

A Generalized Framework for Legacy Application Modernization to the Cloud

*Thesis submitted in partial fulfillment of the requirements for the award of degree
of*

Master of Engineering

in

Software Engineering

Submitted By

Suman Jain

(801331029)

Under the supervision of:

Dr. Inderveer Chana

Associate Professor



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

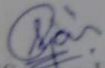
PATIALA – 147004

July 2015

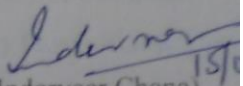
Certificate

This is to certify that the work presented in this thesis entitled "*A Generalized Framework for Legacy Applications Modernization to the Cloud*", submitted in partial fulfilment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department, Thapar University, Patiala, is an original record under guidance and supervision of *Dr. Inderveer Chana* and refers other researcher's work duly mentioned in the reference section.

The work presented in this thesis has not been submitted for award of any other degree of this or any other university.


(Suman Jain)

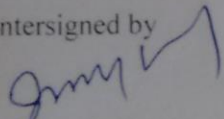
I hereby certify that the above statement made by the candidate is correct and true to the best of my knowledge and belief.


(Dr. Inderveer Chana) 15/11

Associate Professor

Computer Science and Engineering Department

Countersigned by

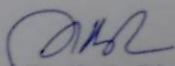

(Dr. Deepak Garg)

Head

Computer Science and Engineering Department

Thapar University

Patiala


(Dr. S. S. Bhatia)

Dean (Academic Affairs)

Thapar University

Patiala

Acknowledgement

This thesis work has been realised with the kind support and guidance of many individuals throughout the course work. I would like to extend my grateful thanks to all of them.

Foremost, I would acknowledge this endeavour to our almighty God for the knowledge and blessings he bestowed upon me to show right direction for this research.

I would express my sincere gratitude towards my respected supervisor, Dr. Inderveer Chana for the consistent guidance and encouragement I have received. She helped me with every aspect of the thesis right from the conceptual understanding to the resources arrangement.

I am also obliged to Dr. Deepak Garg, Head, Computer Science and Engineering Department, Thapar University for his constant supervision and guidance in completing this research. I also offer my gratitude to all the faculty and staff members of Computer Science and Engineering Department for the cooperation and help they have offered.

I would like to express my heart-felt appreciation to my family for the continuous encouragement with love and support they have offered to me.

Also, I wish to thank all my friends and colleagues for their advices, support without which this work would not have been accomplished.

Suman

The ever-growing demand of cloud computing services has presented a major impact on the technologies being used for the IT service provisioning. This change has been benefited in the form of increased business agility, scalability, flexibility and reduced operational costs of the computing systems.

The pay-as-you-use model of computing as utility model has been facing obstacles in myriad of traditional on premise computing systems which are offering needful services with legacy computing environment. These systems have been built on older programming environments with sequential programming methods but the valuable business logic and data being handled cannot be discarded. This legacy problem is one of the main challenges for cloud computing to deal with. When re-architecting and reengineering these legacy systems, parallel computing approaches also need to be assessed for resource optimization in cloud environments.

The proposed approach presents a generalized framework for modernization of the legacy applications to the cloud. Beginning from the source application assessment to the target infrastructure decision making process, essential quality characteristics are also considered to handle this multi-dimensional issue. The decision support module is followed by the elaboration of available modernization approaches and their associated risks with implementation phases. Additionally, this work explores the usage of parallel programming methodologies to deal with existing sequential approaches and large datasets are introduced with Hadoop architecture. The configuration and deployment of applications with parallelism presents a powerful manner of these cloud computing technologies.

Table of Contents

Certificate	i
Acknowledgement	ii
Abstract	iii
List of Figures	vi
List of Tables	viii
Chapter 1 Introduction	1
1.1 Cloud Computing.....	1
1.1.1 Cloud Service Layers	3
1.1.2 Cloud Deployment Models	4
1.1.3 Cloud Actors	5
1.1.4 Cloud Service Providers	6
1.1.5 Benefits of Cloud Computing	7
1.2 Legacy Systems	9
1.3 Challenges of Cloud Computing for Legacy Systems.....	11
1.4 Parallel Computing in Cloud	12
1.4.1 Hadoop	13
1.5 Research Motivation.....	16
1.6 Thesis Outline.....	17
Chapter 2 Literature Review	19
2.1 Legacy Applications Modernization to the Cloud	19
2.2 Related work.....	22
2.2.1 Impact of cloud computing on software development	23
2.2.2 Existing Migration Techniques for the Cloud.....	23
2.2.3 Industrial Contribution for Modernization.....	26
2.2.4 Use of Parallel Computing Approaches for Cloud Migration.....	26
2.2.5 Data Migration to the Hadoop/MapReduce	27
2.2 Analysis	29
2.3 Conclusion	30
Chapter 3 Research Gap and Problem Formulation	31
3.1 Gap Analysis	31
3.2 Research Objectives	32
3.3 Conclusion	32
Chapter 4 The Proposed Framework	33

4.1	Architecture of Framework	33
4.2	Decision Support Module	33
4.2.1	Source Application Assessment	35
4.2.2	Quality of Services.....	37
4.2.3	Target Cloud Assessment	39
4.3	Selection of Modernization Approach.....	41
4.3.1	Replacement (full/Partial).....	41
4.3.2	Wrapping to other services	42
4.3.3	Reengineering	42
4.3.4	Migration to Cloud services	43
4.4	Implementation of Selected Approach	48
4.5	Conclusion	49
Chapter 5 Case Study.....		50
5.1	Existing Sequential Implementation.....	50
5.2	MapReduce Approach	51
5.2.1	Amazon Web Services	52
5.3	Results	61
Chapter 6 Conclusion and Future Scope		64
6.1	Conclusion	64
6.2	Thesis Contribution	64
6.3	Future Scope.....	65
References		66
List of Publications		72
YouTube Link.....		73

List of Figures

Figure 1.1: Cloud Computing Framework.....	2
Figure 1.2: Cloud Service Models	3
Figure 1.3: Cloud Actors.....	6
Figure 1.4: Cloud Computing Service Providers	7
Figure 1.5: Comparison of Resource Provisioning.....	8
Figure 1.6: Evolution of an Application	10
Figure 1.7: Hadoop Architecture	14
Figure 1.8: MapReduce Programming Model	15
Figure 2.1 Evolution of Cloud Computing	20
Figure 2.2: Extract, Transform and Load Process Model.....	28
Figure 3.1: Problem Formulation Process	31
Figure 4.1: Application Modernization Framework	34
Figure 4.2: Architecture Layers in an Application	36
Figure 4.3: Scalability in an Application	38
Figure 4.4: IaaS Technology Stack.....	45
Figure 4.5: Migrating to SaaS.....	46
Figure 4.6: PaaS Migration.....	47
Figure 5.1: Amazon Web Services	53
Figure 5.2: Amazon EMR Interaction	55
Figure 5.3 Signing Up Process for AWS	55
Figure 5.4: Logging in to AWS Console	56
Figure 5.5: AWS Management Console Home	56
Figure 5.6: Data Bucket Creation in Amazon S3	57
Figure 5.7: Folders in Data Bucket	57
Figure 5.8: Data Files Uploaded to Bucket.....	57
Figure 5.9: Cluster Creation.....	58
Figure 5.10: Cluster Configuration	58
Figure 5.11: Hardware Configuration.....	59
Figure 5.12: Hadoop Configuration.....	59
Figure 5.13: Security Key Pair.....	59
Figure 5.14: Mapper-Reducer Configuration	60

Figure 5.15: Launching the Cluster	60
Figure 5.16: Resource Provisioning by Amazon	61
Figure 5.17: Graphical Representation of Performance Analysis	62

List of Tables

Table 1.1: Legacy System Characteristics	10
Table 2.1: Status of Modernization Projects	21
Table 2.2: The Systematic Review Process	22
Table 2.3: Data Migration Tools for Hadoop Framework	29
Table 4.1: Cloud Service Model Selection Criteria	39
Table 4.2: Replacement Approach.....	41
Table 4.3: Wrapping Approach.....	42
Table 4.4: Reengineering Approach	43
Table 4.5: Comparative Analysis of Leading Cloud Providers	48
Table 4.6: Implementation Tasks.....	49
Table 5.1: Validation of Modernization Framework	52
Table 5.2: Amazon Instances and Use Cases.....	54
Table 5.3: Running Time of Sequential and Parallel Approaches.....	62

Chapter 1

Introduction

This chapter introduces the research area of the thesis as modernization of legacy applications, involving the issues concerned with the migration in cloud. The ultimate goal of the research work is to incorporate its findings in a framework of modernization of any application into cloud. It will address multiple benefits and challenges involved in cloud adoption.

1.1 Cloud Computing

Utility based provisioning of computational resources over the network on rental basis unlike centralised computing is the basic idea of cloud computing. It is rather difficult to give a clear and precise definition for cloud computing because of it being continuously evolving technology. According to various researched, three vital characteristics of cloud computing are virtualization of IT resources, scalability and pay-as-per-you-go utility model. Vaquero et al [37] describe cloud computing as-

“Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs.”

Cloud computing has brought a revolutionary change in computing paradigm and consumers of software industry, system developers and business means. The traditional on-premises enterprise systems are the existing methods where each system requires its own database instance, and manual network connectivity among all the instances and resources need extra costs, time and money. On every new customer site too, the installation and setup of the application is also an overhead for the business. This extra money spent for setting up, consolidation and deployment of system sometimes may exceed even the total software product price. Once installed, server administration and maintenance of the operational system are also supplementary.

A variety of anytime anywhere cloud computing services using a terminal (A smart phone, web browser or any device) is being used by the customers which in turn reduces

the over-all cost of IT service provisioning. Multi-tenancy in the cloud for database resources enables small enterprises and start-ups to develop in consolidated, managed and scalable computing environments.

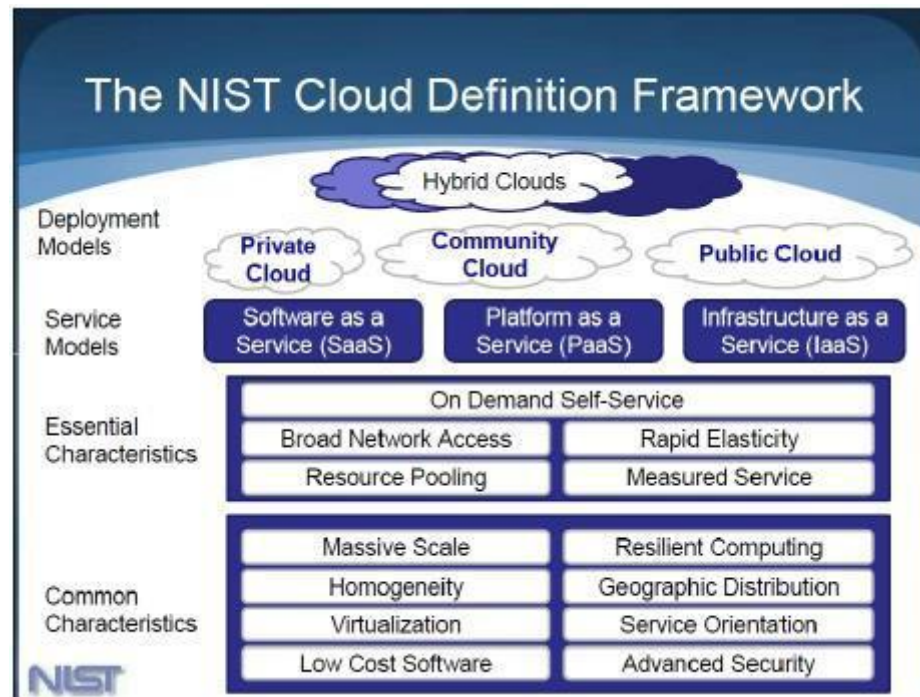


Figure 1.1: Cloud Computing Framework [34]

The represented cloud computing framework is proposed by NIST as discussed in further sections [34].

Cloud Computing illustrates a paradigm shift in for IT resources like hardware capabilities, storage power, and software applications. Instead of running these resources cost intensively on-premises they are provided on-demand of the consumer via networks (i.e. most commonly the internet) and only charged with regard to their actual usage [3] [4]. This approach promises a solution for the long-time ambition of turning computing in a utility such as water or electricity [18] [19]. Thereby cloud computing refers to both; the services delivered, as well as the hardware and software in the data center, which actually provide them and often called a Cloud [4]. The essence of cloud computing is characterised by the outlook of end users, who are actual consumers of the provided IT services while being oblivious of underlying technical particulars. All the implementation and configuration details are abstracted away rather being provided on a pay-as-per-you-use basis instead of acquiring as a whole.

1.1.1 Cloud Service Layers

Cloud ontology defines the incorporation of essential components of IT infrastructures such as platforms, storage, applications in the form of a stack of service layers of cloud computing [15]. All the services offered by the clouds are being distributed among the layers presented in Figure 1.2.

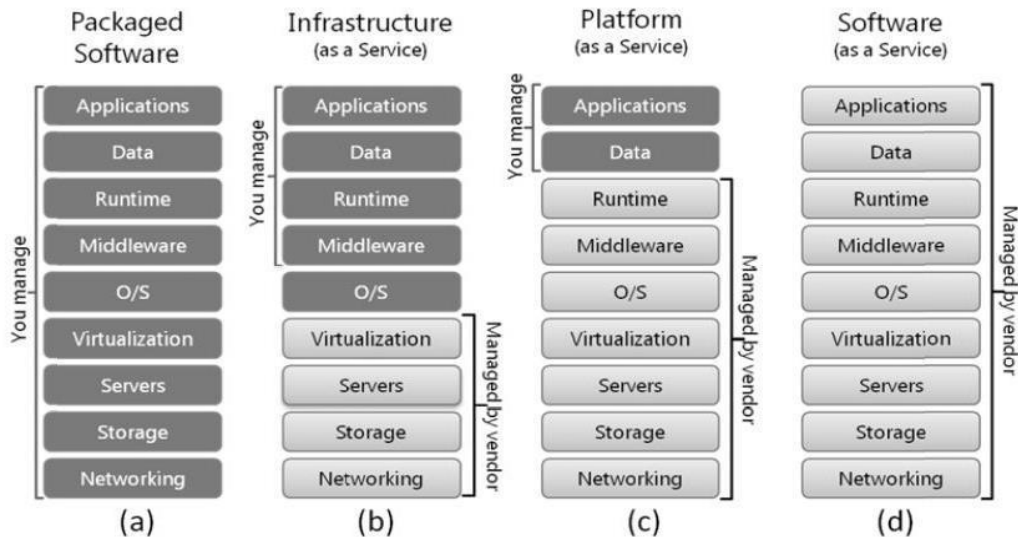


Figure 1.2: Cloud Service Models

- Infrastructure as a service (IaaS):

Infrastructure layer describes the fundamental resources forming the base for upper layers. It is mainly responsible for underlying hardware, virtualised resources (Processing and storing capabilities) and communication as a service. In other words, Application (SaaS) and Platform (PaaS) are dependent upon these infrastructure resources to build new cloud applications. IaaS can be termed as the service of software resources (OS, file system, virtualization) and associated hardware (application server, storage and connecting network) as a service. Traditional hosting of web application has been revolted completely without any advance long run commitments. IaaS provider has the role only to manage the data center working; development and deployment of the software service has to be performed by the user only.

- Platform as a service (PaaS):

This layer provides programming language environment with required APIs to implement and deploy SaaS applications without complexity and expense of

purchasing underlying infrastructure and facilitating all the needed services for software development life cycle totally on the internet. For developers, PaaS is very useful to work with as it has caching, authentication, built-in scaling, load balancing and other services for communication. It speeds up the development and deployment process of different cloud applications so that developers can focus on functionality more instead of environment and configuration management. PaaS cloud is built on top of services provided by the lower SaaS layer, IT environment is lesser prone to be focussed upon but technical composition [13]. Most popular examples are Google App Engine and Microsoft Azure.

The set of tools available with PaaS provider are used to leverage their software stack for application development. For an instance, if the developing application requires usage of Open Source Software (OSS) and MapReduce implementation for parallel computing; PaaS tools are beneficial to provide needed Hadoop nodes for the same.

- **Software as a service (SaaS):**

It contains various IT services which can be treated as alternatives for some of the locally existing applications. Generally they are being accessed by some web browsers with a nominal fee to use the shared resources on the cloud. Gmail and Microsoft Office 365 are the very common SaaS examples.

It is focused more upon the separation of software in use and its ownership. A set of distributed functionalities to deliver a particular software service is being configured and a model for all small scale and large organizations is proposed. Applications running on cloud are in direct service to the customer in SaaS layer [35].

1.1.2 Cloud Deployment Models

Following cloud frameworks are available for deployment of applications in the cloud-

- **Private cloud**

The entire cloud infrastructure is delivered exclusively for a particular organization and its subsystems (business units, employees, users); management and operations along with ownership can be in charge of organization or some third party or their combination. Hence, existence of cloud is to be both on and off premises.

- Public cloud

The use of cloud infrastructure is generally open for public use and shared among numerous consumers publically. Management, operation and ownership can be by an academic, government body, business entity or a combination of them.

Existence is completely on cloud provider's premise. Supported languages, development framework, runtime environment, pricing models and other services are considered while selecting the suitable public platform to deploy the application. Three major public cloud platforms are Amazon AWS, Microsoft Azure and Google App Engine.

- Community cloud

For a particular organizational group with a specific concern or requirements, a cloud infrastructure is provisioned. The management, operation and ownership could be performed by one or more organisations of the community, a third party or some other possible combination of them. Existence of cloud can be either on or off premises manner.

- Hybrid cloud

This computing environment can be a mix of private cloud, public cloud, some on-premises services. On the basis of computing requirements and costs changes, a more flexible business model is facilitated with changing workloads.

For an instance, an organisation can use its own private cloud for critical and sensitive work profiles, as well as uses a third-party acclaimed public cloud for development and testing workloads such as Google Compute Engine. It is significantly beneficial for highly dynamic workloads and another main use case is big data analytics.

1.1.3 Cloud Actors

As per NIST (National Institute of Standards and Technology, Agency of U.S. Department of Commerce) definition [42], a conceptual reference model is presented with five main actors as cloud consumer, cloud provider, cloud carrier, cloud auditor and cloud broker in cloud computing mode as depicted in Figure 1.3.

A cloud consumer requests services on the cloud to cloud provider in a direct way or through a cloud broker. A cloud auditor is responsible for conducting the periodic independent audits and contacting other actors to collect important information. A cloud carrier is the intermediary person who provides transport of services among cloud

consumers and cloud providers over a network connection. A cloud broker is responsible for performance and appropriate delivery of the cloud services and relationships among cloud providers and cloud consumers.

