Migration of Legacy Applications to the Cloud

A Review on Methodology and Tools for Migration to the Cloud

Alexis Wallskog Pappas
Abstract

Many organizations have legacy applications and strive to modernise them in order to react on changes and adapt to the new environment, the cloud. The enticements are quite a few but the risks are lurking as well. How to migrate a legacy application to the cloud is an unanswered question for many organizations. We look at how research has answered this question and the methods and tools they provide. The research partially answers the question of migration of legacy application to the cloud. The methods and tools are still quite granular, not that automated and is very dependent of what type of legacy application and the aim of the end result is.
Acknowledgements

I wish to thank my supervisor Pedher Johansson for words of encouragement and constructive criticism.

Friends, classmates and family have been at my side trough three years of hard work, long days, many laughs which has led up to this thesis.

Last but definitely not least I want to thank Malin Sundberg for supporting and pushing me to complete this project and my studies.
# Contents

1 Introduction  
   1.1 Background  
   1.2 The Cloud  
   1.3 Cloud migration  
2 State-of-the-Research  
3 Discussion  
4 Conclusions  
References
1 Introduction

Over the past 10 years cloud computing has evolved to become one of the clearest trends as a platform for software applications/systems. Since it promises unparalleled levels of scalability and flexibility [16], gives the organizations vast opportunities for more efficient solutions and the possibility to reach more customers with new business models. Although cloud computing is a new technology, it is in many ways an unknown territory, and still under development. The fact that the industry faces a new paradigm shift is nothing new. This has been the case several times throughout the history of computer science, but it introduces new software engineering challenges [21]. Many seek to find the best way to migrate their old on-site applications/systems to the cloud in a general and service provider independent way but how to do this is an unanswered question for many organizations [22]. This has highlighted the importance of providing processes and frameworks to support a systematic migration to the cloud [19, 13]. This paper will investigate what different studies have come up with, what the differences and commonalities among them are. Based on this, the aim is to see if there is a practice covering most of the aspects for migration of legacy application to the cloud.

1.1 Background

Since 2007 a number of projects and research have been funded or partially funded by EU's Seventh Framework Programme for Research (FP7) [5]. The purpose of these are to investigate how to ease migration of legacy software applications to the cloud. Each project and research have approached it from different angles. This paper focuses on the results from some of the research done within these EU funded projects. To get a context for the research there is a need to describe these projects.

The REMICS (REuse and Migration of legacy applications to Interoperable Cloud Services [11]) project has its starting point in Architecture-Driven Modernization (ADM) by Object Management Group (OMG) [4]. REMICS strive to develop efficient languages, agile methods and tools to provide a model-driven methodology, which significantly improve the baseline ADM concepts, to support migration to the cloud [11].

MODAClouds has the main goal to provide methods, a decision support system and the high-level design. As an extension of the REMICS project the MODAClouds project aims to leverage the results of REMICS [10].

The ARTIST (Advanced software-based seRvice provisioning and migraTIon of legacy SofTware [1]) project addresses that most non-cloud applications are not suited for the cloud for a number of reasons, e.g. scalability. ARTIST applies model-
driven engineering techniques for a structured approach to modernisation of legacy software to the cloud. This includes a number of methods and tools that not only assist the migration, they also help the application to take full advantage of cloud features, e.g. scalability; elasticity [1]. The ARTIST project consists of three main phases that act as the baseline for all migration tools developed: a) Pre-migration: Migration Feasibility Assessment, b) Migration: Application Discovery & Understanding and Migration, and c) Provisioning: Testing, Verification & Certification [1, 14].

1.2 The Cloud

Cloud computing, or just the cloud, is a service of delivering shared resources, software and information over a network (typically the internet) to computers and other devices as a utility. The cloud focuses on maximizing the effectiveness of shared resources and by that achieve coherence and economies of scale. This is achieved by multiple users sharing the resources and dynamically reallocate the resources per demand. By using the cloud, organizations move from investing and managing hardware, e.g. computer resources, to buying it as a service and paying for the amount they use it. [20, 12]

The notion of the cloud or cloud computing is to group together several forms of service models and types of implementation. The US National Institute of Standards and Technology (NIST) define cloud computing as follows.

