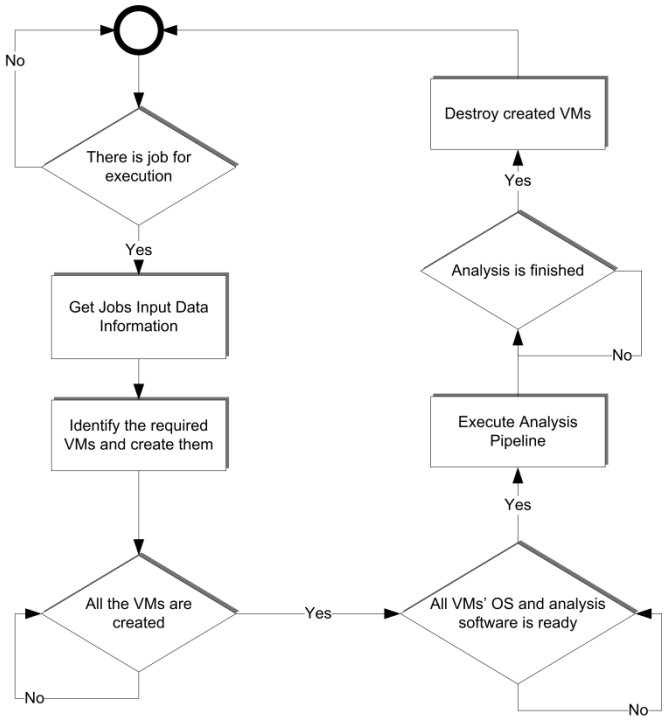

Fig. 2. The task of resources allocation and job analysis initialization

Initially, the workflow receives the data needed for both the orchestration process and the calculations within the VMs. The first step is the calculation of the appropriate computational resources (CPU speed, number of CPUs, RAM size etc.) required for each VM, depending on the size of the input data. This is achieved by running a python script that identifies the size of the problem and chooses the appropriate service offerings. An empirically specified rule that performs the mapping between the problem size and the necessary resources is employed. For example, when executing the problem addressed in [27] the resulting rule is depicted in Table 1:

Table 1. Empirical rule for the resources allocation for a specific test case

Case	Problem size	Resources
1	<= 75000	1 core, 2 GHz and 1 GB RAM
2	(75000, 150000)	1 core, 2 GHz and 2 GB RAM
3	>=150000	1 core, 2 GHz and 4 GB RAM

All python scripts, like this one, are run with the use of the Taverna "Tool" service which allows the execution of commands locally on the machine the workflow is run on (Workflow manager). The outcome of this procedure is the number of the VMs and the template (predefined data analysis software) and the service offerings (operat-

ing system, memory, CPU and storage) for each VM. Figure 3 depicts the segment of the workflow that invokes the script for the calculation of the requested resources.

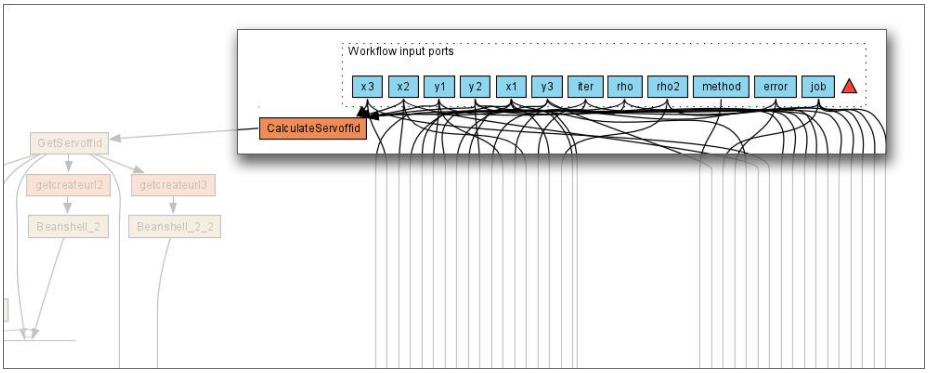

Fig. 3. Workflow segment for calculating the required resources regarding input data

