SELF-PREDICTION OF PERFORMANCE METRICS FOR DBMS WORKLOAD

Basit Raza
42-FBAS/PHDCS/F08

Submitted in partial fulfillment of the requirements for the PhD degree in Computer Science at the faculty of Basic and Applied Sciences International Islamic University, Islamabad, Pakistan.

Dr. Muhammad Sher (IIU)
Dr. Mian Muhammad Awais (LUMS)

February, 2014
Title of Thesis: Self-Prediction of Performance Metrics for DBMS Workload
Name of Student: Basit Raza
Registration No. 42-FBAS/PhDCS/F08

Accepted by the Department of Computer Science and Software Engineering, INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science.

Viva Voce Committee

Supervisor:
Professor Dr. Muhammad Sher
Dean FBAS, HOD CS& SE
International Islamic University (IIU)
Islamabad, Pakistan.

Co-Supervisor:
Dr. Mian Muhammad Awais
Associate Professor
Lahore University of Management Sciences (LUMS)
Lahore, Pakistan.

External Examiner – I:
Professor Dr. Khalid Rashid
Senior Advisor, COMSATS Institute of Information Technology (CIIT)
Islamabad, Pakistan.

External Examiner – II:
Professor Dr. Muhammad Abdul Qadir
Dean Faculty of Computing
Muhammad Ali Jinnah University (MAJU) Islamabad, Pakistan.

Internal Examiner:
Dr. Ali Daud
Assistant Professor
International Islamic University (IIU)
Islamabad, Pakistan.

Dean FBAS & HOD CS& SE:
Professor Dr. Muhammad Sher
International Islamic University (IIU)
Islamabad, Pakistan.

February 21, 2014
This dissertation is submitted to International Islamic University, Islamabad, Pakistan in partial fulfillment of the requirement of the degree of Doctor of Philosophy (Computer Science).
DECLARATION

It is certified that work done on this PhD Computer Science research thesis is purely conducted by us. All the material is prepared by us and has not been copied from anywhere; however some text and figures have been used which are properly referenced.
DEDICATIONS

This work is dedicated to my sweet and beloved parents and my family whose constant support and guidance enabled me to achieve this milestone.

Basit Raza
ACKNOWLEDGEMENTS

First of all I pay my submissive gratitude in the domain of Almighty Allah who is the supreme authority over the universe and nothing goes unaccounted under his domain. I am thankful to Almighty Allah that has bestowed me the strength and qualities due to which we have been able to complete my research.

I express my profound gratitude to my respectable and beloved parents and my family for their sincere care, support and prayers that encouraged me a lot, in getting through the research work.

Especially I am grateful to my supervisor Professor Dr. Muhammad Sher (IIU) for his ideas, valuable suggestions. He has always guided whenever I stuck and encouraged me during the whole period. I am also thankful to my co-supervisor Dr. Mian Muhammad Awais, Associate Professor (LUMS) for his suggestions and good ideas that help us to move in right direction.

Special thanks to my foreign supervisor Professor Dr. Naomie Salim, Universiti Teknologi Malaysia (UTM) for her valuable suggestion and ideas to complete my PhD research. I would like to thank to Soft Computing Research Group (SCRG) of UTM. I would like to appreciate many people who have supported, guided and advised me in my research at SCRG lab during my stay. Many thanks to all fellows namely Yogan Kumar, Waleed, Oye David and Adil Yousif for their help and support.
ABSTRACT

Database Management System (DBMS) workload involves homogenous as well as heterogeneous data and concurrent users. Humans are incapable to manage the versatile data and dynamic behavior of DBMS workload. There is a need of fast computations of current server’s loads and requirements, AI algorithms and machine learning techniques. Autonomic computing technology using types of workload Decision Support System (DSS) or Online Transaction Processing (OLTP) and its performance requirement can help servers, adaptation of the workloads. If we know the type of workload, we can design such systems that predict the identified workload performance and adapt the changes in the behavior of the workload. For managing the workload, we have to face number of problems for the DBMS to better perform. Before executing, we can predict and control the workload to tune the DBMS. Predicting performance of the workload is important for tuning a DBMS and makes the DBMS aware of itself making it autonomic. The optimizer and DBMS can tune itself accordingly. Evolving behavior of workload can be handled by making the system adaptive.

We have developed a framework called Autonomic Workload Performance Predictor (AWPP) for predicting the performance of the workload making it adaptive to the changing behavior of the workload. Case-based reasoning approach is applied and results are compared with other well-known machine learning techniques to observe the accuracy and effectiveness and significance of AWPP framework. MySQL database management system is being used to execute different benchmark workload to validate the proposed workload performance prediction framework. For training and testing TPC-H and TPC-C like queries are used as our representative workload. We have taken the various benchmark workloads of DSS and OLTP for experimentation. CBR approach produced effective, accurate and significant results while predicting the performance of workload using the information available before executing a workload and adapting the workload on evolution. These predictions will be helpful for optimizer and DBMSs algorithms as well as for workload management, capacity planning, system sizing.
# TABLE OF CONTENTS

List of Figures .............................................................................................................. x
List of Tables ............................................................................................................... xii
List of Abbreviation .................................................................................................... xiii

## CHAPTER 1 Introduction
1.1 Background and Motivation .................................................................................. 2
1.2 Knowledge Gap ..................................................................................................... 5
1.3 Research Objectives ............................................................................................. 6
1.4 Organization of Thesis .......................................................................................... 13

## CHAPTER 2 Literature Review Autonomic Workload Management
2.1 Autonomic Workload Management in DBMSs .................................................. 14
2.2 Taxonomy of DBMS Workload management ...................................................... 17
2.3 Workload Monitoring .......................................................................................... 19
   2.3.1 Self-Inspection ........................................................................................... 19
2.4 Workload Performance Tuning .......................................................................... 27
   2.4.1 Self-Configuration ..................................................................................... 27
   2.4.2 Self-Optimization ...................................................................................... 29
   2.4.3 Self-Prediction .......................................................................................... 31
2.5 Workload Adaptation ......................................................................................... 37
   2.5.1 Self-Adaptation ......................................................................................... 37
2.6 Problem Statement .............................................................................................. 48
2.7 Summary .............................................................................................................. 50
# Table of Contents

## CHAPTER 3  Research Methodology

3.1  Research Operational Framework .................................................. 51  
3.2  Research Design and Development .............................................. 53  
   3.2.1  Analysis and selection of input features of DBMS workload .......... 53  
   3.2.2  Analysis and selection of performance features of DBMS workload ...... 53  
3.3  Conceptual Framework of AWPP .................................................. 54  
3.4  Proposed Solution ................................................................. 54  
3.5  Proposed Framework AWPP ...................................................... 55  
3.6  Research Approach ............................................................... 59  
3.7  Machine Learning and Lazy Learning Techniques ............................ 63  
3.8  Implementation and Integration .................................................. 63  
3.9  Experimental Setup ............................................................... 64  
3.10  Performance Evaluation of the proposed framework ....................... 66  
3.11  Validation of Results ............................................................. 67  
3.12  Summary ............................................................................. 68  

## CHAPTER 4  Autonomic Workload Performance Prediction Framework

4.1  Workload Performance Prediction .................................................. 69  
4.2  Workload Performance Feature Selection for Prediction .................... 70  
4.3  Workload Performance Predictor Module ..................................... 72  
4.4  Machine Learning and Lazy Learning Techniques ............................ 75  
   4.4.1  Workload Performance Prediction using SVM ......................... 75  
   4.4.2  Workload Performance Prediction using CBR .......................... 77  
4.5  Experiments and Results .......................................................... 87  
4.6  Comparison of CBR with Other Machine Learning Approaches .......... 91  
4.7  Validation of Results; T-Test ..................................................... 96  
4.8  Summary ............................................................................. 97
# Table of Contents

## CHAPTER 5  Enhanced Workload Adaptive Framework

- 5.1 Workload Adaptation ................................................................. 98
- 5.2 Enhanced Workload Adaptation Architecture ............................ 100
- 5.3 Machine Learning and Lazy Learning Techniques ................... 101
- 5.4 Results and Discussion ......................................................... 103
- 5.5 Validation of Results; Non-parametric procedures with post-hoc tests ........................................... 105
- 5.6 Summary .............................................................................. 110

## CHAPTER 6  Conclusion and Future Work

- 6.1 Summary of Research .......................................................... 111
- 6.2 Research Findings ................................................................. 113
  - 6.2.1 Design and Development consideration .............................. 114
  - 6.2.2 Performance Evaluation of the proposed framework ........ 116
- 6.3 Conclusion ........................................................................... 116
- 6.4 Limitation of the Study .......................................................... 119
- 6.5 Future Work ......................................................................... 120
- 6.6 Closing Note ......................................................................... 122

## REFERENCES ................................................................................. 123

## APPENDICES ................................................................................. i-xiii
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE NO</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The strategy canvas</td>
<td>9</td>
</tr>
<tr>
<td>1.2</td>
<td>Summary of Research Contributions</td>
<td>12</td>
</tr>
<tr>
<td>2.1</td>
<td>Taxonomy of Autonomic Workload Management</td>
<td>18</td>
</tr>
<tr>
<td>2.2</td>
<td>Fishbone diagram of workload performance prediction problem</td>
<td>49</td>
</tr>
<tr>
<td>2.3</td>
<td>Fishbone diagram of workload adaptation of new trends problem</td>
<td>49</td>
</tr>
<tr>
<td>3.1</td>
<td>Research Operational Framework</td>
<td>52</td>
</tr>
<tr>
<td>3.2</td>
<td>Conceptual Framework of AWPP</td>
<td>54</td>
</tr>
<tr>
<td>3.3</td>
<td>Autonomic Workload Performance Predictor (AWPP) Framework-Block Diagram</td>
<td>56</td>
</tr>
<tr>
<td>3.4</td>
<td>Query Processing Steps</td>
<td>59</td>
</tr>
<tr>
<td>4.1</td>
<td>Workload Predictor Module</td>
<td>74</td>
</tr>
<tr>
<td>4.2</td>
<td>SVM Classification Model</td>
<td>76</td>
</tr>
<tr>
<td>4.3</td>
<td>Case I Representation for Prediction of performance features in CBR</td>
<td>78</td>
</tr>
<tr>
<td>4.4</td>
<td>CBR Taxonomy</td>
<td>81</td>
</tr>
<tr>
<td>4.5</td>
<td>CBR Classification Model</td>
<td>82</td>
</tr>
<tr>
<td>4.6</td>
<td>The pseudo-code of Performance Prediction algorithm</td>
<td>84</td>
</tr>
<tr>
<td>4.7</td>
<td>Precision, Recall and F-measure using SVM</td>
<td>88</td>
</tr>
<tr>
<td>4.8</td>
<td>Precision, Recall and F-measure using CBR</td>
<td>89</td>
</tr>
<tr>
<td>4.9</td>
<td>Comparison of F-measure between SVM and CBR</td>
<td>90</td>
</tr>
<tr>
<td>4.10</td>
<td>Comparison of Accuracy between SVM and CBR</td>
<td>91</td>
</tr>
<tr>
<td>5.1</td>
<td>Workload Adapter Module</td>
<td>100</td>
</tr>
<tr>
<td>5.2</td>
<td>The pseudo-code of Adaptation algorithm in CBR</td>
<td>102</td>
</tr>
<tr>
<td>5.3</td>
<td>F-measure before and after Adaptation using SVM and CBR</td>
<td>103</td>
</tr>
</tbody>
</table>
List of Figures

5.4 Accuracy before and after Adaptation via SVM and CBR……………. 104
6.1 The sequence of design and development phases leading to the proposed framework……………………………………………….. 113
6.2 Autonomic Workload Performance Predictor (AWPP)…………….. 115
## LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE NO</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Summary for classification techniques and their limitations............</td>
<td>25</td>
</tr>
<tr>
<td>2.2</td>
<td>Summary for workload performance prediction techniques and predicted attributes</td>
<td>36</td>
</tr>
<tr>
<td>2.3</td>
<td>Summary for workload adaptation techniques and their limitations</td>
<td>44</td>
</tr>
<tr>
<td>3.1</td>
<td>Research Framework</td>
<td>58</td>
</tr>
<tr>
<td>3.2</td>
<td>Actual vs. Predicted class</td>
<td>67</td>
</tr>
<tr>
<td>4.1</td>
<td>Feature Selection for Performance Prediction</td>
<td>71</td>
</tr>
<tr>
<td>4.2</td>
<td>An example case representing the new case</td>
<td>85</td>
</tr>
<tr>
<td>4.3</td>
<td>Similarity Measure Formulae</td>
<td>86</td>
</tr>
<tr>
<td>4.4</td>
<td>Experimental setting using SVM</td>
<td>87</td>
</tr>
<tr>
<td>4.5</td>
<td>Comparison of Precision, Recall, F-measure &amp; Accuracy for Key writes using different algorithms</td>
<td>93</td>
</tr>
<tr>
<td>4.6</td>
<td>Comparison of Precision, Recall, F-measure &amp; Accuracy for Workload size using different algorithms</td>
<td>93</td>
</tr>
<tr>
<td>4.7</td>
<td>Comparison of Precision, Recall, F-measure &amp; Accuracy for Execution time using different algorithms</td>
<td>93</td>
</tr>
<tr>
<td>4.8</td>
<td>Comparison of Precision, Recall, F-measure &amp; Accuracy for Key writes using different similarity measures</td>
<td>95</td>
</tr>
<tr>
<td>4.9</td>
<td>Comparison of Precision, Recall, F-measure &amp; Accuracy for Workload size using different similarity measures</td>
<td>95</td>
</tr>
<tr>
<td>4.10</td>
<td>Comparison of Precision, Recall, F-measure &amp; Accuracy for Execution time using different similarity measures</td>
<td>95</td>
</tr>
<tr>
<td>4.11</td>
<td>Paired Samples T-Test for performance prediction</td>
<td>96</td>
</tr>
<tr>
<td>5.1</td>
<td>Workload Performance 11 Features and 6 Classifiers</td>
<td>106</td>
</tr>
<tr>
<td>5.2</td>
<td>Algorithm and Ranking by Friedman procedure</td>
<td>106</td>
</tr>
<tr>
<td>5.3</td>
<td>The P-values for alpha = 0.05 for Holm and Shaffer</td>
<td>108</td>
</tr>
<tr>
<td>5.4</td>
<td>APVs obtained in example by Holm, Nemenyi, Shaffer &amp; Bergman...</td>
<td>109</td>
</tr>
</tbody>
</table>
# LIST OF ABBREVIATION