"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 a minimal management effort or service provider interaction." [20]

These service models are divided into three types:

The **Software as a Service (SaaS)** approach, where the service provider makes the software available in the form of an internet service, e.g. Microsoft Office 365 [9], Apple iCloud [3]. The service provider manages and controls the underlying cloud infrastructure, the customer access and use the software and is able, to some extent, configure application settings. The customer pays for the amount it uses the software or through a monthly fee. [16, 20, 21]

The **Platform as a Service (PaaS)** approach, where the service provider makes a development environment and/or an API available, e.g. Microsoft Azure [8], Google App Engine [6]. This includes the operating system as well as a set of services dedicated to development, testing, deployment and hosting. The customer/organization merely create the applications within the existing development environment. Just as for SaaS the customer does not control or manage the underlying cloud infrastructure but the customer controls the deployed software and configuration settings for the software-hosting environment. [16, 20, 21, 22]

The **Infrastructure as a Service (IaaS)** approach, where the service provider
offers infrastructure for computing, storage and other fundamental computing resources. This is done through a virtualization platform, i.e. virtual machines and network management, where customer/application developers can procure and operate. In this service model the customer can more freely choose what to deploy and run on the cloud infrastructure. IaaS is seen as the most basic form of a cloud service. [16, 21, 22]

There are also different type (or deployment models) of clouds, each meeting specific needs. There are three main types of clouds, public, private and hybrid. In the case of a service provider wishing to provide a cloud service to public users over the internet, e.g. Amazon Web Services (AWS) [2], Google Cloud Platform [7], Microsoft Azure [8], we talk about a public cloud [21, 13, 20]. The opposite of that is when an organization installs its own farm of servers and deploys a cloud computing infrastructure for its exclusive use [21, 20]. In some cases an organization wishes to combine a private and public infrastructure to gain more flexibility in the distribution and larger control over sensitive information. This is called a hybrid cloud [21, 20]. A less common type of cloud is a community cloud [20]. In this case the cloud infrastructure is provisioned exclusively for a specific community of customers from organizations with common concerns [20].

1.3 Cloud migration

What is migration of legacy application to the cloud? Pahl et al. defines cloud migration as follows; "Cloud migration is the process of partially or completely deploying an organization’s digital assets, services, IT resources or applications to the cloud” [22].

Historically, software migration has been the interest of the software developers and to some extent the business side of the organization. The role of the business side of the organization were mainly to set the economical boundaries for the migration project. Migration to the cloud put more focus on the business side of the organization and involves more parties than the developers. A product delivered through the cloud gives the organization both new opportunities to increase their income but on the other hand they also have to keep control of the expenses. Most likely the organization has to change their business model which is something that effects the majority of the company and in particular will have an effect on the choice of technical solution. [14, 15, 18]

Quite early in the migration process the stakeholders and product owners have to decide if it makes sense to migrate the legacy application at all [21, 18]. The general focus of the research today is the migration of legacy application to make it a cloud-enabled software, i.e. adapt the application to be suited for the cloud. The opposite to that is cloud-native applications, software specifically implemented for the cloud or legacy applications re-written from scratch to suit the cloud [15]. Cloud-native applications themselves and the migration process from legacy application to cloud-native application are not a part of the scope in this paper.


## 2 State-of-the-Research

According to Comella-Dorda et al. [17] there are two main ways how to modernise a legacy application: white-box and black-box modernisation. White-box involves re-structuring, re-architecting and re-implementing the application while the black-box modernisation focuses on the input/output of the application. The white-box approach requires a knowledge of the internal parts of an application. [17] This is something that Jamshidi et al. [19] confirms with their *Cloud Reference Migration Model* (Cloud-RMM). The Cloud-RMM uses both a bottom-up approach that identifies low-level activities and techniques which are then categorised to form generalised tasks and processes and a top-down approach that forms a framework consisting of process, task, activities and techniques. Closely related tasks, in terms of input and output artifacts, identifies categories that together with involved activities form key processes for migration. Jamshidi et al. consider any activities that are a part of the on-site installed software and that preserve the core functionalities after the migration to the cloud platforms to be regarded as migration tasks [19]. In table 1 the processes and tasks that are included in the Cloud-RMM migration framework are presented [19].

<table>
<thead>
<tr>
<th>Processes</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning</td>
<td>Feasibility study, Requirements analysis,</td>
</tr>
<tr>
<td></td>
<td>Decisions of providers and services, Migration strategies</td>
</tr>
<tr>
<td>Execution</td>
<td>Code modification, Architecture extraction,</td>
</tr>
<tr>
<td></td>
<td>Data extraction, Transformation</td>
</tr>
<tr>
<td>Evaluation</td>
<td>Deployment, Testing, Validation</td>
</tr>
<tr>
<td>Crosscutting concerns</td>
<td>Governance, Security, Training, Effort estimation,</td>
</tr>
<tr>
<td></td>
<td>organizational change, Multitenancy</td>
</tr>
</tbody>
</table>