The next step in the workflow is to invoke the appropriate HTTP methods of the Cloudstack Server API for creating the calculated VMs. Before a HTTP call is issued during the workflow execution, a python script must be run, with inputs from previous steps, to generate the URL required for that specific action, like deleting VMs. This applies to all HTTP API requests to Cloudstack. In order to ensure that the VMs are indeed running before moving to the next step a loop structure is used in the workflow that continuously requests new VMs status and stops when all the VMs are up and running. In Figure 4 the workflow loop structure is denoted with a double-line rectangular.

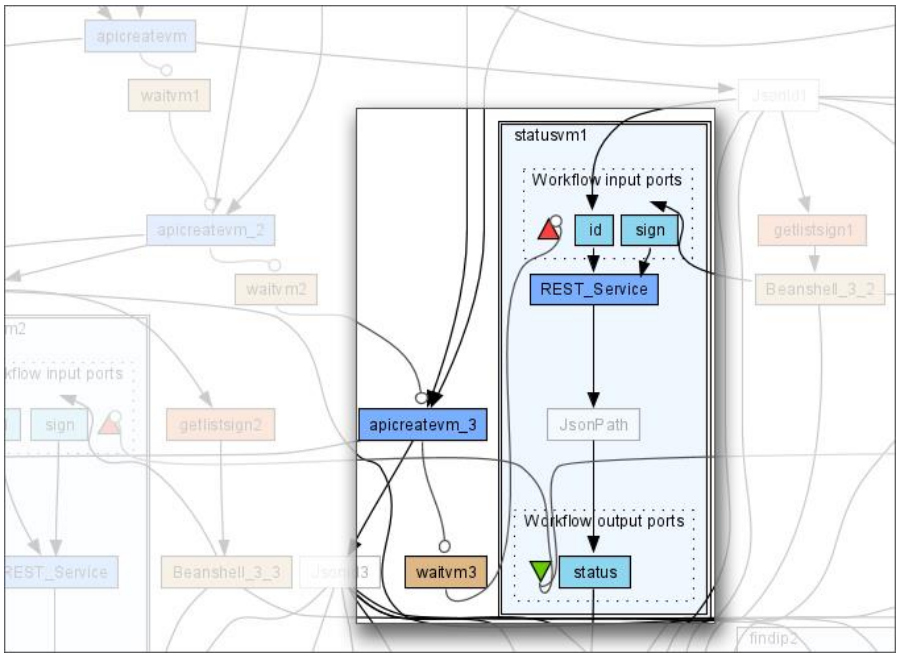

Fig. 4. Workflow segment for creating new VM

Before starting the execution of the analysis workflow, it is necessary that the VMs have been booted and are ready to use. In this direction, each VM template is empowered with a REST service that checks if all the services needed from the analysis software are up. As above, a loop structure is used for requesting the status of the operating system and the analysis software that is installed in the new VM (Fig. 5).