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWPP</td>
<td>Autonomic Workload Performance Predictor</td>
</tr>
<tr>
<td>AC</td>
<td>Autonomic Computing</td>
</tr>
<tr>
<td>ADBMS</td>
<td>Autonomic Database Management System</td>
</tr>
<tr>
<td>ASM</td>
<td>Active System Management</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>APV</td>
<td>Adjusted p-values</td>
</tr>
<tr>
<td>BI</td>
<td>Business Intelligence</td>
</tr>
<tr>
<td>CBMGS</td>
<td>Customer Behavior Model Graph</td>
</tr>
<tr>
<td>CCA</td>
<td>Canonical Correlation Analysis</td>
</tr>
<tr>
<td>CBR</td>
<td>Case-Based Reasoning</td>
</tr>
<tr>
<td>CRT</td>
<td>Classification and Regression Tree</td>
</tr>
<tr>
<td>DSS</td>
<td>Decision Support System</td>
</tr>
<tr>
<td>DBMS</td>
<td>Database Management System</td>
</tr>
<tr>
<td>DW</td>
<td>Data Warehouse</td>
</tr>
<tr>
<td>DBA</td>
<td>Database Administrator</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>DW</td>
<td>Data Warehouse</td>
</tr>
<tr>
<td>EQMS</td>
<td>External Queue Management System</td>
</tr>
<tr>
<td>fp</td>
<td>false positive</td>
</tr>
<tr>
<td>fn</td>
<td>false negative</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machine</td>
</tr>
<tr>
<td>KNN</td>
<td>K-Nearest Neighbor</td>
</tr>
<tr>
<td>KCCA</td>
<td>Kernel Canonical Correlation Analysis</td>
</tr>
<tr>
<td>MPL</td>
<td>Multiprogramming Level</td>
</tr>
<tr>
<td>NNN</td>
<td>Natural Neural Network</td>
</tr>
</tbody>
</table>
### List of Abbreviation

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLTP</td>
<td>On Line Transaction Processing</td>
</tr>
<tr>
<td>OLTP</td>
<td>Online Transaction Processing</td>
</tr>
<tr>
<td>PAQRS</td>
<td>Priority Adaptation Query Resource Scheduling</td>
</tr>
<tr>
<td>PQR</td>
<td>Predictions of Query Runtime</td>
</tr>
<tr>
<td>PI</td>
<td>Progress Indicator</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>PFV</td>
<td>Performance Features Vector</td>
</tr>
<tr>
<td>PMM</td>
<td>Priority Memory Management</td>
</tr>
<tr>
<td>PSP</td>
<td>Psychic Skeptic Prediction framework</td>
</tr>
<tr>
<td>QEP</td>
<td>Query Execution Plan</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>QP</td>
<td>Query Patroller</td>
</tr>
<tr>
<td>REDWAR</td>
<td>RELational Database Workload Analyzer</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SPSS</td>
<td>Statistical Package for Social Sciences</td>
</tr>
<tr>
<td>SLO</td>
<td>Service Level Objective</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>SAN</td>
<td>Storage Area Networks</td>
</tr>
<tr>
<td>TPC</td>
<td>Transaction Processing Council</td>
</tr>
<tr>
<td>TCO</td>
<td>Total cost of Ownership</td>
</tr>
<tr>
<td>TD</td>
<td>Training Data</td>
</tr>
<tr>
<td>tp</td>
<td>true positive</td>
</tr>
<tr>
<td>tn</td>
<td>true negative</td>
</tr>
<tr>
<td>WFV</td>
<td>Workload Features Vector</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Computers are evolving from an unreliable, static unit to reliable and dynamic unit. This dynamic behavior is going to evolve day by day. Computer usage is increasing and in the same ratio complexity is also increasing. Due to this complexity, dynamism and huge data, manageability is becoming difficult. In this regard, one of the main hurdles for the future of information technology is the manageability issue. This issue motivates towards intelligent or autonomic computing that will help IT professional to focus on high value tasks rather than performing hindrance operational tasks. These types of tasks will be performed through intelligent or autonomic way having low cost, reduction of labor and increased efficiency.

Database vendors such as IBM DB2, Oracle and SQL Server and the research community have been working on self-managing and autonomic database management systems since 2001 when IBM introduced the term Autonomic Computing (AC) for the first time. The systems or software which can manage them and have the ability to change them according to intended situation with minimum or no human interaction are called autonomic systems or software. AC is a self-managing computing model like
human body's autonomic nervous system. The basic purpose of the AC is to create such systems which have the capability to run themselves hiding the complexity from the user. It is an evolutionary process rather than revolutionary process because it starts from basic level and continue through managed, predictive, adaptive and finally the autonomic level (Parashar and Salim 2005). It is eminent to incorporate it in different areas of computer science and is helpful in workload management in DBMSs.

Prediction of workload plays an important role. It is the knowledge that system has before executing a query or workload that how it will behave after its execution. Whenever a workload is executed systems gather information about the workload and its internal behavior but they can not do anything about controlling the workload. Prediction can help in this regard. Hence system would have the knowledge about the performance of the workload all the time. Important is to forecast the performance of the workload that will help in decision making, system sizing and capacity planning. Behaviour of the workload may change any time. The evolving behavior of workload is adapted in to the system to adjust itself according to the newly arriving circumstances or changes.

In this research work our first focus is to highlight the importance of prediction with respect to workload management such as how do systems can manage the workload by deciding about the workload. How workload and changing behavior of the workload can be monitored and what will be performance of workload prior its execution.

1.1 Background and Motivation

The research in the context of workload management has been carried out for the last many years and different database vendors have developed and incorporated number of tools and techniques for this purpose. These tools and techniques are related with
workload classification, performance prediction, adaptation and configuration of various parameters (Raza et al. 2011). Data volume and complexity is increasing day by day and due to huge and complex data, human beings are incapable of managing data accurately and efficiently. This motivates researchers and database vendors to work on self-managing or autonomic workload models and frameworks, who have the ability to recognize, predict and adapt according to the changes autonomically. The autonomic technology has a high potential to be incorporated in current DBMSs. Tuning the workload can greatly improve the performance of the system. Knowing information about the workload type and its performance before executing it can help in predicting the behavior of the workload, its management and other decision making.

Autonomic workload performance prediction is expected to manage the workload in an efficient and responsive way. These predictions will be helpful for resource allocation, system sizing and capacity planning and also with these prediction DBMS algorithms can optimize itself accordingly. The incorporation of AC characteristics in the systems can greatly enhance the performance of the system. There is a need of such prediction systems that will collect necessary information about workload (type, performance & others) itself and manage workload with minimal human intervention. The workload management have several issues that includes identification and decision about problematic queries, identification of resource oriented and resource contention queries accurate workload classification, performance prediction and adaptation.

Number of researchers (Zewdu et al. 2009; Elnaffar, 2002), database and data warehouse vendors (IBM Corporation, 2003; Microsoft SQL Server) have contributed in the area of workload classification, workload performance prediction and workload
adaptation. A number of research activities are done regarding classifying and characterizing the DBMS and Data Warehouse (DW) workload. The focus of the work is on identification of workload and characterization of the workload by using statistical and data mining techniques (Zewdu et al. 2009). In this regard, (Elnaffar 2002) identified the classes of the workload, workload mix and then from the mix, how much concentration of workload of one type and other type.

Several researchers have also conducted research in the context of performance prediction of database and data warehouse workload (Ganapathi et al. 2009). Performance of the workload is important factor in tuning the DBMS. A prediction model is proposed (Dan et al. 1995; Sapia et al. 2000) for the query execution time in a warehouse environment. The research is carried out to predict the performance of workload using queuing model for predicting the performance for the next interval based on current and predicted parameters (Menasce et al. 1999, 2001, 2003). Another research is conducted to predict query runtime in the form of binary tree by developing (PQR) tree (Gupta et al. 2008) that predicts runtime on the basis of history. Ganapathi et al. (2009) predicted the performance variables for the queries and Elnaffar et al. (2002, 2004) predicted the shifts from DSS to OLTP workload. Thereska et al. (2005, 2006, 2007) examined the area to predict the performance and provided the direction towards self-managing system by developing a test-bed that uses what-if models. The database vendor (Microsoft SQL Server) has developed resource advisor for their product to predict the response time and throughput.

Workload adaptation is an active area of research and number of researchers conducted research in the area of adapting the DBMS workload. Self-adaptive DBMSs
have been developed to efficiently achieve its service level by taking advantage of its resources and through workload control (Adam et al. 2006; Niu et al. 2006). The Quality of Service (QoS) model has been developed (Krompass et al. 2008), the QoS controller adapts the workload and its changes, using performance model.

1.2 Knowledge gap

Literature is surveyed with respect to the workload classification, workload prediction and workload adaptation. The research has been conducted to classify or characterized the workload by applying different techniques. Workload is classified in the form of clusters by applying number of methods, but none of them have made any actual performance prediction that how the classified workload will perform. DBASs are incapable of monitoring or controlling the workload for decision making and DBMSs have no ability to perform all these activities itself. The research is done in predicting performance attributes of workload and they predicted one performance feature (such as time ranges), two features (such as throughput and response time) and few other attributes. Some other important attribute of workload performance exist such as workload size, Input/Output, execution time and others that are to be predicted and have a great impact on performance. There is a need of exploring the features that affect on the performance of the workload. These features can be of different variable names in different databases. Researchers used HP Neoview database, Microsoft SQL Server, International Business Machines (IBM) DB2 for workload prediction. Based on literature survey of last decade from (2000 – 2011) on workload prediction no implementation and experiments have been performed on database by applying case-based reasoning approach. Similar work performed by (Ganapathi et al. 2009) through developing Kernel
Canonical Correlation (KCCA) model. The KCCA model performs computation constantly for new query and then projection is done. It calculates the distance through K-Nearest Neighbor (KNN) and selection of the value of K is also tricky. The model is not adaptive as well, to the changing behavior of the workload. There are number of questions before executing the workload that how the workload can be managed by deciding, how much time it has to wait, or if it is a problematic query, should we kill it? What will be the performance of the queries after execution? There are system sizing problems, such as how many CPUs or disks are required for executing the workload with some time constraint and how much network bandwidth will be needed? Capacity planning problems are another issue. How can we know whether to upgrade or downgrade the system?

We are the first who have developed the Autonomic Workload Performance Predictor (AWPP) framework that performs all the tasks autonomically such as to detect the workload that is important component of autonomic databases. Features have been identified that are effective to the performance of DBMS. The research focus is not to classify the workload into its type but to make actual performance prediction and adapting the workload according to the change and environment.

### 1.3 Research Objectives

In this research work, we have designed and developed a framework for workload performance prediction that learns based on the experience or reasoning. It consequently, adapts according to the new trends or changes in the behaviour of the workload. The desired characteristics of the proposed framework are to be effective, adaptive and accurate. The main objective is to design and develop a framework for workload
performance prediction and adaptation to tune the database management system. The workload performance predictor and adapter framework will have all the features of monitoring, predicting and adaptation of the workload. The objective of this work is to predict the performance of workload of different types rather than the way how the workload is classified in DSS or OLTP. Here the type is assumed whether it is DSS or OLTP and then the performance of each type of workload is predicted. The predictor module will have the ability to predict the performance of the given classified workload itself with less or without any human intervention and hiding the complexity from the end users. The workload trend may change any time and can be adapted through adapter module. The objective is to incorporate autonomic characteristics in DBMS such as self-inspection, self-configuration, self-prediction and self-adaptation making DBMS aware of itself that makes it autonomic. It facilitates end users or any other system by providing the trend reports, exception reports and reports of actual performance vs. goals. The predictor will predict the current workload with resource utilization and due to this resource utilization, maximum performance and throughput can be achieved.

The organizations that are using DBMS may find the proposed methodologies and frameworks very useful in the development of the predictor and adapter. This framework can be helpful for database vendors as well to incorporate in their products and benefit. As the predictor and the adapter will perform all of its tasks with less or no human involvement so the DBA can do some other useful task, ultimately reducing the Total Cost of Ownership (TCO). There will be first time development cost of the predictor and the adapter but it will pay in terms of time, efficiency, effectiveness and accuracy. The research focuses on the following:-
• Study, design and develop an Autonomic Workload Performance Predictor (AWPP) for workload performance prediction and adaptation on evolution.

• Improve the effectiveness, accuracy and significance of the pre-detection stage by appropriate sub set of feature selection that best represents.

• Develop and apply machine learning techniques for predicting the performance of workload to make DBMS autonomic. This prediction will be helpful for DBMS resource management, capacity planning and system sizing.

• Improve the effectiveness, accuracy and significance of the workload performance prediction.

• Develop and apply machine learning techniques for adapting the new changes and trends of workload.

• Design and develop AWPP framework that can adapt the changes in workload and improves the performance.

• Evaluate and compare the proposed solution with other techniques based on performance measure, effectiveness, accuracy and significance.
We have developed strategy canvas, an action framework for building a compelling blue ocean strategy. The strategy canvas is shown in Figure 1.1 captures the current state of the research with respect to blue ocean strategic move. The horizontal axis captures the factors that the research industry competes on and invests in and the vertical axis captures the offering level ranging from low to high.

![Figure 1.1: The strategy canvas](image)

The scope of the study is limited to the following:

1. The data used in the research is TPC-H benchmark data (http://www.tpc.org). It is widely used by other researchers of this field as well (Zewdu et al. 2009; Elnaffar S, 2002)

2. The study focuses on MySQL database and does not consider any other database or Data Warehouse.
3. Representative machine learning techniques and lazy learning techniques are deployed as proof of concept to the proposed framework.

