Using a Hardware Simulation for Automatic Software Performance Model Parameterization

Sebastian Weber
sebastian.weber@fzi.de
FZI Forschungszentrum Informatik

Abstract

While the fulfilment of functional requirements during software re-engineering or maintenance can likely be monitored with existing test cases, checking whether quality requirements (e.g. performance) are still satisfied requires additional effort. A developer would have to either measure the target system or analyse it based on models. The precise parameterization of these models is usually based on measurements which require an executable software and the target hardware. This paper proposes an approach to use hardware simulations for the automatic parameterization of performance models to remove the need for the target hardware. The results show that the accuracy of the hardware simulation for the chosen hardware requires improvements, if these results are intended to replace measurements. Nevertheless they show the applicability of the approach.

1 Introduction

Modeling and analysis of a system composed of software and hardware can be achieved with different approaches. The approach used to implement the hardware-simulation-based parameterization is the Palladio Approach [3] which aims at component-based software, because it offers comprehensive and extendable modeling and analysis tooling. The Palladio Component Model (PCM) consists of five different model types.

The repository model contains software components that can require and provide interfaces. If a component provides an interface it has to provide implementations for the methods of this interface. These implementations are called Service Effect Specification (SEFF) and describe which resources (e.g. HDD or CPU) are required for the execution of this implementation. The amount of work required on a resource is described on a high level of abstraction as work units which can for example be mapped to number of cycles for a CPU.

In the assembly model the components can be combined to a system according to the interfaces they require and provide. Also the interfaces provided by the system are specified. The allocation model describes how the elements of the assembly model are deployed on hardware which is modelled in the resource environment model. The last model is the usage model which defines how the system is used. The simulation of these models can be used to analyse different quality properties of the system. In this paper only the performance of the system is considered. The simulator used is SimuLizar [2].

As hardware simulation gem5 [1] was chosen, because it supports a multitude of different hardware targets and simulation granularities. Besides its high configurability it is actively developed and has a permissive license. A hardware specification and an executable are required to run a simulation. The hardware specification primarily consists of specifications of the CPU, cache, RAM and memory. Depending on the chosen granularity, different parameters can be adjusted.

The contribution of this paper is the combination of the Palladio approach with the hardware simulation gem5, which was done during a master's thesis [4], to automatically parameterize Palladio models based on given hardware simulation inputs. This allows developers to quickly and easily adapt the model of a system to changed hardware or software by simply adapting the required hardware simulation inputs in the models. Deployment and measurement of the system are not required as long as the chosen hardware simulation is accurate enough. Section 2 describes how the integration was implemented, Section 3 presents evaluation results and Section 4 concludes this paper with a summary and outlook.

2 Implementation

To be able to utilize gem5 for the parameterization of a PCM, the required input data for gem5 has to be stored in related models. The executables are stored in the SEFFs of the repository model instead of a specification of a demand in work units. The hardware specifications are attached to the corresponding resource containers. Possible parameters of the executables are stored in the usage model.

Palladio simulates the execution of the system under the given usage model. Whenever a method of the system is called, it forwards this call with possible parameters to the corresponding component. If this component needs resources according to the specification of the SEFF, this resource demand in work units will be scheduled on the corresponding resource indicated by the allocation of the component. If the resource demand is given as an executable it will be processed by a SimuLizar extension first.

The extension resolves the hardware specification based on the resource the demand should be scheduled on. The executable is already given in the SEFF and possible parameters are retrievable from the call. Subsequently the executable, the parameters and the hardware specification are forwarded to gem5.

During the execution of the hardware simulation, the simulation of Palladio is stopped. When the hardware simulation is finished, the results of it are converted to a number of work units which can be returned to Palladio. At this point the simulation of Palladio continues with the computed number of work units. By caching the results the number of time-consuming hardware simulation executions is reduced to one per hardware simulation input. This is viable due to gem5 being deterministic, so it always yields the same result.

3 Evaluation

The evaluation of the approach focuses on the achieved accuracy of gem5 and on the increased simulation time of Palladio when using the hardware-simulation-based parameterization. To evaluate the accuracy, the execution of three example applications is simulated with different parameters. These applications were the calculation of a MD5 hash from a file (MD5), the conversion of an audio file from "Waveform Audio File Format (wav)" to "Apple Lossless Audio Codec (alac)" (Alacconvert) and the computation of Fibonacci numbers (Fibonacci).

Table 1 shows the number of measured and simulated cycles together with their ratio with the given parameter or parameter properties in the first column. The ratio of the measured and simulated number of cycles converges to approximately 0.75 with increasing input size. This indicates that the hardware specification either was not parameterized precisely enough or does not contain the necessary parameters, e.g. the latency of cpu instructions. Due to the simulation times being between 10000 to 30000 times higher than the execution time of the software on the target hardware, no larger parameters were considered.

The simulation time of Palladio was about 150 times higher when evaluating resource demands with the hardware simulation for MD5 and Alacconvert, but these results are not transferable to larger applications or inputs. Nevertheless they show the high simulation times of hardware simulations on low levels of abstraction and their significant effect on software simulations on high levels of abstraction. Despite the induced overhead this approach should be cheaper and faster than building and measuring a test system for the model parameterization. If a test system is available it could be used in the parameterization process instead of a hardware simulation to reduce the simulation time. Using the hardware-simulation-based parameterization only for parts of the model and a manual parameterization for the rest can also reduce the induced overhead.

Parameter	Measurement	Simulation	Ratio
	MD5		
1MB	8.569.324	8.383.694	1,02
2MB	15.291.831	16.558.028	0,92
3MB	21.704.249	24.730.282	0,88
	Alacconvert		
1 sec	22.145.658	27.741.100	0,8
2 sec	38.884.680	53.229.985	0,73
	Fibonacci		
10	952.276	184.843	5,15
20	1.253.866	562.234	2,23
30	35.216.364	46.083.982	0,76
35	380.132.623	509.121.790	0,75

Table 1: Measured and simulated number of cycles for program execution

4 Conclusion

This paper discussed the usage of a hardware simulation to automatically parameterize a software performance model. The evaluation shows the applicability of the approach despite of the improvable accuracy. Future work is to test more precise parameterizations of the hardware specification to improve the accuracy. Furthermore other hardware simulation on the same or higher levels of abstractions should be used. This might reduce the simulation time without degrading the accuracy too much.

References

[1] N. Binkert et al. "The gem5 simulator". In: *ACM SIGARCH computer architecture news* 39.2 (2011), pp. 1–7.

[2] M. Becker, S. Becker, and J. Meyer. "SimuLizar: Design-Time Modeling and Performance Analysis of Self-Adaptive Systems". In: *Software Engineering 2013*. Ed. by S. Kowalewski and B. Rumpe. Bonn: Gesellschaft für Informatik e.V., 2013, pp. 71–84.

[3] R. H. Reussner et al. *Modeling and simulating software architectures: The Palladio approach.* MIT Press, 2016.

[4] S. Weber. "Co-Simulation of Hardware and Software in the Palladio Component Model". MA thesis. Karlsruhe Institute of Technology (KIT), 2022.