Principles of Computing Resources Planning in Cloud-Based Problem Solving Environment

Kirill Borodulin and Gleb Radchenko

South Ural State University, Chelyabinsk, Russia
borodulinkv@susu.ru

Abstract. “Problem-solving environments” recently became a widely accepted approach to providing computational resources to solve complex eScience problems. This approach represents a problem as a workflow, orchestrating a set of various computational services. The existing cloud computing resources planning methods do not take into account a relation between such services, problem domain specifics, predicted workflow execution timespan, etc. On the other hand, usage of cloud system provides efficient HPC resources usage, distributing tasks on the most suitable resources. Therefore, we need to develop algorithms that provide efficient cloud system resources usage and take into account domain-specific information of the problem.

Keywords: cloud · problem-solving environment · workflow · cloud planning

1 Introduction

Nowadays cloud systems become the primary provider to solve problems in physics, biological, and social research. These problems are characterized by structural complexity which causes different resources (informational, software or hardware) to be integrated into a single solution. The problem often can be represented as a workflow, orchestrating a set of actions represent computational parts of the whole problem.

Each action can require a different amount of computing resources and a specific execution environment to work. Comparing to “classical” approach to HPC resources provision, cloud computing systems provide more efficient usage of computational resources, providing each task a unique execution environment and mapping them to the most suitable resources. Also, cloud system can provide a user-friendly web interface for computational research tasks submission.

To provide scientists and engineers a transparent access to the computing resources a Problem Solving Environment (PSE) concept is commonly used. A PSE is a software solution that warps and provides a problem-oriented access to computational resources to solve a specific class e-Science problems. PSE uses the language of the target problem domain, and users do not need to have a specialized knowledge of the underlying hardware or software to submit and get a solution of their computational problem [1].

A PSE problem domain consists of a finite set of task classes. Each task class is a set of tasks that have the same semantics and the same set of input parameters and output data. This limits the class of problems that can be solved using the PSE. On the other hand, this would allow using a domain-specific information of task for selecting a set of computing resources and planning the task's execution in the cloud system.

The aim of the paper is to describe the principles of computing resources planning in Cloud-based Problem Solving Environment. It's required to solve the following tasks for gain the aim of paper:

1. Analyze related solutions for the planning of execution of problem-solving workflow's.
2. Define a structure of cloud system for problem-solving environment's deployment.
3. Describe a scheme of an approach for the computing resources planning in Cloud-based problem-solving environment.

The paper is organized as follows. In section 2 we present the review of the systems provide workflow's execution and methods used to support the resources scheduling for workflow execution. In section 3 we describe the main components of a cloud-based PSE. In section 4 we provide the scheme of an algorithm for the planning of computing resources in the Cloud-based problem-solving environment. In section 5 we summarize the results of our research and provide further research directions.

2 Scheduling Methods in Workflow Systems

A lot of scientific groups in e-Science and engineering fields use workflow platforms to provide scientific computations. Authors of the papers [3–5] describe an analysis of modern (Scientific Workflow Management Systems (SWMS) that support e-Science approach, such as Pegasus [6], Kepler [7], Taverna [8], Galaxy [9]. The Pegasus system [6] is widely used to solve problems in such fields as astronomy, bioinformatics, climate simulation, etc. the System contains 4 components: Planner (Workflow Mapper), Workflow Manager, Task dispatcher (Job Scheduler), Monitoring system (Monitoring Component). Workflow is presented as DAX-file (XML-representation of directed acyclic graph). Pegasus uses HTCondor [11] task scheduler. This scheduler provides computing resources management in cluster computing systems, GRIDs [12], and cloud computing systems [13]. Authors of paper [14] present the cloud-based Code Execution Framework. The paper describes an architecture of the framework to run a problem-solving environment in the cloud system, like Amazon EC2 cloud or OpenStack. These systems can use numerous optimization algorithms to provide scheduling the resources in the computing system, including Grid or cloud system. The algorithms like Improved Differential Evolutionary Algorithm combined with the Taguchi method, Multi-Objective Evolutionary Algorithm based on NSGA-II, Case Library and Pareto Solution based hybrid GA Particle Swarm Optimization, Auction-Based Biobjective Scheduling Strategy [15] have the main drawback that they do not use information about previous executions.

Problem-oriented scheduling (POS) algorithm, that takes into account both specifics of the problem-oriented jobs and multi-core structure of the computing system nodes is proposed in [16]. The algorithm allows one to schedule execution of one task on several processor cores with regard to constraints on the scalability of the task.

The paper [17] describes the scheduling algorithm PO-HEFT for a workflow-based problem-solving environment, which would effectively use a domain-specific information (such as task execution time, scalability limits, and the amount of data transfer) for prediction of cloud computing environment resources load. The paper also presents a model for cloud problem-solving environment. These algorithms are designed for scheduling the workflow actions' execution - when the task has to run to provide a minimal makespan, for example. The reviewed algorithms don't select the set of computing resources required for task's execution and don't distribute the services on the computing nodes to utilize the maximum of computing resources.

