Improving IaaS Cloud Analyses by Black-Box Resource Demand Modeling

Henning Groenda
FZI Forschungszentrum Informatik
Haid-und-Neu-Str. 10-14
76131 Karlsruhe
groenda@fzi.de

Christian Stier
FZI Forschungszentrum Informatik
Haid-und-Neu-Str. 10-14
76131 Karlsruhe
stier@fzi.de

ABSTRACT

In Infrastructure as a Service (IaaS) Cloud scenarios, data center operators require specifications of Virtual Machine (VM) behavior for data center middle- and long-term planning and optimization. The planning is usually supported by simulations. While users can leverage white-box application knowledge, data center operators have to rely on metrics at the level of resource demands provided by virtualization and cloud middleware platforms. Existing simulations for data center planning do not combine both viewpoints and either require white-box knowledge or focus on short-term predictions using statistical estimators. Our approach allows modeling varying resource demand of black-box VMs based on the Descartes Load Intensity Model (DLIM). The black-box VM models are integrated in the SimuLizar performance simulator complementing the existing grey- and white-box models in order to improve reasoning on (de-) consolidation decisions.

Keywords

Performance Prediction, Modeling, SimuLizar, DLIM, Palladio, Design-Time

1. INTRODUCTION

In Infrastructure as a Service (IaaS) Cloud scenarios, data center operators and users have different insights into performance metrics on deployed Virtual Machines (VMs). Operators require specifications of VM's resource demand and its variation for data center middle- and long-term planning and optimization. Data center customers focus on maintaining service levels for their users by dynamically adapting their applications and number of VMs to the experienced user load intensity. Both benefit from simulation-based planning at design-time to optimize their data centre or application infrastructure.

Operators have only limited insight into deployed VMs and their main information sources are virtualization and Cloud middleware platforms. Cloud customers have white-box application knowledge but do not know co-located VMs competing for resources. Knowledge on the dynamic variation of resource demand allows placement with less resource conflicts while maintaining high utilization levels. This knowledge cannot be assumed for all VMs - at least some of them will remain black boxes due to separation of concerns.

Existing simulations for data center planning do not combine black- and white box modeling viewpoints. Pure black-box approaches usually focus on short-term predictions using statistical estimators for consolidation and elasticity decisions. Application-centric simulations require white-box knowledge and cannot take resource conflicts with other customers into account.

In this paper, we present our approach to extend a white-box simulation by modeling varying resource demands for black-box VMs. Our approach bases on prior work by Lehrig et al. [5, 6] for white-box simulations. Their approach uses the Descartes Load Intensity Model (DLIM) for modeling variation and addresses workload evolution for white-box applications. We address the following assumptions (A) and limitations (L) made by Lehrig et al.: A1) the simulation has a maximum simulation time as its only stop condition, A2) the evolution duration is less than or equal to the maximum simulation time, and L1) the evolution duration is always scaled to the maximum simulation time, and L2) only the first Usage Scenario is evolved. Beyond this relaxation, we contribute a model based on DLIM for describing the resource demand of black-box VMs. The black-box model can be integrated with grey- and white-box models and supports IaaS scenarios from the operator perspective.

The improved evolution mechanism is shipped with the Palladio 4.0 release. An implementation of the black-box VM models complementing the existing grey- and white-box models is shipped with the CactoSim 2.0 release[1]. Both improve reasoning on (de-)consolidation decisions.

This paper structure is as follows: Section 2 provides information on addressing the existing assumptions for white-box models and how the new functionality can be used. Section 3 points out how black-box resource demand specifications are realized enabling their application in own scenarios. Section 4 concludes the paper.

2. RELAXING ASSUMPTIONS FOR SIMULIZAR'S USAGE EVOLUTIONS

This section summarizes the state of prior work and our adaptations for relaxing the assumptions on SimuLizar's Evolution mechanism. The mechanism allows to change load or work parameters during simulations based on DLIM Sequence specifications (concept see Lehrig et al. [5, 6]; implementation by Sintef, Norway). This enabled dynamic load variations. The four addressed assumptions and limitations are: A1) the simulation has a maximum simulation time as its only stop condition, A2) the evolution duration is less than or equal to the maximum simulation time, and L1) the evolution duration is always scaled to the maximum simula-

[1] www.cactosfp7.eu

tion time, and L2) only the first Usage Scenario is evolved.