In the process of developing the Cloud-RMM Jamshidi et al. have used the migration type classification by Andrikopoulos et al. [15]. The research done by Andrikopoulos et al. is based on the migration of a on-site application with a three-layered architecture [15]. The three-layered architecture comprises of a data layer, business layer and presentation layer. They address the general challenges on a per- and cross-layer basis when migrating to the cloud. The conclusion that they have made is that there are four types of migration:

**Type I: Replace** component(s) with cloud offerings. This is the type of migration that is least invasive. One or more (architectural) components, data and/or business logic, are replaced by cloud services. To use Google App Engine Datastore in place of a local MySQL database is an example of this migration.
Type II: Partially migrate some of the application functionality to the cloud. Migrating a set of architectural components from one or more layers implementing a particular functionality to the Cloud [15].

Type III: Migrate the whole software stack of the application to the cloud. This is the classic example of migration to the cloud. Where the application is encapsulated in virtual machines and is run on the cloud [15].

Type IV: Cloudify the application: a complete migration of the application takes place. The application functionality is implemented as a composition of services running on the cloud. As in the case of component replacement (Type I migration), cloudification requires the migration of data and business logic to the cloud [15].

Mohagheghi et al. [21] addresses gaps in the current migration methodology that the REMICS project has identified. The lack of comprehensive methodology, addressing dedicated design patterns in architecture migration methods, emphasises the need of integration between model-based development and migration tools and methods. Further they foresee the necessity for platform independent modelling for cloud computing. [21]

In Alonso et al. [14] they investigate the first of the three core phases in the ARTIST project, the Pre-migration phase, and focuses on the costs, Return Of Investment (ROI), efforts and what the process may imply for the organization. The ARTIST pre-migration phase evaluates the feasibility of the migration through technical and business analysis. This provides information of potential cost and the efforts required of the organization to carry out the migration to the cloud. The evaluation is made in three main steps: 1) Maturity assessment, 2) Technical Feasibility Analysis, and 3) Business Feasibility Analysis. What Alonso et al. proposes in their Cloud Modernization Assessment Framework [14] are three tools that matches these steps. A Maturity Assessment Tool (MAT) to analyse the current (initial) and desired (final) situation of the application that is going to be migrated. Unlike other maturity approaches the MAT focuses on two perspectives, business and technical in both situations (initial and final). The Technical Feasibility Tool (TFT) assists the early evaluation of the technical feasibility of the migration that consists of identifying the components of the legacy application that are affected by the migration and to estimate the efforts required to migrate these components. The third and final tool, Business Feasibility Analysis Tool (BFT) aims to support decision makers to estimate costs, benefits and operational risks of migration to cloud deployment. The purpose of the three tools is to advise software companies whether to migrate the application or to start development from scratch. [14]

Succeeding the research that Ahmad and Babar have performed in [16, 19] they proposed a new framework, the Legacy-to-Cloud Migration Horseshoe in "A Framework for Architecture-driven Migration of Legacy Systems to Cloud-enabled Software" [13]. This framework purposes to support a process-driven and incremental migration of a legacy application to a cloud-based service architecture [13]. The Legacy-to-Cloud Migration Horseshoe framework has been developed from the Cloud-RMM
and shares some structural similarities. In table 2 the complete set of processes and activities belonging to the Legacy-to-Cloud Migration Horseshoe framework is presented [13].

<table>
<thead>
<tr>
<th>Processes</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture Migration Planning</td>
<td>Feasibility study, Requirements analysis, Decisions on cloud providers, Migration strategies development</td>
</tr>
<tr>
<td>Architecture Recovery and Consistency</td>
<td>Consolidation of source code, Extraction of patterns and styles, Legacy architecture Description, Architecture consistency conformance</td>
</tr>
<tr>
<td>Architecture Transformation</td>
<td>Architecture change implementation, Architecture property preservation, Architecture transformation patterns</td>
</tr>
<tr>
<td>Architecture-based Development of Cloud-enabled Software</td>
<td>Service architecture description, Application of patterns and styles, Cloud enabled code generation, Code consistency conformance</td>
</tr>
</tbody>
</table>

The architecture migration planning process with its activities results in a migration plan. This migration plan is in turn the input to the architecture recovery and consistency process. The aim of this process is to deliver a legacy architecture, based on the legacy code, to the architecture transformation process. The outcome of the transformation process is a cloud-service architecture. The final process in the horseshoe is the architecture-based development process. Here the cloud-enabled source code is generated. [13]

