Assisted migration of enterprise applications to the Cloud – A hybrid Cloud approach

Adrián Juan-Verdejo
CAS Software A.G

Abstract

Cloud Computing is a relatively new paradigm with the potential to transform how IT hardware and software are designed and purchased. Computing is no longer purchased as typical products but delivered as a service over the Internet from large data centres. However, despite the potential benefits associated with the migration of enterprise applications from an in-house data centre into a Cloud infrastructure, there are still some issues that hinder the process. In this respect, a migrated application should satisfy specific enterprise policies related to privacy and security, as well as provide an acceptable quality of service. Because of the complexity of enterprise applications today with respect to the large number of components deployed in multi-tier architectures, the complexity of the interactions between them and the components' dependence on stored data, the migration process has to be assisted. Therefore it is presented a framework able to automate the decision-making process related to the migration of different components to a virtualized Cloud system based on a model which factors in benefits and disadvantages associated with the migration subject to the fulfilment of the SLA and the respect of policy constraints. The presented approach envisions an application hosted both partly on-premise and on the Cloud. Resulting migrated applications need to be compliant with enterprise and security policies' constraints while minimizing costs and ensuring performance in terms of wide area network communications and response times. The framework relies on a model to efficiently decide on the component placement. A model which takes into account several factors such as enterprise policies, dynamic performance bottlenecks, data sensibility, performance, cost savings from migration, data flows between application components, and the spread and variability of users. On the one hand the migration could lower the SLA (Service Level-Agreement) while on the other hand the deployment on the Cloud would provide the ability to handle peaks in workload and a higher reliability due to the higher level of replication, the existence of multiple fault domains and the deployment of the application on multiple Clouds. With respect to the *latter, it will be assessed the deployment on different Clouds depending on the constraints imposed by them. The presented framework relies on algorithms and a formal decision model in order to assist developers in migrating existing applications. With the aim of empirically evaluate the overall framework, a prototypical implementation of the framework will be used in order to support the migration to a Cloud-based environment of an existing product configurator. The empirical validation will be used to further improve the input model on which the decision-making process was founded.*

1 Introduction

Due to the fast-paced development of both processing and storage technologies, and the great adoption of the Internet, computing resources have become ubiquitously available at a cheaper price while offering powerful solutions. This trend has enabled the delivery of computing as a utility, with its resulting potential to significantly transform the information technology (IT) world (Fox & Griffith, 2009). Over the last few years, not only early adopters but an increasing number of enterprises are being attracted to Cloud-based deployments due to the promises of reducing the costs associated with their IT-related activities while maintaining, if not increasing, the quality of the service provided. Both the potential advantages and the initial success stories of Cloud Computing adoption inspire enterprises to migrate their existing applications to a Cloud-based architecture. However beneficial, the adoption of this new paradigm by enterprises implies a fundamental change in information technology provision that affects how the services they offer are devised, developed, deployed, scaled, tested, improved, updated, and even charged for (Armbrust et al., 2010). Furthermore, in addition to the technical implications of the embracement of Cloud Computing, the socio-technical and organizational effects of the adoption have to be considered (Khajeh-Hosseini, Sommerville, & Sriram, 2010). Enterprise decision makers should take a holistic approach to analyze the different implications related to the adoption of this relatively new technology. This approach considers both the technical and business-related consequences of the adoption.

From a socio-technical viewpoint, there are legal and regulatory constraints — related to enterprise-specific policies, industry-specific laws and regulations, and national privacy requirements – that have to be respected after the application and enterprise data have been migrated. On the other hand, from a more technical standpoint, the Quality of Service (QoS) requirements of the end-users have to unarguably be met in order to conform to the Service-level agreement (SLA). As a result of this service contract, developers usually have to cope with stringent requirements with regard to availability, performance and delay, while at the same time they are required to respect the aforementioned data privacy regulations. Moreover, costs in terms of wide-area communications resulting from the actual migration have to be contemplated as well. All these issues call for a framework to simplify the task of migration of legacy applications to the Cloud.

