

Florian Fittkau

# Simulating Cloud Deployment Options for Software Migration Support



**Anchor Academic Publishing**

*disseminate knowledge*

**Fittkau, Florian: Simulating Cloud Deployment Options for Software Migration Support, Hamburg, Anchor Academic Publishing 2015**

Buch-ISBN: 978-3-95489-393-5

PDF-eBook-ISBN: 978-3-95636-352-8

Druck/Herstellung: Anchor Academic Publishing, Hamburg, 2015

**Bibliografische Information der Deutschen Nationalbibliothek:**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

**Bibliographical Information of the German National Library:**

The German National Library lists this publication in the German National Bibliography. Detailed bibliographic data can be found at: <http://dnb.d-nb.de>

All rights reserved. This publication may not be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

---

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Bearbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Die Informationen in diesem Werk wurden mit Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden und die Diplomica Verlag GmbH, die Autoren oder Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für evtl. verbliebene fehlerhafte Angaben und deren Folgen.

Alle Rechte vorbehalten

© Anchor Academic Publishing, Imprint der Diplomica Verlag GmbH  
Hermannstal 119k, 22119 Hamburg  
<http://www.diplomica-verlag.de>, Hamburg 2015  
Printed in Germany

## Abstract

Cloud computing is emerging as a promising new paradigm that aims at delivering computing resources and services on demand. To cope with the frequently found over- and under-provisioning of resources in conventional data centers, cloud computing technologies enable to rapidly scale up and down according to varying workload patterns. However, most software systems are not built for utilizing this so called elasticity and therefore must be adapted during the migration process into the cloud. A challenge during migration is the high number of different possibilities for the deployment to cloud computing resources. For example, there exist a plethora of potential cloud provider candidates. Here, the selection of a specific cloud provider is the most obvious and basic cloud deployment option. Furthermore, the mapping between services and virtual machine instances must be considered when migrating to the cloud and the specific adaptation strategies, like allocating a new virtual machine instance if the CPU utilization is above a given threshold, have to be chosen and configured. The set of combinations of the given choices form a huge design space which is infeasible to test manually. Simulating the different deployment options assists to find the best ratio between high performance and low costs.

For this purpose, we developed a simulation tool named CDOSim that can simulate those cloud deployment options. CDOSim integrates into the cloud migration framework CloudMIG Xpress and utilizes KDM models that were extracted by a reverse engineering process. Furthermore, it is possible to use monitored workload profiles as a simulation input. Our evaluation shows that CDOSim's simulation results can support software engineers to sufficiently accurately predict the cost and performance properties of software systems when deployed to private and real world public cloud environments such as Eucalyptus and Amazon EC2, respectively. Thus, CDOSim can be used for the simulation of cloud deployment options and assists to find the best suited cloud deployment option for existing software systems.



# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>                      | <b>1</b>  |
| 1.1      | Motivation . . . . .                     | 1         |
| 1.2      | Approach . . . . .                       | 1         |
| 1.3      | Goals . . . . .                          | 3         |
| 1.4      | Document Structure . . . . .             | 4         |
| <b>2</b> | <b>Foundations and Technologies</b>      | <b>5</b>  |
| 2.1      | Foundations . . . . .                    | 5         |
| 2.2      | Involved Technologies . . . . .          | 12        |
| <b>3</b> | <b>Simulation Input</b>                  | <b>21</b> |
| 3.1      | Overview . . . . .                       | 21        |
| 3.2      | MIPIPS . . . . .                         | 22        |
| 3.3      | Instruction Count . . . . .              | 27        |
| 3.4      | Weights per Statement . . . . .          | 40        |
| 3.5      | Network Traffic . . . . .                | 41        |
| 3.6      | SMM Workload Profile . . . . .           | 43        |
| 3.7      | Enriched KDM Model . . . . .             | 43        |
| 3.8      | Adaptation Rules . . . . .               | 44        |
| 3.9      | Configuration . . . . .                  | 44        |
| <b>4</b> | <b>Simulation Output</b>                 | <b>47</b> |
| 4.1      | Costs . . . . .                          | 47        |
| 4.2      | Response Times . . . . .                 | 47        |
| 4.3      | SLA Violations . . . . .                 | 47        |
| 4.4      | Rating . . . . .                         | 48        |
| <b>5</b> | <b>CloudSim Enhancements</b>             | <b>49</b> |
| 5.1      | Overview . . . . .                       | 49        |
| 5.2      | Enhanced CloudSim Meta-Model . . . . .   | 49        |
| 5.3      | CPU Utilization Model per Core . . . . . | 51        |

