Towards Process-Oriented Cloud Management with Case-Based Reasoning

Mirjam Minor and Eric Schulte-Zurhausen

Wirtschaftsinformatik, Goethe University Frankfurt,
60325 Frankfurt am Main, Germany
{minor,eschulte}@cs.uni-frankfurt.de

Abstract. The paper is on a novel cloud management model based on Case-based reasoning. Cloud resources are monitored and (re-)configured according to cloud management experience stored in a case-based system. We introduce a process-oriented, multi-tier cloud management model. We propose a case representation for cloud management cases, define similarity functions and sketch adaptation and revise issues. A proof-of-concept of this ongoing work is given by a sample application scenario from the field of video ingest.

Keywords: Process-oriented CBR, Cloud Computing, Cloud Management, Configuration Problems.

1 Introduction

Cloud management deals with management methods for provisioning and use of cloud services [2]. It is of vital importance to achieve rapid scalability of cloud services which is one of the main characteristics of cloud computing according to the NIST definition [5]. Cloud management addresses monitoring and configuration methods for cloud systems considering technical, organizational and legal aspects. The monitoring methods include measuring technical parameters like the utilization of physical resources, observing the quality of service in compliance *Service Level Agreements* (SLAs), and predicting the system behavior. SLAs are specifications of the terms of use of a service. The configuration methods include traditional network management methods like switching on and off the physical resources, managing virtual resources like *Virtual Machines* (VMs) or managing virtual network facilities.

From a technical point of view, cloud management is a resource management problem that can be solved by a multi-dimensional optimization approach [20] balancing resource consumption with other optimization criteria like performance or costs for SLA violations. However, decisions in cloud management have to be taken immediately. Solutions with a lower computational complexity than multi-dimensional optimization are advisable. *Case-based Reasoning* (CBR) has been considered for intelligent cloud management recently in the literature [11]. The work of Maurer et al. applies CBR to implement automatic cloud management following the MAPE reference model (Monitor - Analyse - Plan -

L. Lamontagne and E. Plaza (Eds.): ICCBR 2014, LNCS 8765, pp. 303–312, 2014.

Execute) [6], which originates in autonomic computing. A case in cloud management records a cloud configuration with current workloads to be processed as a problem situation. A solution describes the optimal distribution of work on the optimal number and configuration of cloud resources while maintaining SLAs. Maurer et al. use a bag of workloads to schedule the work, which makes it difficult to predict future workloads and to achieve stable configurations for a few future time steps.

In this paper, we make use of the workflow paradigm to address this gap by a case-based cloud configuration approach that takes into consideration the workloads to be scheduled next. The bag of workloads is replaced by the set of ongoing workflow instances. A workflow is "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" [4]. The currently active tasks represent the workload of the cloud system. This process-oriented cloud management approach provides further benefits in addition to better prediction capabilities: It allows to use process-oriented modelling and monitoring tools instead of conventional cloud management tools, which are usually not aware of business processes and which are fairly frequently command-line oriented. The use of modelling tools also for cloud management tasks is more convenient for administrators who prefer graphical tools. Workflow reasoning supports the configuration task of cloud management by well-informed suggestions or provides even an automated cloud management approach based on previous experience. This paper extends previous work on a very early version of the proposed cloud management solution [15] by introducing the workload concept, elaborating a running sample in a video ingest application and refining some other definitions.

The paper is organized as follows: First, we discuss some related work in Section 2. In Section 3, we present on a multi-tier model for process-oriented cloud management implementing different cloud layers. In Section 4, we introduce a case-based approach for task placement on cloud resources. Section 5 provides a proof-of-concept for the approach by means of an application scenario on video ingest workflows. Finally, we conclude the paper in Section 6.

2 Related Work

Many commercial cloud systems still use quite straight-forward algorithms for cloud management. Frequently, cloud management activities are chosen following simple rules based on observations on the number of open connections [8] or on the CPU utilization. In contrast, our approach considers the characteristics of workloads like CPU intensive, storage intensive, memory intensive or network intensive tasks.