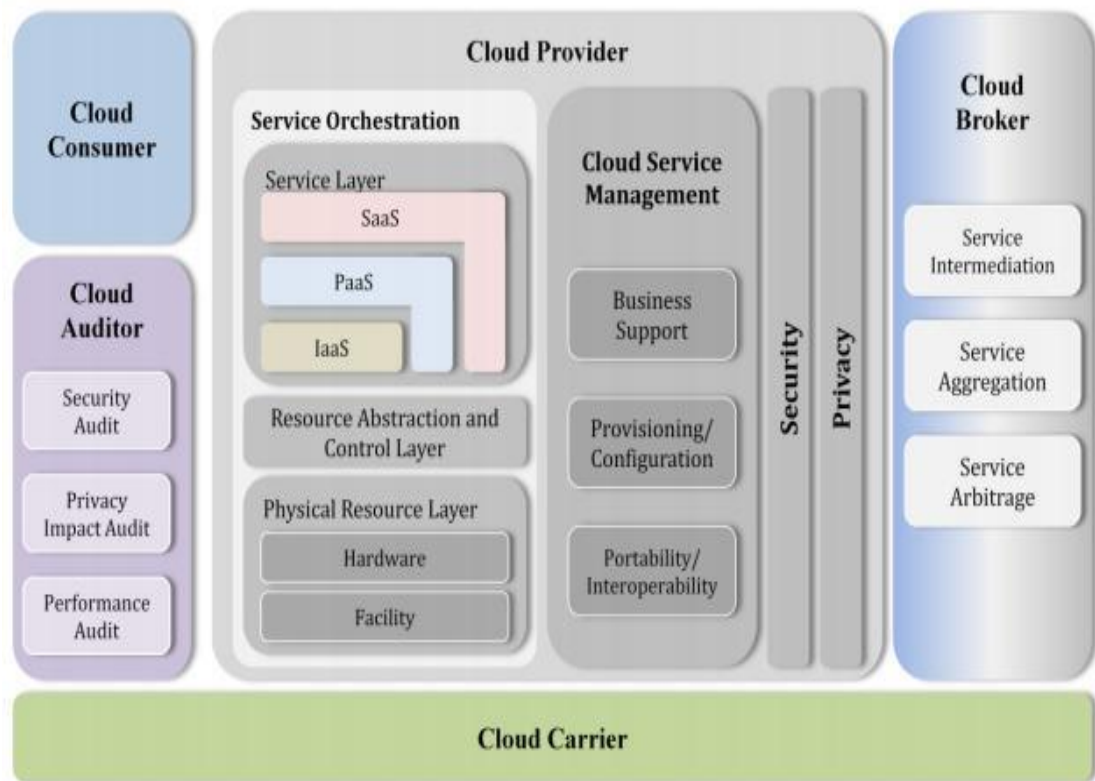


Figure 1.3: Cloud Actors [43]

1.1.4 Cloud Service Providers

Cloud computing has been beneficial for all the actors associated with it. As it has been most promising technology over the years, there is a wide range of cloud service providers for IT service provisioning around the globe. A company offering some sort of cloud services is called service provider. There are many enterprises such as amazon, which is first major service provider. Figure 1.4 represents some of main cloud service providers.

Selection of appropriate cloud service provider for applications is one of the major decisions to be taken when cloud adoption is considered. Table 1.1 lists comparison details among three most popular cloud platforms with selection criteria for suitable applications.

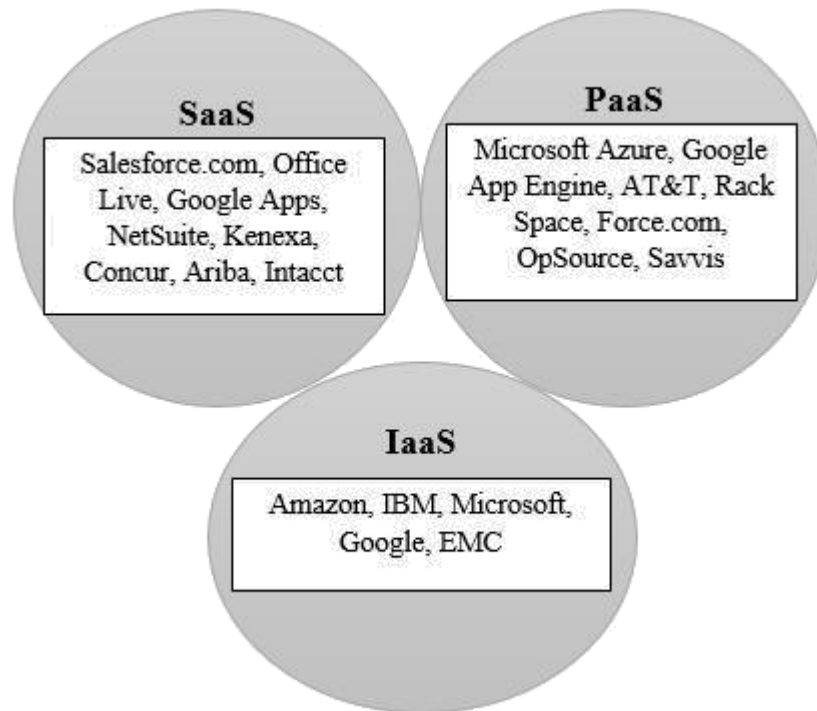


Figure 1.4: Cloud Computing Service Providers

1.1.5 Benefits of Cloud Computing

- Capacity On Demand-

In a traditional enterprise, during peak working hours, the working load can be tedious to manage. Hence, inadequate computing capacity can lead to increase application downtime which is not end user favourable. Figure 1.5 explains the usage pattern of IT resources in both traditional and cloud applications which explains the inefficient distribution of resources with workload in on premise applications. The dedicated resources often retain underutilized or sometimes over-utilized which leads more cost effective model of IT computing.

- Zero Upfront investments-

Traditional organizations bear large upfront investment costs of setting up their own data centers to facilitate software applications. Along with the initial setup expense, for long run, adequate cooling mechanisms, UPS supplies (for power), and other expensive installations are needed. Instead of all these costs, on a rental basis, cloud computing offerings are much efficient to adapt and easier to manage. There are various flexible models to incorporate clouds and minimise time to market for new start-ups and ready to move enterprises.

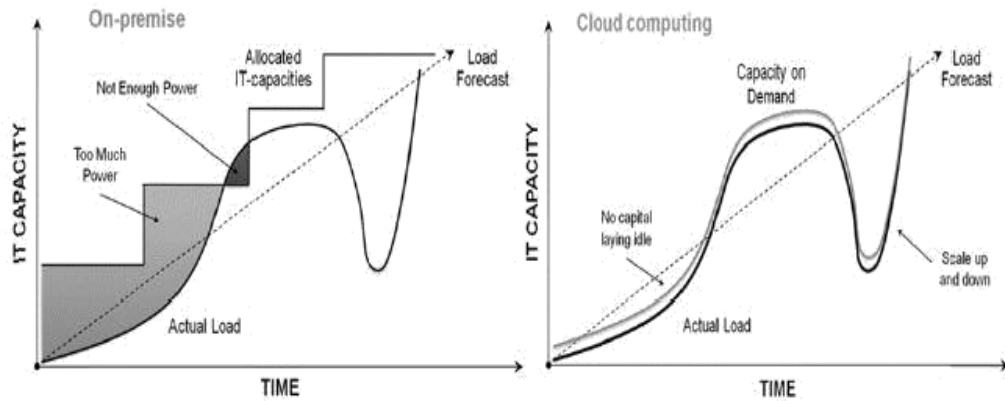


Figure 1.5: Comparison of Resource Provisioning

- Core Application as Main Focus

A standard packaged software contains application with underlying components such as operating systems, networking, data storage, middleware etc. the whole stack has to be managed simultaneously for smooth run of applications. To reduce this configuration and management overhead, cloud computing provides cloud layers (SaaS, PaaS and IaaS) to sub-divide the applications on the cloud and those additional components are being taken care by cloud only. Hence, applications attain to be focused more by the developers without hassling much about computing environment.

- Better sales potential

In comparison with traditional on premise systems, customers and end users find cloud apps as more flexible and attractive solutions for their applications. For example, A SaaS application requires only a web browser with network as compared to a desktop application that would be needing technical skills, hardware resources and other installation necessities. For lower financial resources, cloud computing provides new opportunities to link with new customers on the go.

- Easier customer maintenance

Independent application vendors have to deal with each customer personally on issues such as installation assistance, maintenance and support queries. This maintenance overhead gradually increases with more number of application users. On the other hand, cloud platforms facilitate centralized access to the systems where deployment, updates are carried out concurrently with almost zero downtime.

- Better QoS features

Cloud platforms are widely popular for providing high availability, scalability and security addressing techniques. Data redundancy is used to achieve high availability and data replicas are used in case of any emergencies even in a natural calamity. To minimise data latencies, geographical data distribution is provided for better end-user experience. Additionally, public cloud providers deliver many features and integrated services in a cost efficient manner.

1.2 Legacy Systems

Legacy applications are those enterprise or business software systems that have been for long enough in business and are still pervasive in organizations. These are typically written in older programming languages like COBOL, FORTRAN or may be running on platforms that are no longer supported but perform necessary services in business activities [17]. Software management of such applications is constantly being challenged by upcoming risks such as tight infrastructure budgets, high maintenances of investments and configured resources, and lack of adequate documents.

The term ‘evolution’ of any system is a broad term including an activity of adding a new database instance to the system to the complete re-implementation of the system [21]. Most of large successful enterprises are depending upon old legacy technologies which can be decade before tested hardware and software systems, though important for system’s long-run. Figure 1.7 shows an evolution paradigm of software application considering modernization and replacement as main phases throughout its life.

Due to the emergence of cloud computing, software architects are finding new approaches to adapt it for the traditional back-office applications without investing in new IT infrastructure and setups. A white paper by IBM concerning legacy modernization depicts that “any software system in any programming language or any platform can be termed as legacy once it has been implemented and running live.”

Legacy term is usually addressed for those applications which are deployed in old mainframe environments and resist new modifications [18]. Jonathan Arvin Adolfo depicts an insight into his survey: (i) 80% of the IT applications are still using legacy platforms (ii) 200 billion lines of legacy code are being used still nowadays in more than ten thousand large mainframe firms.

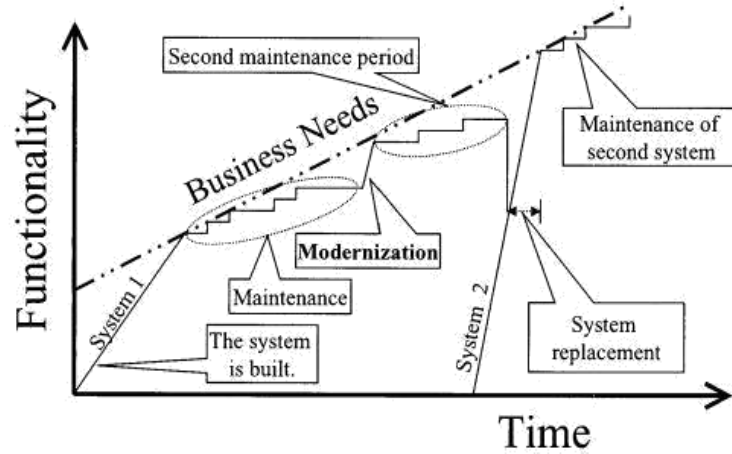


Figure 1.6: Evolution of an Application

Software systems are critical assets for business organizations and enterprises as they are the main knowledge expert systems acting throughout the life of an organisation. These complex systems are considered to be kept updated periodically to smoothen business practices, required modification affects the application’s complexity too [43]. These existing on-premises systems can be fragile for repetitive changes but for implementing newer and more efficient technologies, various practices are used.

The legacy applications have some key aspects which may relate to a modernization process when turning it into cloud application such as scalability, maintenance, security and code architecture. Table 1.1 lists out main characteristics of legacy systems along with the relevance to the cloud computing.

Table 1.1: Legacy System Characteristics

Legacy Characteristics	Correspondence to Cloud Computing
Scalability	Capability to spread workload across many hosts lacks usually in legacy applications and one of major benefit to opt for cloud environment.
Security	Cloud applications have built-in access controls to decrease vulnerabilities. Legacy systems have traditional access mechanisms.
Monolithic code style	Obsolete language being used in older applications with block of system layers intertwined.
Maintenance	Total cost of running legacy systems gets increased due to support and maintenance of traditional computing ways.

1.3 Challenges of Cloud Computing for Legacy Systems

Cloud adoption being a buzzword in IT industry also contains some issues to be considered while making decisions for modernization to the cloud. Legacy systems may encounter some limitations when targeting cloud to be their next platform. The following points are the major cloud computing challenges for traditional applications-

- Performance constraints

Adoption of cloud computing involves some consequences in terms of performance unpredictability because of resource sharing and virtualization. Since application resources are pooled across many locations and users are oblivious of their exact locations, running an application results in bandwidth issues and high latencies. Cloud providers have to deploy some special caching techniques to compensate these limitations.

- Compliance requirements

In many cases such as healthcare insurance or similar applications, government policies do not support cloud services due to data security issues. These enterprises have to analyse the liability of concerned applications while considering cloud adoption. For example Sarbanes-Oxley Act restricts accounting data going live on the cloud.

- Availability

Even though cloud providers ensure high degree of availability in service level agreements, outages are still a part of cloud paradigm. Two main kinds of outages are temporary and permanent ones. Temporary involves for a short duration of time like for few hours, cloud service remain unavailable. Permanent outage refers to cloud cutting off from the business completely. Cloud consumers have to analyse these issues while making decisions for cloud adoption. Service level agreements, using clouds for non-critical applications and keeping secondary on premise backups are some of the precautions to be taken.

- Privacy and security

These are the most discusses issues of cloud computing. However, because of data centralisation and service oriented techniques, it has been improved. Still, because of having full control on their data, consumers have to rely on cloud providers for data security. Also, data transportation has the limitation of data leakage on the go

for which encrypted data tunnelling and virtual private network set up are potential solutions. Private clouds are also motivated by privacy and security purposes.

- Multi-tenancy

Cloud computing SaaS model provides the multi-tenant application over the network where a single system copy is shared among various users. On the other hand, software systems in traditional computing environment are open to any customizable changes as each customer has its own copy of application installed. So, limitations on customizing systems lead to extra development efforts in the cloud.

- Vendor lock-in

Cloud service providers design platforms to provide computing environment to implement legacy systems on the cloud. Due to poor integration and interoperability concerns, once migrated, systems have to stick to respective cloud provider.

- Licensing

Cloud computing requires commercial license for software supporting SaaS applications. Regular licenses for traditional software systems do not match as they usually limit the respective computers. A proper cloud licensing model is needed for cloud applications even if no supporting application is not used by the cloud service. This is also a matter for software vendors to make changes to their selling policies for the cloud.

- Technical constraints

There are plenty of compatibility issues encountered when migration process starts, because of variety of programming environments on the cloud and traditional computing systems. Many cloud models do not support some older programming constraints for example no operating system other than Windows Server 2008 is supported by Microsoft Azure. There may be a significant number of changes to be made to legacy systems to overcome these technological limitations. Google App Engine does not allow .NET applications but Python, Java and Go runtime applications.

1.4 Parallel Computing in Cloud

Cloud technologies have elasticity as one of the key characteristics as elasticity.

Computing resources can be scaled up and down as required in running time. IaaS

providers such as Amazon EC2 [46] service allows the user to add as many virtual instances and use them similarly as any other hardware resource being used in traditional enterprise. While moving to cloud, there may be a sound possibility of parallel environment for the application in cloud to optimize the resources with simultaneous execution of jobs that can be parallelized and produce an integrated output in the end. This cost-effective model of parallel computation in the cloud is facilitated by programming models such as MapReduce by Google. For large data and computation intensive applications, suitable cloud environment configured and installed provide a highly efficient computing environment to lower the administration and operational costs significantly. Google's MapReduce, Yahoo's Map-Reduce-Merge [45] and Microsoft's Dryad [49] are leading customized data processing frameworks in the cloud built upon the parallel programming, fault tolerance and optimization [48]. Since these frameworks allow their customers to access, allocate and control a range of virtual machines (VMs), data processing frameworks like Hadoop, open source implementation of Google's MapReduce have begun to be used widely in infrastructure services. According to parallel data processing requirements, a set of compute jobs is distributed among separate resources to be assigned as particular tasks.

In traditional parallel approaches, applications are executed on compute clusters, grid infrastructures and super computers with the focus on mainly computing power to be parallelized. For more efficiency optimization, both application and data need to be move to the cloud. With the increase in data volume of data intensive systems, inevitability in data access also gets higher.

Distributed file systems such as Google File System (GFS), Hadoop Distributed File System (HDFS) allow Google MapReduce and Hadoop to access by storage systems on heterogeneous compute nodes.

1.4.1 Hadoop

Hadoop is the open source software platform by Apache Software Foundation which draws inspiration from GFS. It has become standard framework that allows large datasets to be processed across distributed compute clusters [50]. It is designed with the idea of scaling up from a single server to thousands of machines to deliver high availability and efficient computing service on the go. It has a library built for the detection and handling application layer failures to deliver fault tolerant services.

Apache Hadoop implements Google's MapReduce runtime where HDFS is used to access data which maps local disks of compute nodes to a single file system to disperse the data across all the nodes. It is currently being used by big corporations like Yahoo, eBay, LinkedIn and Facebook. Key attributes of Hadoop include fault tolerance, built-in redundancy, automatic scaling and move to data computation. Main components of Hadoop are as shown below in Figure 1.7.

- Hadoop Distributed File System

A distributed file system architecture to run on any commodity hardware which had Hadoop architecture preinstalled. It implements the write once-and read multiple times approach. The HDFS architecture is based upon a single master node and multiple worker nodes, also termed as name node and data nodes [50]. A master server consists name node, data node, job tracker and a task tracker. Job tracker does the job of monitoring and scheduling tasks and accordingly instructing to task tracker. Job tracker and task tracker work along with each other by distributing tasks efficiently to compute nodes. HDFS has been proved to be highly fault-tolerant file system as faults are detected and recovered through automatic replication of data blocks.