|          |   |            |
|----------|---|------------|
| 5.4      | Starting and Stopping Virtual Machine Instances on Demand . . . . . | 51         |
| 5.5      | Delayed Cloudlet Creation . . . . .                                 | 52         |
| 5.6      | Delayed Start of Virtual Machines . . . . .                         | 52         |
| 5.7      | Timeout for Cloudlets . . . . .                                     | 52         |
| 5.8      | Improved Debt Model . . . . .                                       | 53         |
| 5.9      | Enhanced Instruction Count Model . . . . .                          | 53         |
| 5.10     | History Exporter . . . . .  | 53         |
| 5.11     | Dynamic Host Addition at Runtime . . . . .                          | 54         |
| 5.12     | Method Calls and Network Traffic between Virtual Machine Instances  | 54         |
| <b>6</b> | <b>MIPIPS and Weights Benchmark</b>                                 | <b>57</b>  |
| 6.1      | Features . . . . .  | 57         |
| 6.2      | Design . . . . .  | 57         |
| 6.3      | Example Output . . . . .  | 59         |
| <b>7</b> | <b>CDOSim</b>   | <b>61</b>  |
| 7.1      | Features . . . . .  | 61         |
| 7.2      | The Simulation Process . . . . .                                    | 62         |
| 7.3      | Design . . . . .  | 63         |
| <b>8</b> | <b>Evaluation of CDOSim</b>   | <b>65</b>  |
| 8.1      | Goals of the Evaluation . . . . .                                   | 65         |
| 8.2      | Methodology . . . . .   | 66         |
| 8.3      | Basic Experiment Setup . . . . .                                    | 67         |
| 8.4      | E1: MIPIPS Benchmark Evaluation . . . . .                           | 74         |
| 8.5      | E2: Accuracy Evaluation for Single Core Instances . . . . .         | 80         |
| 8.6      | E3: Accuracy Evaluation for Multi Core Instances . . . . .          | 99         |
| 8.7      | E4: Accuracy Evaluation for Adaptation Strategy Configurations . .  | 104        |
| 8.8      | E5: Inter-Cloud Accuracy Evaluation . . . . .                       | 109        |
| 8.9      | Summary . . . . .   | 112        |
| <b>9</b> | <b>Related Work</b>   | <b>115</b> |
| 9.1      | GroudSim . . . . .  | 115        |
| 9.2      | Palladio . . . . .  | 115        |
| 9.3      | SLastic.SIM . . . . .   | 115        |
| 9.4      | iCanCloud . . . . .   | 116        |
| 9.5      | Byte Instruction Count for Java . . . . .                           | 116        |

|   |            |
|---|------------|
| 9.6 Measuring Elasticity . . . . .                    | 117        |
| 9.7 Dhrystone Benchmark . . . . .                     | 117        |
| 9.8 Cloudstone Toolkit . . . . .                      | 118        |
| <b>10 Conclusions and Future Work</b>                 | <b>119</b> |
| 10.1 Conclusions . . . . .                            | 119        |
| 10.2 Future Work . . . . .                            | 119        |
| <b>References</b>                                     | <b>121</b> |
| <b>A Glossary</b>                                     | <b>i</b>   |
| <b>B Ecore Model for MIPIPS and Weights Benchmark</b> | <b>iii</b> |
| <b>C KDM example</b>                                  | <b>v</b>   |
| <b>D Rating Algorithm</b>                             | <b>xi</b>  |
| <b>E Attachments</b>                                  | <b>xv</b>  |





## List of Figures

|    |  |    |
|----|--|----|
| 1  | Users and providers of cloud computing taken from Armbrust et al. [2]  | 8  |
| 2  | CloudMIG approach taken from Frey et al. [19]  | 10 |
| 3  | CloudSim architecture taken from Calheiros et al. [10]   | 13 |
| 4  | CloudMIG Xpress overview taken from Frey et al. [19]   | 14 |
| 5  | Extracted CloudSim meta-model  | 15 |
| 6  | Layers of KDM taken from Pérez-Castillo et al. [54]  | 16 |
| 7  | Example of determining the median of response times during phases of low CPU utilization in the dynamic approach | 30 |
| 8  | Enhanced CloudSim meta-model   | 50 |
| 9  | CPU utilization model example  | 52 |
| 10 | New scheduling example   | 55 |
| 11 | Java packages of the MIPIPS and weights benchmark  | 58 |
| 12 | GUI of the MIPIPS and weights benchmark  | 58 |
| 13 | Activities in CDOSim's simulation process  | 62 |
| 14 | Java packages of CDOSim  | 63 |
| 15 | GUI of CDOSim  | 64 |
| 16 | Deployment configuration for Eucalyptus  | 70 |
| 17 | Deployment configuration for Amazon EC2  | 72 |
| 18 | The used day-night-cycle workload intensity  | 73 |
| 19 | Average CPU utilization of allocated nodes in SingleCore.1 experiment  | 83 |
| 20 | Median of response times in SingleCore.1 experiment  | 84 |
| 21 | Average CPU utilization of allocated nodes in SingleCore.2 experiment  | 84 |
| 22 | Median of response times in SingleCore.2 experiment  | 85 |
| 23 | Average CPU utilization of allocated nodes in SingleCore.3 experiment  | 86 |
| 24 | Median response times in SingleCore.3 experiment   | 87 |
| 25 | Average CPU utilization of allocated nodes in SingleCore.4 experiment  | 87 |
| 26 | Median response times in SingleCore.4 experiment   | 88 |
| 27 | Average CPU utilization of allocated nodes in SingleCore.5 experiment  | 89 |
| 28 | Median response times in SingleCore.5 experiment   | 90 |
| 29 | Average CPU utilization of allocated nodes in SingleCore.6 experiment  | 91 |
| 30 | Median response times in SingleCore.6 experiment   | 92 |
| 31 | Average CPU utilization of allocated nodes in SingleCore.7 experiment  | 92 |
| 32 | Median response times in SingleCore.7 experiment   | 93 |
| 33 | Average CPU utilization of allocated nodes in SingleCore.8 experiment  | 94 |