4. TPC-H and TPC-C like queries are used for experiments.

The significance of the research is discussed as follows:

1. Workload performance prediction
   a. We predicted performance features to make DBMS aware of its resources such as Bytes sent, Bytes received, Workload size and others using MySQL.
   b. Our approach uses history and it learns from experience based on reasoning that does not require to perform entire computation and the machine learning is also not needed.

2. Workload Adaptation
   a. AWPP is able to accept the new trends of workload.
   b. Adaptation of evolving behavior of workload.

The research finding are expected to lead to a better understanding on the workload management and provides better approach in designing more effective predicting and adaptive systems. The research could benefit both researchers and practitioners.

This research forms an Autonomic Workload Performance Predictor (AWPP) framework for DBMS using machine learning techniques and CBR approach. The framework consists of workload features extraction, workload performance prediction and workload adaptation. The research contributions are as follows:

1. The research builds framework for workload performance prediction that predicts performance of the workload using machine learning technique.
2. Enhancement in the proposed framework by identifying and adapting the trends of workload, when the behaviour changes using lazy learning technique.

3. Incorporation of autonomic computing characteristics in proposed framework AWPP to make it autonomic or self-managing.

4. The research designed and developed the CBR based solution (single approach solution) for all the three components (workload feature extraction phase, workload performance prediction phase and workload adaptation phase) of AWPP to achieve objective of the research. The CBR is selected because it learns from experience based on reasoning where no machine learning is required. It works on direct experience. In CBR data elicitation is not required.

Figure 1.2 highlights the research contribution in the form of hierarchy. The main contribution is the proposed performance prediction framework and its enhancement to be adaptive that has the ability to adapt the new changes in the workload. The design of the proposed framework is based on the philosophy “case-based reasoning”.

A new theoretical framework is proposed to guide in the design and development framework for CBR adaptation for new workload evolution. The proposed framework is a step forward in addressing adapting new workload changing behavior.
A representative framework is developed as a proof of concept to demonstrate the capability of the proposed theoretical framework.
1.4 Organization of Thesis

The thesis is organized into six chapters. Chapter 1 provides an overview of the research and organization of the thesis. Chapter 2 provides literature review that leads to the formulation of the research problem. Chapter 3 presents research methodology, while the AWPP framework is discussed in detail in chapter 4 that focuses on the performance prediction of the workload. Chapter 5 presents the enhanced workload adaptive framework and chapter 6 concludes the research and provides future work.
CHAPTER 2

LITERATURE REVIEW AUTONOMIC WORKLOAD MANAGEMENT

This chapter begins with a background review of autonomic computing and its incorporation in DBMSs. The taxonomy of autonomic workload management is also described briefly. The literature review focuses on the predictive and adaptive performance tuning in DBMS by examining the existing frameworks and the knowledge gap in the present research work is highlighted. Issues in the existing frameworks and models for workload performance prediction and adaptation are also discussed. Finally, this chapter concludes by highlighting the research problems.

2.1 Autonomic Workload Management in DBMSs

Database vendors such as IBM DB2, Oracle and SQL Server and different researchers have been working on self-managing and autonomic database managements systems (Elnaffar (2002, 2003, 2006); Martin et al. (2006); Lahiri (2003) and Agrawal, (2004)). The term autonomic is first time introduced in 2001 by IBM. Autonomic computing (AC) is important to be incorporated in different areas of computer science and is equally important in workload management in DBMSs. AC has some characteristics and a few or all of these should be there for the system to be autonomic
The research identified some shortcomings in DBMSs with respect to AC characteristics (Elnaffar et al. 2003). The three selected DBMSs such as DB2, Oracle and SQL Server are surveyed (Raza et al. 2008, 2010a, 2010b) and the achievements of current DBMSs are highlighted and evaluated by overcoming their shortcomings. Current DBMSs are on certain level of AC, having some autonomic characteristics in their components, tools, and utilities (Raza et al. 2009). An increase in data volume, complexity and heterogeneity in the workload has attracted the attention of the database researchers to the need of managing the workload properly for better performance. In another survey (Raza et al. 2011) different tools, techniques and algorithms are explored and observed from the autonomic perspective of workload management in DBMSs and Data Warehouses (DWs). There are different aspects of autonomic workload management, such as self-inspection, self-healing, self-optimization, self-prediction, self-configuration and, self-adaptation (Parashar et al. 2005; Koehler, 2003; White et al. 2004; IBM Corporation, 2003). The research in the context of workload management has been carried out for many years and different database vendors have developed and incorporated a number of tools for this purpose. These techniques and tools are related with classification, prediction and adaptation of different parameters for the workload. This motivates researchers and database vendors to work on self-managing workload models and frameworks, which have the ability to recognize, classify, predict and adapt to workloads changes autonomically.

Autonomic computing systems always work in certain environments and have five components, such as Negotiation, Execution, Observation, Deliberation and Failure Recovery (Koehler et al. 2003). An autonomic system consists of seven elements and all
these elements are connected through loop interface (Hausi A, 2006). These elements are: Autonomic Manager, Managed Element, Monitor, Analyze, Plan, Execute, and Knowledge Base. AC systems have various issues and challenges for the future that work with minimal human intervention. IT industry realizes that AC has a lot of challenges and recognizes that meeting these challenges is imperative; otherwise these systems will be difficult to manage. These challenges include Conceptual challenges, Architecture challenges, Middleware challenges and, Application challenges in an autonomic system (Parashar et al. 2005). AC is becoming necessary for future networks as well as in DBMSs to increase efficiency and become less error prone as in AC systems no or less human intervention is required. Databases, being the integral parts of organizations, have a greater need to be autonomic. Autonomic workload management should have AC properties such as self-inspection, self-configuration, self-optimization, self-prediction and self-adaptation. A survey (Raza et al. 2011) was conducted on the DBMS workload with respect to the autonomic characteristics.

Self-inspection in autonomic workload management supports better decision making by using the knowledge of its resources, limits, intensity, etc. In order to achieve efficiency in workload management, the configuration of different components should be performed in a self-managed way. Self-prediction in workload management helps to forecast the different aspects such as resource demand, workload frequency and memory requirements. Self-adaptation allows adaptation to the changes in workload according to the available resources and environment. In the following section the workload management is organized in to taxonomy based on qualitative study and on the literature.
2.2 Taxonomy of DBMS Workload Management

The main objective of this research work is to design and develop a framework for performance tuning for workload management called an Autonomic Workload Performance Predictor (AWPP) that is used to predict the performance of the workload in database systems. Autonomic workload management is expected to manage the workload in an efficient and responsive way. The autonomic technology has a high potential to be incorporated in current DBMSs. The challenges in the workload management includes identification and decision about problematic queries, identification of resource oriented and resource contention queries, workload characterization, prediction (Dayal et al. 2009) and adaptation (Wasserman et al. 2004).

The AWPP framework will have all the features of monitoring, prediction and adaptation of workload. The predictor will predict the current workload with maximum resource utilization and due to this resource utilization the maximum performance and throughput can be achieved. Allocation and re-allocation of resources will lead to maximum output. The prediction will help the optimizer and DBMS algorithm to optimize them accordingly. The workload adaptation is required when the behaviour of the workload changes. The predictor will have the ability to adapt and manage the given workload itself with less or without any human intervention and hiding the complexity from the end users. We believe that this AWPP will collect the necessary information about workload, such as type, intensity, resource etc. itself and manage workload with minimal human intervention.

The workload management taxonomy is shown in the following Figure 2.1.
Chapter 2

Literature Review Autonomic Workload Management

Figure 2.1: Taxonomy of Autonomic Workload Management

- Workload Monitoring
  - Workload Performance Tuning
    - Self-Optimization
      - Self-Prediction
        - Performance Modeling approach
          - Heuristic approach
            - Threshold approach
              - Heuristics
              - PAQRS Algorithm
              - PMM Algorithm
              - M & M Algorithm
              - Query Patroller
              - Active System Management (ASM)

- Workload Adaptation
  - Self-Adaptation
    - Performance Modeling approach
      - Threshold approach
        - Heuristic approach
          - PAQRS Algorithm
          - PMM Algorithm
          - M & M Algorithm
          - Query Patroller
          - Active System Management (ASM)

- Workload Monitoring
  - Self-Inspection
    - Workload Performance Tuning
      - Self-Optimization
        - Self-Prediction
          - Performance Modeling approach
            - Heuristic approach
              - Threshold approach
                - Heuristics
                - PAQRS Algorithm
                - PMM Algorithm
                - M & M Algorithm
                - Query Patroller
                - Active System Management (ASM)

- Workload Adapation
  - Self-adaptation
    - Performance Modeling approach
      - Threshold approach
        - Heuristic approach
          - PAQRS Algorithm
          - PMM Algorithm
          - M & M Algorithm
          - Query Patroller
          - Active System Management (ASM)
2.3 Workload Monitoring

Performance tuning plays a significant role for DBMS efficiency. Monitoring workload is a key consideration in performance management because of the complexity and diversity of the workload. A number of researchers have focused on monitoring the workload by classifying and characterizing it on the basis of their characteristics. Early research conducted on workload characterization aimed to tune the system in many aspects such as capacity planning, scheduling (Menasce et al. 1999), system sizing (Jain et al, 1991) and configuration (Ferrari et al. 1983).

2.3.1 Self-Inspection

Self-Inspection is a characteristic of autonomic computing that has the ability to make intelligent decisions which are based on self-awareness. There are a number of tools and techniques that are used to examine the workload at all times. Research in the context of self-inspection of workload management has been conducted by various researchers. The self-inspection mechanism analyzes the resources and low level characteristics of the workload using machine learning techniques. Information about resources demand can be taken from the system monitoring tools rather than on assumption from previous knowledge. The inspection mechanism should identify as well as quantify the workload intensity in mixed types of workload as this information provides the basis for tuning and modeling of the system. A research is conducted by (Elnikety et al. 2004) on monitoring the workload and developed a GATEKEEPER for admission control and scheduling of E-commerce workload to improve response time and stability. Another group of researchers (Chandramouli et al. 2007) developed the tool PREDATOR for query suspension and resumption. The induction of asynchronous
checkpoints for cardinality in a query is used. The research is conducted by (Chaudhuri S., 2007) for query suspension and resumption to handle workload in an efficient manner. In case of suspension and resumption, only selective information from the past is saved rather than saving all the previous work as happened in other techniques. Due to this technique, overhead can be reduced and it also reduces the running time of restarted query. In Stop & Restart technique, checkpointing is used to improve the performance (Chaudhuri S, 2007). As the technique saves only the remaining part of the suspended query, so there will be less memory wastage with faster restart. The Merge-Pipeline algorithm is used in this technique, which is more efficient than the Current-Pipeline algorithm. The overheads for monitoring, in this technique, are low. The focus of the approach is on regenerating all the results rather than generation of remaining results. Bruno et al. (2007) proposed a framework for online tuning that examines the current workload at all times and the change in the physical design.

As DBMS has become an integral part of many organizations that often require tuning to be performed for optimal performance of DBMS (Lo et al. 1998). For a system like DBMS, identifying the characteristics of workload can help in tuning and configuring the system in an effective way. The characterization of workload has been recognized by a nonprofit organization, Transaction Processing Performance Council (TPC). TPC deals with the production of benchmarks to measure the performance of a system through synthesized workload for conducting experiments (TPC Benchmark Standard Specification). The workload classification is very important for workload management because if it knows the type of workload by monitoring or self-Inspecting, it can manage in a better way by tuning the performance (DB2 Universal Database, 2000;
Oracle9i Database, 2001; Adaptive Server™ Enterprise Performance and Tuning Guide, 1999; Packer et al. 2002). Other researchers have worked on self-Inspection of the workload by monitoring and classifying the DBMS workload (Yu et al. 1992; Elnaffar, 2002 and Zewdu et al. 2009).

**Statistics**

Early research focused on statistical techniques for the characterization of queries. Research work has been done on workload characterization that is based on architectures for analyzing the performance degradation problems (Keeton et al. 1998; Ailamaki et al. 1999). Barroso et al. (1998) discussed the work done on the behaviour of the DSS and OLTP types of workload. The workload is analyzed on the basis of characteristics of TPC-C and TPC-D of real production database workloads. Different researchers have used TPC benchmark databases for experimentation analyzes of proposed approaches (Hsu et al. 2001, TPC Benchmark C, 2001). The static and dynamic techniques are applied. The methods used are basic statistics which includes summaries (average, distributions), the non-homogeneous poisson process and histograms. The purpose is to understand memory behaviour in different architectures, enhance buffer replacement and concurrency control. Other techniques such as Numerical fitting are applied by using attributes such as the number of parameters of Database (DB) transaction, application and SQL statement for characterizing workload.

Early research was focused on automating the system. The research has been carried out on the structure and composition of the SQL statement in relational databases (Yu et al. 1992). They also studied the queries runtime behaviour and other complexities to evaluate design trade off. A model has been developed called the Relational Workload
Analyzer database (REDWAR) for workload characterization. The purpose of the REDWAR tool is to analyze the SQL statement in terms of “WHERE” or “GROUP BY” used in the statement. The workload management in early systems is done through the DBA or automatically and skilled DBAs are required to manage the workload (Lightstone S et al, 2002). With the incorporation of autonomic technology the trend has changed from the automatic to the autonomic level (Muller et al. 2006; Diao et al. 2005).

**Numerical Fitting**

Many studies and research have focused on exploring the characteristics of DSS and OLTP (Hsu et al. 2001; Barroso et al. 1998; Keeton et al. 2000). Numerical fitting techniques are applied and different methods of Non-linear regression are used, such as the Levenberg-Marquardt method to model the arrival pattern of the transactions. The combination with other statistical techniques is performed to get better results. The arrival time is used as a parameter for DB transaction for the interactive type of workload.