2 Motivation

In an effort to provide a solution for these concerns, the architecture is based on a hybrid deployment in which some parts of the application are migrated to the Cloud whereas some others are hosted on-premises within the boundaries of the enterprise to which the service is being delivered. (Calheiros, Ranjan, Beloglazov, De Rose, & Buyya, 2011). A framework is supplied in order to help professionals by providing them with a mechanism to increase application deployment flexibility, as components and data can be migrated to the Cloud while some other parts of the application are kept locally. The decision on what to migrate hinges on a strategy that strives for balance in performance, cost reduction, availability, data privacy, and enterprise policies.

As an example, let us assume that an enterprise aiming for cost reduction plans the migration to the Cloud of an application accessing patients' healthcare data. Due to national privacy requirements, databases which store sensitive data (back-end tier) have to be kept locally. This design decision might affect the performance of those application components which frequently retrieve data from local databases. The system could suffer from increased delays and response times because of the existence of interdependent parts of the application which run at separated locations. Therefore, in this case the wiser solution might be to keep locally both the data and the components which access them very often, in order to enhance performance. On the other hand, less sensitive components would be migrated to the Cloud and can be remotely accessed. As evidenced by this example, the decision of what components to migrate or to keep locally is a challenging one. Hence, the presented framework will base the deployment decision on a component placement model.

The described model for component positioning applies to typical enterprise applications today, which usually present a multi-tiered architecture. These applications are usually composed of lots of components which interact and depend on each other in a non-trivial manner. These facts motivate the delivery of a model to improve the migrating decision by taking into account many factors, namely data privacy, legal and regulatory constraints (e.g., enterprise-specific constraints), dynamic performance bottlenecks, cost reduction due to migration, data flows between application components (transactions could suffer from increased delays), increased reliability due to the presence of multiple fault domains, wide-area communication costs of migrations, and spread and variability of users.

The model and algorithms are evaluated through the application of the introduced framework to realistic migration scenarios. Specifically, the case study uses a product configurator as the enterprise application targeted to be migrated to the Cloud.

At an early stage, the migration process takes into account two locations, local versus Cloud data-centre. However, the described approach can be extended to the migration to multiple Cloud locations running in different Cloud providers' architectures. Optionally, the different characteristics of the Clouds available in the market can be incorporated into the model. The different set of characteristics offered by Cloud providers can be used to decide which Cloud suits the application to be migrated better.

3 Related Work

The related work section will start with a general description of cloud computing together with references to some positioning papers. Next, the reasons for a company to adopt this relatively new paradigm will be described in detail using case studies and best practices. Finally, some frameworks and simulation tools related to the migration of legacy applications to the cloud will be listed.

For some years now Cloud Computing has been a promising area for both scientists and professionals. Cloud Computing emerged as a natural evolution of a combination of virtualization, utility computing and distributed computing (Armbrust et al., 2010). Despite the initial mismatch and overabundance of Cloud Computing definitions, an agreement has been reached as to specify Cloud Computing based on the US National Institute of Standards and Technology definition of Cloud Computing (US NIST, 2009). According to their definition, Cloud Computing is *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*. A system offering a Cloud Computing service should present five essential characteristics; namely, on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. In literature new Cloud-based service models are presented as everybody strives to offer *everything* as a Service. Nevertheless, only three service models are commonly accepted, namely Software as a Service, Platform as a Service, and Infrastructure as a Service. Nowadays research institutions tend to build upon the NIST definition striving for standardization (Vaquero, Rodero-Merino, Caceres, & Lindner, 2008) (Vouk, 2008). Some of these surveys base their line of argument on companies' reports which aim at providing a global overview of the opportunities for IT Cloud services (IDC, 2010). Those company reports state how Cloud Computing is an appealing topic not only for academia but for enterprises as well.