Some work has already been done on the automated placement of VMs on physical resources. Experiments have shown that VM placement decisions should consider the characteristics of the tasks [10]. In the All4green project [1] , energy consumption profiles are investigated to design Green-SLAs between data centers and end users considering the conservation of resources via SLAs. Both approaches are not aware of future workloads. In CloudBay [20], software agents negotiate on available resources like VMs with other software agents. The agents lease and configure resources to orchestrate virtual appliances automatically. The act of sale is not in the scope of our work which is restricted to manage resources in a technical sense.

The placement of jobs on VMs has been studied in the field of High Performance Computing [16,9]. Similar to the job placement problem is the application placement problem [17] which is the problem to to allocate applications with dynamically changing demands to VMs and how to decide how many VMs must be started. In contrast to such bag of work approaches, workflow tasks are executed in a given order.

The problem to assign workflow tasks to computing resources has been investigated in Grid Computing [19]. Wu et al. [18] describe a cloud computing approach inspired by Grid Computing to solve the task placement problem for scientific workflows with meta-heuristics. This Task-to-VM assignment is implemented by SwinDeW-C, a cloud workflow system. A unified resource layer is defined that is not reusable for our approach since we aim to involve heterogeneous cloud environments which are required in recent hybrid cloud solutions, for instance, to combine private and public cloud resources.

3 A Multi-Tier Cloud Management Model

Cloud management has to deal with complex topologies of resources. A multi-tier model separates physical from virtual resources in different layers. It allows to cascade configuration activities from layer to layer. The model is an extension of the cloud management model from Maurer at al. [11], which consists of three layers for hierarchical configuration activities. We place a workflow tier on top of the three tiers from Maurer et al.'s model in order to achieve a process-oriented perspective (see Fig. 1). The *physical machine tier* at the bottom is manipulated by configuration activities like to add and remove compute nodes. The *virtual machine tier* allows activities like to increase or decrease incoming and outgoing bandwith of a VM, its memory, its CPU share, or to add or remove allocated storage by x%. Further, VMs can be migrated to a different physical machine or moved to and from other clouds in case of outsourcing/insourcing capabilities. The *application tier* is dedicated to management activities for individual applications. The same set of activities as for VMs can be applied but with an application-specific scope. Obviously, the migration and insourcing/outsourcing activities refer to the placement of applications on VMs at the application tier. The *workflow tier* adresses the placement of workflow tasks on VMs called *task placement*. Two management activities can be conducted at the workflow tier:

- Task migration
- Task tailoring

Task migration means that a workflow task is scheduled on a different VM for execution. The initial placement of a task on a VM is a special case of migration.

The tailoring of tasks addresses the adaptation of tasks of an ongoing workflow as follows: Tasks can be replicated by splitting the corresponding input and output data in case the monitoring and prediction values advise to do so. This may require an additional task that aggregates the output data resulting from the replicated tasks. The task tailoring is a shallow form of agile workflow technology [12] which allows to structurally adapt ongoing workflow instances.

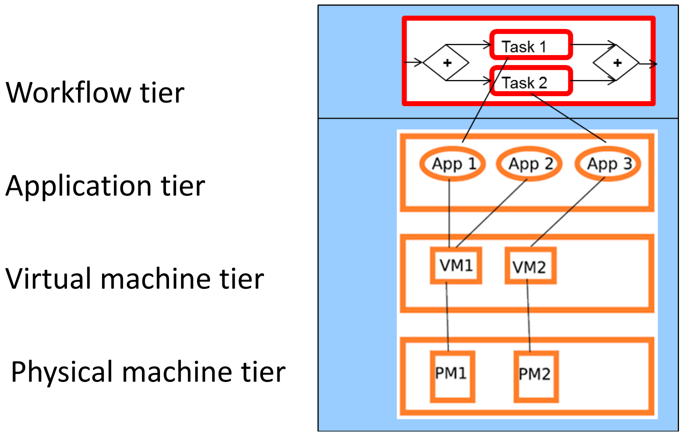

Fig. 1. Process-oriented model for cloud management extending Maurer et al. [11]

4 CBR for Task Placement

A case records an experience of a solved problem [13]. In cloud management, a case refers to solving a cloud management problem like to avoid an impending SLA violation or to schedule workflow tasks to be triggered next.

