International Journal of Cloud-Computing and Super-Computing
Vol.2, No.2 (2015), pp.37-46
http://dx.doi.org/10.21742/ijcs.2015.2.2.05

QoS based Workload Design Patterns in Cloud Computing: A Literature Review

Sukhpal Singh[1*] and Inderveer Chana[2]

[1,2]*Computer Science and Engineering Department, Thapar University, Patiala, Punjab, India-147004*
[1]*ssgill@thapar.edu,* [2]*inderveer@thapar.edu*

Abstract

Provisioning of appropriate resources to cloud workloads depends on the Quality of Service (QoS) requirements of cloud workloads and workload design patterns. Based on application requirements of cloud users, discovery and allocation of best workload – resource pair is an optimization problem. Acceptable QoS cannot be provided to the cloud users until provisioning of resources is offered as a crucial ability. In this paper, we have identified workload design patterns and mapped with the cloud workloads based on their QoS requirements for better provisioning of resources. Further, process of cloud workload design patterns has been discussed along with problems in patterns and their terminology in cloud. Based on existing design patterns, we have designed a workload design pattern for cloud workloads.

Keywords: *Cloud Computing, Cloud Workloads, Workload Patterns, Design Pattern, Quality of Service, Resource Management, Workload Design Pattern*

1. Introduction

Cloud computing offers pay per use based services such as Infrastructure as a Service, Platform as a Service and Software as a Service through different cloud providers [1]. Cloud provider provides the data and computing resources dynamically to the cloud users based on workload patterns. This is a challenging task to maintain the required Quality of Service (QoS) level of service to fulfill the expectations of cloud consumers by executing workloads [2, 3]. In our earlier work [3, 4, 11, 18], we have identified various research issues related to QoS and SLA for cloud resource scheduling and have developed a QoS based resource provisioning technique (Q-aware) to map the resources to the workloads based on used requirements described in the form of SLA. Further, resource scheduling framework (QRSF) has been proposed, in which provisioned resources have been scheduled by using different resource scheduling policies (cost, time, cost-time and bargaining based). The concept of QRSF has been further extended by proposing energy-aware autonomic resource scheduling technique (EARTH), in which IBM's autonomic computing concept has been used to schedule the resources automatically by optimizing energy consumption and resource utilization where user can easily interact with the system using available user interface. In this work, we have identified cloud workload design patterns and mapped with the cloud workloads based on their QoS requirements for better provisioning of resources.

To analyze cloud workload patterns, existing pattern based approaches have been discussed here. The simplest way to describe a pattern is that it delivers a proven solution to a common problem individually documented in a consistent format and commonly as part of a larger collection. Design patterns are patterns rotated around the design of automatic systems in the Information Technology (IT) world. Design patterns [4-7] are helpful because of the following reasons:

ISSN: 2205-8567 IJCS

a) Workload design patterns represent field-tested solutions to common design problems.
b) Workload design patterns organize design intelligence into a standardized and easily referenced format are generally repeatable by most IT professionals involved with design.
c) Workload design patterns can be used to ensure consistency in how systems are designed and built can become the basis for design standards.
d) Workload design patterns are usually flexible and optional (and openly document the impacts of their application and even suggest alternative approaches).
e) Workload design patterns can be used as educational aids by documenting specific aspects of system design (regardless of whether they are applied).
f) Workload design patterns can sometimes be applied prior and subsequent to the implementation of a system.
g) Workload design patterns can be supported via the application of other design patterns that are part of the same collection.
h) Workload design patterns enrich the vocabulary of a given IT field because each pattern is given a meaningful name.

1.1. Our Contributions

We have identified design patterns and mapped with the cloud workloads based on their QoS requirements for better provisioning of resources. Further, process of cloud workload design patterns has been discussed along with problems in patterns and their terminology in cloud. Based on existing cloud workload design patterns, we have designed a workload design pattern for cloud workloads. This research work is an extension of our previous research work [3, 4].

The motivation of this research work is to identify workloads design patterns and mapped with the cloud workloads based on their QoS requirements for better provisioning of resources. The organization of rest of this paper is as follows: Section 2 presents state of the art of workload design patterns. Section 3 describes the pattern identification process. Sec. 4 presents the problems in design patterns. Section 5 describes the pattern terminology. Sec. 6 presents the workload design pattern template. Section 7 describes the mapping of cloud workloads with design patterns. Section 8 presents the conclusion and future scope of this research work.