Enterprises are attracted by the benefits of migrating to a Cloud-based architecture. However, Cloud Computing not only represents a chance to technically improve modern data centres but it also entails an important change in how services are both provisioned and used (Fox & Griffith, 2009). Therefore, professionals have to consider both the benefits and risks of migrating to a Cloud-based deployment. Moreover, more subtle changes have to be considered as well, such as the organizational changes caused once the responsibilities of the IT department are shifted outside of the organization to external companies, e.g. Amazon (Khajeh-Hosseini, Sommerville, et al., 2010). In literature researchers strive to illustrate the real-life implications of the migration of existing applications to the Cloud by performing case studies with different approaches either technically oriented, more business-oriented or something in between.

On the one hand, some take a technical approach on how to tackle the challenges related to the carrying out of the actual migration or the implications for the software architecture of planning the migration of an application. Some authors (Babar & Chauhan, 2011) focus on the analysis of the requirements of the application to be migrated in order to identify the architectural modifications needed to Cloud-enable an existing application. They state that current architecture evaluation methods do not effectively assess the architecture decisions before their implementation as they do not draw attention to Cloud-related quality features such as scalability and accessibility. Likewise, security aspects have to be borne in mind and integrated in the migration process of legacy systems (Rosado, Gómez, Mellado, & Fernández-Medina, 2012). Due to the fact that each service provider has its own identity management system, a user needs to have multiple digital identities. Therefore, they suggest the use of federated identity management in order to uniquely identify users for each of the services accessed (Seung, Lam, Li, & Woo, 2011).

On the other hand, some authors stress the economic and operational implications of including Cloud Computing in an organization, yet ignoring the inherent technical challenges (Marston, Li, Bandyopadhyay, Zhang, & Ghalsasi, 2011). Following this trend, (Hajjat et al., 2010) study how the overall migration cost is the result of a complex combined effect of applications characteristics in terms of workload intensity, storage capacity, growth rate, and the cost of software licenses. They conclude that horizontal partitioning between in-house and Cloud deployments can be very beneficial for certain applications. (Khajeh-Hosseini, Greenwood, & Sommerville, 2010) shows a 37% system infrastructure cost reduction. Moreover, they exemplify the operational implications of the migration as approximately 1 out of 5 support calls for the analyzed system were avoided.

Furthermore, some authors avoid taking sides and have adopted a more holistic approach as they abandon the typical reductionist approach and try to emphasize the study of complex systems. (Mohagheghi & Sæther, 2011) have developed an agile, model-driven, tool-supported methodology for the migration to the service Cloud paradigm. They built upon SMART (Lewis, Morris, Smith, & Simanta, 2008) and included business processes into their methodology. Nevertheless, they did not incorporate legal and regulatory constraints. Following this holistic trend, some researchers supply frameworks which consider multiple criteria in order to assess the migration to the cloud.

The *CloudGenious* framework takes into account a large set of heterogeneous criteria and their interdependencies. However, they apply their framework to single-tiered applications. Therefore, they leave out a lot of current enterprise applications which are more complex than that (Menzel & Ranjan, 2011). A similar approach is taken in *Cloudward Bound* as (Hajjat et al., 2010) present a modelling technique for the migration of enterprise applications to the Cloud. Nevertheless, they forgot to include the socio-technical implications of the migration. As validation, they evaluate their algorithms in the migration of an ERP system to the Cloud. Cloud Motion (*CMotion*) is another framework devised to help researchers to solve the issues related to the migration of heterogeneous composite applications. The very nature of these applications, which consist of heterogeneous components noncompliant with any specific interface, hinders the migration process. *CMotion* effectively addresses this problem whereas it does not incorporate enterprise or government policy support into the decision system (Binz, Leymann, & Schumm, 2011). *CloudMIG* is a framework which automates parts of the migration process, such as the extraction of application requirements, the selection of a Cloud-provider and finally the generation of the target architecture (Frey & Hasselbring, 2010). However, they still have to improve their framework to enable it to evaluate migrated applications. Following the same trend, *CloudFlex* focuses on the technical issues related to the migration of a typical three-tiered enterprise application. *CloudFlex* uses load balancing to solve problems related to dynamic performance bottlenecks which appear after the migration but disregards the socio-technical effects of the migration (Seung et al., 2011). The different frameworks explained are often validated using ad-hoc validation methods. In an attempt at facilitating these tasks, some researchers have presented their own framework for modelling, simulation, and experimentation.

