Florian Fittkau

Simulating Cloud Deployment Options for Software Migration Support

Master's Thesis

Bibliographic information published by the German National Library:

The German National Library lists this publication in the National Bibliography; detailed bibliographic data are available on the Internet at http://dnb.dnb.de .

Copyright © 2012 Diplom.de
ISBN: 9783961162475

https://www.diplom.de/document/280022

Florian Fittkau

Simulating Cloud Deployment Options for Software Migration Support

Diplom.de

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 accurate 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

List of Figures

List of Tables

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.

2 Foundations and Technologies

Sections 2.1 to 2.2 provide an overview of the foundations and technologies that will be used in later sections.

2.1 Foundations

The following Sections 2.1.1 to 2.1.5 describe the foundations.

2.1.1 Cloud Computing

Cloud computing is a relatively new computing paradigm. Therefore, many definitions for cloud computing exist. Here, we use the National Institute of Standards and Technology (NIST) definition by Mell and Grance [42] because this definition has become a de-facto standard.

The NIST definition for cloud computing defines five essential characteristics that a service must fulfill in order to be a cloud service, for example, on-demand self-service. Furthermore, it describes three different service models. These are IaaS, Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). They differ in the levels of abstraction with regard to configuration and programming options. Clouds can be deployed according to four different deployment models. These are public clouds, private clouds, hybrid clouds, and community clouds. In addition, Armbrust et al. [2] define different role models for users and providers of cloud computing services.

Essential Characteristics

The NIST definition for cloud computing defines five essential characteristics that a service must fulfill in order to be a cloud service. These are listed and described below.

1. On-demand self-service

A user can rent computing capabilities like storage and computing time on demand in an automatic way without human interaction of the service provider.