**Prediction Model**

Early research was focused on predicting the hit ratio of buffer in the database. A number of researchers conducted research for characterizing the access pattern (Dan et al. 1995) and on the user access behaviour (Sapia et al. 2000). Another approach that predicts the user behaviour in a Multidimensional Information System Environment defines the user’s access patterns to caching of OLAP systems. The approach is based on the Markov Chain model for the OLAP queries and provides the predicted patterns (Sapia et al. 2000). The purpose of research is to predict buffer hit ratio, to make predictive prefetching and enhance the caching. The attributes used are think time and sequence of
executing queries of a session on database query using an interactive type of workload by combining other statistical techniques.

**Clustering Techniques**

There are a number of researchers who characterized the DBMS workload on the basis of different features for tuning the DBMS. For tuning the system people used clustering techniques for getting the classes of transactions that are grouped according to the resource utilization (Yu, et al. 1994). Dan et al. (1995) focused on the access pattern for the buffer hit ratio.

Nikolaou et al. (1998) applied the clustering approach to the workload in which workload is divided into classes which have similar characteristics. The research presented two environments, CLUE and HALC, for clustering. The CLUE classifies the OLTP type of workload by converting it into classes according to the database patterns, whereas HALC is a batch mode heuristic clustering algorithm that can handle a large volume of data. The on-the-fly clustering algorithm is introduced based on a neural network that can be used for online systems. Traditional batch-mode can not perform well and may degrade the system performance. An Adaptive K-Means algorithm is implemented for experiments (Chinrungrueng et al. 1995). The heuristic clustering algorithm (HALC) and Adaptive K-Means produced better results for clustering as compared to the K-Means algorithm. The number of parameters, such as number of database calls, number of locks, and number of references on database transaction is used for experiments. The workload type used is batch and interactive (Yu et al 1994; Dan et al 1995). The workload is classified into two types, DSS and OLTP, in MySQL (Zewdu
et al. 2009) using statistical and data mining techniques such as hierarchical clustering, classification and regression tree (CRT).

**Unsupervised data mining technique**

Analysis of characterization techniques is presented for BI workload (Wasserman et. al., 2004). The approach is based on some resource related attributes such as CPU consumption, sequential and random I/O rate and joins degree. Sizing technique works by collecting input data from the user. The input data is validated and resource demand identified for each workload class. The total resource demand is calculated by aggregating the resource demand for each class and finding the hardware configuration for it. Proposed characterization technique by (Wasserman et al. 2004) is performed only with TPC-H benchmark data and mostly the values of parameters are based on their assumptions.

**Decision Tree Induction**

Elnaffar et al. (2002a; 2002b) focused on workload management and developed classifiers that automatically classify the workload into two types Decision Support System (DSS) or Online Transaction Processing (OLTP). An induction tree is used for building the classifiers. On the basis of workload type, resources and other requirements can be handled. The autonomic system has the ability to recognize the workload type and then perform resource allocation accordingly. A model is developed for classification on the basis of workload characteristics that classify it and identify the change in the type of workload. Elnaffar et al. (2004) has extended the work by predicting the change in shift of workload from OLTP to DSS and vice versa through a predictive model namely the Psychic Skeptic Prediction framework (PSP).
Classification and Regression Tree (CRT)

Zewdu et al. (2009) conducted research using MySQL database by classifying the workload into the DSS and OLTP types. The identified features are good for classifying the workload. The hierarchical clustering, classification and regression tree for characterization is applied and developed a model for autonomic databases that can characterize the workload. MySQL variables are identified that can be helpful to categorize the workload into its types either DSS or OLTP. Identified four variables that include query ratio of Select and Update/Insert/ Delete (Com_ratio), number of users logging (Innodb_log_writes), the number of query cache hits (Qcache_hits) and number of statement executed (Questions) are effective in classification.

Table 2.1 depicts the attributes selected by different researchers to characterize the workload in different types of workload such as DSS, OLTP and, Business Intelligence (BI), and the techniques they applied for validation and their limitations.

Table 2.1: Summary for workload classification techniques and their limitation

<table>
<thead>
<tr>
<th>Technique used</th>
<th>Input parameters</th>
<th>Type of workload used</th>
<th>Purpose of Research and Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics</td>
<td>Arrival time,</td>
<td>Interactive</td>
<td>Examined the characteristics of the</td>
</tr>
<tr>
<td></td>
<td>cache miss rate,</td>
<td></td>
<td>workload of production</td>
</tr>
<tr>
<td></td>
<td>memory footprint,</td>
<td></td>
<td>database of ten world’s largest</td>
</tr>
<tr>
<td></td>
<td># of references</td>
<td></td>
<td>corporations, compared to TPC-C and</td>
</tr>
<tr>
<td></td>
<td>to a memory block</td>
<td></td>
<td>TPC-D.</td>
</tr>
<tr>
<td></td>
<td>read/write page</td>
<td></td>
<td>In certain cases, TPC-C and TPC-D</td>
</tr>
<tr>
<td></td>
<td>accesses, CPU</td>
<td></td>
<td>falls outside the scope of behaviour</td>
</tr>
<tr>
<td></td>
<td>demands,</td>
<td></td>
<td>depicted by the production workloads.</td>
</tr>
<tr>
<td></td>
<td>deadlocks,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>number of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>transactions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>completed, # of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>different pages</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>accessed per</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>transaction</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Numerical Fitting

<table>
<thead>
<tr>
<th>Lo et al. 1998; Hsu et al. 2001.</th>
<th>Arrival time</th>
<th>Interactive</th>
<th>Examined the database performance on simultaneous Multithreading (SMT) processors. Few real workloads characteristics are not reflected by either of the benchmarks</th>
</tr>
</thead>
</table>

### Prediction Models

<table>
<thead>
<tr>
<th>Dan et al. 1995; Sapia et al. 2000.</th>
<th>Think time, sequence of queries in executing</th>
<th>Interactive</th>
<th>The purpose of research is to predict buffer hit ratio, to make predictive prefetching and enhancing the caching.</th>
</tr>
</thead>
</table>

### Clustering

<table>
<thead>
<tr>
<th>Yu et al. 1994, Dan et al. 1995, Chinnungrueng et al. 1995, Nikolaou et al. 1998.</th>
<th>No of database Calls, No of locks, No of references</th>
<th>Batch Interactive</th>
<th>To optimize the performance of the system the distribution of the requests to different servers for workload balancing.</th>
</tr>
</thead>
</table>

### Decision Trees Induction

<table>
<thead>
<tr>
<th>Elnaffar et al. 2004</th>
<th>Queries Ratio, Pages Read, Rows Selected, Throughput, Number of Locks Held, Ratio of Using Indexes (%)</th>
<th>BI workload</th>
<th>Performed the analysis of characterization techniques based on some resource related attributes.</th>
</tr>
</thead>
</table>

### Hierarchical clustering and classification Regression Tree (CRT)

<table>
<thead>
<tr>
<th>Zewdu et al. 2009</th>
<th>Query Ratio, InnoDB_log_writes, Qcache_hits, Questions</th>
<th>DSS OLTP</th>
<th>Selected Classification variables are less affective, Less accurate, The existing four attributes are not enough to represent all the characteristics of DSS and OLTP. Classification is not done by taking advantage of Machine Learning (ML) and Artificial Intelligence (AI) techniques</th>
</tr>
</thead>
</table>
2.4 Workload Performance Tuning

Workload performance tuning is an important aspect of workload management. Performance prediction plays a significant role in performance tuning. The performance prediction makes the DBMS aware of itself, making it autonomic. The self-predicting systems have the ability to predict based on history according to the available resources and environment. Various researchers (Elnaffar et al. 2004; Narayanan et al. 2005; Gupta et al. 2008; Ganapathi et al. 2009; Akdere et al., 2011) have conducted research to predict the DBMS workload performance attributes. These predictions are helpful for configuring and optimizing the system. The research with respect to self-configuration, self-optimization and self-prediction is discussed as follows.

2.4.1 Self-configuration

The self-configuration is the ability of a DBMS to configure and reconfigure itself dynamically according to the given goals/ objectives and changing conditions of the workload. A model has been developed for the E-commerce environment (Menasce et al. 1999). The Customer Behaviour Model Graph (CBMG) or state transition graph is introduced that represents similar navigational patterns for groups of customers who perform the same activities. There are some limitations of the model, for example, there are maximum session drops when there are huge sessions or maximum load. The other limitation is that the model has no mechanism to manage or recover these drop sessions. The workload model and its parameters are identified and presented through a clustering algorithm for workload characterization. The technique is evaluated with different experiments and results. The QoS level in the E-commerce application is discussed by dynamically monitoring and tuning (Menasce et al. 2001). The technique identified the
best configuration parameters by combining the hill climbing technique with the analytical queuing model. The experiments are performed to evaluate the technique by making comparison of QoS levels. The technique for QoS of the E-commerce workload can handle dynamic workload and short term fluctuations. The technique uses heuristic optimization with the predictive queuing model and provides better results. It used a reactive approach rather than a proactive. The technique used the hill climbing method for searching, but when it gets stuck, a sub-optimal solution is achieved. Menasce et al. (2003) have designed controllers that use analytic performance models with combinatorial search techniques. This modeling technique is used to identify the best configuration for the given workload. The model is used to predict QoS parameters of the workload. The effectiveness of the technique is presented through the simulation and the experiments. The basic reasons for performance problems in the On Line Transaction Processing (OLTP) workload through different metrics are identified (Weikum et. al., 1994). The workload management is done by performing, system and database configuration.

The storage technique as discussed in (Anderson et al. 2002) is introduced for Storage Area Networks (SANs) (Ward et al. 2002). They presented two algorithms for designing a cost effective SAN. These algorithms provide snapshots, mirroring, backup and configuration. The algorithms are evaluated through the experiments over different design problems and showed their effectiveness. Additionally, these algorithms can characterize the workload by using a storage workload estimator (Ward et al. 2002).
2.4.2 Self-Optimization

The self-optimization is the characteristic that optimizes the process of modifying a system or some components so that they work more efficiently with minimum resources. The research in the area of the self-optimization of workload management has been carried out by different researchers. The optimization is achieved through observing the incoming requests and proper resource utilization.

The QoS management technique is presented by (Krompass et al. 2008) in which they applied an economic model that is used to handle the individual requests of BI and OLTP workload proactively. A systematic way is provided to arrange different requests by dividing into different classes based on cost and time limit. The proposed model calculates the cost of a request by differentiating the under achieving and marginal gains of a Service Level Objective (SLO) threshold. The framework is evaluated to observe its effectiveness by performing experiments on different workloads. The framework provided for QoS in workload management is beneficial for OLTP and BI workload. The framework used an economic model with two economic cost functions (Opportunity Cost, Marginal Gains), where penalty information is added with the queries. The penalty information is used to make the efficient order of pending query execution. The scheduling policy used for the OLTP workload in this framework is enhanced by considering the combined effect of priorities and service level objectives rather than merely considering priority.

The microeconomics concepts are used in the research (Luo et al. 2006) to devise the resource allocation framework for a multiuser environment. The main purpose of this framework is to reduce the response time of running queries. As compared to a single
query progress indicator (PI), this framework considered the impact of concurrent running queries on each other named as a Multi-Query SQL PI. The proposed multi-query progress indicator is used for workload management. The effectiveness of the framework is presented by experiments and implementing the framework in Postgres SQL. A Multi-Query SQL PI is distinguished from the previous work by considering the impact of queries over each other. It has good prediction and adaptation ability. In cases when it has been provided with some wrong information, it makes corrections by mean of an estimation mechanism. When the statistics are wrong or obsolete, then the multi-query PI will not provide the correct estimates. Different researchers have worked on resource allocation techniques to optimize the workload (Schroeder et al. 2006; Mehta et al. 1993; Davison et al. 1995). Different resource allocation techniques for assigning resources to the workload efficiently have been proposed in the research.
2.4.3 Self-Prediction

The systems that monitor all the time and provide accurate predictions about the workload change or resource demand are called self-predicting systems. These systems are dependent on the identification of workload that provides workload information. On the basis of this information, previous history and mathematical models, a self-predicting system predicts the future. The system itself predicts tuning, acquisition planning and resource allocation with the help of internal system knowledge. The self-prediction is the characteristic of ADBMS to predict the future on the basis of historic and statistical data. The research has been done in the context of workload prediction and a number of researchers have predicted the performance of DBMS workload. The research is carried out in predicting only one performance attribute, such as the time ranges (Gupta et. al., 2008), and research reveals that researchers have predicted the two performance attributes, such as throughput and response time (Agrawal et al. 2004). The research is conducted that predicts the multiple performance metrics for DBMS workload (Ganapathi et. al., 2009).

Binary Tree

A framework called the Psychic-Skeptic Prediction Framework (PSP) has been proposed by (Elnaffar et al. 2002, 2004) which is used to predict workload shifts from DSS to OLTP workload. The PSP framework consists of three components which are Training Data Model, Psychic (offline) and Skeptic (online). The PSP is offline. However, when it estimates the time interval for expected shifts using historical workload models it works online. The Psychic uses the polynomial regression technique on consolidated scenario and builds an offline prediction model. As compared to other
online prediction techniques it has less overhead. The PSP architecture has self-optimizing and self-healing characteristics making it autonomic. The PSP framework is limited to scheduled tasks and does not have the ability to manage drastic workload change. Gupta et al. (2008) proposed a predicting model for the execution time of a query in a warehouse. The model used a query execution plan (QEP) and system load to predict the query execution time through binary tree implementation. The approach is validated and it can be incorporated into commercial DBMSs. The model predicts on the basis of historical data in the form of binary tree Predictions of Query Runtime (PQR) tree. The PQR trees build in two steps, which are obtaining a PQR tree and time range for a new query. After constructing the PQR tree, it is applied and updation is done periodically. Finally, the execution time of the workload is predicted.

**What-if model**

Thereska et al. (2005, 2006, and 2007) have developed a test bed ‘Ursa Minor’ which is used to predict the workload and provide a direction towards the self-managing system. The Ursa Minor has two major components, Observer and Stardust, and is based on a what-if model (Thereska et al. 2005). Ursa Minor test bed is a cluster based storage system. It is designed in such a manner that it can easily be incorporated into existing as well as new systems. The modeling infrastructure has also been devised. The Observer is an expectation based model to perform predictions in the common region of operations. The Stardust is the other infrastructure that is developed for the shared, distributed systems. It monitors the service centre and critical path of requests while ignoring background and maintenance activities. Ursa Minor uses object-based storage which exposes more information about the stored data. The Ursa Minor provides scalability
using cluster-based technique and dynamic adaptive behaviour through online choice. The re-encode process of Ursa Minor takes some extra time for the system but the throughput increases up to three times (Abd-El-Malek et al. 2005).