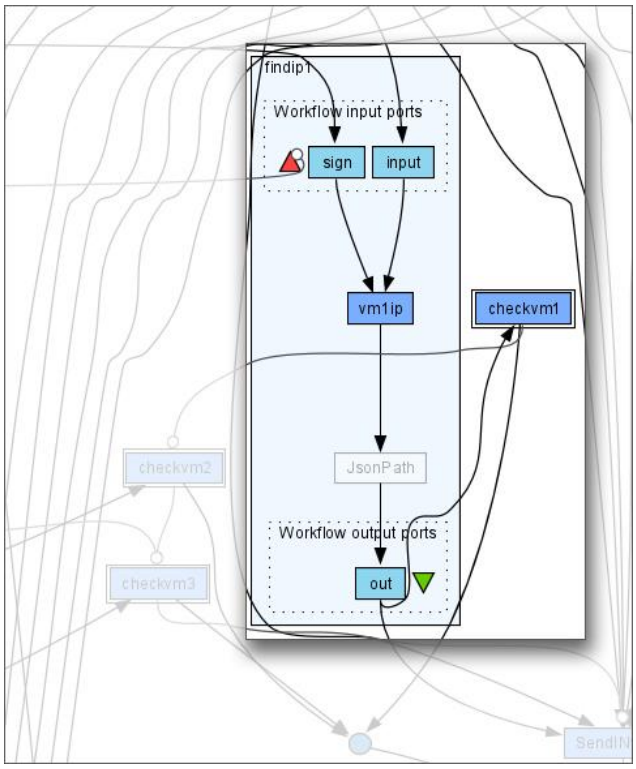

Fig. 5. Workflow segment for ensuring that VM's software is ready

After the Workflow manager ensures that the VMs are booted and the software is ready, it executes the Analysis Workflow, which is presented in the next subsection. When the analysis pipeline is finished, the Task Monitor is responsible for updating the Database with the results and marking the analysis job as completed. Once again a script is used to communicate with the Database and wait until that final status change takes place. Then the VMs can be deleted. First, the VMs must be stopped. A workflow loop structure is also used in this case to check VMs' status until they shut down. After, a HTTP call is made to Cloudstack to delete the stopped VMs (Fig. 6).

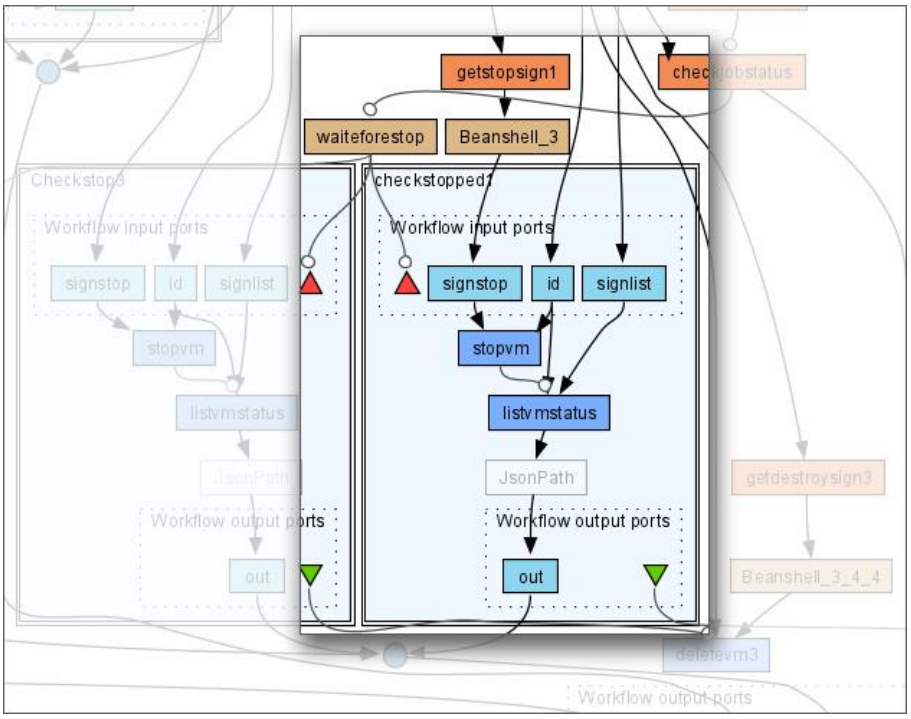

Fig. 6. Workflow segment for stopping and destroying VMs

4.2 Analysis Workflow

The analysis workflow is the segment of the entire workflow that is responsible for executing the analysis pipeline. Its structure depends on the complexity of the data analysis and the involved steps. As example, in Figure 7, we present a small analysis workflow with three VMs that are solving in parallel large-scale differential equations using the Interface Relaxation methodology [28]. The computational resources and the number of VMs are tightly related to the size of the input. The resources allocation method that was used in the first step of the Resources Allocation Workflow has been tested in [27] providing significant performance optimization, especially in large datasets.

