

Chapter 1

A Roadmap for Software Engineering for the Cloud: Results of a Systematic Review

Abhishek Sharma

University of Calgary, Canada

Frank Maurer

University of Calgary, Canada

ABSTRACT

This chapter presents the results of a systematic review from existing literature in software engineering for cloud-based applications and describes what the research community thinks about the effects of introducing cloud computing into a software development process. In this systematic review, the authors describe the challenges cloud computing poses for software development. They particularly investigate whether agile methodologies are beneficial or not in developing software that will be deployed in the cloud. In their results, they found that industry practitioners in their blogs and self-reported reviews indicate that agile development and cloud computing goes well together and that further investigation is required to confirm this claim.

INTRODUCTION

Cloud computing is a relatively recent term that is built upon decades of research in various fields within Computer Science from networking over distributed computing to web and software services. Cloud is emerging as a model that works on the principle of everything-as-a-service (XaaS) (Banerjee et al., 2011). While cloud-based solu-

tions are adopted in Industry with a great pace, there appears to be little research on the impact of this new technology on software development processes.

There are various software engineering issues that have to be addressed when the cloud becomes the target deployment environment. For example, when we plan to migrate software testing to cloud (Parveen & Tilley, 2010) we need to decide upon the characteristic of the application that will be under test and type of testing to be performed.

DOI: 10.4018/978-1-4666-4301-7.ch001

Another example can be the adequacy of current software process models when development involves cloud (Guha & Al-Dabass, 2010).

WHAT IS CLOUD COMPUTING?

Before presenting our review, let us understand what cloud computing is. The US National Institute of Standards and Technology (NIST) defines cloud computing as:

A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.-The NIST definition of cloud computing, 2011

The NIST definition is one of the most comprehensive definitions and is widely referenced. Furthermore, this definition describes cloud computing as having five essential characteristics, three service models and four deployment models. The essential characteristics are:

- **On-Demand Self-Service:** Computing resources can be acquired and used at any time without requiring human interaction with each services provider. Computing resources include processing power, virtual machines, storage etc.
- **Broad Network Access:** Above mentioned resources can be accessed over a network using heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
- **Resource Pooling:** The computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dy-

namically assigned and reassigned according to consumer demand. There is a sense of transparency in it, since a consumer can decide the amount of resources that are required. The customer generally has no control or knowledge over the exact geographical location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center).

- **Rapid Elasticity:** A user can quickly scale out on resources, and these resources can be rapidly released to scale in. To the consumer end, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
- **Measured Service:** Cloud systems automatically control and optimize resource use by leveraging a metering capability on metrics like CPU hours, bandwidth, storage etc.

The above-mentioned characteristics apply to all the clouds, but abstraction at service level differ at the provider level. The level of abstraction is defined in the service models provided by NIST:

- **Software as a Service (SaaS):** Users can access the application, which is offered as a service, using thin client interfaces such as web-browsers. An example of SaaS is web-based email. Cloud users have no control at this level. The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities. One possible exception to these rules is the presence of limited user-specific configuration settings.

14 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/roadmap-software-engineering-cloud/77696

Related Content

Filter/Wrapper Methods for Gene Selection and Classification of Microarray Dataset

Norreddine Mekour, Reda Mohamed Hamouand Abdelmalek Amine (2019). *International Journal of Software Innovation* (pp. 65-80).

www.irma-international.org/article/filterwrapper-methods-for-gene-selection-and-classification-of-microarray-dataset/230924

Fuzzy Adaptive Controller for Synchronization of Uncertain Fractional-Order Chaotic Systems

Amel Bouzeriba (2018). *Advanced Synchronization Control and Bifurcation of Chaotic Fractional-Order Systems* (pp. 190-217).

www.irma-international.org/chapter/fuzzy-adaptive-controller-for-synchronization-of-uncertain-fractional-order-chaotic-systems/204801

Adaptive Replacement Algorithm Templates and EELRU

Yannis Smaragdakisand Scott Kaplan (2010). *Advanced Operating Systems and Kernel Applications: Techniques and Technologies* (pp. 263-275).

www.irma-international.org/chapter/adaptive-replacement-algorithm-templates-eelru/37953

Model Based Process to Support Security and Privacy Requirements Engineering

Shareeful Islam, Haralambos Mouratidis, Christos Kalloniatis, Aleksandar Hudicand Lorenz Zechner (2012). *International Journal of Secure Software Engineering* (pp. 1-22).

www.irma-international.org/article/model-based-process-support-security/69391

Designing Mobile Aspect-Oriented Software Architectures with Ambients

Nour Aliand Isidro Ramos (2012). *Handbook of Research on Mobile Software Engineering: Design, Implementation, and Emergent Applications* (pp. 526-543).

www.irma-international.org/chapter/designing-mobile-aspect-oriented-software/66485