With the aim of assisting researchers in the evaluation of the performance of Cloud provisioning policies, application workload models and resources performance, there have been some attempts to develop simulators of Cloud Computing architectures. CloudSim (Calheiros et al., 2011) models VM allocation, network and data centre energy consumption but they focus on a lower level of abstraction, the Infrastructure as a service (IaaS) level. On the other hand, the presented approach works on the Software as a Service (SaaS) level.

4 Approach

4.1 Overview

As stated in the Motivation section, the main goal of this work is to deliver a framework able to automate the decision-making process related to the migration of different components to a virtualized Cloud system. Given the complexity of the migration strategy selection, the process is based on a model (*see* 4.2)4.2 which factors in benefits and costs associated with the migration subject to the fulfillment of the SLA and the respect of policy constraints. It will help developers to perform the efficient migration of existing enterprise applications by adopting a hybrid migration approach. According to this approach, some of the applications' components are kept locally while others are migrated to a Cloud-based infrastructure. The prototypical implementation of the framework determines a migration strategy by using a decision support tool based on a multi-criteria model for component placement. Those criteria consider not only applications' technical characteristics but they also take into account business-related constraints and requirements. The applicability, validity and effectiveness of both the overall framework and the presented model are evaluated.

In order to validate the described approach the migration of an enterprise will be evaluated. The framework will be employed for the migration of a legated Product Configurator system to a hybrid cloud deployment. The Product Configurator is the real application intended to be migrated to a new Cloud-based deployment and the findings obtained during the validation will provide feedback in order to enhance the component placement model and the decision support part of the framework.

Additionally, this work might consider the specific characteristics of the different Cloud providers in order to incorporate them into the component placement model. The migration of a component to a certain Cloud provider instead of to another can be more beneficial due to the characteristics offered by that specific provider. Moreover, this functionality offers different alternatives in order to avoid the vendor lock-in problem.

4.2 Modelling of components placement

The proposed model can be further explained by the description of the criteria taken into account for the modelling process. Namely, the following:

Legal and regulatory constraints: often disregarded in literature but very relevant to the approach taken by this work. In this classification, enterprise-specific constraints, industryspecific laws and regulations, and national privacy requirements are contemplated.

Dynamic performance bottlenecks (*see Dynamic Performance Bottlenecks,* also called choke points (Seung et al., 2011)).

Cost reduction thanks to the migration because of the migration of both computation and storage to a Cloud environment.

Data flows between application components Increased transactional delays between components which interact very often. The transaction sizes and frequency will be taken into account together with the transaction delays means and variances. Moreover, the transaction origin and destination will be considered (internal users, external users, different components).

Wide-area communications cost of the actual migration arising from the migration of both data and application state to the Cloud nodes.

Spread and variability of users

1 Within or outside the enterprise premises

Data privacy and sensibility. Due to the impossibility of migrating sensitive data (e.g. credit card numbers), those components highly dependent on these data could be forced to be kept locally in the environment of the frequently accessed data.

The model performs application discovery tasks in order to detect dependencies and interactions between components. Moreover, the model needs input related to the predicted response times of these interactions and how much traffic will happen between the application components.

1.1 Example of the migration of a typical enterprise system

With the aim of clarifying the approach taken, an example of the migration of a typical enterprise system will be presented in two figures. This example is not founded on any empirical result but it intends to illustrate the presented approach by showing a specific scenario. Figure 1, depicts a typical example of an enterprise system running off-premises, everything runs on the client side and the provider must deploy the application on the client's infrastructure. As in typical enterprise systems today, the system consists of many interdependent components, namely the *migration target component 1*, *2*, and *3*, the *s1.1* and *s1.2 components*, and the *client-side component*. Moreover, the system interacts with two database management systems, namely *DBMS1* and *DBMS2*. The *migration target components 1, 2,* and *3* are the components intended for migration and they depend on other components and databases. The *client-side component* presents the data to the user. Finally, the *s1.1* and *s1.2 components* are two components which do not belong to the application to be migrated but to system 1 (*s1*). Nevertheless, the *migration target component 2* interacts with them and therefore *s1.1* and *s1.2* are of relevance to the migration strategy selection. The system does not use virtualization and is not highly scalable.