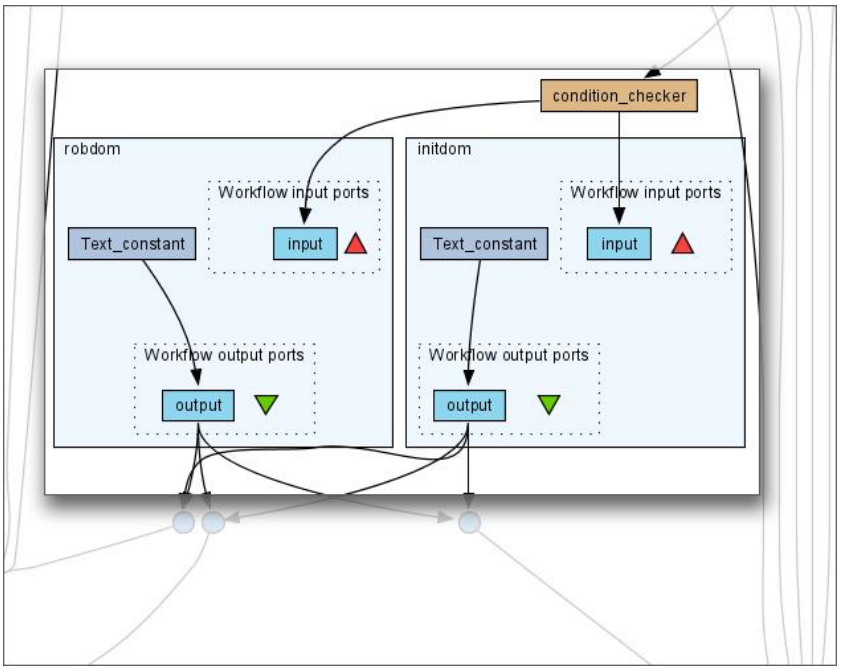

Fig. 7. Analysis workflow example

4.3 Workflow Execution

The overall workflow described in the previous paragraphs has been designed using the Taverna Workbench. The outcome is depicted in Figure 8. The Taverna workflow was stored in T2flow file format in order to be loaded in the Taverna Server that acts as the Workflow manager of the proposed architecture.

The execution of the workflow is handled by the Taverna Server. The Taverna Server is running on a VM on Cloudstack and can have local access to all VMs used during the problem execution. The whole communication with the Taverna Server is performed through REST API calls. More specifically there are three steps needed in order to initiate and run a workflow.

- POSTing a T2flow document describing the desired workflow process. A workflow id is returned that allows us to handle further requests.
- PUTing all input information and data needed by the workflow in their respective locations, as these are described by the workflow's input ports.
- PUTing the value 'Operating' to the workflow's status.

After the execution of the workflow has begun, the Task Manager will periodically check the workflow's status through a HTTP request, until it has the value 'Finished'. This means that the execution of the problem has been completed and the VMs have been destroyed. Then, the job is also marked as 'Complete' in the database by the Task Monitor. When everything is done, the Task Manager checks the Database again

for other pending jobs, and if there are, it chooses the next in the queue and repeats the process described above.