Assumptions A1 and A2 as well as limitation L1 allowed sampling of the Evolution specification at each integer time unit of the underlying DLIM Sequence demand model. The sample was used to adapt the values within the simulation at each $\frac{\text{maximum simulation time}}{\text{duration DLIM Sequence+1}}$ point in simulation time.

The relaxed version supports non-scaled evolutions, i.e. that evolutions are specified for weeks or months and simulation and evolution time progresses with the same speed. Evolutions that have a shorter execution time than the simulation can be repeated. Non-scaled evolutions support sampling at arbitrary intervals.

We relaxed A1, A2, and L1 with full legacy support by extending the Usage Evolution model. We added the attributes *repeatingPattern* and *evolutionStepWidth* to the class Usage. If the repeatingPattern attribute is false - its default value - then the prior behavior is experienced. If it is set to true the relaxed version is experienced. The evolutionStepWidth allows to set the points in simulation time when a sample is taken from the evolution specification. The default value is 1. evolutionStepWidth is only considered if repeatingPattern is true. This allows a seamless transition for legacy models.

Limitation L2 inhibited evolutions for more than one usage scenario. The relaxed version supports any number of scenarios. Model changes were not required as they already supported multiple evolution scenarios.

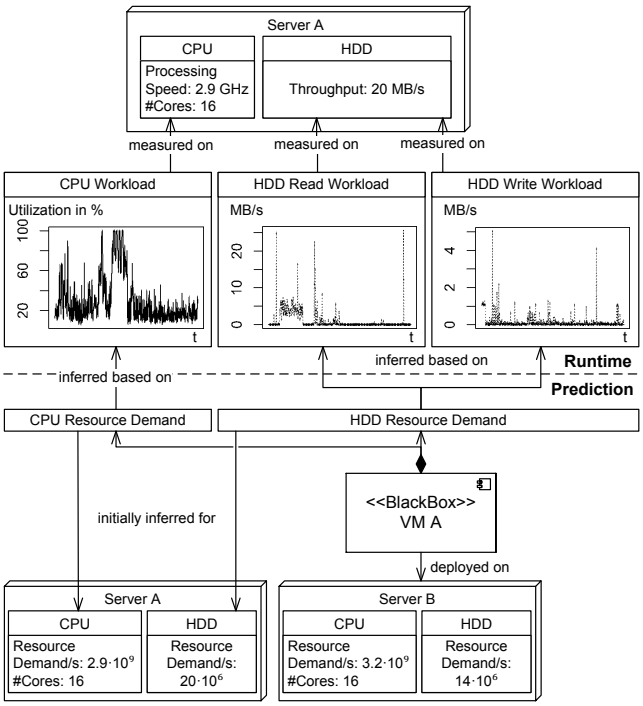

Figure 1: Exemplary mapping of trace-based workload models to prediction models

3. SIMULATING BLACK-BOX DATA CENTER RESOURCES

This section describes how a combination of Usage Evolution and DLIM [9] enable black-box resource demand specifications. The following summarizes a typical context before the details of the combination are explained.

This black-box resource demand modeling is part of the CACTOS project, which creates a methodology and tools for large-scale data centre planning and runtime management [7]. CACTOS stores models for VMs and updates them at runtime for utilization optimization and supports the transfer to prediction time for what-if-analyses. The prediction uses the simulator SimuLizar [1, 2] and Palladio [3]. The resource demand modeling process is split in two steps. CACTOS infers resource demand and stores it in black-box models in the first step. In the second step, those models are transformed to Palladio models. This allows handling concepts like VMs, which are not available in Palladio, and still use existing simulation capabilities.