|    |  |     |
|----|--|-----|
| 34 | Median response times in SingleCore.8 experiment . . . . .                               | 95  |
| 35 | Average CPU utilization of allocated nodes in MultiCore.1 experiment                     | 101 |
| 36 | Median response times in MultiCore.1 experiment . . . . .                                | 101 |
| 37 | Average CPU utilization of allocated nodes in MultiCore.2 experiment                     | 102 |
| 38 | Median response times in MultiCore.2 experiment . . . . .                                | 103 |
| 39 | Average CPU utilization of allocated nodes in Adaptation.1 experiment                    | 105 |
| 40 | Median response times in Adaptation.1 experiment . . . . .                               | 106 |
| 41 | Average CPU utilization of allocated nodes in Adaptation.2 experiment                    | 107 |
| 42 | Median response times in Adaptation.2 experiment . . . . .                               | 107 |
| 43 | Average CPU utilization of allocated nodes in PredictionAmazon.1<br>experiment . . . . . | 110 |
| 44 | Median response times in PredictionAmazon.1 experiment . . . . .                         | 111 |
| 45 | Ecore model for MIPIPS and weights benchmark as UML class diagram                        | iii |

## List of Tables

|    |  |     |
|----|--|-----|
| 1  | Overview of the preconditions for each instruction count derivation approach . . . . . | 28  |
| 2  | Example weights . . . . .  | 37  |
| 3  | Contained weight benchmarks . . . . .  | 42  |
| 4  | Simulation configuration parameters . . . . .  | 44  |
| 5  | Our Eucalyptus server . . . . .  | 69  |
| 6  | Our Eucalyptus configuration . . . . .   | 69  |
| 7  | Used instance types in Amazon EC2 experiments . . . . .                                | 71  |
| 8  | Default simulation configuration . . . . .   | 74  |
| 9  | Results for comparison MIIPS.1 . . . . .   | 76  |
| 10 | Results for comparison MIIPS.2 . . . . .   | 76  |
| 11 | Results for comparison MIIPS.3 . . . . .   | 77  |
| 12 | Results for comparison MIIPS.4 . . . . .   | 77  |
| 13 | Results for comparison MIIPS.5 . . . . .   | 78  |
| 14 | Overview of the relative error values for each scenario . . . . .                      | 113 |



# 1 Introduction

## 1.1 Motivation

Cloud computing is emerging as a promising new paradigm that aims at delivering computing resources and services on demand. To cope with the frequently found over- and under-provisioning of resources in conventional data centers, cloud computing technologies enable to rapidly scale up and down according to varying workload patterns. However, most software systems are not built for utilizing this so called elasticity and therefore must be adapted during the migration process into the cloud [46].

Here, the selection of a specific cloud provider is the most obvious and basic cloud deployment option. Furthermore, the mapping between services and virtual machine instances must be considered when migrating to the cloud and the specific adaptation strategies, like allocating a new virtual machine instance if the CPU utilization is above a given threshold, have to be chosen and configured. The set of combinations of the given choices form a huge design space which is infeasible to test manually [25].

The simulation of a cloud deployment option can assist in solving this problem. A simulation is often faster than executing real world experiments. Furthermore, the adaptation to the software system, that shall be migrated, requires less effort at a modeling layer. The simulation can be utilized by an automatic optimization algorithm to find the best ratio between high performance and low costs.

## 1.2 Approach

We begin with defining the fundamental concept of a cloud deployment option and describe our simulation approach.