DB Resource Advisor is used to predict the response time and throughput dynamically (Agrawal et al. 2004; Narayanan et al. 2005; Microsoft SQL Server 2005 Books, 2007). The advisor predicts the workload using a what-if model without using the configuration description. This helps the advisor to guess the status of the resources. Identified components required for self-prediction are low level instrumentation, end to end transaction tracing and parameterized models of hardware resources. It provides accurate trends of response time in transactional tracing. The experiments are performed on OLTP workload and it is observed that Resource Advisor accurately predicts the changes in the workload. The Resource Advisor presents a modular architecture in which CPU, buffer and storage models are integrated to predict the response time and throughput by identifying the required key components (Agrawal et al. 2004). In Resource Advisor, use of an end-to-end tracing technique benefits visualization and understanding the performance of the system. The resources are properly allocated on the basis of continuous monitoring. By using these models, Resource Advisor provides an accurate prediction and the best performance results. There are some limitations in the tool, such as when the size of the buffer pool is lower, then it has high overheads per transaction. This overhead can be reduced by using some other appropriate techniques. Finally, the tool is evaluated through a prototype implementation in SQL Server.
Exploratory model and Confirmatory model

Workload models for Autonomic DBMS (ADBMS) are becoming necessary for the systems (Martin et. al., 2006). In the research an exploratory model is used to monitor the given workload and a confirmatory model is used to analyze the workload. The machine learning and data mining techniques are used to develop the exploratory and the confirmatory models for ADBMS, as it is becoming the general need for models. Elnaffar et al. (2002, 2004) presented workload models for autonomic DBMSs with examples and discussed a number of issues about the proposed technique such as definition and storage of the workload model, effectual monitoring of managed element and incremental maintenance of the models. In the research an autonomic model is developed for some specific scenario but has not provided any general solution.

Regression Techniques

Dayal et. al., (2009) carried out research to predict the attributes that used regression techniques. Covariate vectors are defined for query plan and performance metrics through dependent and independent variables. There exists a relationship between the dependent and independent variables, so a change in the independent variable has an effect on the dependent variable. Regression analysis enables to estimate the value of the dependent variable on the basis of independent variables. Independent variable estimation is called regression function. Dependent variables are characterized around it, representing probability distribution. Linear regression and least squares used are parametric and non-parametric (Lloyd, 1982). Results through this method are not satisfactory and predicted the poor results; even in some cases the negative amount of predicted time is recorded. There are some limitations of the linear regression technique.
When the relationship between the independent and dependent variables is linear it implements a statistical model and the results are optimized. However, for nonlinear relationships linear regression is not good and the results are not optimized.

**Principal Component Analysis**

Principal Component Analysis (PCA) is used for the eigenvalue decomposition or singular value decomposition of a data matrix. Number of researchers applied PCA (Dayal et. al., 2009; Ganapathi et. al., 2009) for predictive analysis. PCA can be implemented individually for the two data sets. PCA has the limitation that it does not have the ability to identify the relationship between the two sets. The correlation can not be identified between the data sets, results presents that by applying PCA the results are not satisfactory (Ganapathi et. al., 2009).

**Canonical Correlation Analysis (CCA)**

The CCA can solve the problem of PCA. As CCA deals with Euclidean vector spaces, when the Euclidean dot product is high the queries are considered to be similar (Ganapathi et. al., 2009). One of the limitations is that, different queries which can textually be same but have different performance.

**Kernel Canonical Correlation Analysis (KCCA)**

The Kernel Canonical Correlation Analysis (KCCA) that is based on CCA that applies kernel function instead of the Euclidean dot product (Bach and Jordan 2003). Kernel functions are used in machine learning techniques providing better results (Shawe-Taylor and Cristianini 2004). The existing algorithms are evaluated for long running queries and introduced as an approach to manage workload through resource usage prediction of queries (Dayal et al. 2009; Ganapathi et al. 2009). The approach uses
KCCA which identifies the correlation between query properties and performance metrics on the training set. On the basis of a statistical relationship it predicts the performance of incoming queries. Query features are identified through machine learning algorithms and compute similarity measures between each pair of query feature vectors. Identified query features are used by the KCCA model to find its coordinates on query projection and performance projection. K-Nearest neighbour algorithm is applied for this purpose. Required predicted metrics are achieved through mapping of these metrics with performance projection. KCCA model adapted by the prediction framework does not predict all queries and has no ability to perform continuous retraining. The proposed model does not provide any adaptation strategy.

Table 2.2 represents the performance prediction techniques, their limitations and predicted attributes.

Table 2.2: Summary for workload performance prediction techniques and predicted attributes

<table>
<thead>
<tr>
<th>Attributes predicted</th>
<th>Techniques Used</th>
<th>Input</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput</td>
<td>What-if model: Narayanan et al. (2005)</td>
<td>Instrument a private copy of the SQL Server source code to track the use of CPU, memory, and I/O resources.</td>
<td>In Resource Advisor when the size of buffer pool is lower than Resource Advisor has a high overhead per transaction. Continuous monitoring also affects the performance of the system.</td>
</tr>
<tr>
<td>Response Time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time Ranges</td>
<td>Binary Tree: Gupta C et al. (2008)</td>
<td>query plan; optimizer’s estimate of the query cost; load feature vector</td>
<td>Ignored any time of day considerations. Data skewness problems and can not handle sudden changes in workloads behaviour.</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>Kernel Canonical Correlation Analysis: Ganapathi et al. (2009)</td>
<td>number of nested sub-queries, total number of selection predicates, number of equality selection predicates, number of non-equality selection predicates,</td>
<td>In KNN the value of K is tricky to decide how many cluster for the workload. Not predicted other performance features such as the Memory,</td>
</tr>
<tr>
<td>Records Accessed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Records Used</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk I/O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message Count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message Bytes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.5 Workload Adaptation

In DBMSs there is versatility of workload and the behaviour of workload change over time. In autonomic DBMS, in order to meet Service Level Objectives (SLOs), performance management the workload adaptation is performed. Workload adaptation techniques have also been applied in various area of computer science. Menasce et al. (1999, 2001, 2003) conducted research in the area of web services and workload adaptation. Various researchers (Brown et al. 1994; Pang at el., 1995; Horzyk et al. 2005; Schroeder et al. 2006; Niu et al. 2006, 2009; Qiang et al. 2009; Qian et al. 2010) carried out research in the area of workload adaptation in different DBMSs.

2.5.1 Self-Adaptation

The self-adaptation is the capability of the system to adapt changes dynamically. There is versatility in the DBMS workload and the adaptive system can handle the change of the workload. The self-adaptation is a way to convert an old system into a
target system to achieve maximum efficiency. This property has the capability of adapting all the required changes in the system according to the environment. There are different adaptation approaches for workload management. One is the heuristic approach, the second is the threshold approach and the third is a performance modeling approach. The performance modeling approach is the best approach as compared to the threshold and heuristic approaches. The self-adaptation has been achieved by performing different experiments and developing various frameworks and models, some of which are discussed below.

Heuristic approach

i. PAQRS Algorithm

Pang et al. (1995) devised a Priority Adaptation Query Resource Scheduling (PAQRS) Algorithm based on the Priority Memory Management (PMM) algorithm that deals with multi-class query workload. The algorithm reduces the missed deadlines according to the miss distribution defined by the administrator. The algorithm works by allocating memory and assigning priorities by considering resource usage, workload characteristics and performance experience. Whenever the workload changes, a new multiprogramming level (MPL) is calculated. The memory reallocation and priority adjustment is done accordingly. Two techniques, a miss ratio projection and resource utilization heuristics are used to calculate new MPL. In the case of the miss ratio projection method, previous MPL and miss ratio are used as parameters. The PAQRS is used to schedule the complex type of workload and reduces the number of missed deadlines thereby making efficient use of the system resources. It has a bias control mechanism, which regulates the distribution of missed deadlines among different query
classes. The MPL and memory is allocated on the basis of regular and reserve group quota. The priority of regular queries is higher than reserve queries. By doing this, PAQRS makes adjustments between the miss ratio and the target distribution. The PAQRS can not handle transactions and is limited to a workload consisting of mixed queries. Its performance degrades with the increased workload fluctuations. The adaptation mechanism of the PAQRS is not up to the mark and needs to be improved.

ii. M & M Algorithm

The technique suggested by (Brown et al. 1994) is used to alter the MPL and memory allocation. This technique handles workload by achieving per class response time goals using heuristics. Different classes have dependency which elevates the problem of assigning shared resources due to competition. For each class, MPL is assigned and memory settings are done using the M&M algorithm. The M&M algorithm is a simple, robust and responsive algorithm that works well for different workloads, configurations and memory requirements. It has a mechanism for disk buffer classes that allows it to deal with the interdependence among classes. It uses the feedback mechanism that adjusts the class goals, when these goals are violated or exceeded.

Threshold control approach

i. Query Patroller

In DB2, the flow of requests is controlled proactively and dynamically by streamlining the requests according to the available resources and workload through Query Patroller (QP) (IBM Corporation, DB2 Query Patroller Guide, 2003; Lightstone, 2002). This strategy helps to execute the small and high priority queries without any delay. The QP provides information about the completion of requests and finds the future
trends, workload of users, as well as the frequently used indexes and tables. Different classes are defined with respect to their size. It prioritizes queries according to user privileges and suspends high load queries so that they can be cancelled or scheduled to run after peak hours. The QP assigns privileges of resources at user and system level through DBA. The QP works as a gatekeeper that monitors the given workload, and after analysis and prioritization it schedules the incoming requests. On the basis of a profile provided by the administrator, it limits the flow of long running queries to avoid saturation and ensures better resource utilization.

ii. **Teradata’s Active System Management (ASM)**

ASM (Ballinger, 2002) controls the workload encountered in the DBMS by allocating resources to a different allocation groups. The ASM takes the preventive approach for the change in workload and handles the exceptions as defined with admission control using two parameters that are MPLs and the number of users. In ASM the resource allocation of the performance classes may not be accurately mapped to the allocation groups and the resources demand for each class can not be predicted well.

**Performance model approach**

i. **QoS controller**

Menasce et al. (1999, 2001, 2003) devised a QoS controller for e-commerce applications that has the ability to manage workload. They proposed that QoS requirements can be achieved by adjusting different configuration parameters within a system. These adjustments are done through the QoS controller by considering three performance goals which are average response time, average throughput and probability of rejection. The QoS controller adapts the workload changes using a performance model
based approach and predicts the performance through a queuing model. The queuing model uses the current and predicted workload values as parameters and predicts the performance for next interval. The architecture and its implementation for cluster based web services has been discussed (Pacifici et al. 2005). The web services workload is divided into different service classes for each gateway. The resources are allocated to different services with respect to MPLs for each gateway. There is a feedback control component that maintains the service classes in a performance model. This model considers that all requests have the same size, however in DBMS different requests have different size and resource demand.

ii. **External Queue Management System (EQMS)**

Schroeder et al. (2006) proposed to use, the response time, as Service Level Agreement (SLA) that is incorporated in the proposed framework to achieve QoS. They proposed a framework for scheduler, the External Queue Management System (EQMS) which is used to limit the multiprogramming level on concurrent requests. The EQMS consists of three components, scheduler, MPL Advisor and performance monitor. The EQMS self-optimizes in an adaptive way. This scheduler is independent from the internals of the DBMS. The framework has a feedback loop that is used to execute more than one query at a time. During the process, some information like available resources and number of executing requests is used by the feedback loop. The EQMS provides a self-tuning and adaptive response through a scheduler and MPL advisor. This helps to handle the changes in system load or workload dynamically and provides better results due to its feedback loop. It works well for all type of workload due to its core idea which is reducing contention by imposing a limit on the MPL.
iii. **Artificial Neural Networks**

The ability of Artificial Neural Networks (ANNs) is explained by their better performance in comparison with Natural Neural Networks (NNNs) (Horzyk et al. 2005). The NNN works on training data and tuning it according to the environment. In the same way in the process of adaptation ANNs works on the training data based on history and is provided initially first time. Qiang et al. (2009) suggested the performance forecasting based on a workload control scheme by using layered queuing network modeling techniques to build a performance model of DBMS.

iv. **Kalman Filter**

An autonomic DBMS model (Niu et al. 2006) has been proposed in the context of workload management by providing a new direction in research for automatically managing DBMSs workload. The proposed framework (Niu et al. 2006, 2007, 2009) has two components, workload detection and workload control. The workload detection finds the changes and provides information about the workload. The research contributes by designing a general framework for performance oriented workload adaptation, prototype implementation of the framework (Query Scheduler), a cost-based performance model for workload control plans and improved accuracy of performance prediction through the Kalman filter. The experiments for Query Scheduler are performed on stable workload which is not suitable for a dynamic environment where the workload changes rapidly, such as in OLTP or OLAP. During the experiments, the total cost of a query is used as a parameter that may generates error prone results. It is confined to linear workload; however, in the real environment most of the time workload is non-linear.
Qian et al. (2010) evaluated the workload adaptation architecture proposed by (Niu et al. 2009). They are able to control the user’s satisfaction rate and optimized control of the resources and make the architecture operable. There are some problems while adapting the workload with some conflicts about the user satisfaction rate and optimized control of the resources. Therefore, an accurate and fast adaptation process is required.