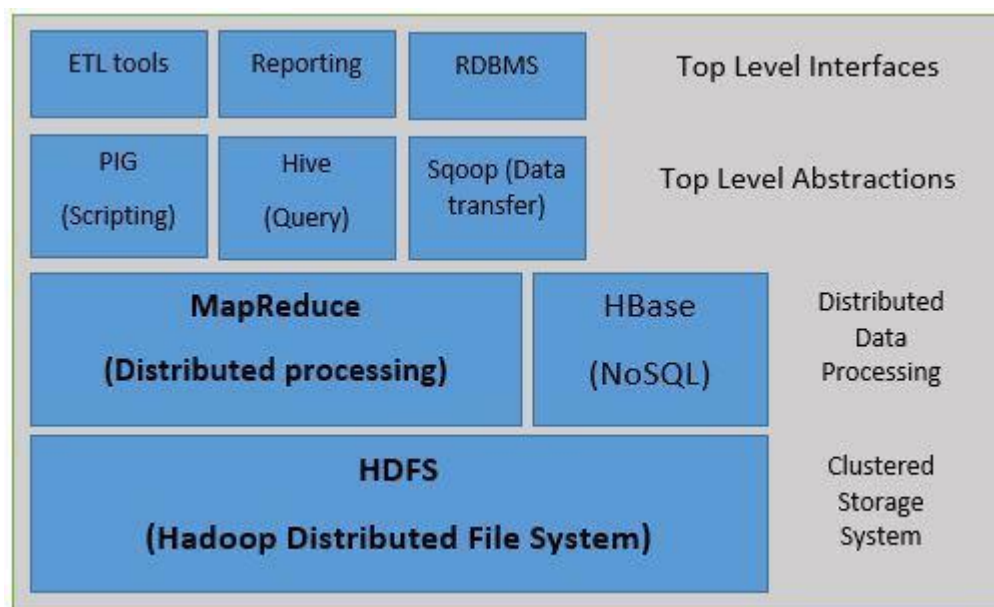


Figure 1.7: Hadoop Architecture

- MapReduce

MapReduce is a straightforward and scalable programming model for processing huge amount of datasets, parallel programming and fault handling. In order to finish

the jobs in limited time, input data need to be distributed on a large number of machines for computation [23]. The application developers only need to define a map function and input data is applied to it, which further generates some intermediary key pairs. Then, a reduce operation is used aggregate these key pairs to generate a final derived data output. MapReduce paradigm can be leveraged by a utility computing model provided by major cloud providers such as Amazon, which offers through PaaS approach, further including overhead of setting up Hadoop system. It provides high throughput when run in clustered environment for distributed computing application. Hadoop MapReduce engine also provides essential features such as monitoring, scheduling, prioritizing and failover of tasks. MapReduce frameworks allow developers without much knowledge on distributed computing to write applications that take advantage of distributed resources [23]. Hadoop MapReduce is an implementation of such a model. When a MapReduce function is called, following actions are executed as shown in Figure 1.8:

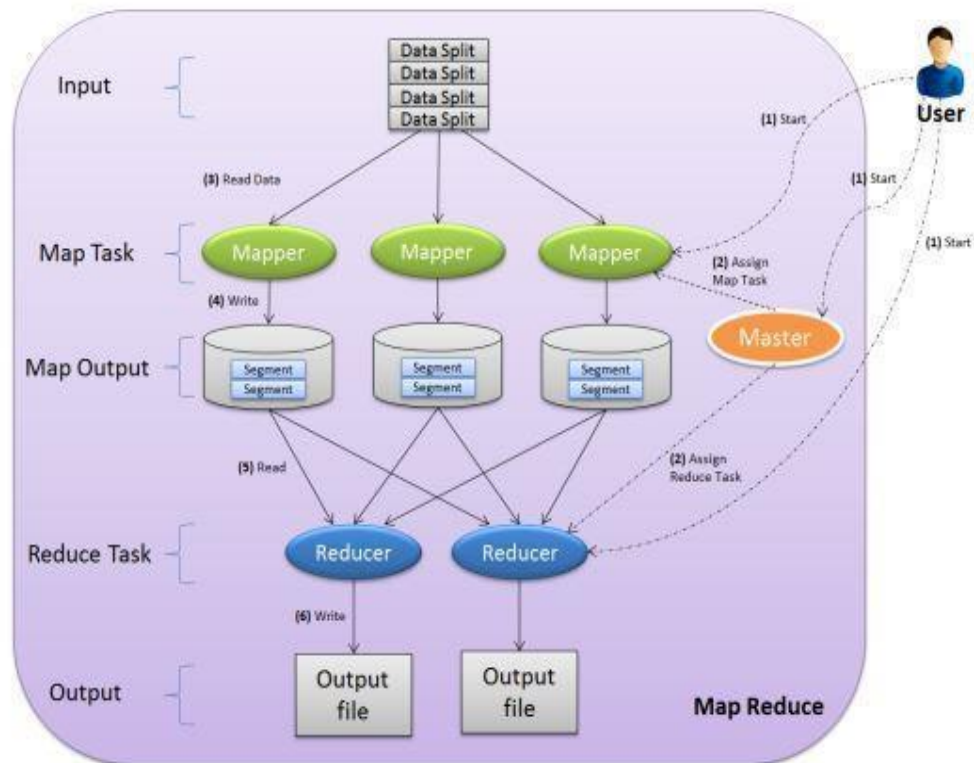


Figure 1.8: MapReduce Programming Model

- First, input data is divided into M partitions and starts programs on a cluster instance of different machines.

- Among them, one master program distributes a Map or Reduce task to other workers. Any idle worker can be assigned a task by the master program.
- In the assigned Map task, partition of input data and generation of key-value pairs is carried out and these temporary pairs are kept in memory buffer.
- Periodically, these pairs are written to the local disk and their location is transmitted to the master program.
- If workers are being assigned Reduce task by the master, Reducer read the entire buffer for key-value pairs using remote procedure calls. Then, temporary data is sorted out based on the key.
- Finally, reducer deals with all the data and for each key and associated set values, a user defined Reduce function is called. The output generated by Reduce function is the final output of the program.
- After processing all the mappers and reducers, master program returns the resulted output to the user's programs and stored in individual files.

1.5 Research Motivation

Information and communication technology has made lives easier in numerous ways with digitization of practises in many industries such as e-health, banking, e-governance, entertainment, communication etc. [11]. Application of newer technologies such as grid computing, distributed computing, cloud computing, users consuming these computational resources are getting increased exponentially. The perception of computing being the 5th utility (after electricity, water, telephony and gas) is already being realised with the technological developments for IT services.

Due to this promising belief of better and more convenient IT services, fundamental economics of IT industry is being affected towards this belief of newer way of computing. It leads to the 'modernization of enterprise application' being a buzzword for most of the organizations today. The advantages of cloud computing are compelling enterprises to consider the shift from on-premises to cloud for their services. Databases, applications and infrastructures are targeted to be migrated from traditional system environments to cloud. It facilitates the delivery of software services in a flexible, economical and easy to maintain way to the consumers using a terminal (smartphone, browsers etc.) anytime, anywhere.

Moving an on-premises software application into cloud, changes the approach how services are being sold and maintained along with management of their application architecture, operational profile and business models. The expenses are significantly reduced because of per-tenant subscription and software providers are targeting “long tail” markets [36]. Cloud environments are more accessible and manageable which leads to the rise in software engineering activities concerning modernization of legacy applications to the cloud. Although there is a lack of enough support systems to determine when and how cloud computing can be helpful for the legacy applications and should be adopted. Potential adopters encounter these conflicts and need to follow a well-formed framework to assess their on-premises applications and decide the feasibility of cloud migration. This thesis aims on a generalised framework to be proposed which encompasses all the possible modernization approaches for existing systems and evaluation criteria to decide what taxonomy of aspects are to be assessed upon.

1.6 Thesis Outline

- As shown in Figure 1.9, Chapter 1 introduces topic to reader and points out the related base concepts with brief introduction.
- Chapter 2 provides an overview of related work beginning with basic terminology necessary in context of this work followed by an introduction of recent publications addressing the issue of application migration to the Cloud.
- Then, Chapter 3 the gaps among the existing researches are addressed and the current problem has been figured out. The problem formulation is then followed by main research objectives in the end.
- Based on this Chapter 4 shows the proposed framework for the modernization of legacy applications with the detailed components.
- Regarding the realization aspect of this Master's thesis, Chapter 5 illustrates a case study where migration of a sequential application to the cloud with MapReduce is performed to the cloud.
- Finally, this work is concluded in Chapter 6 by answering the initially stated research questions and briefly discussing thesis contribution and future work.

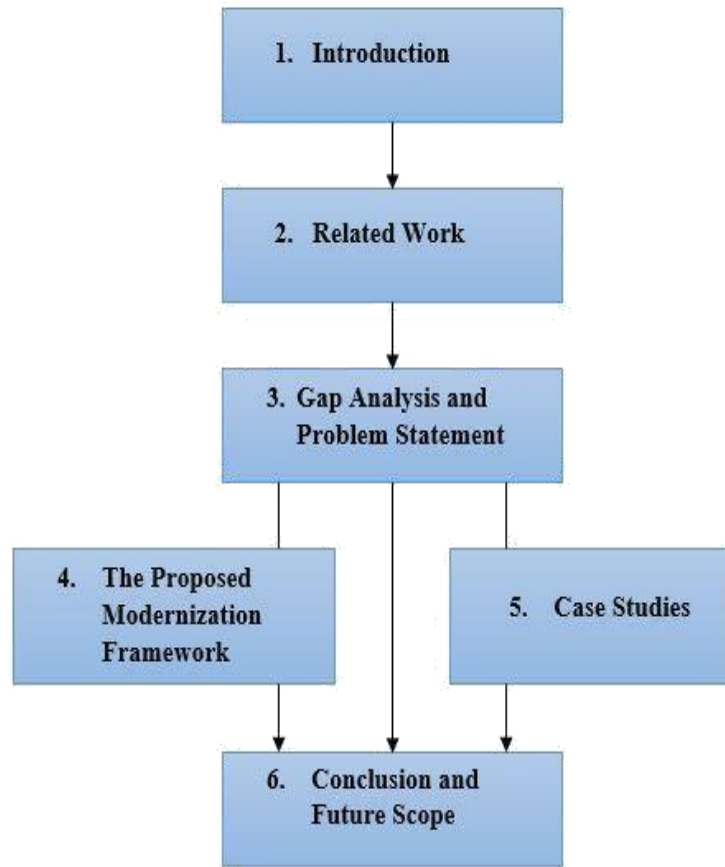


Figure 1.9: Thesis Outline

This section discusses the key concepts of the underlying technologies for this research work. It is followed by the recent work in the field of application migration to the cloud and finally the basis of this research is presented to the reader. This chapter sets the context for the rest of the thesis with provision of detailed background concepts. First, cloud computing origin and later how migration of enterprise applications has been evolved lately has been discussed in detail.

2.1 Legacy Applications Modernization to the Cloud

Cloud computing is being considered a mature IT paradigm, covering all the computing assets, applications delivered as a service over the network and hardware and software systems providing data centres for these services. Figure 2.1 depicts the evolution of cloud computing with the idea of utility computing to provide computing as a service.

Because of IT industry's initiative, computing mechanisms have been changed drastically to employ new innovations. Most of the research work focuses on the underlying methodologies and technical background of cloud computing [11] [13] [30],

Cloud computing is an emerging business mode providing "everything-as-a-service" (XaaS), including virtualized infrastructure and resources, as well as software applications with middleware platforms being provided and consumed in the cloud [35].

A technical report from University of California at Berkeley [3] describes the main concepts of cloud computing with upcoming challenges in different research opportunities. One of these major areas is software engineering which focuses on the main issues in developing software for the cloud and migration approaches already discussed by the research community. Cloud computing proposes many state of the art solutions beneficial for older applications to be moved to a cloud environment. The selection criteria of the strategy is dependent on particular requirements and characteristics of the enterprises and concerning issues.

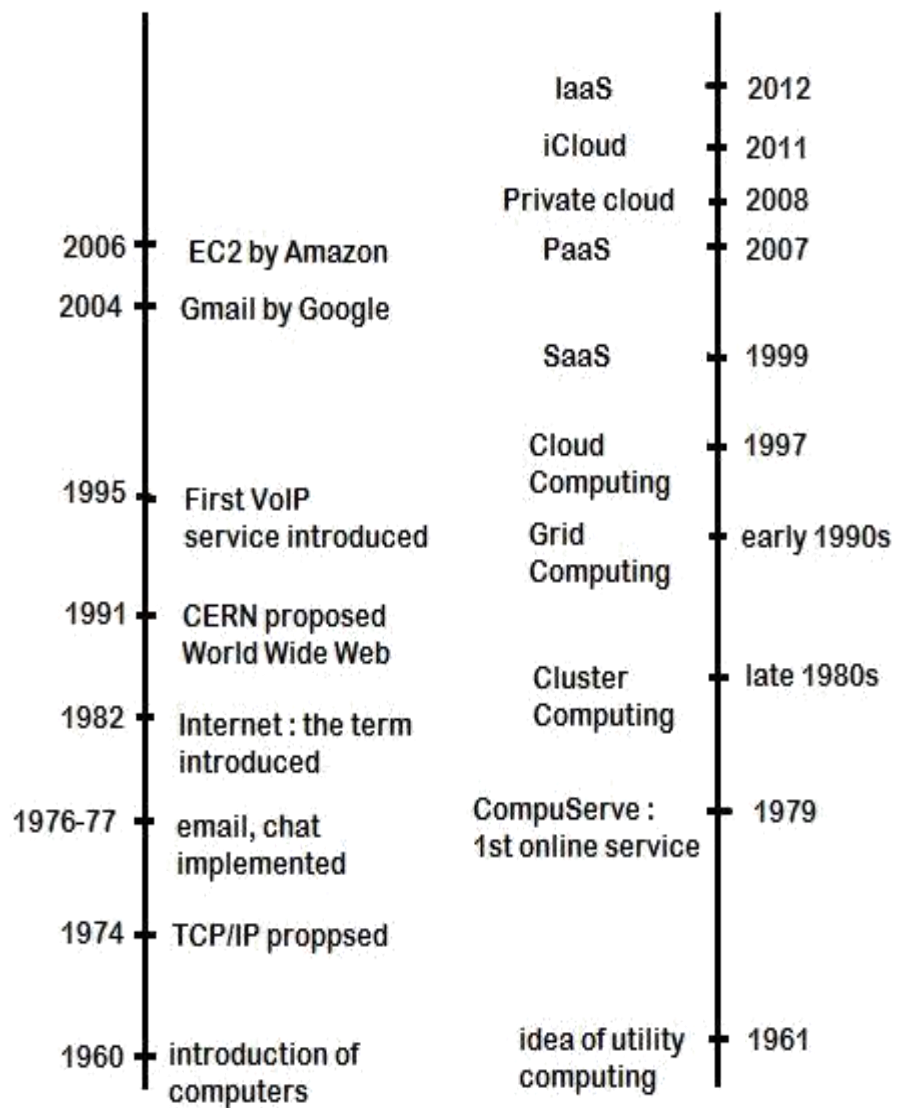


Figure 2.1 Evolution of Cloud Computing

Conferring to the software maintenance standard of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) Institute], the term adaptive maintenance is illustrated as “the modification of a software product, performed after delivery, to keep a software product usable in a changed or changing environment” [2]. Hence, software migration or application retirement are two main possible suppositions of the ISO/IEC maintenance process circle and application migration is considered as an exceptional instance of adaptive maintenance. Thus, to retain present systems capable for improvement and future-proof, migration to more efficient and up-to-date architectures [33].

Legacy system modernization is a term used when an older application requires more pervasive changes than usual maintenance efforts while having business importance worth preserving [31]. For decades, it has been considered to be a common approach not limited only to cloud migration purposes. For example a system still running on a Sun Micro operating system or an AS/400 platform. It also includes integration of applications with some newer applications or changing interfaces. Legacy modernization majorly involves five aspects as mentioned below-

- Present functionality and application architecture assessment.
- Selection of which components to be kept unchanged, which to be replaced by off-the-shelf ones and which have to be re-written.
- New functionality needed in target software system.
- Platform selection criteria
- Decisions on interconnection of all the parts together- newer and already existing ones in organisation.

The modernization methodology by Microsoft white paper [43] also presents four strategies to be migration, redevelopment, replacement and integration or extension. Also, single approach cannot be applied to all kinds of projects, as it is possible to use combination of suitable strategies for single legacy application.

A Nexaweb report [41] on modernisation projects depicts the status of on-going and underway projects.

Table 2.1: Status of Modernization Projects

Criteria	Yes	No	Not Sure
Defined Modernization map for application?	55.6%	30.6%	13.6%
Actively Modernizing legacy apps?	65.7%	28%	6.3%
Scheduled, but not started.	46%	44.7%	9.3%
No project started nor scheduled.	16%	71.8%	12.2%

2.2 Related work

In this section, a formal literature review process has been conducted for rational assessment of research area and presented in Table 2.2. A fair assessment of research topic using rigorous review and auditing to remove the researcher bias. A total of 135 resources were analysed out of which 74 were selected for review purpose.

Table 2.2: The Systematic Review Process

Objective:	To identify the existing techniques for the modernization of the legacy systems and the available approaches.
Main Question:	What are the available approaches for legacy system modernization with cloud perspective with the automation tools proposed? How parallel processing computations can be efficiently migrated to cloud?
Intervention:	Case studies, surveys conducted, other research in the domain of cloud adaptation by the systems.
Control:	No study to be used as control.
Population:	Research related to software reengineering for the cloud
Results:	Case Studies, models, surveys and approached used for reengineering
Application:	Software Reengineering approaches for the systems for cloud, parallel programming applications in the cloud
Keywords:	Cloud computing, legacy systems, migration, reengineering, service oriented architecture
Source Selection Criteria:	Availability of research papers on the web; search engines
Source Search:	Google Scholar and other engines like ACM and IEEE.
Methods:	Automatic search, through bibliographic references of papers, related conferences and journals
Source Engines	Springer, Elsevier, IEEE, ACM;
Study inclusion criteria	Available full papers related to cloud computing, software modernization and parallel programming.
Results Summarization:	Analysis of all research papers and summarization of data into Text reports and then results are formed.

2.2.1 Impact of cloud computing on software development

- Armbrust et al [3] described key concepts and vision of cloud computing with particular emphasis on elasticity being main driver for economic model. The faster growth of workloads to reach millions of users rapidly with lower investments and increased flexibility has influenced on the business organizations majorly.
- Kim et al [14] presented an extensive research on issues of cloud computing mainly availability and security being most challenging. Three categories of cloud users as consumers, small companies and large organizations are described. Key issues being faced by these users are security, integration, compliance, performance, environment settings, outage, and cost concerns. These expectations are the main challenges for cloud service providers and need to be managed.
- Rimal et al [15] contributed a detailed study of various cloud providers and possible similarities and differences among them. They have used a comprehensive taxonomy about key aspects of cloud computing and further presented a comparative results about many existing cloud service suppliers such as Amazon, Force.com, GoGrid, Google App Engine, SunCloud etc. The features considered for the comprehensive study for all are load balancing, interoperability, fault tolerance, security, storage issues, computing architecture and virtualization management.

2.2.2 Existing Migration Techniques for the Cloud

- Babar et al [18] provided observations about migration of an existing open source application, Hackystat into cloud environment with accordingly demonstrated guidelines and suggestions. Required modifications in architecture of the system were separating database layer and introducing a new controller layer. Identification of phases when migrating a service oriented application to the cloud system and related experiences and challenges are thoroughly discussed.
- An incremental migration technique had been proposed by Brodie and Stonebraker [7]; termed as *Chicken Little* methodology. It is a step by step approach in contrast to big bang strategy where a newer system has to be developed from scratch and finally system is replaced in the last step. In *Chicken*

Little methodology, the exchange of information between legacy and target application is performed via Gateways, while both the systems are operational until the migration process gets over.

- Wu et al. [22] proposed another approach named as butterfly methodology for certain mission critical applications to minimize the downtime. Parallel running of both legacy and target applications does not require any gateways because of no need of data consistency provisioning. Bisbal et al. included an overview of legacy software migration including *Chicken Little* and *butterfly* approaches in their research [21].
- Hasselbeing et al. [19] present a technique to migrate integrated legacy applications into multi-tier systems with a *Dublo* pattern, partial Duplication of Business Logic. It involves a step by step pattern where partially duplicated business logic is realised by components of legacy code deployed on target server.
- Lately, model driven approaches are being followed for software modernization processes. IBM's Service Oriented Modelling and Architecture (SOMA) utilises strategies which is used to transfer legacy assets to service oriented architectures [27]. With the phases like code analysis, service identification and model transformation a prototype of a java application is migrated into web. Due to lack of support for many other programming languages, it cannot propose universal approach for migration.
- Fleurey et al. [53] proposed another model oriented approach which includes reverse engineering, transformation analysis and code generation along with the model transformation phases. Automation, flexibility and reliability are the main concerns for this approach.
- REMICS (Reuse and Migration of legacy applications to Interoperable Cloud Services) [54] is a focused on the legacy application migration to service cloud for SaaS alone. It does not deal with other cloud service models and applications already built on service oriented architecture.
- Cloud Adoption Toolkit [63] is a decision support framework proposed for cloud adoption processes by organizations. It focuses on identification of different enterprise concerns by performing analytical study about the organization and source application to be migrated. Key concerns are stakeholder impact analysis, responsibility and cost modelling, and technology suitability analysis.