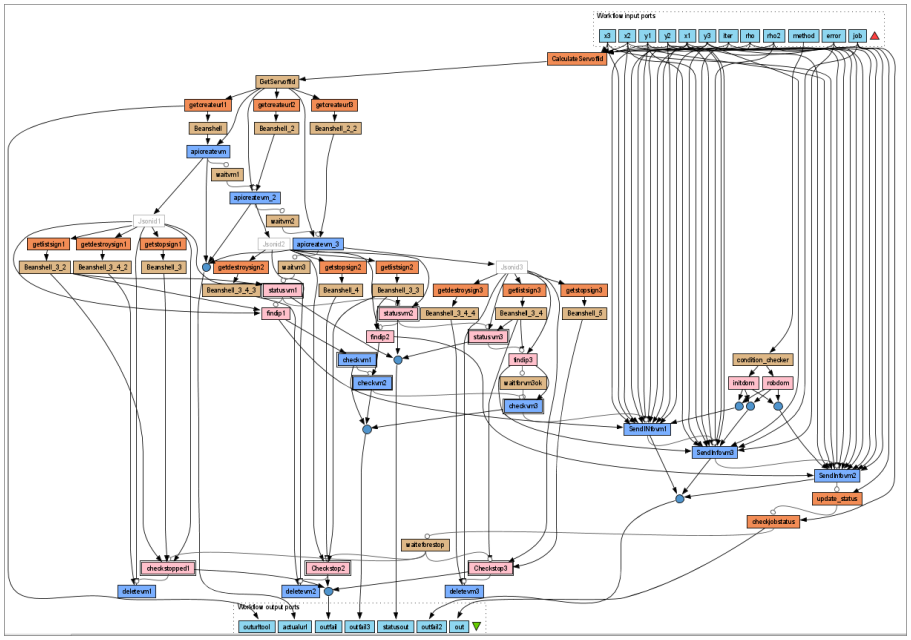

Fig. 8. The workflow for resource allocation and analysis execution as depicted in the Taverna Workbench

5 Conclusion and Future Work

One of the cornerstones of cloud computing is the elastic computational resources that it provides on demand. This enables users to quickly scale up and down the reserved resources according to user, business or performance requirements. The proposed architecture aims to achieve a certain level of dynamic scalability and flexibility in applications executing complex and highly demanding Big Data Analytics processes.

The proposed solution is based on two axes, the first one is the calculation of the required computational resources in terms of Virtual Machines and the second is the depiction of the whole process including virtual infrastructure preparation in workflow.

The proposed system introduces a step in the analysis pipeline where it calculates the number and size of the VMs that are needed on the cloud infrastructure for the execution of the analysis. The calculation is made taking account both the input data and the available resources in the cloud infrastructure.

For the execution of the analysis, a workflow with two segments is used. The first segment, Resources Allocation Workflow is composed of two steps in order to firstly

identify the resources requirements and after to create the appropriate VMs for the actual analysis execution. The second segment is the Analysis Workflow that realizes the analysis pipeline in the cloud infrastructure (VMs and networks) that is dynamically created in the first segment of the workflow.

By the use of workflow coordinated resources allocation, the aforementioned approach can increase the performance in both terms of time and cost efficiency. Nevertheless, the proposed system is subject for further improvements. The authors have already started the research in the empowering of the calculation algorithm with rules that depict user or business requirements. Also, a self-improvement time-scheduling algorithm is going to be added for better allocation of resources in systems with high rate of analysis requests but with limited hardware resources.

6 References

1. Gantz, J., & Reinsel, D.: The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. IDC iView: IDC Analyze the Future, 1-16. (2012)
2. Ji, C., Li, Y., Qiu, W., Awada, U., & Li, K.: Big data processing in cloud computing environments. In: International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN), IEEE (2012)
3. Deelman, E., Gannon, D., Shields, M., & Taylor, I.: Workflows and e-Science: An overview of workflow system features and capabilities. Future Generation Computer Systems, 25(5), 528-540 (2009)
4. Da Xu, L.: Enterprise systems: state-of-the-art and future trends. IEEE Transactions on Industrial Informatics, 7(4), 630-640(2011)
5. Talia, D.: Clouds for Scalable Big Data Analytics. Computer, 46(5), 98-101 (2013)
6. Kagadis, G.C., Kloukinas C., Moore K., Philbin J., Papadimitroulas P., Alexakos C., Nagy P.G., Visvikis D., Hendee W.R.: Cloud computing in medical imaging. Medical Physics, 40(7), 070901 (2013)
7. Chen, H., Chiang, R. H., & Storey, V. C.: Business Intelligence and Analytics: From Big Data to Big Impact. MIS quarterly, 36(4), 1165-1188 (2012)
8. Howe, D., et al.: Big data: The future of biocuration. Nature, 455(7209), 47-50 (2008)
9. Demchenko, Y., Grosso, P., de Laat, C., & Membrey, P.: Addressing big data issues in scientific data infrastructure. In: 2013 International Conference on Collaboration Technologies and Systems (CTS), pp. 48-55. IEEE, New York (2013)
10. Padhy, N., Mishra, D., Panigrahi, R.: The survey of data mining applications and feature scope, International Journal of Computer Science, Engineering and Information Technology, l.2(3), 43-58 (2012)
11. Iosup, A., Ostermann, S., Yigitbasi, M. N., Prodan, R., Fahringer, T., & Epema, D. H.: Performance analysis of cloud computing services for many-tasks scientific computing" IEEE Transactions on Parallel and Distributed Systems, 22(6), 931-945 (2011)
12. Mell, P., & Grance T.: The NIST definition of cloud computing. NIST Special Publication 800-145 (2011).
13. Baset, S. A.: Open source cloud technologies. In: 3rd ACM Symposium on Cloud Computing, p. 28, ACM (2012)
14. Williams, D. E.: Virtualization with Xen (tm): Including XenEnterprise, XenServer, and XenExpress. Syngress (2012)