2. Workload Design Pattern: State-of-the-Art

The industry-driven evolution of cloud computing tends to complicate the common underlying architectural concepts of cloud offerings and their implications on hosted applications. Design patterns are used for documentation of architectural principles and to make good solutions to reoccurring (architectural) cloud challenges reusable [8]. To capture cloud computing best practice from existing cloud applications and provider-specific documentation, an elaborated pattern format is used to enabling abstraction of concepts and reusability of knowledge in various use cases. Elasticity empowers cloud users to reserve and release cloud resources dynamically and based on the currently experienced workload [9]. In industry, cloud providers already make use of pattern based descriptions and provide vendor-specific pattern formats and graphical notations. These patterns can be used to model cloud applications hosted on these providers' clouds. If a certain number of resources have been provisioned, a human system manager shall be notified via e-mail. The administrator has to approve further resource provisioning via a management user interface to control the costs generated by the application [10].

Design patterns will be used to tackle the complexity of the cloud environment, a well-established concept to describe pieces of knowledge to capture architectural styles [10]. The patterns are interrelated in a structure manner and ensure an abstraction of

implementation detail and use case specifics to describe reusable solutions to reoccurring problems. Any researchers already proposed several patterns capturing cloud architecture principles and management styles common for different cloud providers [11]. Using just one format enables the categorization of cloud providers regarding the patterns they support and sets the context for architectural patterns to be implemented by application developers. The use of a common pattern format, further, eases perception [12] and, thus, increases cloud technology adoption. This pattern template is used to capture the knowledge from various experts. Following are the benefits of workloads design patterns [7, 13, 14-16]:

a) Presentation of information in a pattern format enables application architects to quickly absorb a large body of knowledge and to understand the impact that cloud computing, associated technologies, and products have on application architectures. The patterns, thus, capture how architectural decisions impact the functional and non-functional requirements of cloud based applications.
b) Cloud patterns provide abstract knowledge about cloud products to application developers. A pattern set of distinct semantic of design pattern interrelations furthermore aids developers during the discovery and use of design patterns appropriate in their concrete use cases. Rapidly changing and evolving cloud application implementations and available cloud offerings may be easier coped with, because abstract template solutions, contained in the pattern descriptions, may be applied to different cloud technologies.
c) Abstract pattern-based descriptions of concepts can be used in education and ensure a longer persistence and relevance of obtained knowledge, because of their focus on abstracted information rather than on concrete implementations and offerings.

Pattern-based descriptions have been used frequently to describe good solutions to reoccurring problems during the architectural design of applications. G. Hohpe *et al.*, [7] described patterns on how IT reconsolidation may reduce the environmental impact of the IT architecture by restructuring application architectures and supported processes. However, to be applicable in the context of cloud computing each pattern would have to be evaluated and possibly needs to be revised. In the field of cloud computing, patterns are gaining momentum. In industry, they are often used in a less formal way to present architectural guidelines.

3. Workload Design Pattern Identification Process

From the literature [17-19], the identification process of design patterns is shown in Figure 1. The components of the workload design pattern identification process are described below:

3.1. Customization of the Pattern Format

As shown in Figure 1, this format allows the omission of optional sections, which are then summarized with mandatory sections as indicated in the word template. Each pattern is identified by a unique name and has an icon associated with it. Then, a driving question states the problem solved by the pattern to enable readers to quickly check if the pattern fits to problems at hand. The expression of cloud types and cloud offerings in the pattern format has proven to be very valuable.

3.2. Collection of Information Sources

In this step, different information sources are collected. An excel spread sheet contained in the pattern will be used. For each information source, general information, such as document names, hyperlinks etc. are collected. Further, the

cloud computing relevant information obtained from the sources is summarized. This summarization should be expressed respecting the context of the information source and its domain specific terms. Examples for information sources are provider guidelines, knowledge about existing applications, books, and journals.