The following shows how to infer resource demand from workload information in runtime environments such as data centers and store it in CACTOS models. Figure 1 provides an overview. Every black-box component has a set of resource demands describing its behavior. In the example, VM A's load is described in terms of the CPU utilization and HDD accesses observed on Server A.

Execution-platform independent demand specifications allow reasoning on re-location decisions. Palladio and other predictive performance models therefore use these kinds of resource demands. Resource demands specify the amount a service requires of a specific *type* of resource for execution. Different approaches for resource demand estimation are available to derive resource demand estimations from system traces, see Spinner et al. [8]. The mapping in Figure 1 between runtime and prediction model uses these techniques to derive the resource demands of a black-box VM from the measured CPU utilization and HDD throughput.

The DLIM instances in the prediction model capture the varying resource demand of a VM. Each DLIM instance models the resource load intensity at which the black-box VM issues load on a resource of the underlying execution environment. The DLIM instances are contained in the specification of the VM. The black-box VM model of CACTOS consists only of the presented elements. The concept and mapping works regardless if you want to use CACTOS models or not. The models are execution environment independent and therefore allow evaluating resource load if the VM runs on another server. Figure 1 provides an example where VM A is deployed on Server B instead of Server A. The faster CPU of Server B processes the resource demand of the VM in shorter time.

The derivation of the resource load intensity limits the resource load intensities' validity to similar server hardware platforms. Different hardware platforms, e.g. AMD and Intel processors for CPU, process the same instructions with different performance [4]. Black-box VMs reference the environment used to derive the workload model. This allows users of the simulation-based prediction to reason on the acceptable accuracy for their use-case.

In Palladio and SimuLizar's white-box view, only user requests cause the system and its parts to use resources. The simulation takes care of contention and delay effects. The simulation assumes that the arrival rate and the interactions of users with the system are known and fully specified. This assumption does not hold for black-box VMs.

The simulator does not directly simulate the extended

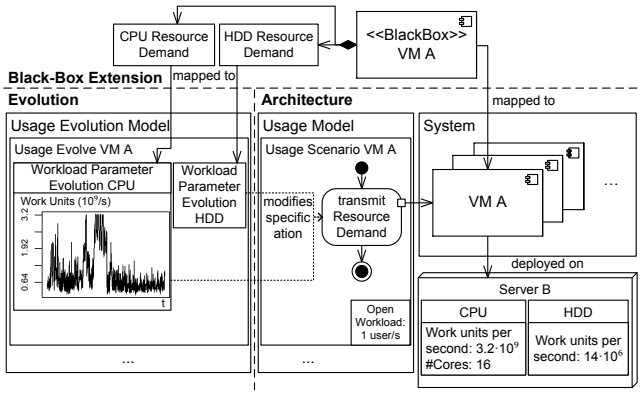

Figure 2: Mapping of the prediction models to the simulated Palladio model.

black-box model. A refinement transformation maps the black-box model to the simulation kernel, generating a Load Evolution model [6] besides the corresponding architecture models. Figure 2 illustrates how the elements in the extended prediction model correspond to the elements in the simulated models. Each black-box VM maps to a distinct Usage Evolve, Usage Scenario, Component and Component Assembly element. In case of VM A, its Usage Scenario VM A contains the activity *transmit Resource Demand* which transmits the requested resource demands to VM A. VM A then requests the demand from its deployment environment. Usage Scenario VM A is designed as an Open Workload issuing the demand as originally requested and the step width specified for the evolution. Resource contention can cause overlapping processing of these requests without the need to change the evolution model. Usage Evolve VM A is responsible to modify the specifications of these resource demand parameters in Usage Scenario VM A before the demand is transferred to VM A. Usage Evolve VM A contains one Work Parameter Evolution per DLIM instance.