- In 2012, CloudGenius, a framework for an automated procedure [64] of cloud migration has been proposed. It mainly focuses on single tier web services and assisting them to choose a combination of appropriate virtualization instance and IaaS service on the cloud. Based on the technical requirements of a web server and technical issues in traditional data center, the framework does support the cloud migration process but some issues and decision making criteria have not been covered thoroughly.
- Tran et al [62] presented a cost estimation model as cloud migration point (CMP) for cloud applications. They have evaluated the model empirically with six different small scale migration projects for cost and effort estimation. It provides a size and effort estimation mode for migration projects and validates with the CMP.
- Zhou et al. [10] proposed a novel approach for enterprise software reengineering for the cloud. A five step ontology process with generation of code, database and framework ontology has been identified to find potential service candidates in traditional application to migrate on the cloud.
- Prodan [72] provides a useful system of categorizing and classifying the type of clouds which are typically encountered in practice. The research work that PaaS is “relatively new and immature” is indication of just how fast cloud technology is moving because of the number of mature PaaS offerings like Google App Engine and Microsoft Azure which are now available only two years later. In spite of that, his work provides a useful survey of some of the differentiating characteristics of the available clouds systems which can help the software architect better understand which of those features might be useful.
- A practical approach for cloud migration has been proposed in [73] where requirements and limitations of infrastructure as a service are discussed. Three legacy case studies are demonstrated and possible obstacles are covered such as database size, programming libraries etc. however the approach lacks when it comes to migration to SaaS as it does not provide the scenarios for which SaaS is to be taken as a suitable option or replacement by off the shelf components is preferred. Also, the missing factor is complete assessment of the legacy application for selection of appropriate cloud provider.

2.2.3 Industrial Contribution for Modernization

- **CISCO**

A legacy migration white paper provided by CISCO Systems [65] depicts the significant amount of information about the considerations of a migration project usability of cloud computing, delivery and deployment models are discussed following evaluation of the application characteristics such as architecture, dependency and scalability. The research also focuses on the decision making criteria and performance requirements. Though it outlines major aspects about cloud migration, it does not present any organisation level readiness for cloud adoption by the company.

- **IBM**

IBM has presented a research to overcome the challenge of selecting whether an application can be a cloud candidate or not when considered for migration [66]. It categorises the application domain into critical and analytical systems where complex applications have low benefit and ease of migration as compared to big data analytical applications. Hence, readiness of analytical applications to be migrated to cloud is more than mission critical applications. However, the white paper has focused only on the private cloud domain, it still provides needful information about the candidature for the cloud.

- **HP**

Hewlett Packard has proposed a brief about Cloud transformation consultancy program where major cloud adoption issues are concerned curtly [67]. There are two main models for business and technical perspectives each having respective considerations about cloud migration. Even though the work does not describe the issues in detail, it introduces various domains and challenges about cloud adoption.

2.2.4 Use of Parallel Computing Approaches for Cloud Migration

Although cloud service models have been presented to maximize the technical, business and organizational benefits, cloud computing also comprises of a unique opportunity for analytical tasks and batch processing applications. If application data can support parallelism, system can be modernized to take advantage of several compute instances at a time. It further enhances the cost efficiency of the sequential legacy application. Programming constructs provided by Google's MapReduce with Hadoop as an open-

source counterpart facilitate the parallelism of task execution while hiding the complexity from the users [3].

- Shang et al. [23] have introduced the use of Google's MapReduce in cloud migration for data mining applications. Distributed model involves migration to MapReduce including extra lines of code but it may perform four times faster than a single node model of application. Machine learning applications have greater benefit from parallel processing cloud models.
- Panda et al. [24] practice MapReduce to implement classification and regression learning algorithms on huge datasets through their own tools PLANET. They utilise it to gather information and predict about user experience on a sponsored advertisement click.
- Coffman et al [55] discussed a sequential application implementing best fit bin packing algorithm to convert a multi-processor scheduling. Various sequential implementation of bin packing have been discussed and a comparison is presented in [57].
- MapReduce framework has been implemented in wide range of applications mainly text processing and machine learning. Many improvements of sequential algorithms through parallel processing are proposed in [58] [59].

2.2.5 Data Migration to the Hadoop/MapReduce

Enterprises who are investing in big data analytics are providing largely efficient services as they are focused more on business intelligence and further appropriate tools to extract, process and analyse the information are used. Migrating to a big data environment includes Hadoop as target platform as compared to traditional database systems such as MySQL. Hadoop/MapReduce framework involves support of many tools for migration approach. These tools are also categorized on the basis of their speed, performance rate and cost effect etc. The study reviews existing models and the tools being used for migration to parallel environment.

- Cristian et al. present data replication as an approach to migrate databases with efficiency and transactional replication is preferred for better end user experience and performance improvement [68]. A tool named informatica has been introduced for data integration purposes to migrate and employ big data technologies in the enterprises.

- According to Jeffrey et al., A parallel programming model is presented for dataset processing purposes [69]. MapReduce facilitates the filtering and sorting jobs on different nodes and finally generating an output summary by reducer. This architecture is treated as a cluster computer architecture. MapReduce algorithms such as Breadth First Search (BFS), PageRank etc. to manage the constructs in master/slave directed acyclic graph [70].
- Extract, Transform and Load (ETL) is data migration model used to extract, clean and customize data from different sources to insert into data warehouses as shown in Figure 2.2 [71]. ETL tools are used to migrate the data from heterogeneous resources. Data extraction is independent of the traditional database type. Migration to Hadoop/MapReduce includes pre-processing of the migrated data into Hadoop Distributed File System.

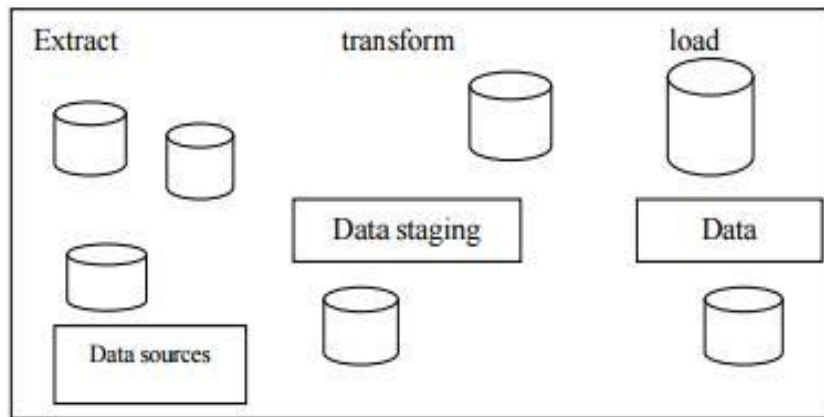


Figure 2.2: Extract, Transform and Load Process Model

Data Extraction includes classification and filtering of the data from sources such as web applications, transactions and enterprises. For example, relational database management systems are XML based which can be mapped with highly similar NoSQL data stores in Hadoop. This unstructured or semi-structured data is further processed (data staging), analysed and transformed. MapReduce is used with Cloudera sqoopis to import relational data to Hadoop. After transformation, data gets loaded into warehouses. It is an open source Java Data Based Connectivity (JDBC) based platform for supporting parallel import and export between various database systems and Hadoop.

- There are various data stores available for data design and implementation with Hadoop. Some of them are listed in Table 2.3:

Table 2.3: Data Migration Tools for Hadoop Framework

Data Migration Tool	Usage
Redix	Key value data store as an open source distributed database
Apache Cassandra	Open source distributed database for large datasets across various commodity servers.
Hbase	Non-relational distributed database, developed as a part of Apache Hadoop project.
Dynamo database	Amazon full proprietary NoSQL database.
Riak	Fault tolerant NoSQL database based on Amazon's Dynamo. It uses hashing as key value scheme in data buckets.
Mongo database	Open source NoSQL document oriented database with JSON like dynamic schema documents
Couchbase	Open source NoSQL document oriented database for interactive applications providing low tenancy and high throughput
Neo4j	Native graph database
Hypergraph database	Intelligent system oriented open source database with object oriented capabilities

2.2 Analysis

The research in the area of the modernization of legacy applications identifies many aspects and issues concerned with the legacy applications. However, some of the researched focus only on a particular cloud service model or some of them propose cloud only solutions for legacy applications. Architecture driven modernization techniques have been used for most of the projects but all types of applications cannot be using architecture driven modernization. There are various modernization approaches which are good alternative to that every application which needs to be move on the cloud. To sum up, cloud computing paradigm is the mature and comprehensive approach to be consider by the existing applications. There is possibility that every system may not be feasible to be moved on the cloud.

2.3 Conclusion

In this chapter, modernization has been proposed as a prerequisite for considering legacy application on the cloud with state of art researches recommended. Many researchers have focused on the architecture recovery from the legacy application and modifications that lead to the newer platform. In next section, research gaps has been identified with research problem formulation at the end of the chapter.

Research Gap and Problem Formulation

Based on the literature analysis in the preceding section, various researches have been identified in cloud computing concerning advantages, issues and consequences of modernization of legacy systems to the cloud. These issues have been recognised by many researchers though a generic solution has not been proposed yet for all kinds of legacy applications. Formulation of the thesis has been done on this basis in subsequent sections. Research objectives are enumerated at the end of this section.

3.1 Gap Analysis

The literature analysis and problem identification process is depicted in Figure 3.1:

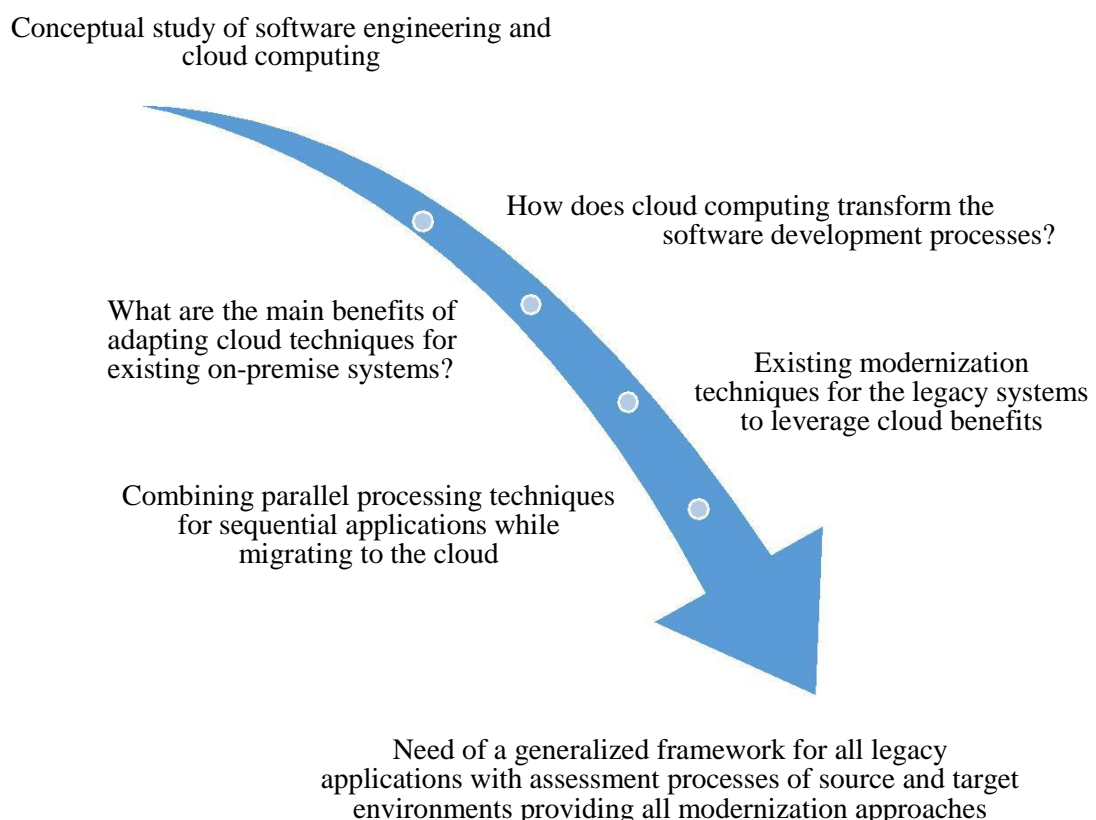


Figure 3.1: Problem Formulation Process

3.2 Research Objectives

The main objective of the work is to analyse the concepts and approaches for the modernization process of legacy applications. Migration is a non-trivial aspect which also depends upon various application components and quality attributes. There may be alternatives such as to migrate the whole software stack into cloud from local data center or to replace the specific components by existing cloud offerings. Main contributions of this work are:

- To present a comprehensive and detailed study of the cloud computing offerings with the perspective of enterprise applications and the existing migration approaches.
- To analyse different cloud platforms for suitable decision making process.
- To develop a framework for any legacy application towards modernization and prototype migration to cloud and validate it using parallel computing techniques such as MapReduce.

3.3 Conclusion

This chapter discusses the formulation of idea behind this research. The reasons why modernization of legacy applications is important, options available to follow and selection criteria to choose a particular alternative are presented in a single framework. This modernization framework has been proposed with decision making module and a set of available strategies for the modernization of legacy applications in the next chapter.

This chapter proposes a generic framework for legacy application modernization. The term modernization is a prerequisite for migrating older applications into the cloud. Cloud based solutions are newer advancements in the technology where legacy systems can be assessed to leverage cloud benefits. There is need of a balanced mix of technical and business aspects along with stakeholder impact analysis for assessment process which further decides the readiness of legacy application to the cloud.

4.1 Architecture of Framework

The proposed modernization framework is based upon three main phases in the whole process, evaluation of application, appropriate strategy selection and implementation of selected approach with suitable testing and execution in the cloud. It does not only focus on question whether a certain application would be a good candidate for cloud, but which modernization approach should be chosen. Figure 4.1 depicts the architecture of proposed modernization framework with modules as decision making, approach selection and modernization process implementation.

This framework also consists of other than cloud alternatives for the applications which are critical or may not need to be migrated to the cloud, not every application can be moved to the cloud. A generalised method to be approached by every kind of application with modernization objective is projected for analysis of possible transformations.

4.2 Decision Support Module

The idea of modernization of older systems to cloud has been relied upon a major process of selection of most suitable approach to follow. There are many researches indorsed for this decision making module which extend the procedure beyond only cost comparison. In 21st century, the newer critical systems do require the resource utilization on an optimum basis with other key needs such as high availability, computing power, storage and business continuity.

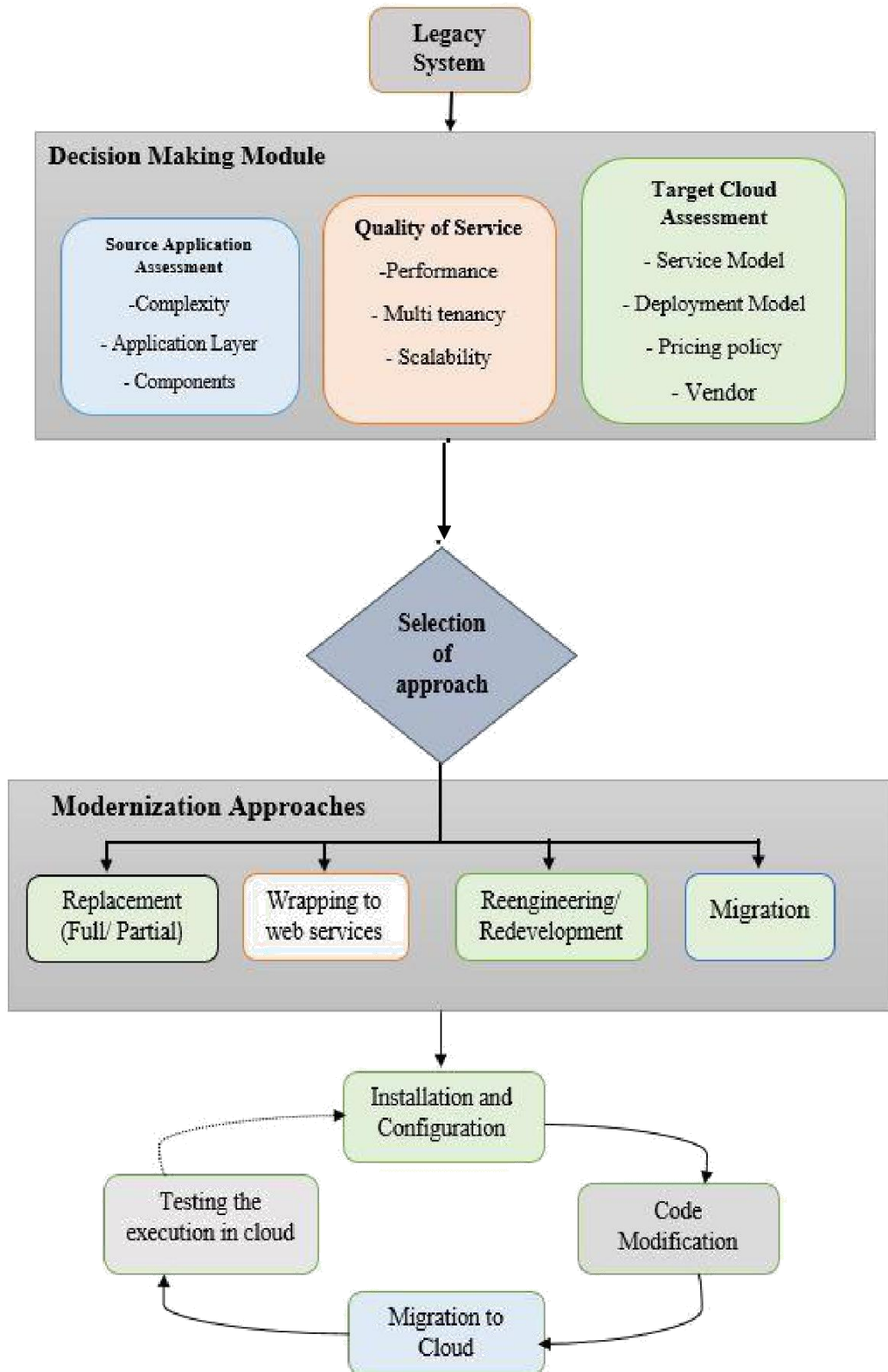


Figure 4.1: Application Modernization Framework

Unlike some of other State of the Art methodologies proposed in [20][21], which are mainly concerned about some selective kind of legacy applications, a more generalised framework is provided to focus on assessment of all applications to the cloud. In source application assessment, there are main three decision making aspects, all of them are influenced with each other and are also affected by the selections of outcomes.

4.2.1 Source Application Assessment

The application aimed for the migration to the cloud, is analysed for some key decisions to be taken for the assessment of the application readiness to be modernized and accordingly select the target cloud environment features. This assessment criteria includes a combination of technical and business aspects related to an organization.