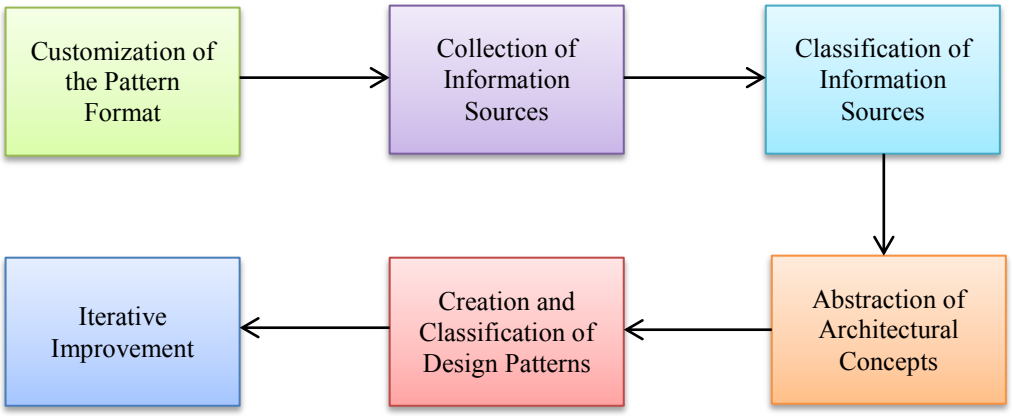

Figure 1. Pattern Identification Process

3.3. Classification of Information Sources

The categorization of architecture domain is done from brief knowledge mined from the different information sources. During the classification process the summaries created for information sources may be split up into multiple entries to allow a fine granular classification.

3.4. Abstraction of Architectural Concepts

Till now, the brief knowledge mined from the several information sources remains related to cloud provider. It, consequently, desires to be distracted to essential architectural concepts. During this step, summaries may again be split up. This allows a grouping of summarized knowledge regarding the abstracted concepts.

3.5. Creation and Classification of Design Patterns

Based on the abstract concepts, design patterns are now compiled from the information sources. Meanwhile the design patterns are not designed, then known and exposed from present solutions, confirming an appropriate level of abstraction is always a challenge.

3.6. Iterative Improvement

Research is shown that patterns evolve continuously through discussion in a community. This community should be heterogeneous, thus, should contain experts and beginners alike to ensure that obtained patterns contain all necessary information while remaining understandable. As per design patterns are generated by several different contributors, an arrangement of graphical components can be pursued during the revision for a more a consistent look-and feel. Therefore, during iterative improvement, graphical designers are involved to obtain easy to understand graphics.

4. Problems in Workload Design Patterns

The classification of problems [20-23] that will be discussed through design pattern into the following domains is based on this information:

4.1. Accounting & Controlling

The use of pay-per-use billing is inherent to cloud computing. Additionally, providers may offer long term reservation of resources at a lower price, such as Amazon Reserved Instances. These are intended for the static workload of customers. Variable pricing depending on the overall utilization of a cloud are also available, such as Amazon Spot Instances. These billing models should be reflected in the application architecture to reduce runtime costs.

4.2. Application Migration

Many existing applications could benefit from cloud computing. To move applications to the Cloud, the applications' infrastructure requirements and how they are respected by the application architecture need to be reconsidered.

4.3. Cloud Integration

When using cloud computing, a company often faces the challenge that different computing environments have to be integrated. For example, legacy applications may reside in on-premise data centers, whereas others use a public cloud.

4.4. Data Storage

The inconsistency of data replicas is well-researched in distributed systems, cloud computing has shown how this concept may be employed to increase the availability and performance of an application to handle large amounts of data and users efficiently.

4.5. License Management

Licenses issued per CPU executing software are hard to measure or enforce in the cloud where hardware virtualization is inherent. This renders the use of many software products employing this licensing complicated to impossible in the cloud. Since virtual servers may be started and stopped dynamically in the cloud, best practices for license management are required to ensure a compliant deployment of cloud resources, for example, during automated scaling processes.

4.6. Monitoring, Analysis, and Reporting

Cloud computing is often considered for the aspect of profitable. The presented resource sharing, yet, can also be exploited for energy savings. In either case, information about cloud resources has to be collected and analyzed. Sharing resources between different projects and products significantly complicates such computations.

4.7. Multi-Tenant Cloud Middleware Sharing

Cloud resources between multiple cloud users, also called tenants, are a very important concept to benefit from economies of scale. Patterns are, therefore, required in this domain to address challenges, such as tenant-isolation, version management, customer-specific application customization, migration of tenants between software versions and runtime environments etc.