4.1 Representation of a Cloud Management Case

The *problem part* of a cloud management case describes a state of the cloud system where an action is required. The state of the system comprises of:

- the cloud configuration,
- the task placement, i.e. the set of ongoing workflow tasks (workloads) and their distribution on the cloud resources
- the actual utilization values measured for the virtual and physical resources
- the utilization values agreed on in the SLAs

The cloud configuration CC is a tuple $\langle PM, VM, VMP \rangle$. PM denotes a set of physical machines. Each physical machine is characterized by a tuple $(bandwidth, CPU, memory, storage, costs\ per\ time\ unit)$ describing the hardware parameters of the physical machine. The costs per time unit specifies an average value for the operating costs, particularly the energy consumed while the machine is running. VM stands for a set of virtual machines. Each virtual machine is characterized by a tuple $(bandwidth, CPU, memory, storage)$ describing the maximum share of the physical resources consumed by the virtual machine running under full workload. $VMP : VM \rightarrow PM$ is the placement function for virtual machines on physical machines. Each VM $vm_i \in VM$ is assigned to exactly one $pm_j \in PM$ but a PM can contain more than one VM, for example $\{(vm_1, pm_1), (vm_2, pm_1)\}$.

The task placement TP is described by a tuple $\langle AT, VM, TPF \rangle$. AT is the set of currently active tasks. Tasks join this set at the moment of being triggered for execution by the workflow engine. A workflow task $wt \in AT$ is described by a tuple $(task\ name, input\ data, execution\ time\ per\ unit, cpu\ usage, memory\ usage, storage\ usage, bandwidth\ usage)$. The task name is a string; input data provides a link to the input data to be processed by the task. The execution time per unit provides an average value for the duration of the task per data unit of a standardized size. The cpu usage is given in Million Instructions Per Second (MIPS), memory usage in gigabytes, storage usage in gigabytes and bandwith usage in megabytes per second. For example, a task "render image" might be described by (render image, 10GB, 27.5 GB/h, 100 MIPS, 4GB, 1GB, 0Mbit/s). VM is a set of virtual machines. $TPF : AT \rightarrow VM$ is the placement function for tasks on virtual machines.

The actual utilization values and the utilization values agreed on in the SLAs are expressed in percentage of the values provided by the cloud configuration.

Further, the event for which an action is required is part of the problem description, for instance the event of observing the CPU utilization of a physical server exceeding a threshold.

The *solution part* of a cloud management case describes the action taken in the past as well as the action recommended by a post mortem analysis for the past situation. The post mortem analysis has been inspired by the work of Gundersen et al. [7] on real-time analysis of events. The recommended action can be determined by an optimization approach, for instance.

4.2 Retrieval of a Cloud Management Case

A query is a partial case describing a recent situation of a cloud system with an event causing a cloud management problem. The retrieval of cases uses a similarity function for cloud management cases following the local-global-pinciple [13]. In this early stage of the work, we are planning to use straight-forward similarity functions, which might be replaced by more sophisticated functions such as object-oriented functions considering classes of tasks. Both, cloud configuration and task placement can be denoted as a bi-partite graph with the nodes of the one graph being mapped to the nodes of the other graph. Hence, graph edit distances [3] are used for the local similarity functions for cloud configurations and for task placements. The edit distance seems a natural approach since it imitates the actual configuration steps. The utilization values are compared by weighted sums for numerical distances. The events are compared by a simple structured similarity function based on a taxonomy of events.

4.3 Reuse of a Cloud Management Case

The reuse of a cloud management case requires an adaptation of the retrieved case to the current situation. Further, it has to be investigated whether the recommended management activities actually solve the problem. The latter can be done by a short-time simulation of the behaviour of the cloud system making use of the workflow execution logic. Based on the estimated execution time of the particular current workloads, a prediction of the workloads in the following n time steps is computed by simulating the tasks to be scheduled next for each ongoing workflow instance. For this sequence of n time steps, the prospective utilization values are simulated and assessed. A rough approximation of such a short-time simulation could be to consider only one future step with the set of all tasks that are subsequent to ongoing tasks. This set can be easily derived from the control flow structure of the workflow instances. The workload of this set of tasks can be approximated based on the estimated size of the output data of the previous task. The characteristics of the application domain might have an impact on the number of future steps to be considered best by the workload approximation.