The workload adaptation mechanism uses classification rules for dividing the workload into classes (Menasce et al. 2003) however the classification rules are not followed. The Query Patroller uses resource demand as a criterion which makes it easy to control resource allocation (DB2 Query Patroller Guide, 2003; Lightstone, 2002) M & M algorithm by Brown et al. 1994), whereas PAQRS by (Pang et al. 1995) and ASM by (Ballinger 2002) uses performance goals for easy performance tracking. The combinations of both criteria are followed to achieve high level performance goal. The workload is detected in two ways, one characterizing the workload that proactively tracks the change in workload before performance is affected in PAQRS (Pang et al. 1995). The other is a performance monitoring approach that takes action when the performance degrades as in M & M algorithm (Brown et al. 1994) and ASM by (Ballinger 2002).

Table 2.3 summarizes different workload adaptation techniques used in DBMS workload and their limitations.
Table 2.3: Summary for workload adaptation techniques and their limitations

<table>
<thead>
<tr>
<th>Adaptation goal</th>
<th>Input used</th>
<th>Techniques Used</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>To achieve per-class response time goals</td>
<td>Complex DBMS workload MPLs and memory allocation</td>
<td>M &amp; M algorithm (Brown et al. 1994)</td>
<td>Deals only the performance objectives not the resource-oriented workload control</td>
</tr>
<tr>
<td>Admission control, allocating memory and assigning priorities</td>
<td>DBMS Workload New MPL, reallocate memory and priorities adjustments.</td>
<td>PAQRS Pang’s algorithm (Pang et al. 1995)</td>
<td>PAQRS can not handle transactions and is limited to workload consisting of mix queries. Its performance degrades with the increased workload fluctuations. The adaptation does not provide good accuracy and the adaptation process is slow. The adaptation mechanism of PAQRS is not up to the mark and need to be improved.</td>
</tr>
<tr>
<td>To perform admission control</td>
<td>Query costs and MPLs</td>
<td>DB2 Query Patroller (QP)</td>
<td>Performance objectives are not used as guides.</td>
</tr>
<tr>
<td>To control the DBMS workload using predefined rules that are based on workload thresholds</td>
<td>MPLs and number of users</td>
<td>Teradata’s Active System Management (ASM) (Ballinger 2002)</td>
<td>Resources are mapped to allocation groups, while mapping resource allocation requirement may not be fulfilled. ASM does not predict resource demand for each performance class.</td>
</tr>
<tr>
<td>To analyze or examining the set of Training Data (TD) and construction of neural network topology and weights computation.</td>
<td>Input Training Dataset</td>
<td>Artificial Neural Network (Horzyk et al. 2005)</td>
<td>Complex to model ANN</td>
</tr>
<tr>
<td>To schedule, MPL setting &amp; performance monitor &amp; self-optimize in an adaptive way.</td>
<td>Response time as an SLA</td>
<td>Queuing Theory (Schroeder et al. 2006)</td>
<td>The delay caused by workload management is too much</td>
</tr>
<tr>
<td>To improve user satisfaction rate.</td>
<td>SLO</td>
<td>Layered queuing network modeling techniques (Qiang et al. 2009)</td>
<td>Their work does not handle the distribution of client arrivals. Performance forecasting accuracy is less.</td>
</tr>
<tr>
<td>To improve the accuracy of the performance prediction, to meet the SLOs</td>
<td>SLO</td>
<td>Kalman Filter (Niu et al. 2006, 2008, 2009)</td>
<td>There are some problems in Niu B adaptation framework, while adapting the workload with some conflicts about user’s satisfaction rate and optimized control of the resources Qian et al. (2010).</td>
</tr>
</tbody>
</table>
In DBMSs, we are interested in developing a framework for performance tuning called Autonomic Workload Performance Predictor (AWPP) which detects the workload, predicts its performance and adapts the changes on evolution of the workload. A number of tools, techniques and models with respect to prediction and adaptation have been developed that provide solutions in each domain. Existing techniques have a number of limitations which have not been addressed and need to be resolved.

The performance of workload or queries depends on number of features, which is the measure of database activities. The prediction of different performance features, such as Workload size, Message sent or Message received, Disk I/O and others have not been predicted. These are important features and are helpful to the DBMS for resource allocation, scheduling, performance prediction and adaptation. There are number of problems in workload management as the existing methods are slow and can not predict and adapt accurately. Effective adaptation and accuracy are the other issues to be solved.

The early research conducted in workload management used traditional techniques to solve the problems. It applied different techniques for workload prediction and workload adaptation. There is a need of technique that solves the performance prediction and adaptation problems with a single approach. According to the best of our knowledge there exists no such solution for workload management that accurately predicts the performance features and adapts according to new trends of the workload. Before executing the workload number of problems arises in deciding, how can we manage the workload by monitoring it? Should we run the workload? Or delay it, and when should it run?
With the growing complexity and heterogeneity it is becoming difficult for humans to manage big data. There is a need of a system that has the capability to manage all the DBMS activities proactively. Before executing the workload, we do not know what type of workload is entering into the system and what will be its performance. If different type of workload enters into the system, what will happen and how should we handle this? There are other DBMS management issues which can not be handled without knowing about the workload before execution.

After studying the problem background, several issues need to be addressed. The main question of workload management is to detect the workload and extract the features of the workload. For workload we considered the following questions:

i) How to detect the workload?

ii) What features are to be extracted that represent the workload?

Secondly, to make the DBMS aware of workload performance, prediction will help in resource allocation, capacity planning, system sizing and improving the existing framework of workload performance prediction by applying machine learning techniques.

For workload and its required resources, following are considered:

1) What is the Execution time requirement for the workload?

2) What is the total workload size?

3) How many bytes are sent and received during execution?

4) How many Disk I/O are performed?

5) What will be the workload cost?

6) How much physical writing on the disk is done?

7) What will be the buffer size?
8) What is the concurrency factor?

Finally, exploring the techniques and approaches for solving workload management problems will be of further interest. Therefore, the questions are as follows:

a) When the behaviour of the workload changes, how can we adapt to these changes?

b) Which existing techniques and approaches give better results?

c) How can we apply techniques to solve workload management problems with respect to performance prediction and adaptation?
2.6 Problem Statement

In this research, main problems in the workload management are identified with respect to workload performance prediction and adaptation by number of causes and their effect. Cause and effect diagram or fish bone is created by Kaoru Ishikawa in 1960 to identify potential factors causing an overall effect.

The workload can not be well managed without knowing prior execution of the workload. DBAs are unable in fine performance tuning and performance of the workload is unpredictable prior execution. DBMS is not aware of what will be happen next with workload and therefore workload can not be controlled. The ultimate effect of these causes is that DBMS is unable to manage workload by itself. One of the workload management problems is workload performance prediction problem that is represented through cause and effect or fishbone - 1 diagram as shown in Figure 2.2.

The other workload management problem is due to uncertainty in the workload. The trend of workload may change any time that can not be monitored. When behaviour of the workload changes, how to handle it. DBMS itself has no adaptive behaviour and can not handle evolving behaviour of workload. The cause and effect for workload adaptation problem is shown in Figure 2.3 by fishbone 2 Diagram.
Cause and Effect Diagram
(Fish Bone) – 1

Figure 2.2: Fishbone diagram of workload performance prediction problem

Cause and Effect Diagram
(Fish Bone) - 2

Figure 2.3: Fishbone diagram of workload adaptation of new trends problem
2.7 Summary

This chapter discusses taxonomies that describe various approaches for workload management. Although the area of workload classification, workload performance prediction and workload adaptation have been investigated by researchers that demands a re-examination. In the literature review number of limitation in the exiting work are highlighted and the research problem is derived. The research problem is presented in the form of cause and effect diagrams.
CHAPTER 3

RESEARCH METHODOLOGY

The chapter presents the methodology followed in this research. This chapter provides details about how the research is designed and the proposed framework is developed. The phase wise discussion of all modules of the framework is provided and at the end experimental design of the framework is presented. System design and development helps to understand proposed solution according to objective of the research. After developing the proposed solution, data is prepared for experiments. Results are obtained by applying machine learning technique and case-based reasoning approach. Performance measures are selected to evaluate the performance of the proposed framework and results are validated.

3.1 Research Operational Framework

The research operational framework is shown in Figure 3.1 which represents the flow of research to be conducted. It consists of problem formulation then system design and development process and implementation and integration process.
Figure 3.1: Research Operational Framework

**Problem Formulation**
- Literature review and current issues of workload management
  - Existing systems analysis
  - Research problem

**System Design and Development Process**
- Analysis and Selection of input features of DBMS workload, AWPP framework
- Analysis and Selection of features of DBMS Workload that are to be predicted
  - Framework for prediction of performance metrics for DBMS workload
- Analysis and Design of DBMS workload for adaptation
  - Adaptation architecture for DBMS workload

**Implementation and Integration**
- Integrate modules of Autonomic Workload Performance Predictor (AWPP)
- Analysis and Comparison on experimental results based on the evaluation metrics
  - Summarize report results
3.2 Research Design and Development

Workload management plays an important role in database management system as well as in data warehouses. Workload management includes many factors which includes workload performance prediction and workload adaptation. If workload performance is predicted in advance then the performance of the DBMS can be enhanced and optimal results can be achieved.

3.2.1 Analysis and Selection of input features of DBMS workload

The identification of input features for the incoming workload is important. The workload is in the form of input data, the Workload Features Vector (WFV) is extracted from the given workload. The workload features that can best represent about the workload are Number of nested sub-queries, Number of selection predicates, Number of equality selection predicates, Number of non-equality selection predicates, Total number of join predicates, Number of sort columns, Number of aggregation columns, Records accessed, and Record used.

3.2.2 Analysis and Selection of performance features of DBMS workload

We identified some features for the prediction of workload that measure the performance of workload. The list of features is: Bytes_received, Bytes_sent, Key_read_requests, Key_reads, Key_write_requests, Key_writes, Query_cost, Workload size, Execution time, Innodb_dblwr_pages_written and Innodb_dblwr Writes.
3.3 Conceptual Framework of Autonomic Workload Performance Predictor

The conceptual framework of the proposed Autonomic Workload Performance Predictor (AWPP) is shown in Figure 3.2.

Figure 3.2: Conceptual Framework of Autonomic Workload Performance Predictor (AWPP)

3.4 Proposed Solution

We have proposed a framework for workload management i.e. Autonomic Workload Performance Predictor (AWPP) that solves the workload management problems.
➢ **Workload Performance Predictor Module**

This is main module of the AWPP framework. Workload predictor module predicts the performance of incoming workload prior execution. For the classified workload, similarity measure, between the workload feature vector and performance feature vector, is performed. The result is checked and compared. If the result is in the repository and match is found, we have the predicted result. If no match is found, analytical and statistical techniques are applied to find similarity between the workload features vector and performance features vector and the predicted results can be obtained.

➢ **Workload Adapter Module**

The generic architecture of workload autonomic computing is used in our proposed framework making it adaptive. The Adapter module of AWPP handles all the changes and behavior of the workload by adapting all the workload. When there are similar cases we can get result from the repository or CBR however, when we have different cases and no match is found, a new case is developed and it is adapted in the repository for future use.

### 3.5 Proposed Framework AWPP

The proposed framework for Autonomic Workload Performance Predictor (AWPP) is shown in Figure 3.3. The AWPP framework consists of three phases. Phase I is feature extraction phase which has two components; one is workload features extraction and the other is performance features extraction. Phase II is prediction component (i.e. Workload Performance Predictor), in which similarity measure of
workload features vector (WFV) and performance feature vector (PFV) is performed. Phase III is adaptation phase (Workload Adapter) in which workload is adapted when the data evolve.

![Figure 3.3: Autonomic Workload Performance Predictor (AWPP) - Block Diagram](image)

**Phase I – Workload Features Extraction Module**

The Workload Features Extraction is the first module of AWPP that extracts the workload features and performance features. The workload features represent the
incoming workload entered into the system. Performance features are the features that are to be predicted for workload performance.

**Phase II – Workload Performance Predictor Module**

The phase II is the predictor module that predicts the performance of the workload. There is a need of self-predicting systems that will help managing the workload dynamically and proactively. The predictor will predict different performance metrics such as Workload size, Execution time, Workload cost etc. By doing this the workload will be executed without any delay with maximum efficiency. The framework will predict workload run time before execution and simultaneously predict multiple performance metrics.

**Phase III – Workload Adapter Module**

Workload Adapter is the third module of the AWPP. It is the adaptation phase. When the workload evolves with the passage of time the autonomic system should have the ability to adapt the changing behavior of workload. It has the ability to absorb workload without any effect on the system performance. There is no need of Database administrator (DBA) or human to intervene the system.
Research Framework

The proposed research framework is shown in Table 3.1.

Table 3.1: Research Framework

<table>
<thead>
<tr>
<th>Phases</th>
<th>Activity</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase I:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workload Feature</td>
<td>1.1 Identification of workload features vector that can be used as an</td>
<td>Workload Features Vector (WFV)</td>
</tr>
<tr>
<td>Extraction</td>
<td>input for the system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.2 Extraction of workload features vector</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase II:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workload Performance</td>
<td>2.1 Identification of performance metrics which are to be predicted</td>
<td>Performance Features Vector (PFV)</td>
</tr>
<tr>
<td>Prediction</td>
<td>that have impact on the performance of workload</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.2 Applying techniques for prediction of multiple features, the</td>
<td>Workload Performance Predictor Module</td>
</tr>
<tr>
<td></td>
<td>technique that support multiple output of parameters</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.3 Building predictor for prediction of performance metrics for DBMS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>workload</td>
<td></td>
</tr>
<tr>
<td>Phase III:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workload Adaptation</td>
<td>3.1 Check when data evolve</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.2 Adapt the data for better prediction</td>
<td>Workload Adapter Module</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.6 Research Approach

Phase I: Workload Features Extraction

The first step is to extract the workload features which are identified. When workload enters into the DBMS, the values generated by the optimizer about the workload are saved. When the workload executes, it is parsed by the parser that generates syntax tree then optimizer generates the query execution plan (QEP). The initial data for training data set is prepared by taking the values of attributes from the QEP. Figure 3.4 shows processing steps of a query.

