Chapter 75
Pattern-Based Cloud Migration: Take Blockchain as a Service as an Example

Zhitao Wan
Ge Lian Corporation, China

ABSTRACT

To migrate on-premises business systems to the cloud environment faces challenges: the complexity, diversity of the legacy systems, cloud, and cloud migration services. Consequently, the cloud migration faces two major problems. The first one is how to select cloud services for the legacy systems, and the second one is how to move the corresponding workload from legacy systems to cloud. This chapter presents a total cloud migration solution including cloud service selection and optimization, cloud migration pattern generation, and cloud migration pattern enforcement. It takes the pattern as the core, and unifies the cloud migration request, the cloud migration service pattern, and the cloud migration service composition. A cloud migration example of blockchain system shows that the proposed approach improves the cloud service selection, cloud migration service composition generation efficiency, migration process parallelization, and enables long transaction support by means of pattern reuse.

INTRODUCTION

With the skyrocketing of the cloud computing deployment, the demands of cloud migration increase dramatically. To migrate the on-premises business systems to the cloud environment faces challenges (Linthicum, 2017). Firstly, the complexity of business systems and the diversity of operating environment lead to complex initial states of the cloud migration. Secondly, the diversity of the cloud computing environment causes the complexity of the target environments. Thirdly, the diversity of cloud migration services worsens the cloud migration. Besides, the cloud migration is also subject to the constraints of time, cost and other non-technical factors. The cloud migration plan generation faces two major problems. The first one is how to plan the target cloud environment and cloud service selection of the legacy systems. The current cloud migration approaches usually adopt experts' recommendation basing on the

DOI: 10.4018/978-1-7998-5351-0.ch075

users' prefers and constraints (Al-Masri & Mahmoud, 2007a, 2007b). The automation mechanisms are usually unavailable. The second one is how to perform the cloud migration, i.e., how to move the corresponding workload from legacy systems to the cloud services. Manual migration is expensive, time consuming and error prone. Recently dedicated cloud migration services emerge in the market for fine granular cloud migration (Huang, Gao, Zhang, & Xiao, 2017).

However, migrating existing on-premises enterprise applications to cloud is still a costly, labor-intensive, and error-prone activity due to the complexity of the applications, the constraints of the clouds, and the limitations of existing migration techniques provided by migration service vendors. It fails to handle complex legacy system migration. Approaches have been proposed to find out the most cost effective solution by composing multiple migration services from different vendors together to complete specific migration task (Fan, Wang, & Chang, 2011; Frey & Hasselbring, 2010). They are sandbox approaches and these only provide one-off solutions by calculating without verified precedent. These approaches adopt exhaustive searching based algorithm with pruning to improve the efficiency. In these sandbox approaches, the metric for migration service selection and composition is usually simple, e.g., only the total cost. In fact, the reliability, privacy and other nonfunctional constraints should be considered as well. Besides, there are other nontechnical factors that will impact the selection. For example, a customer may prefer services from specific vendors. More critically, a case by case solution discovery approach has not explicitly process logic for tracing, benchmarking, debugging and optimization.

Fortunately, pattern has been proven as an appealing approach to accelerate the service composition and alleviate the defects (Ejarque, Micsik, & Badia, 2015; Tilsner, Fiech, Zhan, & Specht, 2011). It can also be applied to the cloud migration service composition to solve the mentioned challenges (Emna, Jmaiel, Dupuy, & Tazi, 2012). Pattern is basing on the fact that an idea has been proven useful in one practical context and will probably be useful in others (Yan, Dijkman, & Grefen, 2010). Even though, every system has its own set of prerequisites, hidden costs, one-off requirements and special case exceptions, the best practices could tell us how to cope with these issues. More specifically, service patterns are defined over services and present the typical ways of composing services to achieve certain goals. The pattern in this chapter refers service composition pattern dedicates to the service composition for the cloud migration. Service composition patterns facilitate the service composition and accelerate the response to the market. This is exactly why the patterns are appealing as a medium to convey solutions.

A complete on-premises application/system migration to cloud generally involves two major phases: the target cloud selection and the migration process. The target cloud service selection and optimization is to find out proper cloud services. The requirements for each node can be described as cloud service parameters. This chapter presents a clustering based approach to select corresponding cloud services, and also introduces a cloud layer based node merging and splitting approach. Thus solving the problem that it is difficult to quickly and automatically select cloud services for complex system migration out of a large number of cloud services. This chapter also presents a cloud migration service composition approach, which adopts pattern based approach by classifying and analyzing of service composition approaches. It can be abstracted as graph generation problem and further the problem of service composition pattern generation is abstracted as graph similarity calculation, which simplifies the pattern generation. In this way, it solves the problem that the cloud migration service composition pattern cannot be effectively generated in the rapid evolution environment that lacks precedence. Besides, this chapter also presents cloud migration service pattern enforcement approach, which improves the service composition generation efficiency, parallel execution efficiency and enables long transaction support by means of pattern reuse and bipartite graph analysis. It solves the problem that it is difficult to quickly form a cloud migra-