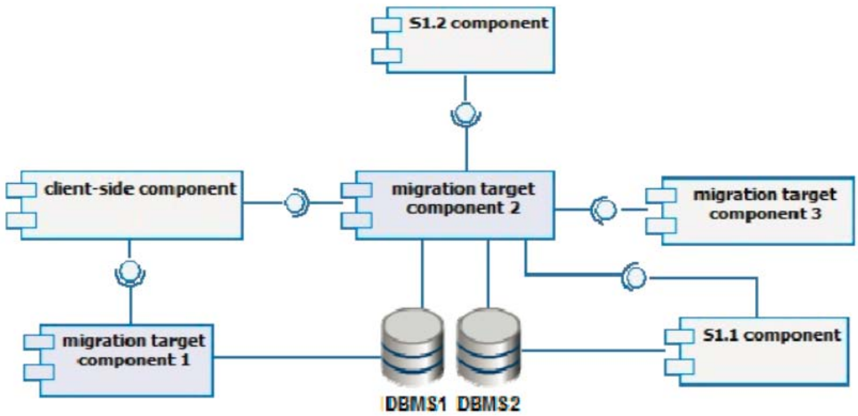

*Figure 1: Typical Enterprise System today, on-premises deployment*2

In Figure 2 a new node is included, namely the Cloud provider. The Cloud provider runs the migrated components, namely the *s1.2 component*, the *migration target components 1* and *3*. On the left side, i.e. the client side, a new component has been created, namely a *broker,* as a result of the migration of parts of the system to the Cloud. Part of the knowledge previously on the migrated components has been transferred to the *broker*. Surprisingly enough, the *migration target component 3* could not be migrated due to its high reliance on database *DBMS1*. It might be because the *migration target component 3* accesses the database (*DBMS1*) too often or because the traffic load between them is very high. Therefore, moving this component out of the local infrastructure could harm the overall system performance. On the other hand, with respect to the migrated components, a component which was not targeted (*s1.2 component* in Figure 2) has been shifted to the Cloud provider along with *migration target components 2* and *3*. This decision could be taken in order to avoid an SLA infringement due to an increased transactional delay in case the *s1.2 component* would have been kept on the client side.

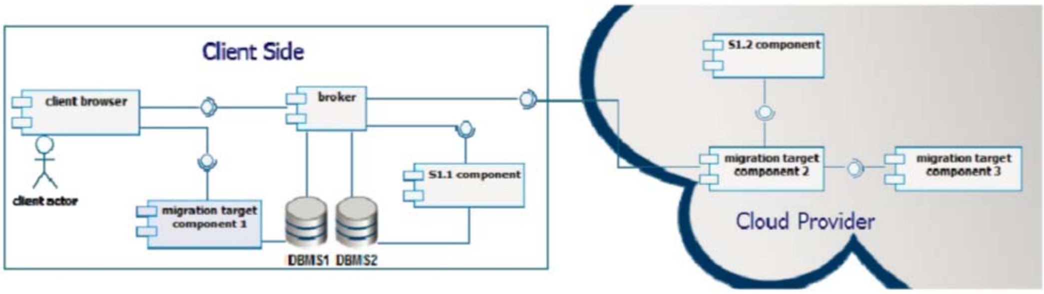

Figure 2: Hybrid migration approach

Dynamic Performance Bottlenecks

Typical enterprise applications today, like the one presented, employ multi-tiered architecture with lots of components which depend on each other. This fact entails some design challenges which have the potential to hinder the migration process. In this respect, it is important to bear in mind applications' performance bottlenecks and their dynamic nature. As an application scales in a horizontal fashion, a dynamic performance bottleneck might move from one part of the system to another.