- Complexity

Estimation of size complexity of the source application has its direct influence on development costs. Lines of Code (LOC) has been a traditional size metric but not applicable to be used in early lifecycle phases [61]. Function point metrics proposed by Albrecht [60] are independent of programming language being used and focus on the functionality of the application. Feature point analysis is a functional size estimation approach to measure complexity of the application and costs. Feature points are similar to functional point estimation but more accurate method for scientific and complex software. Total estimation of feature points of an application includes following characteristics-

- External inputs
- External outputs
- External queries
- Internal logic files
- External logic files
- Algorithms

Each of these feature has a unique complexity value, which is used to estimate cost and efforts attributes for the application. For every general system characteristic F_i , total Feature Point Estimation (FPE) includes

$$FPE = TOTAL_COUNT * (0.65 + 0.01 * SUM (F_i));$$

After calculating FPE, effort is calculated by multiplying the estimated count with productivity measure (in hours).

- **Architecture Layer**

The legacy enterprise system is considered to have a traditional layered architecture followed upon, which further constitutes of three key layers to encapsulate application functionality to logical separation [6]. This layered architecture is very beneficial for most of the complex business applications.

The outline of layered architecture is the representation of the software system into three layers; *data layer*, which is related to database communication and interaction, *business layer* to contain the actual business logic and topmost *presentation layer* which is responsible for the application-user interaction and request handling processes.

The selection depends on the sub-division of the existing application and distribution manner. Most of the migration processed do not support simply replacing any of layer with cloud services without considerable adjustments needed

[5]. For ex. Database which are logically located on the bottom layer in the traditional application architecture, cloud service provider proposes many commonly used cloud databases used on premise, like MySQL, SQL Server, Oracle, there are possibilities of many incompatibilities and version mismatches which may not be resolved by the cloud provider [6].

In this way, the target layer which impacts the migration process, can be any single layer out of these three layers or any combination of them as shown in Figure 4.2.

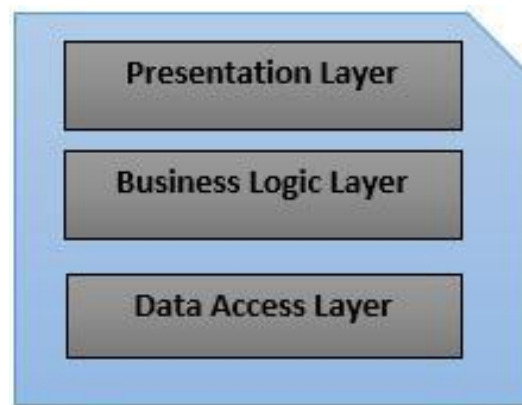


Figure 4.2: Architecture Layers in an Application

- **Application Components**

The representation of the whole application is in terms of multiple components on a single layer too. The layered architecture is not completely appropriate enough to know about the level of modernization needed in the legacy system due to

intricacy of the system. With more detailed separation, components constituting a layer can be a database server or an instance located in a separate data tier or any other small system functionality like bill calculation etc. in large and complex systems, key decisions are taken component wise according to the priority of the component and modification needed. Type of components and their roles are major criteria of them being selected for migration, a database having very high workload is more suitable to be migrated on cloud than another data resource which is dedicated solitary for reporting or analysis activities.

Selection of application components to keep locally or to migrate is a tricky decision depending upon the particular architecture of the application concerned for modernization. The resulted outcome can be:

- Single component
- Multiple components

4.2.2 Quality of Services

When modernization of applications is concerned there are various non-functional requirements that are key drivers for any modernization project that decide the degree and approaches to be followed. These requirements are not explicitly stated, but ideally each modernization process encompasses following aspects-

- **Multi-tenancy**

For a distributed application, the key pattern is how to support multiple instances with on-going multiple requests on the network. There can be two key outcomes where one option is to opt for

- Native multi-tenancy

Single application is being shared by multiple tenants on basis of shared resource provisioning.

- Multiple instances Multi-tenancy

Each tenant is being supported by a dedicated resource instance which further can be hosted in an environment comprising of either dedicated OS/Middleware or shared OS/Middleware on shared hardware. It can facilitate up to dozens of users at a time but with the increasing number of tenants, it is tedious to ensure QoS with increasing work load as well. For small or medium sized enterprises native multi-

tenancy is cost effective with lesser isolation while larger enterprises may go for more isolation (dedicated resources), hence more premium to be paid.

- Performance

Moving to the cloud environment involves the metrics used for performance of legacy application in the cloud. Main performance metrics are-

- Application running time

Time required to execute a certain functionality of the application.

- Session storing/ retrieval time
Time for loading and saving a session

- Response time

Time from end-user perspective

- Scalability

For an application being elastic concerned with all software layers, hardware and virtualization, scalability is the main issue which mostly leads to be the bottleneck in most cloud based applications. There can be three key areas which are most likely to be focused for the scalability criteria, i.e. hardware, application and virtualization level as shown in Figure 4.3 [37].

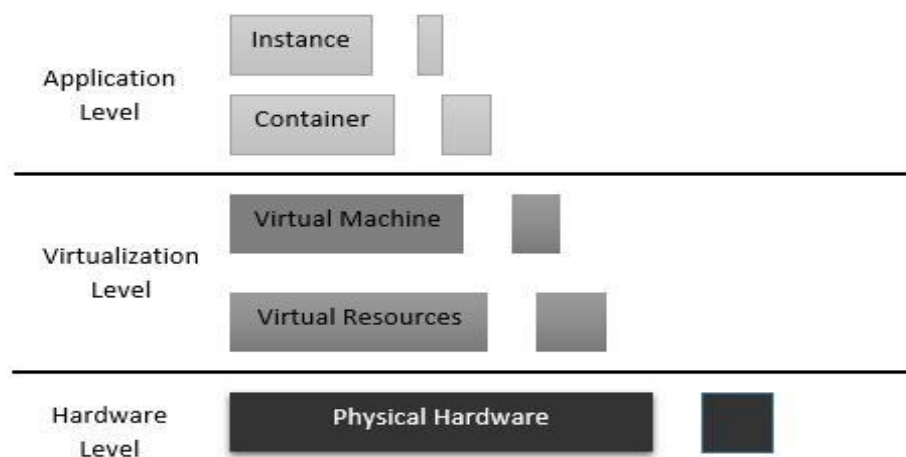


Figure 4.3: Scalability in an Application

Hardware level scaling refers to the future needs of additional physical resources such as a server instance. There can be various hardware instances which can be intended to scale up or down on the go for a cloud application. Virtualization level describes the Virtual machines configuration and management and the possibilities of scaling.

On each level, any of scaling type (vertical, horizontal and hybrid) can be applied. Vertical scaling means adding up new computing resources on demand while horizontal scaling involves assigning new instances or replicas.

On application level, no computing resources are provided so vertical scaling cannot be performed. In a cloud application, adding new instances to existing server increases total number of requests being handled by the server instance, thus instance level scaling is the highest level of scaling in a cloud application. Possible decision outcomes for one or more scalability levels are as follows-

- Instance level
- Container level
- Virtual machine level
- Virtual resource level
- Physical hardware level
- Multiple levels

4.2.3 Target Cloud Assessment

- Service Model

Selection of suitable cloud service model depends upon what kind of service is to be provided on the cloud. It is considered to be most crucial decision while assessing the cloud environment to find most efficient service model. Table 4.1 explains the selection criteria for appropriate service model.

Table 4.1: Cloud Service Model Selection Criteria

SaaS	PaaS	IaaS
<ul style="list-style-type: none"> - Applications not accepted by users enough. - Lacking main functionality vital for business logic - Application is no longer in maintenance budget. 	<ul style="list-style-type: none"> - Application requiring new functionality to be added to legacy system - Change in design and implementation of the application 	<ul style="list-style-type: none"> - Ideal for applications running on legacy hardware and no longer supported operating systems. - Systems likely to take out due to security risks, data integrity risks

- Deployment Model

According to the degree of resource sharing, following deployment models are presented from minimal to maximal sharing of distributed resources.

- Private Cloud
- Community Cloud
- Public Cloud
- Hybrid Cloud

- Pricing Policy

Cost considerations are the main issues for cloud adoption as legacy applications' capital expenses gets converted to operational expenses in a cloud environment. Pay as you go approach to facilitate resources on demand, replaces physical infrastructure expenses. Mainly two pricing models are found for cloud resources.

- Pay-per-use model
- Pay-per-charge(subscription based) model

Pay-per-use model involves billing of cloud resources such as CPU, bandwidth etc. for an instance, per hour cost for Amazon EC2 virtual server instance is \$0.113 [38]. Pay-per-unit model proposed a timespan subscription based billing method. For example for a month, a dedicated Rackspace server costs \$499. These two models facilitates the usage of cloud resources irrespective of any long-term commitments and upfront payments [39].

- Charge-per-use model pricing model includes advance subscription of the resource say a month or year beforehand. These commitments are for long term mandatory users.
- A combination of previous pricing models can also be approached.
- Also, for a certain amount limitations and specific time, some cloud services are free to use.

- Cloud vendor

Cloud service provider and its offering also play vital role in target cloud environment evaluation. Several features such as benchmarks, general reputation, vendor's capabilities, resources, skills along with pricing policies are considered while choosing a particular cloud vendor. A excel-based tool [40] has been created by Info-Tech Research Group which evaluates a certain cloud vendor based upon the expected features and shortlists them on the basis of criteria as viability, support, affordability etc.

4.3 Selection of Modernization Approach

The idea of modernization of legacy systems is resulted from the continuous and required changes from the technological improvements in the IT industry. Main approaches for modifications of the older systems are either rewriting from scratch with modern tools and hardware supports or opt for an already existing ERP solution for the concerned application. There are four main modernization approaches focusing on service oriented architectures and cloud to address the legacy system problem-

4.3.1 Replacement (full/Partial)

This strategy involves replacing the legacy application by ready-made, commercially existing off-the-shelf components (COTS) or re-writing it completely. The selection of this approach requires two main prerequisites-

- The legacy system contains obsolete and difficult to maintain technologies for long but business rules are well defined and valuable for organisation.
- This approach is to be chosen only if all other alternatives are either not feasible or costs for wrapping, migrating to clouds are not justified.

Table 4.2 discusses main strategies and associated risks with the replacement modernization approach.

Table 4.2: Replacement Approach

Strategies	Associated Risk
<ul style="list-style-type: none"> - COTS components are lesser time consuming with lower risk than rewriting the application. - Rewriting from scratch is rather expensive option than using third party components but all the required customizations can be performed. It can be either performed incrementally or all at once. Incremental redevelopment is a good option if organization has well understood structure. 	<ul style="list-style-type: none"> - If key business rules are embedded in the legacy system, COTS modification gets expensive and organisation also loses control over the component code. - Maintenance of the newer system may not be familiar as previous one. <ul style="list-style-type: none"> - The new system may have a lesser functionality compliance.

It is considered to be least desirable approach for modernization because of associated risks and complete retirement of the legacy application.

4.3.2 Wrapping to other services

Wrapping involves a black box modernization technique where the interface of the application is renewed with a new interface for ex. WSDL (Web Service Description Language) for accessibility modifications by users and other components [26]. It is only interface oriented and does not involve any internal complexity [52]. It involves reverse engineering of the system user interactions and wrapping them with new web-accessible interfaces to certain segments. Table 4.3 shows the key strategies and risks associated with the wrapping methodology. It is desirable approach when-

- Legacy system contains high quality code and valuable business logic, advantages of SOA can be comprehended by wrapping a new interface to the application.
- If legacy system is too pricey to be re-written and small in size with reusable capabilities, wrapping proves to be a faster and cost effective solution.

Table 4.3: Wrapping Approach

Strategies	Associated Risk
<ul style="list-style-type: none"> - Collect application user interactions using distinct instrumented middleware. - System interface is reverse engineered to understand dynamic system behaviour and user navigations. <ul style="list-style-type: none"> - Specific navigation paths are extracted to present a model of user's interest with interface and information exchange. 	<ul style="list-style-type: none"> - Current upgrade and maintenance problems remain unchanged. - There is no functionality changes to the legacy system - System internals may need to be analysed which require white box tools.

Wrapping strategies are desirable when a legacy system requires a quick and simple modification of service oriented functionality and system is a rich business valuable asset to the organization.

4.3.3 Reengineering

Analysis and modifications to the legacy system to represent it in a new form are part of reengineering processes [16]. These activities involve reverse engineering, restructuring, redesigning and re-implementation of application. Main strategies and associated risks with reengineering practices are discussed in table 4.4. Two main approaches for the implementation of re-engineering are system and incremental re-

engineering [8].

- **System Re-engineering**

It involves the process to reengineer the entire system full in one go. System includes all the data files, platform, source code etc. after the validation of the newer functionality, the implementation of the system is performed. Older legacy systems are retired instantly if they are centralised and can be subjected to periodic upgrades if decentralised. It is also termed as lump-sum or evolutionary re-engineering.

- **Incremental Re-engineering**

Partial or incremental Re-engineering includes basic criteria as re-integration. For small software changes, which do not require alteration in the source code language, reengineered modules are re-integrated into the existing system. But this is applicable in some of the systems where re-structuring is not needed [9].

For the other cases, functionally cohesive modules are identified, isolated and reengineered. These modules are not integrated in the legacy system. The subsequent system now contains both legacy components and reengineered modules too. An interface between the two modules is needed for careful mapping mechanism. The proposed mechanism for phased re-engineering of the software architecture involves following steps:

- Source architecture specification
- Target architecture specification
- Analysis of legacy source code(any programming language)
- Re-structure the code modules

Table 4.4: Reengineering Approach

Strategies	Associated Risks
<ul style="list-style-type: none"> - White box modernization approach involving changes to legacy code. - Legacy architecture recovery, evolution plan and execution of the plan. 	<ul style="list-style-type: none"> - Component identification, deployment are non-trivial activities.

4.3.4 Migration to Cloud services

The new age cloud based modernization solutions are very popular since last few years. They are capable of quick modernization of the legacy systems to leverage the benefits

from the technological innovations from the computing advances in the IT industry. Cloud service providers are providing these services to overcome the shortcomings of the traditional legacy applications. The following sub-sections discuss various approaches according to the cloud delivery models for legacy modernization-

- Migration to IaaS

Migrating to IaaS involves “lifting” the complete software system from present computing environment to a new operating environment with minimalistic changes made to system and its components. It comprises of all constituents like data stored in database, executable files and binaries and supporting libraries also. It is also called as rehosting a legacy application into cloud [25]. An application with old Windows NT 4.0 server and being used by a limited number of users can service more efficiently when moved to a cloud environment. Reduction of maintenance and support efforts of old hardware assets and associated failure risks are also major drivers for moving on IaaS. To decide whether rehosting can be a good option for any legacy system, selection criteria involves following reasons-

- A mainframe legacy application requiring very high maintenance though it is not supported application in the enterprise.
- System is not a business critical application and has high MIPS consumption; high CPU power.
- New functional requirements are accessible on cloud environment.

Employing infrastructure as a service includes all aspects of application except management of physical hardware and operating system are controlled by application consumer only, as depicted in Figure 4.4. There should be minimal changes to the system components which may be some configurational changes or some database connection string modifications as required. The role of cloud consumer remains almost similar to maintain and update hardware and software assets, and operating system licensing.

This approach may be an easiest or quickest one for migration of systems into the cloud. Though all benefits cannot be leveraged from cloud computing as old legacy code and responsibility of maintenance of application is still present. However, workload from traditional on premise data center gets significantly reduced for sure.

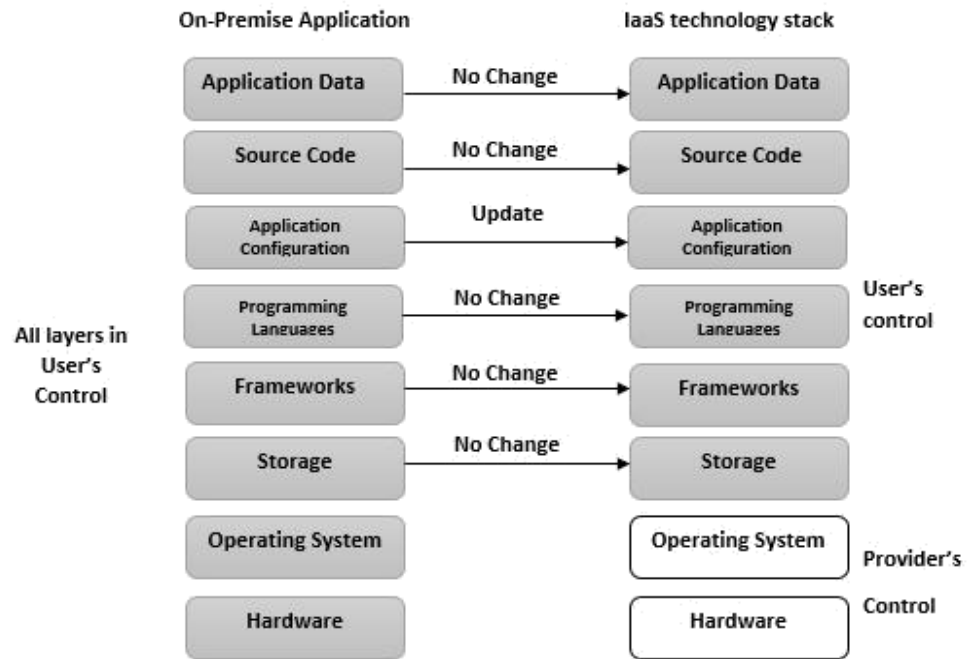


Figure 4.4: IaaS Configuration

- Migration to SaaS

Some applications are suitable to be simply replaced by an already existing cloud service. It is most cost efficient approach where all other approaches as transformation, reengineering, rehosting require time and cost. It includes replacement of the traditional system with an on premise commercial off the shelf component or some alternative present with the cloud provider [17].

The evaluation of SaaS offerings is considerable while a legacy system is targeted to be replaced. Making changes into the software with some off the shelf cloud or on premise components requires modifications to business logic rules too as these modifications can be costly or not feasible in existing scenario. Figure 4.6 describes the management roles of cloud provider on migrating an application to SaaS.

The main benefit from SaaS replacement is the complete management of hardware and system infrastructure by the cloud provider as entire IT stack gets outsourced to it. Before opting for the SaaS replacement of the application, quality check for evaluation of system whether it is still contributing valuable offerings is necessary.