**Definition 1** *In the context of a deployment of software on a cloud platform, a cloud deployment option is a combination of decisions concerning the selection of a cloud provider, the deployment of components to virtual machine instances, the virtual machine instances' configuration, and specific adaptation strategies.*

Definition 1 shows our definition of a cloud deployment option. The deployment of components to virtual machine instances includes the case that new components might be formed of parts of already existing components. By a virtual machine

instances' configuration, we refer to the instance type, as `m1.small` in the case of Amazon EC2, of virtual machine instances, for instance. Furthermore, an example for an adaptation strategy is “start a new virtual machine instance when for 60 seconds the average CPU utilization of allocated nodes stays above 70 %.”

For simulating a cloud deployment option, we basically need a cloud environment simulator. For this purpose, we utilize CloudSim [10]. There are various inputs that are required by CloudSim. For modeling a computation like an application call, named *Cloudlet* in CloudSim, CloudSim mainly requires the *instruction count* of the computation. The instruction count of a Cloudlet is a measure for the work that has to be conducted by the CPU. As a central input for modeling the capacity of virtual machine instances, CloudSim needs the mega instructions per second (MIPS) of the virtual machine instance. MIPS are a measure for the computing performance of the virtual machine instance. CloudSim does neither define a method for deriving the instruction count nor the MIPS. Furthermore, CloudSim does not specify which instructions are meant.

We assume that CloudSim requires instructions on a language level, e.g., *double divide* and *integer minus*, and that these instructions all equally flow into the MIPS value. Hence, we consider MIPS as too coarse grained because different instructions have different runtimes in general. Therefore, we define the measure mega integer plus instructions per second (MIPIPS). The measurement of MIPIPS should be separate from the actual simulation software because it has to be run on the virtual machine instances to measure their MIPIPS, for example. In accordance to MIPIPS, the instruction count unit of a Cloudlet has to be in integer plus instructions. Other instruction types must be converted to these integer plus instructions by weights that will also be measured separately from the actual simulation software.

To rate the suitability of a specific cloud deployment option, the simulation has to compute some information like costs for the given cloud deployment option. Furthermore, the outputs of a simulation run have to be comparable to the outputs of other simulation runs. This leads to the need for a rating approach.

A further requirement for the simulation results from the wide range of programming languages supported by different cloud providers. Infrastructure-as-a-Service (IaaS) providers typically support all programming languages because they are only providing the infrastructure computing resources. Therefore, we need a language independent simulation. For this purpose, we utilize the Knowledge Discovery Meta-Model (KDM) that provides information about the existing software system in a language independent way.

CloudMIG [15] provides a promising approach to assist in a migration project to a cloud environment. There also exists a prototype implementation, called CloudMIG Xpress [18], that implements this approach. Our software, named Cloud Deployment Options Simulator (CDOSim), for realizing the simulation contributes to CloudMIG Xpress as a plug-in. It utilizes workload profiles that can be modeled by the user or can be imported from monitoring data that were recorded by, for instance, Kieker [70].

### **1.3 Goals**

Our main objective is a software that enables the simulation of cloud deployment options on a language independent basis. For this purpose, we define the following goals.

#### **1.3.1 G1: Definition of the Simulation Input**

The definition of the simulation input should be accomplished by goal G1. MIPIPS and instruction count was already described as an input. However, there are more. Furthermore, where appropriate, derivation methods for the input parameter should be developed or defined.

#### **1.3.2 G2: Definition of the Simulation Output**

In goal G2 the output of the simulation should be defined. Furthermore, a metric for comparing the cloud deployment options in respect to the output should be developed.

#### **1.3.3 G3: Development of a Benchmark for Measuring the Computing Performance of a Node in MIPIPS**

In G3 a benchmark for measuring the computing performance of a node in MIPIPS, that can be easily adapted to new programming languages, shall be developed. It shall include a GUI and a console interface because virtual machine instances can often only be accessed via a command shell.

#### **1.3.4 G4: Development of CDOSim**

The last goal is the development of a software that realizes the simulation. Furthermore, it shall be integrated into CloudMIG Xpress as a plug-in. We name this software CDOSim. To achieve the programming language independence, CDOSim shall operate on KDM instances.

### **1.4 Document Structure**

*The remainder of the thesis is structured as follows.* Section 2 outlines the foundations and utilized technologies. Afterwards, Section 3 presents the simulation inputs and how they can be derived (G1). Then, Section 4 describes the simulation output (G2) and a rating approach for rating simulation runs relatively to each other. The enhancements we needed to conduct for CloudSim are listed in Section 5. The following Section 6 describes our MIPIPS and weights benchmark (G3). Our developed tool for simulating cloud deployment options, named CDOSim, is discussed in Section 7 (G4). The following Section 8 evaluates the functionality and accuracy of CDOSim. Then, Section 9 describes related work. The final Section 10 concludes the thesis and defines the future work.