4.8. Security and Compliance

Many of the security issues found in non-cloud applications are also arising in a cloud environment. These concepts have been captured in a pattern format and may be applicable in cloud computing as well. New security threats arise from malicious use of dynamic cloud resources or from other cloud users. Regarding compliance, cloud computing introduces the challenge that a company's IT has to compete with public cloud services. Methods how to address these issues have been investigated but there are currently no practical solutions to extract patterns from.

5. Workload Design Pattern Terminology

From the literature [20] [22] [24], the pattern terminology of design patterns is given below:

5.1. Patterns for Feedback

The feedback of readers new to the cloud domain was especially useful during step 6 (Section 3.6) of the pattern identification process (iterative improvement), because it is difficult for experts to find an adequate level of abstractions during the initial identification and description of patterns.

5.2. Transparency of Pattern Identification

The presented pattern identification process enables transparency of this task and allows other to retrace the steps undertaken to identify a pattern.

5.3. Categorization of Cloud Providers based on Design Patterns

Cloud design patterns enabled a classification of providers regarding the patterns supported by them.

5.4. Patterns for Documentation

Generally, patterns can be used in software documentation to reduce its size and increase its quality. Use of patterns in documentation increased readability, because the well-known concept of a pattern could be perceived easier and quicker while reducing the amount of introductory text required in documents.

5.5. Pattern Vocabulary

The standard vocabulary is very useful to bridge different product user communities.

6. Workload Design Pattern Template

The workload pattern template consists of different parts related to all aspects as shown in Figure 2. The units of workload pattern are described below:

6.1. Design Pattern Name

The name of the workload design pattern is to be considered.

6.2. Requirement

This is a brief statement that offers the important constraint addressed by the workload pattern in the form of a query.

6.3. Summary

The summary table is consisting of statements that jointly offer a brief outline of the workload pattern for rapid reference purposes. Moreover, the summary table offers references to interrelated service-orientation design philosophies and service-oriented architectural categories.

Design Pattern Name
Requirement
Summary
Problem:
Solution:
Application:
Impact:

Figure 2. Workload Design Pattern Template

6.4. Problem

It is the problem for which the workload pattern offers a solution. Problem explanations may also contain common situations that can lead to the problem.

6.5. Solution

This signifies the design solution offered by the workload pattern to solve the problem and fulfill the need.

6.6. Application

This section is devoted to defining how the workload pattern can be used. It can consist of strategies, implementation details, and occasionally even a suggested process.

6.7. Impact

This section describes significances, budgets and requirements related with the application of a workload pattern.

7. Mapping of Cloud Workloads with Design Patterns

Based on the key characteristics of cloud workloads, the workload patterns are mapped with cloud workloads. How the cloud workloads have been distributed among four clusters and their corresponding pattern classification, framework and their description is done in our previous work [3, 4]. Table 1 shows how the cloud workloads have been distributed among four clusters and their corresponding pattern classification, framework and their description [1-4], [7-10], [22-24].

8. Conclusions and Future Directions

In this paper, cloud workload design patterns have been identified and based on their QoS requirements and cloud workload design patterns are mapped with the cloud workloads for better provisioning of resources. We have identified the patterns problems in cloud computing and process of cloud workload design patterns has been discussed. Workload designed pattern template has been designed for cloud workloads. Further, workload patterns can be used for effective resource provisioning and scheduling.

Table 1. Mapping of Design Patterns and Cloud workloads

Cluster	Cloud Workload	Pattern Classifications	Framework	Description
Compute	Performance Testing	On-demand Application Instance	Applications that need scale-out and scale-down capabilities.	Let retail store sites be available during special events.
	Technological Computing	Operative	Executing parallel batch jobs or background applications.	Using schedulers for analytics processing by executing background tasks in parallel.
Storage	E-Com	Simple Storage	Storing large amount of unstructured data.	Company storing legal compliance reports in backup store.
	Storage and Backup	Structured Storage	Storing data in a table structure while not demanding full relational semantics.	A structured storage to maintain a Web application's state (shopping basket information)
Communication	Website	Service Interface (Web and Web Service API)	Exposing application capabilities through UI and Web services.	Company building digital asset management solution exposing APIs to other services.
	Critical Internet Applications	Service-oriented Integration	Invoking external Web services using Web-standard protocols.	Web applications leveraging Web-hosted live meeting services for collaboration.
	Mobile Computing Services	Messaging	Share messages between applications in a scalable, reliable, and asynchronous way.	Web application informing a scheduler to execute a specific task.
Administration	Endeavour Software and Online financial Services	Cloud Deployment	Deploying applications with desired configurations such as scale-out and high-availability requirements.	A retail store configuring Web portal to automatically scale-out when usage exceeds threshold and scale-down as needed.
	Software/Project Development and Testing	Design for Operations	How to make my application operations-ready by providing health status and logging.	Developers designing Cloud applications to be operations friendly through GUI.
	Central Financial services	Service Instance Management	Start, stop, and suspend Cloud apps. Manage service configurations.	A Web application administrator using the service portal to manage application state.
	Productivity Applications	Management Alerts	Sending Immediate Messages, mails, or warnings about resource and billing detail.	Enabling applications to send emails. Default notification on resource usage.
	Graphic Oriented	Service Level Management	Get info on app resource consumption such as processor time, bandwidth.	Looking for billing and resource usage info about the application deployed with billing transparency.