5 Video Ingest as a Sample Application Scenario

We have chosen a sample application scenario on video ingest processes in order to provide a first proof-of-concept for our novel, case-based, process-oriented cloud management approach. Fig. 2 depicts a sample workflow for a video ingest process that is inspired by the work of Rickert and Eibl [14]. Video content on VHS video tapes is digitized and stored in different formats on a tape in the more recent LTO standard. Different transformation steps like creating a preview file, creating a legacy proxy for archiving purposes, and creating an analysis proxy for further processing, for instance, for face detection, are executed as workflow tasks in parallel. Part of the workflow tasks include algorithms with a high computational complexity. A cloud computing solution allows to accelerate the workflow execution by running tasks on different computing resources in parallel.

The cloud configuration that we have chosen for this illustrating example is CC = {PM, VM, VMP}. Let us assume a 3GHz CPU that can handle around 9000MIPS, a 2GHz CPU that is able to handle 6000MIPS and a 1GHz CPU with about 3000MIPS. Further, we assume that we have two PMs $pm_1, pm_2 \in$ PM and a configuration of both PMs $pm_i =$ (3GHz, 4GB, 1000GB, 100MBit/s, 3$ /day). In addition, we assume to have 3 identical VMs $vm_i \in$ VM with the configuration (1GHz, 2GB, 500GB, 50MBit/s). The actual placement of the virtual machines is VMP = $\{(pm_1, vm_1),(pm_2, vm_2),(pm_2, vm_3)\}$.

Let the sample tasks shown in Fig. 3 have a task placement TP = {AT, VM, TPF} with *create preview file*, *create legacy proxy* and *create analysis proxy* being active tasks. Table 1 describes the properties of the tasks. Let the VMs as described in CC be assigned as follows by TPF = {(vm_1, create preview file),(vm_2,create legacy proxy),(vm_3, create analysis proxy)}.

The set of the tasks to be scheduled next would be {detect faces, create QR code} according to Fig. 2.

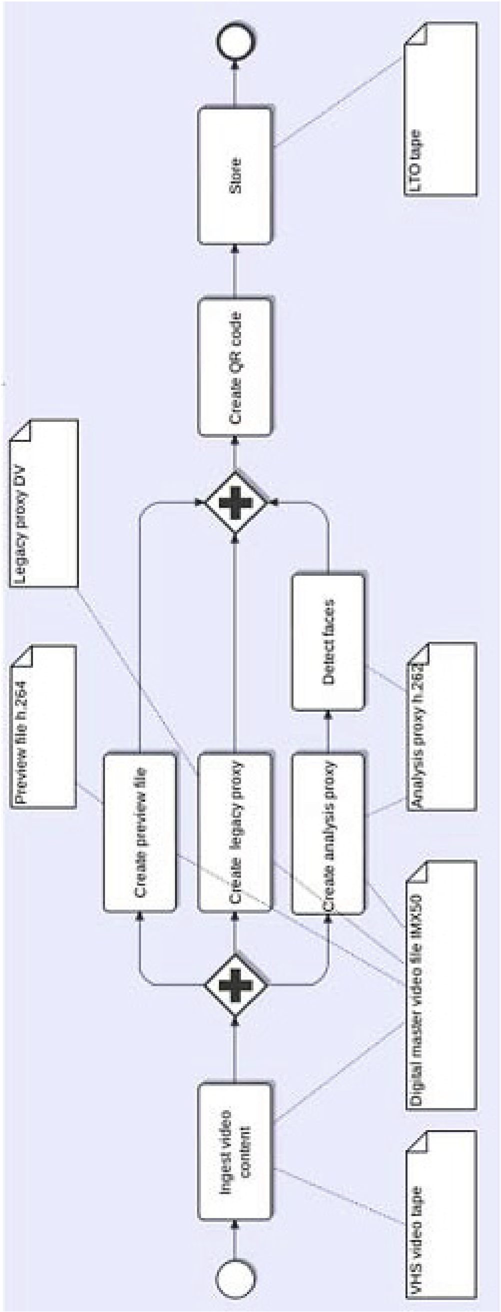

Fig. 2. Example for a video ingest workflow derived from Rickert and Eibl [14]