15. Van Der Aalst, W., & Van Hee, K. M.: Workflow management: models, methods, and systems. MIT press (2004)
16. Curcin, V., & Ghanem, M.: Scientific workflow systems-can one size fit all? In: Biomedical Engineering Conference (CIBEC 2008), pp. 1-9, IEEE (2008).
17. Ko, R. K., Lee, S. S., & Wah Lee, E.: Business process management (BPM) standards: a survey. Business Process Management Journal, 15(5), 744-791 (2009)
18. Chinosi, M., & Trombetta, A.: BPMN: An introduction to the standard. Computer Standards & Interfaces, 34(1), 124-134 (2012)
19. Cambronero, M. E., Dı, G., Macià, H.: A Petri net approach for the design and analysis of Web Services Choreographies. The Journal of Logic and Algebraic Programming, 78(5), 359-380 (2009)
20. Wolstencroft, K., et al.: The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud, Nucleic Acids Research, 41(W1), W557-W561 (2013)
21. Kambatla, K., Kollias, G., Kumar, V., & Grama, A.: Trends in big data analytics. Journal of Parallel and Distributed Computing, 74(7), 2561-2573 (2014)
22. Agrawal, D., Das, S., & El Abbadi, A.: Big data and cloud computing: current state and future opportunities. In: 14th International Conference on Extending Database Technology, pp. 530-533, ACM (2011, March)
23. Vaquero, L. M., Rodero-Merino, L., & Buyya, R.: Dynamically scaling applications in the cloud. ACM SIGCOMM Computer Communication Review, 41(1), 45-52 (2011)
24. Mao, M., & Humphrey, M.: Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In: 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, p. 49, ACM (2011)
25. Herodotou, H., Lim, H., Luo, G., Borisov, N., Dong, L., Cetin, F. B., Babu, S.: Starfish: A Self-tuning System for Big Data Analytics. In: 5th Biennial Conference on Innovative Data Systems Research (CIDR '11), vol 11, pp. 261-272 (2011)
26. Bernstein, D., Ludvigson, E., Sankar, K., Diamond, S., & Morrow, M.: Blueprint for the intercloud-protocols and formats for cloud computing interoperability. In: Fourth International Conference on Internet and Web Applications and Services (ICIW'09), pp. 328-336, IEEE (2009)
27. Korfiati, A., Sfika, N., Daloukas, K., Alexakos, C., Tsompanopoulou P. and Likothanassis, S.: IRaaS: A Cloud Implementation of an Interface Relaxation Method for the Solution of PDEs. In: 2015 International Conference of Parallel and Distributed Computing, part of World Congress on Engineering 2015 (WCE 2015), IAENG, Hong Kong (2015)
28. Korfiati, A., Tsompanopoulou, P. and Likothanassis S.: Serial and Parallel Implementation of an Interface Relaxation Method. In: 6th International Conference on Numerical Analysis (NumAn 2014), pp 167-173 (2014)