Software performance is one of the key non-functional requirements of software and becoming a vital concern for all the stakeholders of software projects, predominantly for users and developers. Smith and Williams (2002) define that Software Performance Engineering (SPE) is a systematic, quantitative approach to construct software systems that meet performance requirements. Software performance plays a vital role in any software product solution. Software solutions for business and other problems originally for standalone application on desktop personal computers, progressively moved to network based client/server and distributed computing solutions due to advancements in network technology.

The advent of Internet and its tremendous exponential growth enabled computing through Internet and a lot of solutions emerged based on Internet exploiting Internet infrastructure. Recent technological proliferation in web services has seen wide usage of the web service technology in software solutions due to its unique interoperability capability. This has provided opportunities for the exploitation of the existing Internet infrastructure by using open standards and thus making the web services technology portable to software solutions.

The new wave of technology in distributed computing paradigm, known as web service, is being used in e-business web portals. In these kinds of web applications, web service performance has very high impact on user
satisfaction. Experiments show that users can tolerate about eight seconds of delay before they either repeat their request or leave the site (Bhatti et al 2000). Poor response times may result in low service usage and it will affect service revenue (Zhichun Li et al 2010).

Amazon reported approximately 1% loss in sales due to 100 ms extra delay (Zhichun Li et al 2010). Another study by Google revealed that 500 ms extra delay in search results may cause for decrease in revenues to the extent of 20% (Ron Kohavi et al 2007). Even worse, users may abandon a service provider and move to another service provider. This has forced service providers to think possibilities for optimization and improving performance of web services. The performance of a single web service has the potential to affect the performance of an entire web process and so it is better to evaluate the performance of the services within a process before making it available for commercial usage (Chandrasekaran et al 2003).

Thus this thesis focuses on performance aware implementation and selection of web services predominantly the software support for implementation and selection of web services in terms of performance.

1.1 MOTIVATION

The vast outreach of Internet technology has created necessity for interoperability between software applications such as healthcare applications, financial applications, telecommunication systems etc. Web service has features like code reuse, interoperability, open standards etc., and it has become an ingredient in software solution for solving the problems of application and data integration.
Web service usage has experienced exponential growth and so there is an increasing need to address research issues on web service such as scalability, availability and performance enhancement etc., which is very much indispensable to satisfy all the stakeholders of software. For instance, web service usage in large scale business applications like stock quote service for stock market, news (RSS) feed service, distributed authentication service for credit cards etc., illustrate the use of web services in mission-critical applications. These applications demand very high level of scalability but without compromising performance and so improving web service performance is always a concern.

Giovanni et al (2005), in their work on performance of web services, stressed the “need to develop models of web services and web service traffic loads to study and predict platform performance under different traffic and service conditions.” Lei Gao et al (2005) pointed out that performance of databases play a role in overall performance of web services.

Moreover Frank Coyle (2002) pointed out that “web services and SOAP-based connections do not currently have the key building blocks for web-based e-business and messaging, transactions, security, identity are key ingredients for web commerce which are to be supported by software platforms”. Vittorio et al (2001) pointed out that responsiveness is a critical issue in the design of a web service and a poorly responsive service can be considered as an unavailable service. The SLA is purported for defining the responsibility relation between the user and provider and to guarantee the quality of service (Dongjoon et al 2005).

So web service provider is liable for the user to provide quality of service mentioned in the SLA. Though web services have been adopted by major software vendors such as IBM, Oracle, Borland, and led to plethora of research activities in web services, there are no studies on web services in
terms of software metrics (Yijun Yu et al 2007). Studies on web services metrics will benefit the development and management of web services and research activities on web services (Yijun Yu et al 2007). Thus this thesis describes two aspects of performance metrics for evaluating the performance, one is time taken for response which is the ultimate aim of any software system and the other is communication payload. Particularly, response time plays a vital role in real-time systems, mobile applications etc., and it is the key non-functional requirement from users. The other metric is size of data transmitted through wire because web services are realized through Internet and communication over the Internet involves both time and cost.

The novel approach in this work is that service traffic load is measured with all possible combinations of software platforms and databases in different tiers of realization of web services. To the best of our knowledge, earlier research works lack in these particular aspects of web services performance and thus this thesis attempts to empirically focus on the following research issues:

(i) What is the extent to which, software platforms support for web services particularly in terms of performance?

(ii) What is the extent to which, databases support for web services in terms of performance?

(iii) How to select a web service efficiently by applying SLA (Service Level Agreement) performance requirements parameters ranking strategy in cloud broker?
1.2 CONTRIBUTION OF THESIS

This thesis is a milestone in performance aware implementation and selection of web services. This thesis provides quantitative results of performance support provided by various software technologies involved in implementation of web services. The quantitative results are expressed using a novel set of software metrics. This thesis proposes a novel methodology to migrate from legacy software systems to web services with the use of software metrics knowledge gained during reverse engineering process. A new technique, service level agreement parameters ranking strategy contributes for efficient web service selection in cloud brokers. The experimental results contribute a lot to determine performance of web services and also to design web services with predictable performance.

The performance metrics data collected through implementation of web services by using standard software platforms and collection of databases provide empirical observation of performance aspects of each method. This will give insight into technical aspects of implementation with the focus on performance. Performance enhancement techniques may be explored based on the metrics data.

A novel methodology is proposed to migrate from legacy software systems to web services by exploiting the advantage of .NET compliance to many programming languages and using software metrics knowledge to choose among wrapping or redevelopment.

In cloud broker, match-making functionality that is matching between customer requirements and cloud service is an essential part of cloud broker functionalities. This work contributes to efficient web service selection through service level agreement parameters ranking strategy to understand the priority desired by customers. This work deployed tree data
structure and sparse matrix for storing and organizing priority data and a set of metrics suite proposed for analysis of priority data.

1.3 ORGANIZATION OF THESIS

The rest of the thesis is organized as follows.

Chapter 2 discusses literature review in evaluation of performance of web services.

Chapter 3 provides foundation for the thesis in detailing software metrics, need for measuring software, web services, standards involved in web services, and web services architecture models.

Chapter 4 discusses empirical research in software engineering, its significance, various methods and threats to validity. This chapter also discusses research methodology of this research work with the focus on empirical research in this work.

Chapter 5 explains empirical evaluation of performance support from software platforms involved in implementation of web services with the use of software metrics proposed for measuring performance.

Chapter 6 explains service traffic load involved in utilizing web services in different combinations of software deployed for implementation of web services.

Chapter 7 explains empirical study of performance support from various standard databases in implementation of web services because database is used in the back end of web services to handle large amount of data.
Chapter 8 explains a technique to migrate from legacy software systems to web services by exploiting programming language agnostic characteristic of .NET and also by using software metrics information gained during reverse engineering process.

Chapter 9 explains service level agreement parameters ranking strategy in selecting a right web service based on parameters priority desired by customers and a set of metrics proposed for analyzing parameters priority decided by customers.

Chapter 10 discusses conclusion and future work in each of the works discussed in the thesis.