A broader management perspective is what Pahl et al. have in their research "A comparison of on-premise to cloud migration approaches" [22]. They investigated the migration process by conducting expert interviews and focus groups with major international cloud solution providers and independent consultants. The research has focused on determining the principle cloud migration process and they show differences between the cloud models IaaS, PaaS and SaaS. [22]

A slightly different approach to migration of legacy application is presented by Gunka et al. [18]. In their case study they use the tools, methods and modelling techniques from the MODAClouds project [10]. This led to an evolutionary migration strategy where the transition to cloud is made in three main steps 1) porting the application to an IaaS, 2) adding load balancing on the presentation and business logic layer, and 3) partially moving the application to a PaaS.

Gunka et al. get support from Wendland et al. [23] in their view of an evolutionary approach on the migration process. Although in Wendland et al. the research revolves around testing and test architecture in the migration process of a legacy application [23].
3 Discussion

In our opinion it is not, in the current state, possible to say that there is a practice covering most aspects of how to migrate legacy application to the cloud. This is not possible even if one would focus on just one area, such as time, cost or security.

In this paper we have seen that interesting progress have been made regarding the development of migration tools and methods. The four migration types, replace, partially migrate, migrate the whole software stack or cloudify as proposed by Andrikopoulos et al. [15] have gained acceptance and are being used by other researchers. This can be exemplified in the work done by Jamshidi, Ahmad and Pahl [19] which have used the results of Andrikopoulos et al. as a foundation in their work on developing a migration framework, and in the research done by Ahmad and Babar [13].

Even though these four migration types have been defined from a migration process for a three-layered application, research shows that they can be applicable on other types of legacy applications. This points out a direction for future work and ongoing migration processes and can be seen as an overhead structure. Still, how to break down each type of migration has to be done in a more specific way since there are too many aspects to a migration, e.g. the legacy applications are too diverse, the intent of the organizations are not conclusive and different cloud services have their own installation processes. In particular the intent of the organization plays a much bigger role in the cloud migration process than in traditional software migration and this steers the technical choices that are made and makes everything more complex. This together with the differences in the cloud service models, SaaS, PaaS and IaaS, makes it hard to create general migration tools and methods.

In a migration process and in the final cloud solution there is always a risk that information which is sensitive for the organization can be exposed. To reduce such risks and better utilize the advantages of cloud computing one should consider a hybrid cloud solution and/or to use an evolutionary approach during the migration process. The evolutionary approach better helps the organization to have control over the sensitive data during the migration whereas the private cloud gives better control of the security for sensitive data in the final cloud solution. To use a public cloud towards the customers gives the opportunity to utilize the advantages of cloud computing, e.g. scalability and availability.

There is a complexity to this area which makes it hard to, in a comprehensive manner, get a grip on it. This together with the fact that there are a lot to gain if one succeeds to find one or more comprehensive tools and methods which in a both technical and business oriented way makes migration from legacy application to the cloud effective is what makes it so interesting to investigate further and continue research on.
4 Conclusions

This paper presents the state-of-the-art on migration to the cloud from a on-site legacy software application. The potential for effective migration to the cloud is large, and some methods, that might be seen as general in some sense, have been developed like the Legacy-to-Cloud Migration Horseshoe framework [13]. However there is no solution that fit all types of legacy software applications migration. The organization has to take in consideration what kind of application they have and what result they aim to achieve with the migration. Decisions made by the organization will determine the choice of methodology and how the migration will be executed. What is certain is that no matter what tools or methodologies an organization choose the migration will require a lot of time, planning and manual work.

In the current situation there are no methods and tools covering most of the aspects on how to migrate a legacy application to the cloud. It partially comes down to the fact that this research area is still relatively new, but the main reason is that a migration to the cloud from a legacy application is complex, more complex than traditional software migrations since more aspects has to be taken in to consideration. There are more parties involved and a larger part of the decisions than before are linked to the business view of the migration.

With this said, research has still come a long way the last couple of years, a clear progress can be seen. This indicates that a lot of discoveries have been made, but there are many more things to be discovered in future research. New discoveries will be made almost no matter what since the interest for this area is large both from the industry and the academia but to keep up the fast pace and rapid development will require that EU funded projects like REMICS and MODAClouds or new projects will continue to receive research funding.

Our assessment is that it is not likely that we in the future will have a practise covering most aspects of a migration to the cloud from a legacy application. It is more likely that a palette of methods and tools will be developed. Which methods and or tools one choose will depend on the type of legacy application, requested cloud service and cloud service provider.
Bibliography