Table 1. Properties of the sample tasks used in the scenario

task name	**create preview file**	**create legacy proxy**	**create analysis proxy**
input data	10GB	10GB	10GB
execution time per unit	1GB/h	12.6GB/h	3.15GB/h
cpu usage	3000MIPS	1000MIPS	1000MIPS
memory usage	4GB	2GB	2GB
storage usage	10GB	2GB	2GB
bandwidth usage	0MBit/s	0MBit/s	0MBit/s

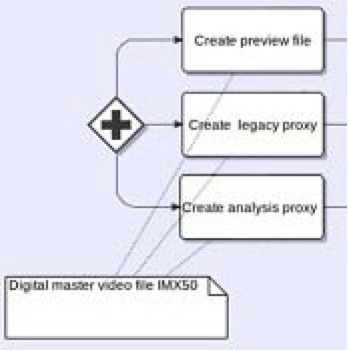

Fig. 3. Example of currently active tasks of the video ingest workflow from Fig. 2

Let us assume that one SLA has guaranteed that all tasks of a user are always provided with CPU times according to the MIPS that are specified in the sample tasks. In case of the "create preview file" task, the task requires 3000MIPS, for instance. However, the virtual machine vm_1 has a CPU of 1GHz only and, thus, can handle 3000MIPS. Since the vm_1 requires some additional CPU time for its operating system, obviously, the SLA will be violated.

If such an event is detected, the CBR process is triggered to retrieve a best matching case from the case base. The resulting case has a cloud configuration CC' and a task placement TP'. Let the set of virtual machines in the retrieved case be VM' = $\{vm_1$', vm_2',vm_3'$\}$ with vm_i' = (0.5GHz, 1GB, 500GB, 50MBit/s). VM' differs from VM with respect to the cpu and the memory values. The edit distance between the according placement graphs is 6 since the following edit operations are required to transform CC into CC':

- Decrease the CPU by 50% for the three VMs.
- Decrease the memory assigned by 50% for the three VMs.

Let the set of tasks from the retrieved case be the same as in the current situation but with half the MIPS numbers. The local similarity function for the task placement makes use of a table of similarity values specified in Table 2.

Table 2. Local similarity function for MIPSs

	3000	6000	9000
3000	1	0.7	0.3
6000	0.7	1	0.7
9000	0.3	0.7	1

A sample solution suggested by the case could be to increase the CPU share for the virtual machine vm_1. This solution can be transfered to the recent situation namely to increase the CPU share for vm_1'.

6 Conclusion

In this paper we have introduced a novel cloud management model with a process-oriented perspective and a CBR approach for experience-based cloud management. A cloud management case describes the state of a cloud system including an event causing a problem. The solution part of the case provides a solution for the problem by cloud management activities. The novel model has been illustrated by a sample application with a video ingest workflow. Our workflow reasoning approach aims at improving the automated management of computational resources of a cloud system for the different video analysis tasks. It takes into consideration cases reporting previous configuration decisions in similar situations, for instance to switch on an additional virtual machine in case of a high utilization of CPU's. The cloud management model is still in an early phase of ongoing work. In future work, we will further specify the similarity functions and adaptation methods, develop an implementation of the proposed model and conduct some lab experiments.

We believe that cloud computing and especially cloud management and cloud configuration issues are an intriguing, novel application field for CBR. The cloud management model provided in this paper is a first step towards case-based solutions for cloud management and will hopefully stipulate further work in future.