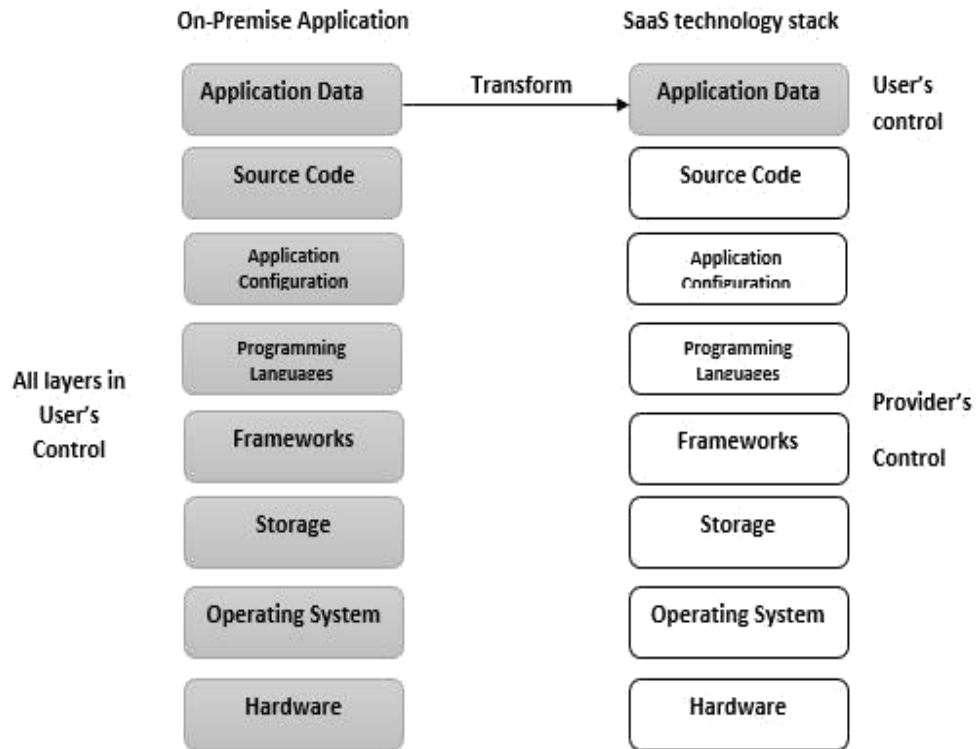


Figure 4.5: Migrating to SaaS

Gartner describes main benefits of SaaS adoption as better operational efficiency, reduction of capital expenses, increasing scalability and freeing up the mainframe data center [44]. Some drawbacks include being locked-in with a certain cloud vendor after cloud embedding with enterprise, modifying internal business logics, though well-structured service level agreements should not be avoided for an acceptable settlement.

- Migration to PaaS

Migration to platform-as-a-service is the approach when there are significant quality gaps between source and target application. It is required when rehosting or replacement of legacy application is not suitable enough as application needs redevelopment from the ground and accordingly possible changes in the business rules.

When scalability is one of the most important concern for legacy application for cloud modernization, PaaS may be a good option. All the legacy code embedded with the existing business logic needs to be extracted for analysis and functionality changes. Second major aspect with PaaS migration is portability issue. An already developed application is subjected to be ported to alternative platform, which can

further leads to vendor lock-in. Maturity of PaaS delivery model is essential feature while selecting a certain service provider to develop a new application. A PaaS provider does the job of managing most of the application stack. Cloud consumer has to perform code modification, configuration and data migration to the newer platform.

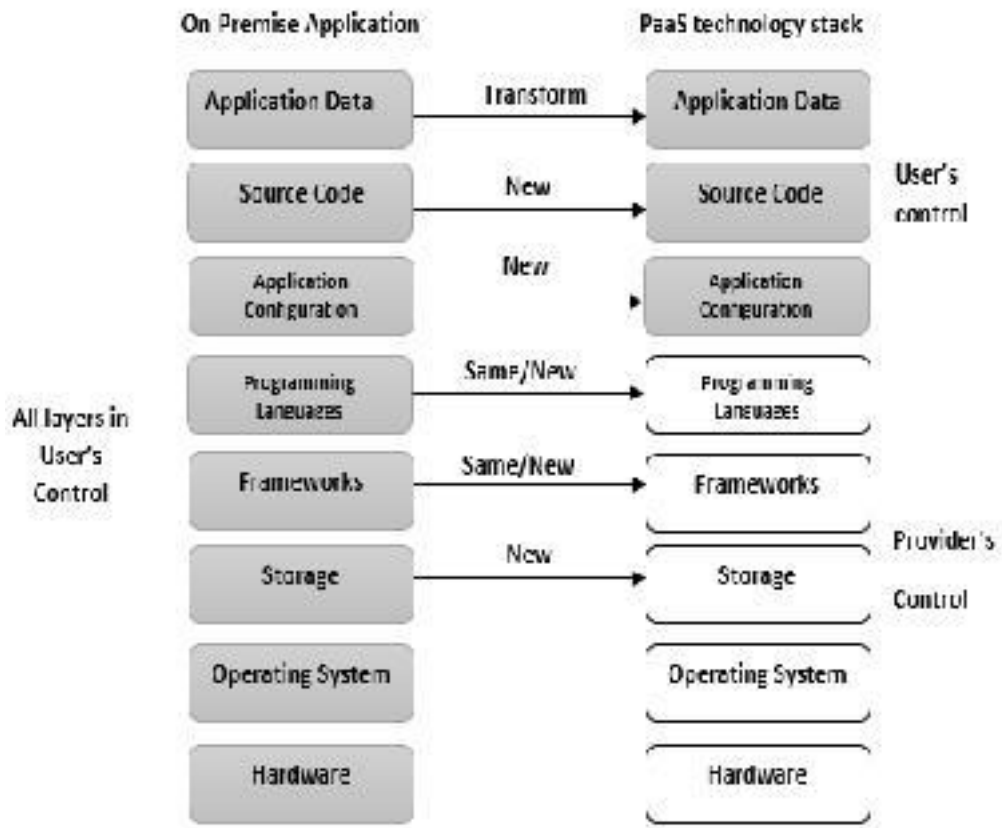


Figure 4.6: PaaS Migration

Rebuilding the application with PaaS is the most flexible way for application developers to incorporate as much as functionality to the existing system. Thus, maximum benefits from cloud computing such as resilience, scalability, built in security methods and service oriented methods are realised.

This approach proposes a unique way to leverage the full potential of PaaS offerings by architecting the application in the PaaS way. Platform available on the cloud offers many scalability options, development framework and tools, language primitives to realise the actual business requirements of the enterprise with efficient resources [51]. Table 4.5 shows the comparison of leading PaaS providers according to their offerings and preference criteria.

Table 4.5: Comparative Analysis of Leading Cloud Providers

Public Cloud	Pros	Cons	Integrated Services	Remarks
Amazon Web Services (AWS)	<ul style="list-style-type: none"> - Great flexibility for developers - Platform and language independent 	<ul style="list-style-type: none"> - Difficult for automatic scalability 	<ul style="list-style-type: none"> - Simple queue Service and Notification service for reliable communication. - Elastic load balancer for EC2 instances. - CloudWatch for monitoring services. 	Fit for mainstream of present applications
Google App Engine	<ul style="list-style-type: none"> - High level domain specific platform - Automatic load balancing, elasticity, integration with others 	<ul style="list-style-type: none"> - Applications in limited environment - Significant reimplementations for migration 	<ul style="list-style-type: none"> - Scheduled Tasks and Task Queues to perform tasks outside of web request periodically. - Built in monitoring support services. 	Good choice for building new applications
Microsoft Azure	<ul style="list-style-type: none"> - Lower layer application framework 	<ul style="list-style-type: none"> - Lack of monitoring capabilities and built in scaling support - Users have to purchase own tools. 	<ul style="list-style-type: none"> - AppFabric dynamic and distributed cache - Content delivery network for distribution over many locations - Access control service for authentication 	Best fit for systems based on Microsoft technologies.

Having considered these available cloud service approaches, there might be requirement of legacy application to be executed in a parallel computing environment deployed in cloud. Parallel computing in cloud is a promising technology to keep reducing computing resources and routine operating time. There are some applications such as text mining applications that may need to be migrated to a parallel processing framework. So the combination of cloud services with parallel computing turns out to be very cost effective especially for traditional data mining applications. In parallel processing, a pool of resources is used by a number of applications with reduced operation and administration costs.

4.4 Implementation of Selected Approach

The main purpose of modernization and migration is to incorporate the specific business needs to the legacy application with suitable approach. In previous sections, decision making module and a variety of modernization approaches have been discussed along with the selection criteria according to the source application.. After the selection of an appropriate modernization approach, implementation is preferred to

be carried out in an incremental manner. There are four main phases of tasks to be performed as depicted below in Table 4.5.

Table 4.6: Implementation Tasks

Category	Associated tasks
Installation and Configuration settings	<ul style="list-style-type: none"> - Development framework and tools - Third party tools (if any) - Appropriate cloud environment
Code Alteration	<ul style="list-style-type: none"> - Changes into the legacy code - Database connection handling - Compatibility concerns
Migration	<ul style="list-style-type: none"> - Prepare application - Prepare database scripts - Migrate the application and database with third party tools
Testing and execution in the cloud	<ul style="list-style-type: none"> - Test cases for each functionality - Special test cases for cloud

These steps also involve some efforts spent on training stage which includes a ground understanding of the application with assessment and appropriate selection of modernization approach.

4.5 Conclusion

This chapter describes the proposed framework and its modules in detail with appropriate approaches available for a legacy system considering for modernization. Not only cloud solutions, service oriented techniques, wrapping techniques and COTS replacement can be feasible for some of the legacy applications to leverage modern technological benefits.

A generalized modernization framework has been proposed in previous chapter, which is further validated in this chapter. It includes a case study of the implementation of an application using best-fit allocation scheme on cloud with parallel processing with Hadoop implementation. Hadoop means that the legacy algorithms for sequential computing need to be redesigned or converted to Map/Reduce algorithms.

5.1 Existing Sequential Implementation

The process of migrating an application into cloud needs benchmarking the application with corresponding versions of source and target application. A baseline benchmarking is required for comparison between application performances in different environments. A typical legacy application execution environment may be a Java runtime and target application is to be considered to run in a cloud with parallel processing. The allocation algorithm involves:

A given list of N items of different sizes $S_i \leq k$ where k is the capacity of the blocks in which these items are to be allocated optimally. The objective is to utilize the storage blocks as minimum as possible. The items are taken independent of each other and approach is to take each item and allocate that to best possible block as possible [59]. There are three kinds of approaches as first-fit, next fit and best fit approach for the allocation of items to the blocks.

- First fit approach: allocate the scanned item to the first available block with capacity to accommodate it.
- Next fit approach: allocate the item into same or previously used block if there is room for it.
- Best fit approach: sort the items first in increasing order and then allocate the items in such a way that each block remains with lowest space free after complete allocation. Best fit approach provides the most efficient allocation out of all these three.

The legacy sequential implementation has adverse performance on large dataset scenarios while running on a single machine. This scalability issue is one of the main reason for cloud computing being adapted by the existing systems. An application using best-fit allocation scheme has various real life implementations such as memory allocation approaches, finding out an optimal sequence of advertisements to be placed in a slot of time gap between series on a broadcasting network and sequential bin packing [56].

A sequential application using one dimensional best fit algorithm is parallelized using MapReduce and implemented on Hadoop clusters in AWS environment.

5.2 MapReduce Approach

Cloud computing paradigm offers three basic delivery models as SaaS, PaaS and IaaS according to the need of users to leverage the cloud services. Enhancement of Scalability is key idea of all three approaches. The proposed modernization framework is used to migrate the legacy sequential best-fit approach to the cloud using parallel computing technique. The cloud solution is using both IaaS (Amazon Web Services) and PaaS (Hadoop) for the facilitation of MapReduce paradigm on the cloud.

The modernization to cloud comprises of main tasks as below-

- To use multiple machine instances on the cloud to handle large datasets.
- Modification of implementation approach for making use of MapReduce paradigm for distributed computation scheme of Hadoop.

The proposed framework for modernization of legacy applications is validated for the same and essential requirements are assessed for both source and target environments. An assessment table is shown in Table where all the decision making parameters are evaluated to find the suitable outcome as per the computing needs. The suitable cloud environment is then configured to migrate the sequential legacy application to run in parallel with dataset mapped across various nodes and integrating the independent results from all of them.

Table 5.1: Validation of Modernization Framework

Selection Criteria of modernization framework	Possible Outcomes	Resulted Outcome for the application
Components	Single / Multiple components	Single Component
Complexity	Feature Points Based	
System Layer	Application/Business/Data/ Multiple layers	Application (Parallel jobs) and Data Layer (MapReduce)
Performance	Execution time with large dataset	Parallel execution of tasks
Multi-Tenancy	Native/Multiple Instances tenancy	Native tenancy
Scalability	Application/Virtual Machine/Hardware Level	Application Level
Service model	SaaS/PaaS/IaaS	PaaS (Hadoop) and IaaS (Amazon Web Services)
Deployment Model	Public/Private/Community/Hyb rid Cloud	Public Cloud (Amazon)
Pricing Policy	Pay-as-per-use or Pay-as-per- charge	Pay-as-per-charge (subscription based)
Cloud Vendor	Specific Policy	Amazon Web Services (fit for most mainstream applications)
Additional Requirements	-	Parallel Processing in MapReduce
Selected Modernization approach	Replacement/ Wrapping/ Reengineering/ Migration to the Cloud	Migration to the Amazon Cloud

5.2.1 Amazon Web Services

Amazon web services (AWS) is the name of cloud computing platform provided by Amazon for services over internet [29]. Since 2006, AWS is publically available for distributed computing framework providing IT resources and services on pay-as-you-use model with easy accessibility, scalability and cost efficiency. With no upfront costs and maintenance expenses, usage charges based on time of utilization are charged by the user. AWS has been geographically distributed in many regions (central hubs) across the world being in Eastern USA, Western USA, Ireland, Singapore, Ireland, Australia and Japan where each region further contains multiple availability zones. Figure 1.9 describes the key services provided by AWS on pay-as-you-go basis:

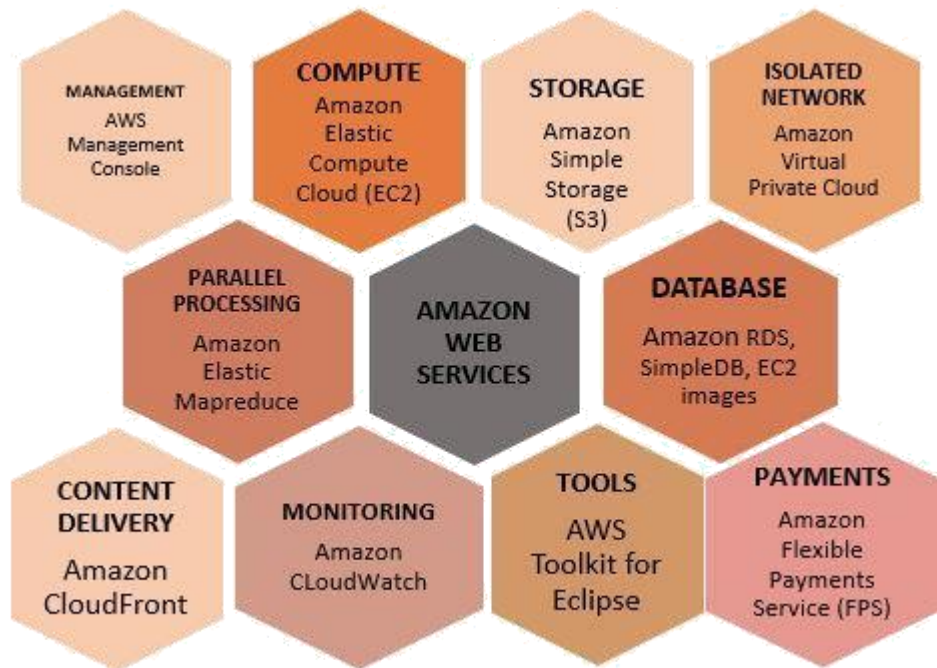


Figure 5.1: Amazon Web Services

- EC2

Amazon Elastic Compute Cloud (EC2) is the infrastructure service provisioning compute capacities on the cloud. Users can manage and control virtual servers through application programming interfaces and tools provided. A simple web interface is accessed to configure the servers like physical machines only from a resizable pool of virtual servers. Task processing, adding, removing and terminating resources is done on the basis of computing demand [32]. These resources can also be scaled up and down for their computing power on usage requirements. Three main components in EC2 are instances as virtual machines to perform computing tasks, virtual machine images (AMIs), and the computing environment for configuration and control mechanisms.

Amazon EC2 provides a range of various instances types to fit different computing needs of the clients and provide various combinations of memory, CPU, networking and storage capabilities. Table 1.3 depicts all the available instances types and their usage scenarios for computing needs.

Table 5.2: Amazon Instances and Use Cases

Instance Types	Use Cases
T2 (Burstable performance instances)	Low-traffic web services, build servers, computing environments, small databases
M4 and M3 (General purpose instances)	Mid-size databases, tasks requiring additional caching memory, cluster computing, Microsoft SharePoint
C4 and C3 (Compute optimized instances)	Web-servers, distributed computing, high performance engineering applications, video encoding.
R3 (Memory optimized instances)	High performance databases, memory caches, SAP deployments
G2 (Graphics and general purpose compute instances)	3D streaming, video encoding, machine learning, server side graphics.
I2 (Storage optimised high I/O instances)	NoSQL databases such as MongoDB and Cassandra, Hadoop and clustering
D2 (Dense storage instances)	Heavy parallel tasks, MapReduce and Hadoop file computing, data warehousing, log processing

- Amazon S3

Amazon Simple Storage Services (S3) is distributed storage infrastructure with high reliability provided for data services. It provides a data model for file system storage in terms of buckets and objects. Bucket acts as a container to hold data and metadata. Every object is assigned a unique key for retrieval purposes. Services offered by S3 include access control mechanisms to make data public or private, identification of resources using URL format and domain names. S3 clients are billed for usage of storage services monthly. These prices may differ according to the location of S3 bucket used.

- Amazon Elastic MapReduce

To process large datasets, Amazon Elastic MapReduce (EMR) is web service using Hadoop for distribution of data across Amazon EC2 instance clusters. It is built on top of Amazon S3 and EC2 and enables developers to write MapReduce programs with more focus on application functionality [47]. Easy to use AWS console and dashboard allows the scaling of computing resources as per the requirements and monitoring the jobs being executed.

It relies on Amazon S3 for application input and output storage and Amazon EC2 for Hadoop cluster as shown in Figure 5.2.

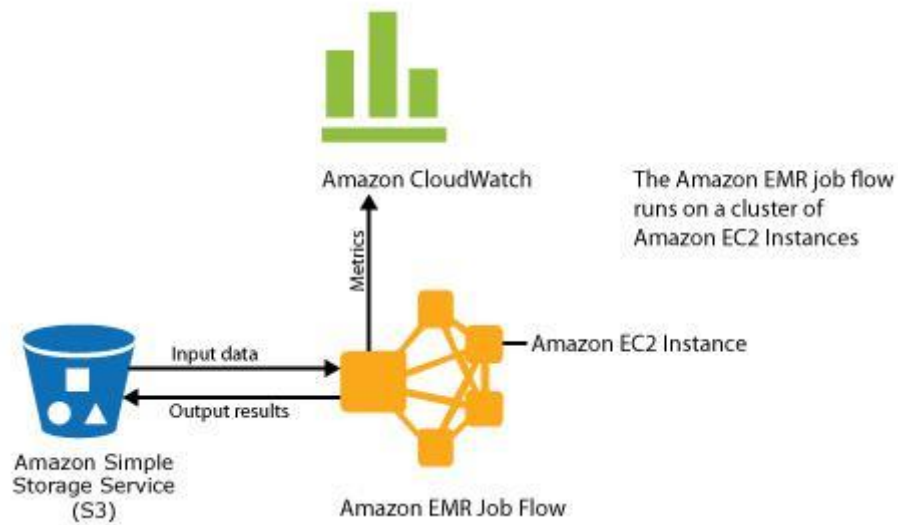


Figure 5.2: Amazon EMR Interaction [47]

Hence, Amazon web services is primarily known as a popular infrastructure as a service as a client can configure the computing resources such as EC2 instances and S3 storage as per the need. AWS also provides compatible PaaS offerings making it IaaS and PaaS in one.