Acknowledgments

One of the authors, Sukhpal Singh, acknowledges the Department of Science and Technology (DST), Government of India, for awarding him the INSPIRE (Innovation in Science Pursuit for Inspired Research) Fellowship (Registration/IVR Number: 201400000761 [DST/INSPIRE/03/2014/000359]) to carry out this research work. We

would like to thank all the anonymous reviewers for their valuable comments and suggestions for improving the paper. We would also like to thank Dr. Maninder Singh [EC-Council's Certified Ethical Hacker (C-EH)] for his valuable suggestions.

References

[1] S. Singh and I. Chana, "Cloud Based Development Issues: A Methodical Analysis", International Journal of Cloud Computing and Services Science (IJ-CLOSER), vol. 2, no. 1, **(2012)** pp. 73-84.
[2] J. Vlissides, R. Helm, R. Johnson and E. Gamma, "Design Patterns: Elements of Reusable Object-oriented Software", Addison-Wesley, **(1995)**.
[3] S. Singh and I. Chana, "Q-aware: Quality of Service based Cloud Resource Provisioning", Computers & Electrical Engineering - Journal – Elsevier, **(2015).** DOI: http://dx.doi.org/10.1016/j.compeleceng.2015.02.003
[4] S. Singh and I. Chana, "QRSF: QoS-aware resource scheduling framework in cloud computing", The Journal of Supercomputing, Vol. 71, no. 1, **(2015)** pp. 241-292.
[5] C. Fehling, F. Leymann, R. Retter, D. Schumm and W. Schupeck, "An Architectural Pattern Language of Cloud-based Applications", Proceedings of the Conference on Pattern Languages of Programs (PLoP), **(2011)**.
[6] C. Fehling, F. Leymann, R. Mietzner and W. Schupeck, "A Collection of Patterns for Cloud Types, Cloud Service Models, and Cloud-based Application Architectures", Report No. 2011/05, University of Stuttgart, **(2011)**.
[7] G. Hohpe and B. Wolf, "Enterprise Integration Patterns: Designing, Building, and Deploying", Addison-Wesley, **(2004)**.
[8] K. Hashizume, N. Yoshioka and E. B. Fernandez, "Misuse Patterns for Cloud Computing", Proceedings of the Asian Conference on Pattern Languages of Programs (AsianPLoP), **(2011)**.
[9] S. Singh and I. Chana, "Consistency verification and quality assurance (CVQA) traceability framework for SaaS", Proceeding of the IEEE 3rd International on Advance Computing Conference (IACC). **(2013)**, pp. 1-6.
[10] S. S. Gill, "Autonomic Cloud Computing: Research Perspective", **(2015)**, pp. 1-3. Retrieved from http://arxiv.org/ftp/arxiv/papers/1507/1507.01546.pdf
[11] S. Singh and I. Chana, "EARTH: Energy-aware Autonomic Resource Scheduling in Cloud Computing" Journal of Intelligent and Fuzzy Systems, **(2015)**, pp. 1-16.
[12] S. Singh and I. Chana, "Introducing Agility in Cloud Based Software Development through ASD", International Journal of u-and e-Service, Science and Technology, vol. 6, no. 5 **(2013)**, pp. 191-202.
[13] S. Singh and I. Chana, "Advance billing and metering architecture for infrastructure as a service", International Journal of Cloud Computing and Services Science (IJ-CLOSER) vol. 2, no. 2 **(2013)**, pp. 123-133.
[14] S. Singh and I. Chana, "QoS-aware Autonomic Cloud Computing for ICT", Proceeding of the International Conference on Information and Communication Technology for Sustainable Development (ICT4SD - 2015), Springer International Publishing, **(2015)**.
[15] S. Singh and I. Chana, "QoS-aware Autonomic Resource Management in Cloud Computing: A Systematic Review", ACM Comput. Surv., vol. 48, no. 3, **(2015)**, pp. 1-39.
[16] S. Singh and I. Chana, "Energy based efficient resource scheduling: a step towards green computing", Int J Energy Inf Commun, vol 5, no. 2, **(2014)**, pp. 35-52.
[17] S. Singh and I. Chana, "Formal Specification Language Based IaaS Cloud Workload Regression Analysis", arXiv preprint arXiv:1402.3034 **(2014)**. Retrieved from http://arxiv.org/ftp/arxiv/papers/1402/1402.3034.pdf.
[18] S. Singh and I. Chana, "Cloud Resource Provisioning: Survey, Status and Future Research Directions", Knowledge and Information Systems, **(2015)**.
[19] S. Singh and I. Chana, "A Survey on Resource Scheduling in Cloud Computing Issues and Challenges", Journal of Grid Computing, **(2015)**.
[20] D. Sarddar, E. Nandi, R. K. Gupta, R. K. Pateriya, A. Abdulmohson, S. Pelluri, and R. Sirandas, "Implement of Dynamic Time Quantum Shortest Load First Scheduling for Efficient Load Balancing" International Journal of Cloud-Computing and Super-Computing, vol. 2, no. 1 **(2015)**, pp. 1-8.
[21] R. K. Gupta and R. K. Pateriya, "Energy Efficient Virtual Machine Placement Approach for Balanced Resource Utilization in Cloud Environment", International Journal of Cloud-Computing and Super-Computing, vol. 2, no. 1 **(2015)**, pp. 9-20.
[22] A. Abdulmohson, S. Pelluri, and R. Sirandas, "Energy Efficient Load Balancing of Virtual Machines in Cloud Environments", International Journal of Cloud-Computing and Super-Computing, vol. 2, no. 1 **(2015)**, pp. 21-34.
[23] V. S. Rathor, R. K. Pateriya, R. K. Gupta, M. Shelar, S. Sane, V. Kharat, and R. Jadhav, "An Efficient Virtual Machine Scheduling Technique in Cloud Computing Environment", International Journal of Cloud-Computing and Super-Computing, vol. 1, no. 1 **(2015)**, pp. 1-14.