4. CONCLUSIONS

We described our extension to SimuLizar's Usage Evolution models with full legacy support that enables the decoupling of evolution specifications from the simulation configuration. The introduced sampling at arbitrary intervals supports using different levels of detail and accuracy within models. This allows (re-)using evolution specifications in different simulations. The support for more than one usage scenario eases modeling multiple evolutions. We discussed how black-box resource demand models for VMs can be inferred based on estimation techniques. The described transformation of these resource demands models to the simulation kernel Palladio allows to model black-, grey-, and white-box VM and applications models with the same approach. This allows taking into account the effect they have on each other if they are competing for resources.

Data center customers and application architects benefit from the improved and decoupled Load Evolution models. They can use and combine black-, grey- or white-box models depending on their need. Additionally, they can trade prediction accuracy with complexity where possible and desired. Data center operators are able to use design-time simulations for planning and what-if analyses even if they use only virtualization and cloud middleware platform data.

In the short term, performance improvements for evaluating the DLIM Sequences at different points in time as part of the evolutions are planned. In the middle term, we want to develop a concept for enabling the setting of default values representing that an evolution has terminated. We are also planning to relax the assumption that evolutions always start at their beginning and enable to sample sections of long-running evolutions or respectively simulations at different intervals within the time span of evolutions. Finally, we aim to improve the accuracy of simulated resource demands by integrating over the sampled interval instead of using a discrete sample.

5. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union's Seventh Framework Programme under grant agreement no. 610711 (CACTOS).

6. REFERENCES

[1] M. Becker, S. Becker, and J. Meyer. Simulizar: Design-time modeling and performance analysis of self-adaptive systems. In S. Kowalewski and B. Rumpe, editors, *Software Engineering 2013: Fachtagung des GI-Fachbereichs Softwaretechnik, 26. Februar - 2. März 2013 in Aachen*, volume 213 of *LNI*, pages 71–84. GI, 2013.

[2] M. Becker, M. Luckey, and S. Becker. Performance analysis of self-adaptive systems for requirements validation at design-time. In *Proceedings of the 9th International ACM Sigsoft Conference on Quality of Software Architectures*, QoSA '13, pages 43–52, New York, NY, USA, 2013. ACM.

[3] S. Becker, H. Koziolek, and R. Reussner. The palladio component model for model-driven performance prediction. *Journal of Systems and Software*, 82(1):3 – 22, 2009. Special Issue: Software Performance - Modeling and Analysis.

[4] M. Kuperberg. *Quantifying and Predicting the Influence of Execution Platform on Software Component Performance.* PhD thesis, Karlsruhe Institute of Technology, 2010.

[5] S. Lehrig and M. Becker. Approaching the cloud: Using palladio for scalability, elasticity, and efficiency analyses. In *Proceedings of the Symposium on Software Performance 2014, 26-28 November 2014, Stuttgart, Germany*, 2014.

[6] S. Lehrig and S. Becker. The cloudscale method for software scalability, elasticity, and efficiency engineering: A tutorial. In *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering*, ICPE '15, pages 329–331, New York, NY, USA, 2015. ACM.

[7] P.-O. Östberg, H. Groenda, S. Wesner, J. Byrne, D. S. Nikolopoulos, C. Sheridan, J. Krzywda, A. Ali-Eldin, J. Tordsson, E. Elmroth, C. Stier, K. Krogmann, J. Domaschka, C. Hauser, P. Byrne, S. Svorobej, B. McCollum, Z. Papazachos, L. Johannessen, S. Rüth, and D. Paurevic. The CACTOS Vision of Context-Aware Cloud Topology Optimization and Simulation. In *Proceedings of the Sixth IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 26–31, Singapore, December 2014. IEEE Computer Society.

[8] S. Spinner, G. Casale, F. Brosig, and S. Kounev. Evaluating approaches to resource demand estimation. *Performance Evaluation*, 92:51 – 71, 2015.

[9] J. G. von Kistowski, N. R. Herbst, and S. Kounev. Modeling Variations in Load Intensity over Time. In *Proceedings of the 3rd International Workshop on Large-Scale Testing (LT 2014), co-located with the 5th ACM/SPEC International Conference on Performance Engineering (ICPE 2014)*. ACM, March 2014.