AWS facilitates the development and deployment of Hadoop programs with no overhead of setting up infrastructure. Hadoop provides a successful open source implementation of Google’s MapReduce model to process large datasets and unstructured data primarily used on Linux clusters. Hadoop distributed file system by Hadoop is used to store and replicate the data using MapReduce over a cluster. Streaming Hadoop job containing Map and Reduce functions is easier with the custom source code uploaded to Amazon S3. The MapReduce implementation is explained step by step as follows-

- Signing up for Amazon web services as shown in Figure 5.3.



Figure 5.3 Signing Up Process for AWS

After signing up once to the AWS console, the login screen for future use is as shown in Figure 5.4.



Figure 5.4: Logging in to AWS Console

- A management console showing all the key services provided by the Amazon is shown as in Figure 5.5. These categories are compute, storage, application, security, database and deployment services etc.

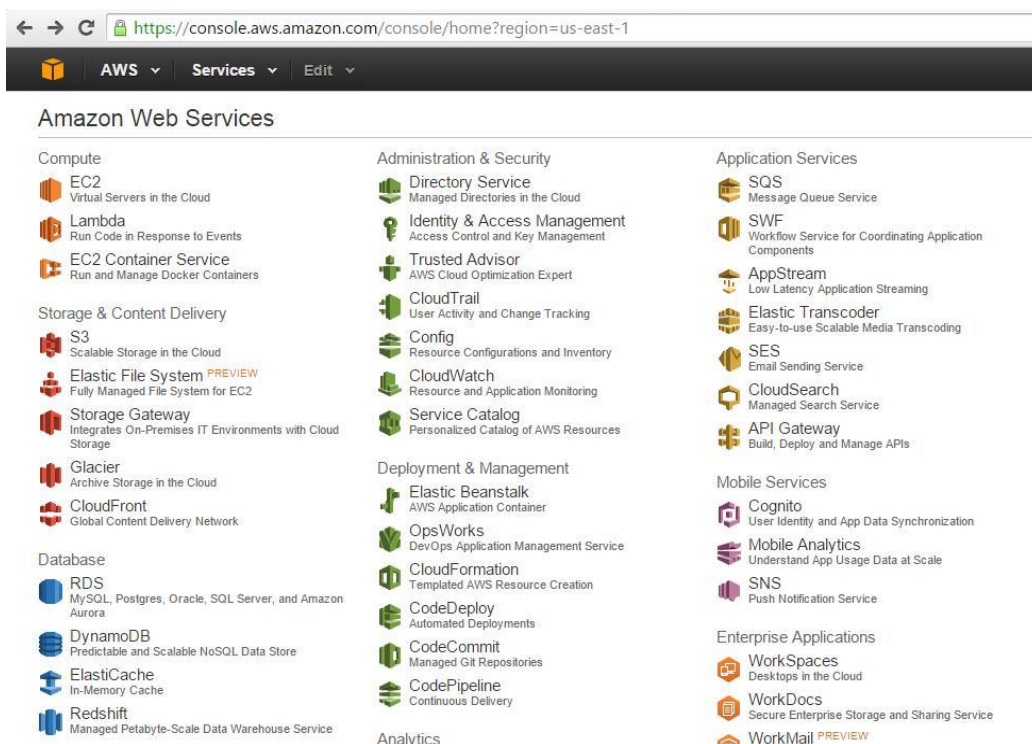


Figure 5.5: AWS Management Console Home

- A data bucket is created in Amazon S3 to host the input, output data and log files for MapReduce process. The data can be sourced from system's HDFS also in the case when S3 is not being used for jobs on elastic compute instance. Figure 5.6 and 5.7 depict the creation of a bucket with the hosted region and the folders in it for data and source file. Source.jar file is the MapReduce code file outsourced from a programming framework such as Eclipse. Figure 5.8 shows the completion of uploading files to the bucket folders.

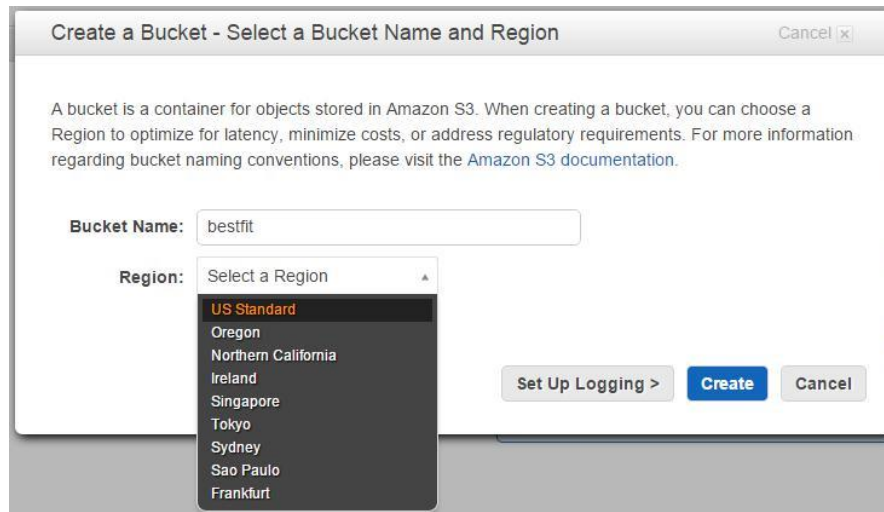


Figure 5.6: Data Bucket Creation in Amazon S3

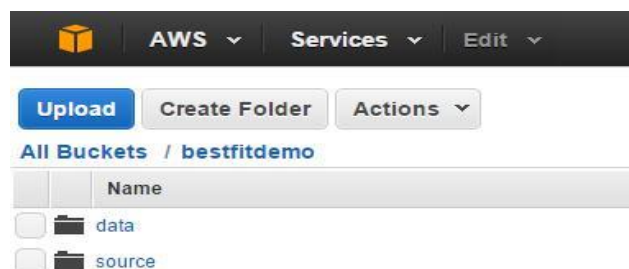


Figure 5.7: Folders in Data Bucket



Figure 5.8: Data Files Uploaded to Bucket

- After uploading the data and source files to S3, data and source files are ready to be used in computing MapReduce tasks. Amazon Elastic MapReduce (EMR) Management Console is used to create new virtual Hadoop clusters as shown in Figure 5.9.

Welcome to Amazon Elastic MapReduce

Amazon Elastic MapReduce (Amazon EMR) is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data.

You do not appear to have any clusters. Create one now:

[Create cluster](#)

How Elastic MapReduce Works

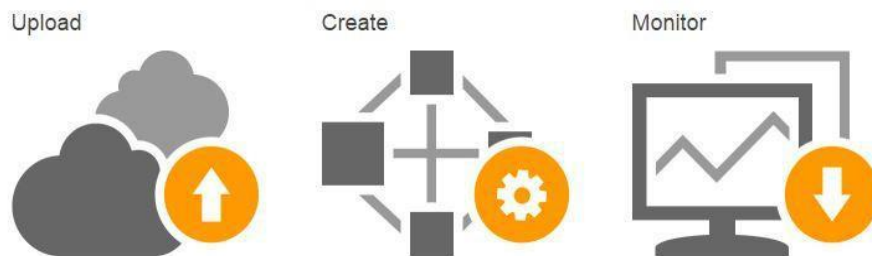


Figure 5.9: Cluster Creation

- Created single node cluster is configured with input and output locations of input data file along with source code file all present in S3 with location path.

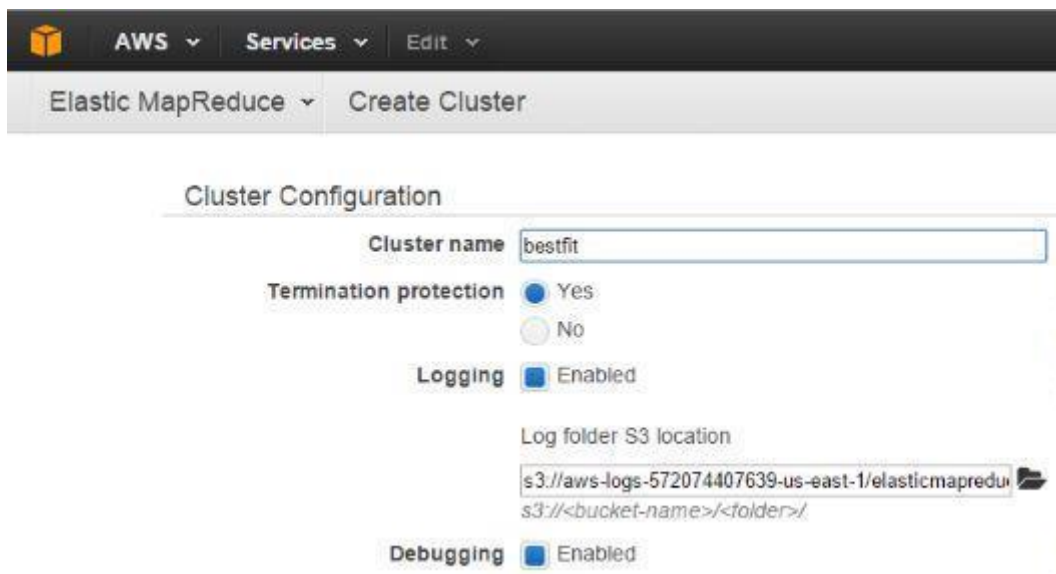


Figure 5.10: Cluster Configuration

- Next, AmazonEC2 instances are configured as per requirements and network setting are selected, if not selected default.

Hardware Configuration

i Specify the networking and hardware configuration for your cluster. If you need more capacity, you can [Request Spot instances](#) (unused EC2 capacity) to save money.

Network vpc-ec061289 (172.31.0.0/16) (default) ▼

EC2 Subnet No preference (random subnet) ▼

Type	Name	EC2 instance type	Count
Master	Master instance group - 1	m3.xlarge ▼	1
Core	Core instance group - 2	m3.xlarge ▼	2
Task	Task instance group - 3	m3.xlarge ▼	0

Figure 5.11: Hardware Configuration

- Required Hadoop version is selected, version 3.8.0 is the latest one provided by Amazon.

Software Configuration

Hadoop distribution Amazon

AMI version

3.8.0 ▼

Figure 5.12: Hadoop Configuration

- For security and access control mechanisms, a Key pair is generated for the account holder to authenticate the Hadoop process. EC2 Key pair is selected as shown in Figure 5.13.

Security and Access

EC2 key pair sumanjain ▼

IAM user access sumanjain

No other IAM users

Use an existing EC2 key pair to SSH into the master node of the Amazon EMR cluster. [Learn more](#)

Proceed without an EC2 key pair

Control the visibility of this cluster to other IAM users. [Learn more](#)

Figure 5.13: Security Key Pair

- An additional step for Map Reduce program is created with the specification of mapper and reducer jar file details and input output data folders in Figure 5.14.

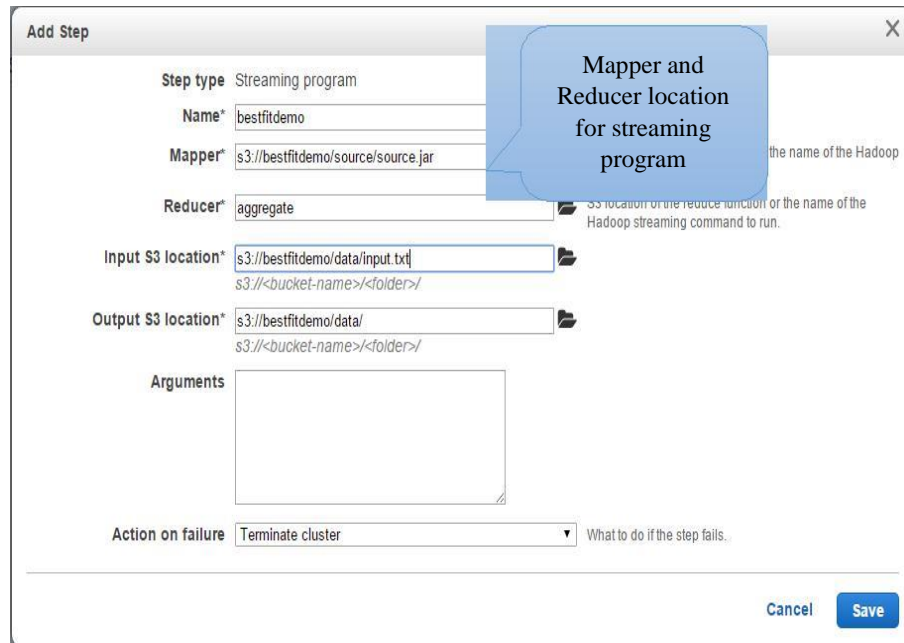


Figure 5.14: Mapper-Reducer Configuration

- After configuration, created cluster is launched (in Figure 5.15) and monitored for processing steps as shown in Figure 5.16.

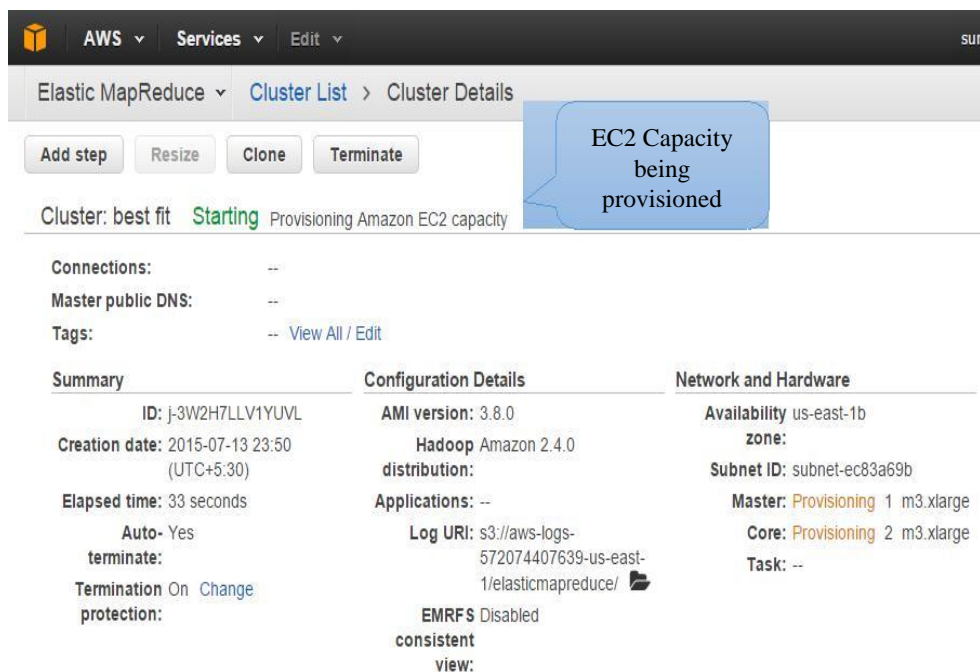


Figure 5.15: Launching the Cluster

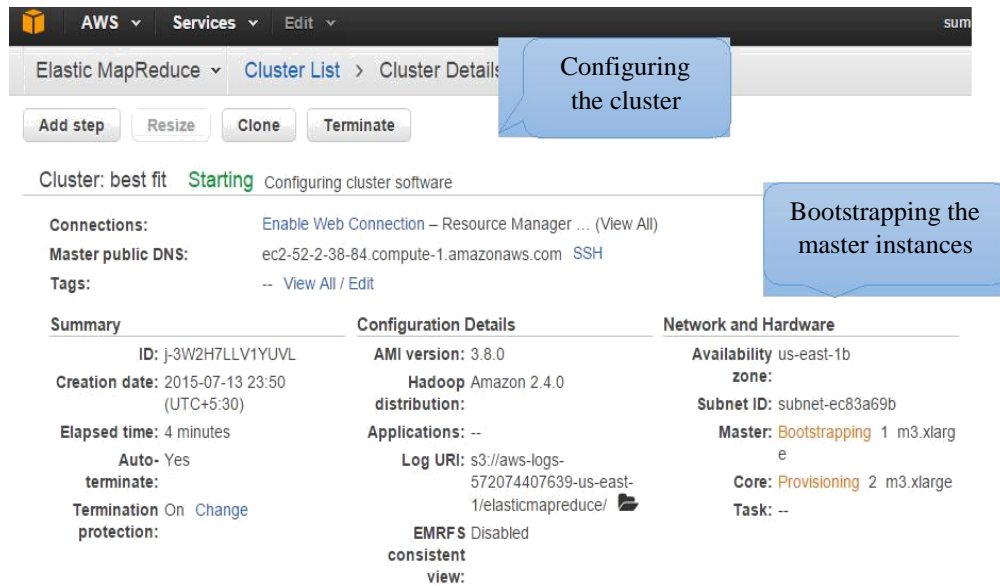


Figure 5.16: Resource Provisioning by Amazon

MapReduce approach for best fit allocation has following phases:

- Collection of the items in a buffer along with their count and the size of block in which these items are to be stored.
- In mapping phase, list of items is sorted in increasing order. Each mapper node process these items independently. For a particular block, non-decreasing ordered list of items is resulted having optimal accommodation.
- After getting mapper phase outputs, for every block, very first combination of items without exceeding its capacity is searched using dynamic programming.
- Extracted items are appended to the block's list and removed from original item's list.

5.3 Results

The system used to access AWS console and conduct the experiments had the following configuration:

Processor: Intel(R) Core (TM) i3-3110M CPU @ 2.40GHz.

Installed memory (RAM): 4.00 GB (3.87GB usable)

System Type: 64-bit operating system with x64-based processor.

The implementation of both sequential and parallel best-fit algorithms in Java environment with dataset ranging from 0.5MB to 1000MB have been performed. These algorithms were tested by running them on Amazon Hadoop 3.8.0 cluster with one master

instance m3.xlarge. Both sequential java application and Hadoop streaming were implemented one by one and their execution time was calculated simultaneously. As shown in Table 5.3, raw execution times for the various dataset file sizes are recorded in each independent test run.

Table 5.3: Execution Time of Sequential and Parallel Approaches

Dataset Size (MB)	Test Run	Sequential Java App (sec)	Parallel Hadoop Stream (sec)
0.5	1	5	28
	2	6	27
20	1	79	176
	2	77	165
100	1	556	309
	2	547	308
1000	1	-	868
	2	-	871

Arithmetic mean execution time is computed for each dataset size and then comparison of both the applications is presented in Table 5.4.

Table 5.4: Mean Execution Time Comparison

Dataset size (MB)	Mean Execution time of Java App (sec)	Mean Execution time of Hadoop streaming (sec)
0.5	5.5	13.5
20	78	170.5
100	551.5	308.5
1000	-	869.5

The derived result displays the performance of variation in dataset size on implementation with MapReduce in Table 5.1. Graphical representation of the retrieved data is depicted in Figure 5.13.