3 Model of Cloud-Based Problem Solving Environment

We can define the following main components of cloud-based problem-solving environment:

1. *Service* is an entity that implements functionality for particular computational action. Service provides transparent access to its functionality and interacts with another system's components by an open protocol (HTTP-based or binary). Service is implemented as a virtual machine file, including a base image and applied software for particular computation action implementation.
2. *Base image* - is a basic image of a computational environment (a Virtual machine) that includes a set of system software tools ("middleware"). The following elements are included in the common middleware set of the base image:
   - execution agent (provides common interface for submission, execution, live cycle management of tasks);
   - agent of monitoring system;
   - storage client.

   Virtual machine is an isolated computational environment that provides a limited set of resources for service's execution. Virtual machines work on *computing nodes* of a computing cluster. Every computing node of a cluster runs an agent of virtualization platform that provides execution for virtual machines and cloud system's agent for virtual machines management. Several virtual machines can share resources of one computing node, such as main memory, computing resources, a bandwidth of the network adapter, local storage of the computing node. We can define a set of performance characteristics for every virtual machine [17].

3. *Workflow executor* is a cloud computing system service that provides services orchestration, executing computational actions by workflow planners request.
4. *Workflow planner* - generates a list of services with the computing resources are required for service's execution and send the list to the Workflow Executor.
5. *Workflow predictor* - predicts task's execution parameters for all tasks in the job on the basis of the domain-specific parameters.
6. *Cloud monitoring system* - monitors the execution of task and computing resources' consumption. After a task is finished the Cloud monitoring system sends the statistics about consumption of resources during task's execution with the specified domain-specific arguments of tasks for further improvement in task's prediction.

4 Workflow-Based Resources Planning in Cloud-Based PSE

We can define four layers in the proposed workflow planning approach consists (see Figure 1): the workflow layer, the service layer, the virtual machine layer and the computing nodes layer. The input parameters for the workflow planning wold be abstract problem-solving workflow, the set of the domain-specific arguments' values, paths for input and result files, and quality-of-service (QoS) description. The following agreements can be defined in QoS:

- workflow execution time;
- maximum amount of computing resources;
- cost of calculation.

4.1 Workflow Planning Layer

The workflow layer implements the transformation of the abstract workflow into the executable job. The data sources of the input parameters are being connected with the certain tasks and sub-flows of the workflow during the transformation. The abstract workflow [3] is executable if input arguments of any task are independent of the result of another task's execution. In other words, the arguments for workflow control nodes (such as decision and fork none) may come only before workflow execution, the influence of the intermediate arguments are prohibited.

4.2 Service Planning Layer

The Service layer provides assignment of particular services to the required computing resources for any task in the workflow. The Service layer performs the assignment on the basis of task executions prediction by Workflow predictor. The Workflow predictor sends to the Workflow planner the following prediction information:

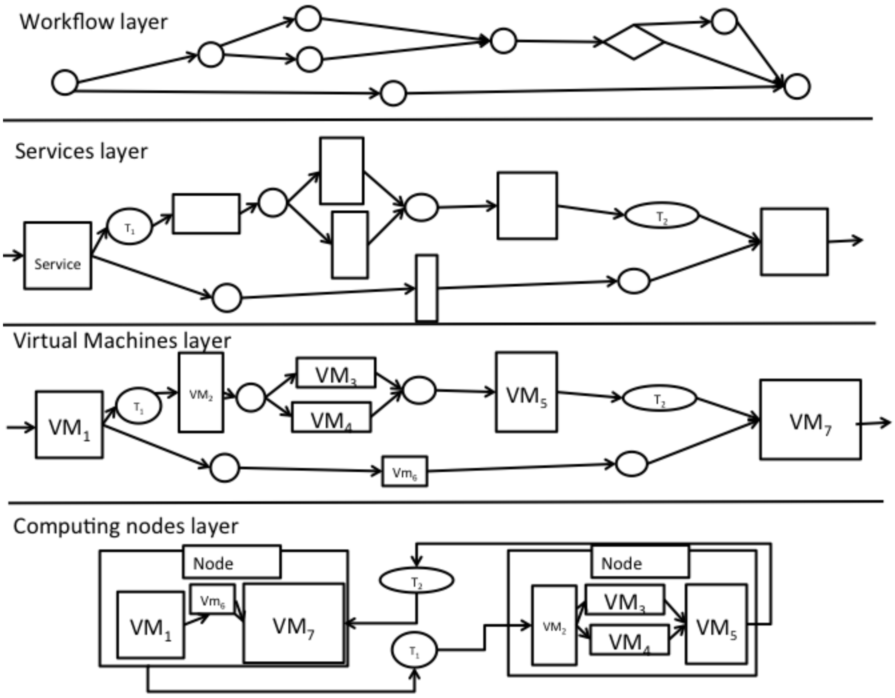

Fig. 1. The layers of the workflow planning approach

- time of the task execution (on the 1 computing core);
- the amount of main memory, needed for the task execution;
- maximum task scaling (how much cores can be provided to the task);
- the amount of the result data;
- prediction accuracy for each value.