Figure 3: Typical three-tiered enterprise application

In Figure 3, a typical three-tiered enterprise application is depicted in order to exemplify the concept of dynamic performance bottleneck and how it affects a migrated application. It does therefore not arise from any empirical finding but it is just an illustrative example.

At the bottom of Figure 3, the back-end layer (Tier-1) consists of a database which interacts with the components in the immediately superior level (Tier-2), which belong the business logic layer. The business logic components query the database in order to get information related to the sales made the previous year. Finally, the layer at the top, the front-end layer, presents the data to the user (Tier-3).

Such a deployment has some effects on the dynamic performance bottlenecks as they might shift to different parts of the system as the system scales out. Let us assume that in Figure 3 the components of the business logic layer (Tier-2), which get their data from the bottom layer, are the current bottleneck of the application. As a result of the utilization of the proposed framework, Tier-2 components might be selected as target components to be migrated to the Cloud in order to keep up with the SLA. Let us assume that due to this decision the system performance keeps on improving until so many requests are sent to the back-end layer that its components cannot respond timely to the requests of the upper-layer. At this point, the dynamic performance bottleneck has been shifted from where it initially was, Tier-2, to the back-end layer. The depicted framework might then decide, by using the presented model (*see* Modelling of components placement), that the bottom layer should be scaled out in an attempt to improve the system performance. However, the requirements related to enterprise policies and data privacy are also factored in the presented model. Therefore, the decision support system could bring about a different migration strategy in order to meet those requirements too.

1.1 **Evaluation**

Aiming at evaluation, a legated Product Configurator will be migrated to a virtualized environment by using the provided framework. As a result the application will be hosted partly on the Cloud and locally. It will be used as a case study to show the advantages of a hybrid approach as a cloudenabling. Moreover the validation will show that the model for component placement effectively works and respect the SLA and enterprise-specific constraints. Firstly, the application used as a case study to exemplify the advantages of the presented approach will be modelled in order to describe it in terms of the nature of the users (internal, external) and its components (back-end, business logic, front-end). Moreover the transactions from the users to the front-end components (thin clients) or business logic components (thick clients), and the transactions between components will be analyzed. Once this model is ready, the typical communications happening in the system will be measured on an end-to-end fashion as well as the transactions between users and components and between different components will be assessed. As a result, communication delays between local components, migrated components, internal users, and external users will be estimated. In addition to the overall transaction sizes and their frequency, the transactions between components will be carefully taken into account as they could have a large effect on the migration strategy. As an example, the migration of a business logic component to a Cloud environment might not be beneficial whereas its migration together with the back-end it deeply rely on in order to satisfy users' requests, may imply great economical gain at little transactions cost. The presented framework will use the model of the application in order to estimate the migration benefits and communication costs in order to recommend a migrated deployment (in case it is beneficial) depending on the maximum delays allowed for the application and their variance. On the one hand, the benefits will be calculated from the migration to the Cloud of both computation and storage (Khajeh-Hosseini, Greenwood, et al., 2010). On the other hand, the communication costs are related to the migration of data and application state to the Cloud (Armbrust et al., 2010). Additionally, it will be verified that the traffic is routed accordingly to the final migrated deployment in order to profit from the new architecture. For example, users external to the architecture are routed to the components on the Cloud as far as it is possible.

2 Research Method

It is argued for a design-oriented information systems approach (Österle et al., 2010), in which a system is build up bearing the conceptual design in mind while taking into account both applicable restrictions and limitations. However, the precise steps might either change as this work evolves and some steps may be executed iteratively until an appropriate outcome is reached. Nine steps have been identified: 1. Conduct a literature review on: Cloud Computing and Internet data centres, case studies, frameworks and approaches to the migration of legacy applications to the Cloud, descriptive modelling techniques, application discovery, and distributed system economics. 2. Conduct interviews or have informal meetings with other member of the scientific and professional communities in order to clearly identify and further define the relevant problems. 3. Choose the correct Hybrid Cloud approach to create a framework for the migration of legacy enterprise applications to the Cloud. 4. Clearly define the modelling of components' placement. 5. Use the migration of a product configurator to the Cloud as a real-world use case to exemplify and evaluate the framework and defined models. 6. Conduct experiments based on a real-world use case to be deployed in an enterprise environment with the resulting realistic evaluation. 7. Statistically analyse the performed measurements and use those results in order to improve the modelling of component placement. 8. (Optional) Study the migration to different Cloud providers depending on the specific requirements of the application. 9. Gather all the experience, observations and experimental results in a doctoral dissertation.