References

1. Basmadjian, R., Niedermeier, F., de Meer, H.: Modelling and analysing the power consumption of idle servers. In: Sustainable Internet and ICT for Sustainability, SustainIT 2012, October 4-5, Pisa, Italy, Sponsored by the IFIP TC6 WG 6.3 "Performance of Communication Systems", pp. 1–9. IEEE (2012)
2. Baun, C., Kunze, M., Nimis, J., Tai, S.: Cloud Computing - Web-Based Dynamic IT Services. Springer (2011)
3. Bunke, H., Messmer, B.T.: Similarity measures for structured representations. In: Wess, S., Althoff, K.D., Richter, M. (eds.) EWCBR 1993. LNCS, vol. 837, pp. 106–118. Springer, Heidelberg (1994)
4. Workflow Management Coalition: Workflow management coalition glossary & terminology (1999), `http://www.wfmc.org/standars/docs/TC-1011_term_glossary_v3.pdf` (last access May 23, 2007)

5. NIST US Department of Commerce: Final version of NIST cloud computing definition published, `http://www.nist.gov/itl/csd/cloud-102511.cfm` (last access September 17, 2013)
6. IBM Corporation: An architectural blueprint for autonomic computing (2006), `http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White %20Paper%20V7.pdf` (last accessed: April 17, 2014)
7. Gundersen, O.E., Sørmo, F., Aamodt, A., Skalle, P.: A real-time decision support system for high cost oil-well drilling operations. In: Twenty-Fourth IAAI Conference (2012)
8. Hammond, B.W.: Getting Started with OpenShift. Google Patents (2003), US Patent 6,637,020
9. Le, K., Zhang, J., Meng, J., Bianchini, R., Jaluria, Y., Nguyen, T.: Reducing electricity cost through virtual machine placement in high performance computing clouds. In: 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC), pp. 1–12 (November 2011)
10. Mahmood, Z.: Cloud Computing: Methods and Practical Approaches Springer (May 2013)
11. Maurer, M., Brandic, I., Sakellariou, R.: Adaptive resource configuration for cloud infrastructure management. Future Generation Computer Systems 29(2), 472–487 (2013)
12. Minor, M., Tartakovski, A., Schmalen, D., Bergmann, R.: Agile workflow technology and case-based change reuse for long-term processes. International Journal of Intelligent Information Technologies 4(1), 80–98 (2008)
13. Richter, M.M., Weber, R.: Case-Based Reasoning: A Textbook. Springer (2013)
14. Rickert, M., Eibl, M.: Evaluation of media analysis and information retrieval solutions for audio-visual content through their integration in realistic workflows of the broadcast industry. In: Proceedings of the 2013 Research in Adaptive and Convergent Systems, RACS 2013, pp. 118–121. ACM, New York (2013)
15. Schulte-Zurhausen, E., Minor, M.: Task placement in a cloud with case based reasoning. In: 4th International Conference on Cloud Computing and Services Science, Barcelona, Spain (2014)
16. Sharma, B., Chudnovsky, V., Hellerstein, J.L., Rifaat, R., Das, C.R.: Modeling and synthesizing task placement constraints in google compute clusters. In: Proceedings of the 2nd ACM Symposium on Cloud Computing, SOCC 2011, pp. 3:1–3:14. ACM, New York (2011)
17. Tang, C., Steinder, M., Spreitzer, M., Pacifici, G.: A scalable application placement controller for enterprise data centers. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, pp. 331–340. ACM, New York (2007)
18. Wu, Z., Liu, X., Ni, Z., Yuan, D., Yang, Y.: A market-oriented hierarchical scheduling strategy in cloud workflow systems. The Journal of Supercomputing 63(1), 256–293 (2013)
19. Yu, J., Buyya, R., Ramamohanarao, K.: Workflow scheduling algorithms for grid computing. In: Xhafa, F., Abraham, A. (eds.) Metaheuristics for Scheduling in Distributed Computing Environments. SCI, vol. 146, pp. 173–214. Springer, Heidelberg (2008)
20. Zhao, H., Li, X.: Resource Management in Utility and Cloud Computing, 1st edn. Springer, Dordrecht (2013)