![Query Processing Steps](image)

Figure 3.4: Query Processing Steps

Phase II: Workload Performance Prediction

The second phase of the AWPP is prediction of performance of the workload. Next step is to predict the performance of the workload of two types that is DSS or OLTP. The DSS workload behaves in a different way as compared to OLTP workload.
The focus of the research is workload performance prediction therefore we are not interested how the workload is classified into two types. It is assumed that the workload is classified into DSS or OLTP. In past, research has been carried out on predicting one attribute or two only. Our interest is to predict more features that affect the performance of the workload.

- **Identify Performance metrics which are to be predicted that have impact on the performance of the workload**

  Based on the knowledge, literature review and the work done by the early researchers, we want to predict the performance of the workload that depends on number of features rather than only one or two. We are expecting to predict more features that definitely affect the DBMS performance and are further useful for good decision making and other objectives for predicting the performance of the workload.

- **Apply techniques for prediction of multiple features, the techniques that support multiple output of features**

  To meet the research objective a number of techniques were studied that support predicting multiple features. Different techniques are analyzed; few of these techniques are limited to predict one or two variable(s). There are some machine learning techniques which support multiple features and case-based reasoning approach. A number of machine learning techniques are applied to solve the problem and to test the proposed approach towards workload management.

- **Building Training Data Model**

  i. We used some standard workload and queries and also designed some other OLTP and DSS workload and executed these workloads.
ii. Training data (workload/ query features) are created by extracting the query features from query execution plan.

iii. Performance features are extracted from executed workload.

iv. The training data are divided into two classes.

When the workload is entered, it is parsed and QEP is generated by the optimizer before query execution. Workload features vector are extracted from each QEP which is generated by the query optimizer. The workload features consist of many nested sub-queries, selection predicates, equality selection predicates, non-equality selection predicates, total number of join predicates, many sort columns and aggregation columns.

The performance features are extracted that affect the performance of the workload which includes Workload size, Query cost, Records accessed, Records used, Bytes received, Bytes sent and Execution time. Initially these features are extracted and stored by executing different workloads. The similarity measure for new workload features and workload performance features is performed by applying machine learning techniques and CBR approach. The performance metrics for a workload are predicted which enables us for decision making about workload management, capacity planning and system sizing.

The framework has input data in the form of workload or queries. The input features consist of number of nested sub-queries, number of selection predicates, number of equality, number of non-equality selection predicates, total number of join predicates, number of sort columns and aggregation columns. The framework is tested through evaluation metrics. The predictor is implemented in Open Source database (MYSQL).
We evaluated this framework by comparing number of different machine learning techniques and CBR approach.

➢ *To build Predictor for prediction of Performance Metrics for DBMS Workload*

The predictor is developed that predicts the performance of the workload. The predictor predicts by finding correlation or similarity among the incoming workload features and the existing workload performance features.

**Phase III: Workload Adaptation**

The third phase of AWPP is workload adaptation. The behaviour of the workload is dynamic that can change any time and also workload evolves with the passage of time. Therefore adaptation of the workload to the system is required.

➢ *To check when data evolve*

After performance prediction of workload the next phase is workload adaptation. The behaviour of the workload may change any time and new cases are also needed to be incorporated so the behavior should be adaptive to accommodate new changes into the system.

➢ *Adapt data for better prediction*

Various machine learning techniques and CBR approach are applied for adaptation of the workload. The results are compared before adaptation and after adaptation. All the functionalities and autonomic characteristics are incorporated in the proposed AWPP framework.

**Data Set**

Transaction Processing Performance Council (TPC) defines transaction processing and database benchmarks and delivers trusted results to the industry. TPC
provides different benchmark data sets such as TPC-C, TPC-H, TPC-R, TPC-W, TPC-DS and postmark. TPC-H is used for the experiments. TPC-H is a Transaction Processing Council decision support benchmark that contains a suite of business oriented ad-hoc queries as well as data modifications that occur concurrently. The TPC-H database and TPC-H queries are standard and related to industry. These are used for decision support systems of huge volume of data and complex queries (TPC-H, 1999).

3.7 Machine Learning and Lazy Learning Techniques

The techniques used for different modules of the AWPP are SVM, CBR, J48, Bayes Net, Naïve Bayes and Simple Cart. The LibSVM is used for SVM and algorithms are designed for CBR approach. Other algorithms are implemented using Weka tool.

3.8 Implementation and Integration

To implement the proposed framework, Autonomic Workload Performance Predictor, first we need to develop an experimental setup. The framework has input data in the form of workload or queries. The framework is tested using artificially generated data that allows examining specific cases as well as arbitrary situations. The standard Transaction Processing Performance Council (TPC) benchmark workload as well as designed queries and workload is used for experiments. The AWPP is implemented in Open Source database (MYSQL). The framework is evaluated by comparing the performance of DBMS using Predictor and Adapter.

The Predictor finds similarity measure between workload feature vector and the performance feature vector and as a result the performance metrics of the workload are
obtained that include Workload size, Elapsed time, Records accessed, Record used, Message send, Message received and Disk I/O etc. The case-based reasoning (CBR) is applied to save all the different cases. When the workload entered is different from the cases stored in the CBR case-base and its performance presents a new case, the new case is stored in the CBR for future use. Once we get the performance of workload then we are able to decide whether this type of workload should run or not. We can also suspend workload for the time being for the queries which are capturing or demanding more resources which we do not have at that time. After developing the setup, different experiments are performed by using number of different machine learning techniques and case-based reasoning approach for number of different workloads. Those performance metrics are considered that are either used in literature and other new proposed features that affect the performance.

3.9 Experimental Setup

The experiments are executed on the machine running Windows 7 Ultimate 32-bit having Intel Core 2 Duo 3.0 GHZ with 280 GB hard disk, main memory of 2 GB and MySQL version 5.1. We use the same benchmark data set used by other researchers of the area namely TPC-H as per specification of the Transaction Processing Council (TPC) (Zewdu et al. 2009; Elnaffar et al, 2002; Ganapathi et al. 2009). Machine learning techniques such SVM, NN, J48 and Simple Cart and CBR approach is applied for the experiments. The data set used is same for all classifiers. The estimated database size is one gigabyte (1 GB). Data in tables of database are created through DBGen (TPC BenchmarkTM). DBGen is used to generate the TPC-H benchmark to populate the
workload or transactions executed by different threads from the client machines being executed one after another in a serial fashion.

We have selected the MySQL DBMS to execute different benchmark workload to validate the proposed AWPP framework. The TPC-H and TPC-C like queries are used as our representative workload. Various benchmark workloads of DSS and OLTP are taken for experimentation. The values of status variables are recorded which provide the information about a particular operation (MySQL Reference Manual). For the experiments, we executed over 100 TPC-H like workloads over MySQL database and recorded the values of status variables of our interest that meets the research objective. The training and testing data is prepared for the experiments purpose. We obtained WFV and PVF through classification by defining classes for all the performance features and developed cases for the case-base for reasoning. We developed the cases from the WFV as input and the corresponding PFV output class of the feature to which it belongs. In MySQL 5.1 there are 291 status variables which are used to distinguish different operations.

We studied various machine learning techniques and CBR approach and prepared training data for the experiments by executing the workloads and storing the features values. Different DSS and OLTP type of workload is executed on MySQL database and status variables are recoded. The workloads that consist of Select statements are mostly DSS workloads and Insert, Update and Delete and OLTP types of workload. We observed that DSS queries are complex and long and therefore take more time as compared to OLTP which executes in very small unit of time.
In this research, number of machine learning techniques and case-based reasoning (CBR) approach is applied and the results are compared with each other.

### 3.10 Performance Evaluation of the proposed framework

The effectiveness of the proposed AWPP framework is evaluated for performance prediction and adaptation through performance measures which are as follows:

1. Effectiveness
2. Accuracy
3. Significance
4. Adaptiveness

The effectiveness is calculated through f-measure. The performance measure such as precision, recall, f-measure and accuracy is considered by early researchers (Wu X and Banzhaf W, 2010) for evaluating an adaptive framework. In Table 3.2, the rows represent the predicted class and columns indicate the actual class. The class labels are positive and negative. The tp is the number of correct predictions that a sample is positive. The tn is the number of correct predictions that a sample is negative. The fp is the number of incorrect predictions that a sample is positive (negative sample being classified as positive) and fn is the number of incorrect predictions that a sample is negative (positive sample being classified as negative). Table 3.2 shows the relationship.
Table 3.2: Actual vs. Predicted class

<table>
<thead>
<tr>
<th>predicted class (observation)</th>
<th>actual class (expectation)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tp</td>
<td>fp</td>
</tr>
<tr>
<td></td>
<td>(true positive)</td>
<td>(false positive)</td>
</tr>
<tr>
<td></td>
<td>Correct prediction</td>
<td>Incorrect prediction</td>
</tr>
<tr>
<td></td>
<td>fn</td>
<td>tn</td>
</tr>
<tr>
<td></td>
<td>(false negative)</td>
<td>(true negative)</td>
</tr>
<tr>
<td></td>
<td>Incorrect prediction</td>
<td>Correct predictions</td>
</tr>
</tbody>
</table>

Precision is calculated through the formula as given in Equation 3.1

\[
Precision = \frac{tp}{tp + fp}
\]  

(3.1)

and Recall is calculated through the formula as given in Equation 3.2

\[
Recall = \frac{tp}{tp + fn}
\]  

(3.2)

We calculated F-measure with the formula as given in Equation 3.3

\[
F - measure = 2 \cdot \frac{precision \cdot recall}{precision + recall}
\]  

(3.3)

The Accuracy is calculated through the formula as given in Equation 3.4

\[
Accuracy = \frac{tp + tn}{tp + tn + fp + fn}
\]  

(3.4)

Significance is measured by performing paired t-test and adaptiveness is calculated through effectiveness and accuracy before adaptation and after adaptation.

3.11 Validation of Results

The results are validated by performing paired t-test and non-parametric procedures with post-hoc test.
3.12 Summary

This chapter presented the research methodology. Selection of workload features and workload performance features are discussed. A conceptual framework for autonomic workload performance predictor has been proposed. The research design and development process is discussed and implementation process is described. Additionally, the proposed framework Autonomic Workload Performance Predictor (AWPP) and its modules are presented. At the end, the data set that is used and the techniques that are applied for experiments are discussed. To measure the performance of the proposed framework, performance evaluation metrics are discussed.
CHAPTER 4

AUTONOMIC WORKLOAD PERFORMANCE PREDICTION FRAMEWORK

This chapter is dedicated to prediction of performance features of the workload. The objective is to predict the performance features that will make DBMS aware of itself. We can monitor the workload and its performance in this way and DBMS can optimize itself by taking advantage of the prediction making it autonomic. As DBMS is growing in size and data volume is increasing day by day, it is becoming difficult to handle the workload. If we know the workload performance by predicting it, we can solve workload management problems. With accurate performance prediction, we can decide about the problematic workload that will cause serious problem in future and will engaged the resources and also will not be completed within due time.

4.1 Workload Performance Prediction

Research has been conducted in predicting the workload performance attributes using different techniques. There are other attributes that need to be predicted that can solve workload management problems. Our Objective is to predict the performance of two types of workload either, DSS or OLTP. The performance can be obtained by predicting number of attributes that contribute in performance. The predicted
performance will be helpful in workload optimization, tuning, resource allocation, query scheduling, system sizing, capacity planning and adaptation. In this work our focus is workload prediction for resource allocation and query scheduling.

4.2 Workload Performance Features Selection for Prediction

Workload is an important entity in the databases and its proper management plays a major role that leads to performance optimization of the workload. Our objective is to predict the performance features that can be further used for resource allocation, scheduling, capacity planning and system sizing. MySQL status variables (291) have been studied for selection of performance features. We noticed that a number of features may be selected for prediction that would be helpful for the performance tuning.

We have also studied a number of features that can contribute in performance and helpful for resource allocation and for self-configuration. Keeping in view of resource allocation, scheduling, system sizing, and capacity planning requirement we found a number of variables that can help us in predicting performance of workload. With the help of these predicted features DBMS can optimize itself based on predicting the values of its resources and in this way it solves the configuration problems. The purpose of predicting these features is making DBMS autonomic.

The precision and recall have been very useful performance measures for information retrieval and extraction. The precision deals with substitution and insertion errors whereas recall deals with substitution and deletion errors. Because of our desire to have a single measure of performance that deals with all three types of errors simultaneously, the f-measure is used. To handle multi-class problem we used the approach of building a set of one-versus-one classifiers, where the target class is
determined by choosing the class that is selected by many classifiers (Duan and Keerthi 2005).

To meet the research objective of database performance tuning, we selected the performance features and calculated f-measure to observe effectiveness. The threshold i.e. f-measure > 80% is considered for selection of features of PFV. The list of all selected features along with purpose of selection is given in Table 4.1.

<table>
<thead>
<tr>
<th>Features</th>
<th>Purpose of Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key_read_requests</td>
<td>To determine the number of requests to read a key block from the cache</td>
</tr>
<tr>
<td>Key_write_requests</td>
<td>To determine the number of requests to write a key block to the cache.</td>
</tr>
<tr>
<td>Bytes_sent</td>
<td>To determine how many bytes are sent</td>
</tr>
<tr>
<td>Byte_received</td>
<td>To determine how many bytes are received</td>
</tr>
<tr>
<td>Key_reads</td>
<td>To determine when the value of the Key_reads is large, then key_buffer_size value is probably too small</td>
</tr>
<tr>
<td>Key_writes</td>
<td>To determine how many physical write on the disk is done</td>
</tr>
<tr>
<td>Workload cost</td>
<td>To determine the cost of workload</td>
</tr>
<tr>
<td>Workload size</td>
<td>To determine the available size</td>
</tr>
<tr>
<td>Execution time</td>
<td>To determine the time, CPU will be busy during this time</td>
</tr>
<tr>
<td>Innodb_dblwr_pages_written</td>
<td>To determine the Disk I/O</td>
</tr>
<tr>
<td>Innodb_dblwrWrites</td>
<td>To determine the Disk I/O</td>
</tr>
</tbody>
</table>
4.3 Workload Performance Predictor Module

This is main module of the AWPP Framework. Workload predictor methodology is shown in Figure 4.1. First of all, when workload enters into the system, workload features are extracted which represent the whole workload. The correlation through similarity measure is found between the workload features vector and performance features vector. The result is checked and compared, if result is in the knowledge-base and match is found, we have the predicted result. In other case, when no match is found, analytical and statistical techniques are applied to find similarity measure between the workload feature vector and performance feature vector to get the predicted results. The predicted results from the repository or CBR can be obtained for similar cases, however for different cases; a new case is developed and afterward adapted in the CBR for next use that is discussed in workload adapter module. Various machine learning techniques are applied to evaluate the work. In our framework the input is our workload feature vector and output is performance feature vector that we want to predict. The workload is analyzed on the following parameters.