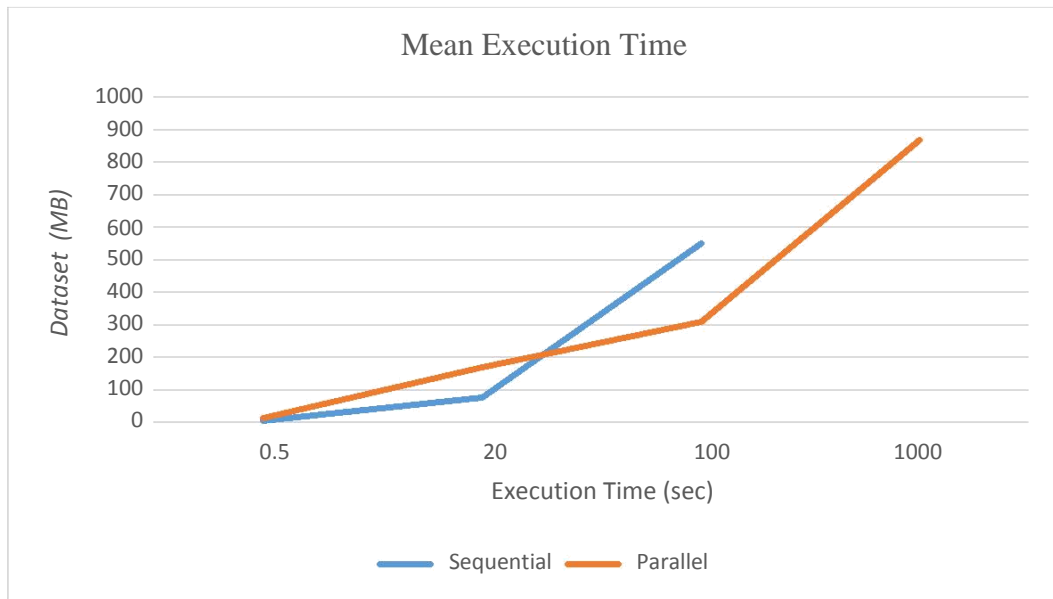


Figure 5.17: Graphical Representation of Performance Execution

According to the results, when the dataset is significantly small in size, Hadoop takes longer than sequential run which is not preferable to be moved to the parallel environment. Hadoop needs time to initialize the dataset which involves a lot of time as compared to previous approach. Hence, for larger datasets, MapReduce framework provides a significantly beneficial approach.

6.1 Conclusion

The rise of cloud computing paradigm has influenced most of the organizations to look after cloud technologies for newer applications to leverage as many benefits of cloud services with appropriate evaluation mechanisms for cloud assessment. The legacy problem depicts the status of organizations having a traditional data center with complex applications with obsolete programming environments, no longer supporting operating systems. With increase in complexity of these older systems, applications are increasing overhead on the management and business roles. Hence, even small and medium IT enterprises have started to search for optimum approaches to benefit from the cloud services with legacy modernization methods.

The main objective of thesis is to solve the legacy problem with different modernization approaches whether cloud is a suitable solution for particular application or not. This need of clear understanding of cloud offerings and corresponding legacy requirements. Every enterprise may not be able to hire consultants for cloud recommendations. Thus to provide a generalised framework which provides all the selection criteria for cloud features and legacy characteristics are the main research goals behind this study. Main approaches include first options where cloud computing may not be needed for modernization of legacy application and later cloud migration techniques to opt for particular delivery model. The proposed framework also involves integration of cloud computing with parallel processing platforms for efficient resource management.

6.2 Thesis Contribution

The contributions made by the proposed work are as follows-

- A review showing significant researches in academic industry for cloud computing migration and parallel processing techniques in cloud has been presented.
- A generic assessment model for legacy applications and sequential systems which may need to be migrate to the cloud or modernize with some newer approach is designed. This framework categorizes all the aspects from source application to

target cloud model along with the essential quality characteristics expected from the systematic process.

- An approach to utilize parallel computing techniques for optimum resources utilization has been implemented using Amazon web services and MapReduce.

6.3 Future Scope

Research area of cloud computing has been growing exceptionally with modern technological advancements to cater high quality cloud services. As we have found some limitations still existing for cloud modernization of all the applications existing out there with traditional enterprises.

- Cloud migration of mission critical systems is not favourable because of security and latency issues. So as technology gets advanced, cloud service providers have to be able for high bandwidth provisioning for assured performances of critical applications also.
- It is the elementary need of cloud modernization field to provide suitable assessment technologies. Also, there is lack of supporting automated tools for the recovery and migration processes, which is a major area to work upon.

References

- [1] *Institute of Electrical and Electronics Engineers, International Organisation for Standardisation, International Electrotechnical Commission: Software Engineering - Software Life Cycle Processes - Maintenance.*, New York: Institute of Electrical and Electronics Engineers, 2006.
- [2] R. Gmnich and A. Winter, "Softwaretechnik-Trends," *Workflows der software-migration*, pp. 22-24, 2005.
- [3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stocia and M. Zaharia, "Above the Clouds: A Berkeley view of cloud computing," Dept of Electrical Eng. and Computer Sciences, University of California, Berkeley, 2009. [Online]. Available: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>.
- [4] D. F. Parkhill, "The Challenge of the Computer Utility.," *Addison-Wesley Publishing Company*, 1966.
- [5] H. M. Sneed, M. Hasitschka and M. Teichmann, "Software Migration. Dpunkt, Heidelberg," *Necker (2008)*.
- [6] V. Andrikopoulos, T. Binz, F. Leymann and S. Strauch, "How to adopt applications for the cloud environment: Challenges and solutions in migrating applications to the cloud.," *Computing* 95(6), pp. 493-535, June 2013.
- [7] M. L. Brodie and M. Stonebraker, "Migrating Legacy Systems: Gateways, Interfaces & the Incremental Approach," *Morgan Kaufmann Publishers Inc., San Francisco, CA, USA*, no. ISBN 1-55860-330-1, 1995.
- [8] M. R. Olsem, "An Incremental Approach to Software Systems Re-engineering," *Software Maintenance: Research and Practice*, 1998.
- [9] M. Antoun and W. Coelho, "A Case Study in Incremental Architecture based Re-engineering of a Legacy Application".
- [10] H. Zhou, H. Yang and A. Hugill, "An Ontology-Based Approach to Reengineering Enterprise Software for Cloud," in *IEEE 34th Annual Computer Software and Applications Conference*, 2010.
- [11] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud Computing and emerging IT platforms : Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer System; Elsevier*, pp. 599-616, 2009.
- [12] S. Jain and I. Chana, "Reengineering process of legacy systems for the cloud: An Overview," in *IEEE International Conference on parallel, grid and distributed computing*, Solan, Dec 2014.

- [13] I. Foster, Y. Zhao, I. Raicu and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," in *IEEE Grid Computing Environment Workshop, GCE'08*, Austin, TX.
- [14] W. Kim, S. D. Kim, E. Lee and S. Lee, "Adoption issues for cloud computing," in *11th International Conference on Information Integration and Web based Applications and Services*, 2009.
- [15] B. Rimal, E. Choi and I. Lumb, "A Taxonomy and Survey of cloud computing Systems," in *Fifth International Joint Conference on INC, IMS and IDC*, 2009.
- [16] Manzella and Mutafelija, *Concept of Re-engineering Life Cycle*, IEEE, 1992.
- [17] x. Meng, J. Shi, X. Liu, H. Liu and L. Wang, "Legacy Applications Migration to the Cloud," in *4th International Conference on Cloud Computing*, 2011.
- [18] M. A. Babar and A. M. Chauhan, "A Tale of Migration to Cloud Computing for Sharinf Experiences and Observations," in *2nd International Workshop on Software Engineering for Cloud Computing*, 2011.
- [19] W. Hasselbring, R. Reussner, H. Jaekel, J. Schlegelmilch, T. Teschke and S. Krieghoff, "The Dublo Architecture Pattern for Smooth Migration Business Information Systems: An Experience Report," in *26th International Conference on Software Engineering(ICSE-04)*, Los Alamitos, California, 2004.
- [20] A. Fuhr, T. Horn and A. Winter, "Model-Driven Software Migration," *G. Engels, M. Luckey, and W. Schäfer, editors, Software Engineering 2010: Fachtagung des GI-Fachbereichs*, Vols. P-159, 2010.
- [21] J. Bisbal, D. Lawless, B. Wu and J. Grimson, "Legacy Information Systems: Issues and Directions," *Software, IEEE*, pp. 103-111, 1999.
- [22] B. Wu, D. Lawless, J. Bisbal and R. Richardson, "The Butterfly Methodology: A Gateway Approach for Migrating Legacy Information Systems," in *3rd IEEE Conference on Engineering of Complex Computer Systems (ICECCS'97)*.
- [23] W. Shang, Z. Jiang, B. Adams and A. E. Hassan, "MapReduce as a general framework to support research in mining software repositories (MSR)," in *6th IEEE International Working Conference on mining software repositories*, Washington, DC, USA, 2009.
- [24] B. Panda, J. S. Herbach, S. Basu and R. J. Bayardo, "PLANET: massively parallel learning of tree ensembles with MapReduce".
- [25] J.-F. Zhao and J.-T. Zhou, "Strategies and Methods for Cloud Migration," *International Journal of Automation and Computing*, pp. 143-152, 2014.
- [26] L. Zheng and S. Wu, "An Infrastructure for Web Service Migration in Clouds," in *International Conference on Computer Applications and System Modelling*, 2010.

- [27] A. Zimmermann, M. Pretz, G. Zimmermann, D. G. Firesmith, I. Petrov and E. El-Sheikh, "Towards Service-oriented Enterprise Architectures for Big Data Applications in the Cloud," in *17th IEEE International Enterprise Distributed Object Oriented Conference Workshops*, 2013.
- [28] W.-T. Tsai, X. Sun and J. Balasooriya, "Service-Oriented Cloud Computing Architecture," in *IEEE International Conference on Information Technology*, 2010.
- [29] J. Varia, "Migrating your Existing Applications to the AWS Cloud," October 2010. [Online]. Available: <http://www.amazon.com>.
- [30] S. Tilley and T. Parveen, "Research Directions in Web Systems Evolution III:," *WSE 2009 Working Session*.
- [31] H. M. Sneed, "Encapsulation of Legacy Software: A Technique for Reusing Legacy Software Components," *Annals of Software Engineering*, vol. 9, pp. 293-313, 2000.
- [32] E. A. N. d. Silva and D. Lucreidio, "Software Engineering for the Cloud: a Research Roadmap," *Brazilian Symposium on Software Engineering*, 2012.
- [33] V. Rajlich, "Comprehension and Evolution of Legacy Software," *Wayne State University*.
- [34] F. Liu, J. Tong, J. Mao, R. B. Bohn, J. V. Messina, M. L. Badger and D. M. Leaf, "NIST Cloud Computing Reference Architecture," *NIST Publications*, 2011.
- [35] A. Lenk, M. Klems, J. Nimis and S. Tai, "What's Inside the Cloud? An Architectural Map of the Cloud Landscape".
- [36] Chong, F., & Carraro, G. (2006). "Architecture strategies for catching the long tail". *MSDN Library, Microsoft Corporation*, 9-10.
- [37] L. M. Vaquero, L. Rodero-Merino and R. Buyya, "Dynamically scaling applications in the cloud.," *ACM SIGCOMM Computer Communication Review*, p. 45, January 2011.
- [38] "Amazon Web Service Inc.: Amazon ec2 pricing, pay as you go for cloud computing services," 2013. [Online]. Available: <http://aws.amazon.com/en/ec2/pricing/>.
- [39] "Rackspace, US Inc.: Pricing for dedicated servers by rackspace.," 2014. [Online]. Available: <http://www.rackspace.com/managed-hosting/dedicated-servers/pricing/>.
- [40] "Info-Tech Group: Cloud vendor shortlisting tool," June 2011. [Online]. Available: <http://www.infotech.com/research/cloud-vendor-shortlisting-tool>.

- [41] “Nexaweb. The CIO ’ s New Challenge Creating a Business Case for Application Modernization.,” 2008. [Online]. Available: http://www.saasmarkets.com/uploads/u_926/appl/757/resources/CIO-ModernizationWhitePaper-Nexaweb-October2008.pdf.
- [42] P. Mell and T. Grance, “ The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology,” 2011. [Online]. Available: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.
- [43] “Microsoft Inc. The Business Value of Legacy Modernization.,” 2007. [Online]. Available: http://download.microsoft.com/download/e/9/7/e9734f87-c581-482a-aaca-2835df48d40e/business_value_legacy_modernization.pdf.
- [44] A. T. Manes, “ Moving Apps and Data to the Cloud : Migration Options,” 2011. [Online]. Available: <http://agendabuilder.gartner.com/CATUS2/webpages/SessionDetail.aspx>.
- [45] H. C. Yang, A. Dasdan, R. L. Hasio and D. S. Parker, ““MapReduce-Merge: simplified Relational Data Processing on Large Clusters,” in *ACM SIGMOD Interantional Conference Management of Data*, 2007.
- [46] A. W. S. LLC, “Amazon Elastic Compute Cloud (Amazon EC2),” 2009. [Online]. Available: <http://aws.amazon.com/ec2/>.
- [47] A. W. S. LLC, “Amazon Elastic MapReduce,” 2009. [Online]. Available: <http://aws.amazon.com/elasticmapreduce/>.
- [48] D. Warneke and O. Kao, “Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 6, pp. 985-989, June 2011.
- [49] M. Isard, M. Budiu, Y. Yu, A. Birrell and D. Fetterly, “Dryad:Distributed Data-Parallel Programs from Sequential Building Blocks,” in *Second ACM SIGOPS/EuroSys European Conference Computer Systems*, 2007.
- [50] “Apache Hadoop. What Is Apache Hadoop?,” 2014. [Online]. Available: <http://hadoop.apache.org/>.
- [51] G. Semones, “Building Cloud-ready, multi-core friendly applications, Part-1: Design principles-Javaworld,” 2009. [Online]. Available: <http://www.javaworld.com/javaworld/jw-03-2009/jw-03-multicore-and-cloud-ready-1.html?page=1>.
- [52] R. C. Seacord, K. Wallnau and J. Robert, “A Survey of Black-Box Modernization Approaches for Information Systems,” in *International Conference on Software Maintenance*, San Jose, California, 2000.

- [53] F. Fleurey, E. Breton, B. Baudry and A. Nicolas, "Mode Driven Engineering for Software Migration in a Large Industrial Context," *Model Driven Engineering Language and Systems*, pp. 482-497, 2007.
- [54] P. Mohagheghi, A. J. Berre, A. Sadovykh, F. Barbier and G. Benguria, "Reuse and Migration of legacy systems to interoperable cloud services-the REMICS project," *Proceedings of Mda4ServiceCloud*, 2010.
- [55] E. G. Coffman, M. R. Garey and D. S. Johnson, "An Application to bin-packing to multi-processor scheduling," *SIAM Journal of Computing*, vol. 7, no. 1, 1978.
- [56] D. S. Johnson and M. R. Garey, "A 71/60 Theorem for Bin Packing," *Journal of Complexity*, pp. 65-106, 1985.
- [57] L. Epstein, L. M. Favrholt and J. S. Kohrt, "Comparing Online Algorithms for Bin Packing Problems," *Journal of Scheduling*, vol. 15, no. 1, pp. 13-21, February 2012.
- [58] R. J. Anderson, E. W. Mayr and M. K. Warmuth, "Parallel Approximation Algorithms for Bin Packing," *Information and Computation*, vol. 82, no. 3, pp. 262-277, 1989.
- [59] A. Gyorgy, G. Lugosi and G. Ottucsak, "Online Sequential Bin Packing," *Journal of Machine Learning Research*, vol. 11, pp. 89-109, 2010.
- [60] Albrecht, A. J., & Gaffney Jr, J. E. (1983). "Software function, source lines of code, and development effort prediction: a software science validation". *Software Engineering, IEEE Transactions on*, (6), 639-648.
- [61] Dolado, J. J. (2000). "A validation of the component-based method for software size estimation". *Software Engineering, IEEE Transactions on*, 26(10), 1006-1021.
- [62] Tran, V. T., Lee, K., Fekete, A., Liu, A., & Keung, J. (2011, September). "Size estimation of cloud migration projects with cloud migration point (CMP)". In *Empirical Software Engineering and Measurement (ESEM)*, 2011 International Symposium on (pp. 265-274). IEEE.
- [63] Khajeh-Hosseini, A., Greenwood, D., Smith, J. W., & Sommerville, I. (2012). "The cloud adoption toolkit: supporting cloud adoption decisions in the enterprise". *Software: Practice and Experience*, 42(4), 447-465.
- [64] Menzel, M., & Ranjan, R. (2012, April). "CloudGenius: decision support for web server cloud migration". In *Proceedings of the 21st international conference on World Wide Web* (pp. 979-988). ACM.
- [65] Cisco Systems. (2010). *Planning the Migration of Enterprise Applications to the Cloud*. Retrieved from http://www.cisco.com/en/US/services/ps2961/ps10364/ps10370/ps11104/Migration_of_Enterprise_Apps_to_Cloud_White_Paper.pdf

- [66] Beaty, K., Kochut, A., & Shaikh, H. (2009). Desktop to Cloud Transformation Planning. *Symposium A Quarterly Journal In Modern Foreign Literatures* (pp. 1– 8). Rome: IBM Watson Research Center. doi:10.1109/IPDPS.2009.5161236
- [67] HP. (2012). HP Applications Suitability Mapping for Cloud. Available Online: <http://h20195.www2.hp.com/V2/GetPDF.aspx/4AA3-3271ENW.pdf>
- [68] M. M. Cristian, "Database replication," *database system journals*, vol. 1, no. 2, pp. 33-38, 2010.
- [69] D. Jeffrey and G. Sanjay, "Mapreduce: simplified data processing on large clusters," vol. 51, *Communications of the ACM*, 2008, pp. 107-113.
- [70] Silva, L. M., & Buyya, R. (1999). Parallel programming models and paradigms. *High Performance Cluster Computing: Architectures and Systems*, 2, 4-27.
- [71] A. Simitsis and P. Vassiliadis, "A methodology for the conceptual modeling of ETL processes," *research gate*, 2014.
- [72] R. Prodan and S. Ostermann, "A survey and taxonomy of infrastructure as a service and web hosting cloud providers", 2009 10th IEEE/ACM International Conference on Grid Computing, pp. 17-25.
- [73] Vu, Q. H., & Asal, R. (2012). Legacy Application Migration to the Cloud: Practicability and Methodology. 2012 IEEE Eighth World Congress on Services, 270–277. doi:10.1109/SERVICES.2012.47

List of Publications

- Published research paper “Reengineering process of legacy applications to the cloud: An Overview” in IEEE International Conference on Parallel, Grid, Distributed Computing, Jaypee University of Information Technology, Solan India. (Dec 2014)
- Accepted research paper “Incremental Architectural Re-engineering for the Cloud’ in IEEE International Conference on Computing, Communication and Automation in Galgotias University, Noida (ICCCA 2015)
- “Legacy System Modernization: A Generalized Framework” has been communicated in International Conference on Computer and Communication Technology MNNIT, Allahabad, India. (ACM).

YouTube Link

- The link for thesis video presentation is:
<https://www.youtube.com/watch?v=GScGwI2Cg8M>