4.3 Virtual Machine Planning Layer

The Virtual machine layer selects type ("instance") for each service from the existing types in the cloud computing system. This layer performs the instances selection, using the prediction of computing resources required for the certain task execution, but (1) Workflow executor can allocate another set of resources for QoS satisfaction, (2) if the prediction accuracy is low, i.e. most likely prediction if false, then executor choose the type of virtual machine which is default for the certain service.

4.4 Computing Node Planning Layer

The Computing node planning layer maps virtual machines' onto computing nodes on the basis of a virtual machine computing resources and a volume of node's local storage. At this layer, planner tends to place related virtual machine from the workflow (which on this layer are presented as Task-to-VM list [18]) on the same node to reduce the amount of data are transferred between the computing nodes.

5 Conclusion

In this paper, we provide a description of an approach for computing resources planning for workflows execution in the Cloud-based problem-solving environment. We present the components of the Cloud-based problem-solving environment. The scheme of workflow planning approach consists of 4 layers: workflow layer, service layer, virtual machines layer, computing nodes layer. Currently, we implement the problem-solving algorithm for planning workflows in the Cloud-based problem-solving environment. For the validation and evaluation of the planning algorithm, it is planned to implement a model of problem-based Cloud platform in the simulation environment, as well as on the basis of real HPC system.

Acknowledgements. The reported paper was supported by RFBR, research project No. 15-29-07959.

References

1. Kobashi, H., Kawata, S., Manabe, Y., Matsumoto, M., Usami, H., Barada, D.: PSE park: Framework for problem solving environments. J. Converg. Inf. Technol. 5, 225239 (2010).
2. Fox, G.C., Gannon, D.: Special issue: Workflow in grid systems. Concurr. Comput. Pract. Exp., 18, 1009-1019 (2006).
3. Deelman, E., Gannon, D., Shields, M., Taylor, I.: Workflows and e-Science: An overview of workflow system features and capabilities. Futur. Gener. Comput. Syst. 25, 528–540 (2009).
4. Taylor, I., Deelman, E., Gannon, D., Shields, M.S.: Workflows for e-Science. Work. e-Science Sci. Work. Grids. 1–523 (2007).
5. Miles, S., Wong, S.C., Fang, W., Groth, P., Zauner, K.P., Moreau, L.: Provenance-based validation of e-science experiments. Web Semant. 5, 28–38 (2007).
6. Deelman, E., Vahi, K., Juve, G., Rynge, M., Callaghan, S., Maechling, P.J., Mayani, R., Chen, W., Ferreira Da Silva, R., Livny, M., Wenger, K.: Pegasus, a workflow management system for science automation. Futur. Gener. Comput. Syst. 46, 17–35 (2015).
7. Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B., Mock, S.: Kepler: an extensible system for design and execution of scientific workflows. Sci. Stat. Database Manag. 2004. Proceedings. 16th Int. Conf. I, 423–424 (2004).
8. Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M.R., Wipat, A., Li, P.: Taverna: A tool for the composition and enactment of bioinformatics workflows. Bioinformatics. 20, 3045–3054 (2004).

9. Goecks, J., Nekrutenko, A., Taylor, J.: Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol. 11, R86 (2010).
10. Wolstencroft, K., Haines, R., Fellows, D., Williams, A., Withers, D., Owen, S., Soiland-Reyes, S., Dunlop, I., Nenadic, A., Fisher, P., Bhagat, J., Belhajjame, K., Bacall, F., Hardisty, A., Nieva de la Hidalga, A., Balcazar Vargas, M.P., Sufi, S., Goble, C.: The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. Nucleic Acids Res. 41, (2013).
11. Thain, D., Tannenbaum, T., Livny, M.: Distributed computing in practice: The Condor experience. Concurrency Computation Practice and Experience. vol. 17, no. 24, 323-356 (2005).
12. Deelman, E.: Grids and Clouds: Making Workflow Applications Work in Heterogeneous Distributed Environments. Int. J. High Perform. Comput. Appl. 24, 284-298 (2010).
13. Juve, G., Deelman, E.: Scientific workflows and clouds. Crossroads. 16, 14-18 (2010).
14. Ludescher, T., Feilhauer, T., Brezany, P.: Cloud-Based Code Execution Framework for scientific problem solving environments. J. Cloud Comput. Adv. Syst. Appl. 2, 1–16 (2013).
15. Pandey, S., Wu, L., Guru, S.M., Buyya, R.: A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In: Proceedings - International Conference on Advanced Information Networking and Applications, AINA. pp. 400407 (2010).
16. Sokolinsky, L.B., Shamakina, A. V.: Methods of resource management in problem-oriented computing environment. Program. Comput. Softw. 42, 17–26 (2016).
17. Nepovinnykh, E.A., Radchenko, G.I.: Problem-Oriented Scheduling of Cloud Applications : PO-HEFT Algorithm Case Study. 2016 39th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2016 - Proc. 196–201 (2016).
18. Yang, Z.W.L.N.Y.: A market-oriented hierarchical scheduling strategy in cloud workflow systems. 63, (2013).