3 Conclusions and Acknowledgements

Although the Cloud offers many benefits, the migration to a cloud-based architecture needs to be carefully planned. Organizations need to take many criteria into account in order to adopt an adequate migration strategy to select the appropriate parts of the application and target infrastructure for the migration. This paper calls for a framework to assist in the migration following a hybrid Cloud deployment in which parts of the application are kept locally and which parts are migrated. Decision making is supported by a model for component placement. Finally, both the framework and models are validated in a real-case scenario, namely a Product Configurator. This research has been supported by the FP7 Marie Curie Initial Training Network "RELATE" for research on Engineering and Provisioning of Service Based Cloud Applications. Grant Agreement No. 264840. **7**

4 References

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., . . . Stoica, I. (2010). A view of cloud computing. *Communications of the ACM, 53*(4), 50-58.

Babar, M. A., & Chauhan, M. A. (2011). *A tale of migration to cloud computing for sharing experiences and observations.* Paper presented at the SECLOUD '11 Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing.

Binz, T., Leymann, F., & Schumm, D. (2011). *CMotion: A Framework for Migration of Applications into and between Clouds*.

Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., & Buyya, R. (2011). CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience, 41*(1), 23-50.

Fox, A., & Griffith, R. (2009). Above the clouds: A Berkeley view of cloud computing. *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS, 28*.

Frey, S., & Hasselbring, W. (2010). *Model-Based Migration of Legacy Software Systems to Scalable and Resource-Efficient Cloud-Based Applications: The CloudMIG Approach*.

Hajjat, M., Sun, X., Sung, Y. W. E., Maltz, D., Rao, S., Sripanidkulchai, K., & Tawarmalani, M. (2010). *Cloudward bound: planning for beneficial migration of enterprise applications to the cloud*.

IDC. (2010). Cloud Computing. An IDC Update. Khajeh-Hosseini, A., Greenwood, D., & Sommerville, I. (2010). *Cloud migration: A case study of migrating an enterprise it system to IaaS*.

Khajeh-Hosseini, A., Sommerville, I., & Sriram, I. (2010). Research challenges for enterprise cloud computing. *Arxiv preprint arXiv:1001.3257*.

Lewis, G. A., Morris, E. J., Smith, D. B., & Simanta, S. (2008). Smart: Analyzing the reuse potential of legacy components in a service-oriented architecture environment: DTIC Document.

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing—The business perspective. *Decision Support Systems, 51*(1), 176-189.

Menzel, M., & Ranjan, R. (2011). CloudGenius: Automated Decision Support for Migrating Multi- Component Enterprise Applications to Clouds. *Arxiv preprint arXiv:1112.3880*. Mohagheghi, P., & Sæther, T. (2011). *Software Engineering Challenges for Migration to the Service Cloud Paradigm: Ongoing Work in the REMICS Project*.

Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Sinz, E. J. (2010). Memorandum on design-oriented information systems research. *European Journal of Information Systems, 20*(1), 7-10.

Rosado, D. G., Gómez, R., Mellado, D., & Fernández-Medina, E. (2012). Security analysis in the migration to cloud environments. *Future Internet, 4*(2), 469-487.

Seung, Y., Lam, T., Li, L. E., & Woo, T. (2011). *CloudFlex: Seamless scaling of enterprise applications into the cloud*. US NIST, M., P. Grance, T.;. (2009). The NIST definition of cloud computing *National Institute of Standards and Technology* (Vol. 53, pp. 50). Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2008). A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review, 39*(1), 50-55.

Vouk, M. (2008). *Cloud computing—Issues, research and implementations*.