[24] M. Shelar, S. Sane, V. Kharat, and R. Jadhav, "Efficient Virtual Machine Placement with Energy Saving in Cloud Data Center", International Journal of Cloud-Computing and Super-Computing, vol. 1, no. 1 **(2015)**, pp. 15-26.

Authors

Sukhpal Singh obtained the Degree of Master of Engineering in Software Engineering from Thapar University, Patiala. Mr. Singh received the Gold Medal in Master of Engineering in Software Engineering. Presently he is pursuing Doctoral degree in Cloud Computing from Thapar University, Patiala. Mr. Singh is on the Roll-of-honor being DST Inspire Fellow as a SRF Professional. He has done certifications in Cloud Computing Fundamentals, including Introduction to Cloud Computing and Aneka Platform (US Patented) by ManjraSoft Pty Ltd, Australia and Certification of Rational Software Architect (RSA) by IBM India. His research interests include Software Engineering, Cloud Computing, Operating System and Databases. He has more than 20 research publications in reputed journals and conferences.

Inderveer Chana joined Computer Science and Engineering Department of Thapar University, Patiala, India, in 1997 as Lecturer and is presently serving as Associate Professor in the department since 2011. She is Ph.D. in Computer Science with specialization in Grid Computing and M.E. in Software Engineering from Thapar University and B.E. in Computer Science and Engineering. Her research interests include Grid and Cloud computing and other areas of interest are Software Engineering and Software Project Management. She has more than 100 research publications in reputed Journals and Conferences. Under her supervision, more than 30 ME thesis and four Ph.D thesis have been awarded and four Ph.D. thesis are on-going. She is also working on various research projects funded by Government of India.