- **Number of nested sub-queries**
- **Number of selection predicates**
- **Number of equality selection predicates**
- **Number of non-equality selection predicates**
- **Total number of join predicates**
- **Number of sort columns**
- **Number of aggregation columns**
Our predicted parameters will be the query performance features. These are the features which represent the performance of the workload. Eleven (11) selected performance features are as follows:

- **Bytes received**
- **Bytes sent**
- **Key read request**
- **Key read**
- **Key write request**
- **Key write**
- **Query cost**
- **Workload size**
- **Execution time**
- **Innodb_dblwr_pages_written**
- **Innodb_dblwr_writes**

The purpose of prediction is to have knowledge about performance of the workload before and during execution, which enables us for decision making about when or how to run the resource contention queries, long running queries and which queries need to be stopped or suspended or suspend for the time being. We use the knowledge available from the workload. Workload performance prediction helps us in workload management, system sizing and capacity planning.
Figure 4.1: Workload Predictor Module
4.4 Machine Learning and Lazy Learning Techniques

Machine learning techniques are considered to evaluate the proposed framework. Well known classification and prediction techniques such as Support Vector Machine (SVM), Bayes Net, Naïve Bayes, J48, Simple Cart and Case-based reasoning (CBR) are selected.

4.4.1 Workload Performance Prediction using SVM

Support Vector Machine (SVM) is a classification technique described in (Hsu et al. 2010; Cristianini et al. 2000). It is a computer algorithm that learns by example for assigning the labels to the corresponding objects. In classification, data are separated as training and testing data. Training data corresponds to the target value with some class and attributes. On the basis of training data, SVM generates a model that predicts the output for the testing data.

In our case we defined seven (7) classes for classification that refer to multi-class problem. SVM deals with binary (two) class and multi-class classification problems. There are a number of ways to handle SVM multi-class classification problem (Duan and Keerthi 2005). One of the common approaches is to build one-versus all classifiers (also known as “one-versus-rest”). Another approach is to build a set of one-versus-one classifiers, where the target class is determined by choosing the class that is selected by the most classifiers. We applied the latter approach in our work.

When a workload enters into the system, first of all workload features are extracted. We trained our data through SVM train and then modeled it. For the testing data we also have some set of data. When we have good training data we can take workload input from the testing data and from the training model we can predict the
performance of the workload. We performed 5-fold cross validation to find the best parameters for classification, and 100 different types of workloads were selected for our experiments. Figure 4.2 shows how SVM performed training and testing and then predicted the workload performance features of the workload.

For applying the SVM technique, we transformed the data into SVM format and scaled on the data. The training of the data was done with the selected features of workload performance and testing was performed.

The LibSVM tool developed by (Chang et al. 2011) was used to train the data on MATLAB. LibSVM is the integrated software for solving classification problems. The SVM model is trained and the kernel function is applied to the new input data that locates space in higher dimensional space to determine the class to which it belongs. There are
different kernels in Support Vector Machine (SVM) (Genton, M et al, 2001), the most commonly used kernels are linear, polynomial, RBF and Sigmoid,

\[
\text{Linear: } K(x_i, x_j) = (x_i^T x_j). \tag{4.1}
\]

\[
\text{Polynomial: } K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0. \tag{4.2}
\]

\[
\text{RBF: } K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \gamma > 0. \tag{4.3}
\]

\[
\text{Sigmoid: } K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r). \tag{4.4}
\]

For our experiments we used Radial Basis Function (RBF) for kernel function. Linear kernel handles only the linear relations. The reason for selecting RBF kernel was that this kernel could handle nonlinear relations between the class labels and attributes. RBF nonlinearly maps the samples into a higher dimensional space.

### 4.4.2 Workload Performance Prediction using CBR

This research investigates the issues of CBR in DBMS workload management. The most important issues addressed are described here.

1. **Case Representation**

   Case representation is important to explore the deeper knowledge of data for achieving objectives.

   We represent the case consisting of workload input features and performance features. The representation of Case I is shown in Figure 4.3 for the prediction of performance features in CBR.
Figure 4.3: Case I Representation for Prediction of performance features in CBR

2. Indexing

In the case-base, for the organization of cases, indices are selected and then searching and retrieving is done on those bases. For efficient retrieval of cases, we divided them into two types i.e. OLTP and DSS.

3. Case-base management

The cases are indexed in such a way that for efficiently retrieving the case(s), relevant case(s) are searched instead of whole case-base with respect to its type i.e. either search OLTP or DSS.
4. Retrieval

Retrieval is performed in such a way that we can reuse the cases from the case-base that solves the current problems. In our approach we manipulate the data in the form of matrices. For N workload we have input features vector $A$ and case-base metrics $B$.

$$A = \begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & x_{17} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} & x_{27} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} & x_{37} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{i1} & x_{i2} & x_{i3} & x_{i4} & x_{i5} & x_{i6} & x_{i7} \end{pmatrix}$$

$$B = \begin{pmatrix} y_{11} & y_{12} & y_{13} & y_{14} & y_{15} & y_{16} & y_{17} \\ y_{21} & y_{22} & y_{23} & y_{24} & y_{25} & y_{26} & y_{27} \\ y_{31} & y_{32} & y_{33} & y_{34} & y_{35} & y_{36} & y_{37} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{i1} & y_{i2} & y_{i3} & y_{i4} & y_{i5} & y_{i6} & y_{i7} \end{pmatrix}$$

We used the distance measure formulae to find similarity measure between the testing data and the training data for retrieval of CBR approach.

5. Reuse

The CBR Reuse phase is applied to devise the solution to the given problem. We developed algorithm to devise the solution. The similarity measures are used for reusing the case if the match case is exact match or $>80\%$, then the algorithm reuses the case and predict the output.
6. **Revise**

Revise case is used in the study when no exact solution or > 80% solution is found. It finds nearest match and updates the values according to the new input case and corresponding output and predicts the output.

7. **Adaptation**

The retrieved cases can be different from the solutions required. When a new case occurs, whose solution is not matched then the case is revised according to domain knowledge to solve the problem and retains in the case-base for future use.

The CBR taxonomy is shown in the Figure 4.4.
Figure 4.4: CBR Taxonomy
In our study, the CBR classification model that represents life cycle of CBR is shown in the Figure 4.5. The incoming workload will be the new case. The new case is matched with the existing knowledge base. If the match is found, the case is retrieved from the case-base and the suggested solution is reused for performance prediction. In other situation, when the case is not matching, the case needs revision and suggests a new solution. The suggested revised solution is confirmed and retained in the case-base or adapted in the case-base as a new case for future purposes. The new case is learned case.

Figure 4.5: CBR Classification Model
As our objective is to predict the performance of workload so the input is incoming workload. We have developed a CBR model to implement the proposed framework and a number of different cases are defined. According to CBR approach the proposed framework consists of cycle of four phases.

We build the training data in the form of cases for CBR. The workload features along with their performance features are stored in each case. When new workload enters in the system, the workload or query is analyzed in the query editor and the workload features are extracted. For predicting the performance of the incoming workload, the cases already stored in CBR are searched. The case-base is in the form of matrix where each row represents the case and column represents the feature. We used indexes to represents each case in the case-base. Initially, the case-base consists of training and testing data prepared from the workload. For the training and testing data, we ran some workload on MySQL and extracted performance features. When a new workload enters into the system, the workload feature vector (WFV) is extracted. The similarity measure of the extracted WFV is calculated with the training data (TD). The similarity measures such as Cosine distance, Euclidian distance and other distance measures are used for this purpose. All the four phases of CBR i.e. (Retrieve, Reuse, Revise, and Retain) is followed in this research. Through similarity measure, when the exact match is found within training data, the case is retrieved and that retrieved case solves the problem and it is reused. On the other side, when an exact match is not found, the case is revised according to the required solution. When the retrieved case is different from our solution, the case is revised and retained/stored in the case-base for future use.
The pseudo code for the workload performance prediction is shown in the Figure 4.6. When workload enters into the system, the workload features are extracted. The workload is categorized into two types that are OLTP and DSS and the performance of each type of workload is predicted. The executed workload is divided into training data and testing data. New incoming test workloads are matched with the train cases stored in the case-base through similarity measure. When the exact match is found it predicts the results. If no exact match is found, the measure of similarity is checked. If it is greater than 80% then reuse the case otherwise revise phase is applied and as a result a new case is developed for the solution for workload performance prediction and is then retained in the case-base for future prediction.

1- Extract the features of incoming workload
2- Find similarity measures between new workload and existing case-base in the Training data (TD)
3- If exact match found value is predicted
   a. Do step 3 for all features for the feature values to predict
4- Else find the percentage of similarity measure
   a. If high percentage (>80%) reuse the case
5- Performance features are predicted

Figure 4.6: The pseudo-code of Performance Prediction algorithm

In the AWPP framework, input is the workload and output is the performance features that we want to predict and adapt the workload on evolution.
We applied the CBR approach for our experiments and compare the results with the SVM and other machine learning techniques. The experiments, results and discussion are described in detail as follows. Table 4.2 represents an example of representing the new case. The Case 0 and Case 1 are the cases stored in the case-base and the New case is developed based on reasoning.

Table 4.2: An example case representing the new case for performance prediction

<table>
<thead>
<tr>
<th>Features</th>
<th>New case</th>
<th>Case 0</th>
<th>Case 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytes_received</td>
<td>4766</td>
<td>3009</td>
<td>1603</td>
</tr>
<tr>
<td>Bytes_sent</td>
<td>89856300</td>
<td>44334</td>
<td>22467190</td>
</tr>
<tr>
<td>Key_read_requests</td>
<td>66</td>
<td>102</td>
<td>6</td>
</tr>
<tr>
<td>Key_reads</td>
<td>26</td>
<td>31</td>
<td>5</td>
</tr>
<tr>
<td>Key_write_requests</td>
<td>11</td>
<td>36</td>
<td>3</td>
</tr>
<tr>
<td>Key_writes</td>
<td>9</td>
<td>36</td>
<td>1</td>
</tr>
<tr>
<td>Query_cost</td>
<td>2.399</td>
<td>2.399</td>
<td>2.399</td>
</tr>
<tr>
<td>Workload size</td>
<td>10</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>Execution time</td>
<td>0.2</td>
<td>0.05</td>
<td>0.3</td>
</tr>
<tr>
<td>Innodb_dblwr_pages_written</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Innodb_dblwrWrites</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

**Performance Analysis**

The proposed approach has been tested and validated. The case-base consists of training cases and testing cases. The performance analysis is performed through precision, recall, f-measure and accuracy. We used different similarity measures, the formulae are given in Table 4.3.
Table 4.3: Similarity Measure Formulae

<table>
<thead>
<tr>
<th>Similarity Measure Name</th>
<th>Similarity Measure Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jaccard Distance</td>
<td>( J(A, B) = 1 - J(A, B) = \frac{</td>
</tr>
<tr>
<td>Chebychev Distance</td>
<td>( D_{\text{Chebychev}}(p, q) := \max(</td>
</tr>
<tr>
<td>Correlation Distance</td>
<td>( d = 1 - r ) where ( r = Z(x).Z(y)/n )</td>
</tr>
<tr>
<td>Cosine Distance</td>
<td>( a \cdot b = |a||b| \cos \theta )</td>
</tr>
<tr>
<td>Euclidean Distance</td>
<td>( d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} )</td>
</tr>
<tr>
<td></td>
<td>( = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} )</td>
</tr>
<tr>
<td>Hamming Distance</td>
<td>( t\text{-test}^{\text{HAD}}(i, j) = \sum_{k=0}^{n-1} [y_{i,k} \neq y_{j,k}] )</td>
</tr>
<tr>
<td>Minkowski Distance</td>
<td>( D = \left( \sum_{i=0}^{n}</td>
</tr>
<tr>
<td>Spearman Distance</td>
<td>( \rho = \frac{\sum_{i=0}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{n}(x_i - \bar{x})^2 + \sum_{i=0}^{n}(y_i - \bar{y})^2}} )</td>
</tr>
</tbody>
</table>

**Experimental Setting**

For the implementation of the AWPP we decided to apply different machine learning techniques and CBR approach. We applied SVM and CBR techniques to predict the performance of workload and other machine learning techniques are also applied for comparison purpose. MySQL status variables are explored for a selection of performance features.
We performed a number of experiments by applying SVM and CBR techniques to predict the performance of workload. We calculated precision, recall and f-measure. The effectiveness is calculated through f-measure (Frequency measure) of workload performance features. We selected eleven features for which performance is to be predicted. For finding the optimization parameters in SVM, LibSVM is used with RBF kernel function. The 5-fold cross validation is performed to find the best optimization parameters. We find best parameter values as $C = 8$ and the gamma parameter of the RBF kernel ($"-G" = 0.0625$) and train the training data set. The experimental setting using the SVM is shown in Table 4.4.

<table>
<thead>
<tr>
<th>S #</th>
<th>Description</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tool Used</td>
<td>LibSVM</td>
</tr>
<tr>
<td>2</td>
<td>Database</td>
<td>MySQL</td>
</tr>
<tr>
<td>3</td>
<td>Cross Validation</td>
<td>5-fold CV</td>
</tr>
<tr>
<td>4</td>
<td>Kernel Function</td>
<td>RBF</td>
</tr>
<tr>
<td>5</td>
<td>Number of classes</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>Number of input feature</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>Number of predicted attributes</td>
<td>11</td>
</tr>
</tbody>
</table>

### 4.5 Experiments and Results

We performed a number of experiments for predicting performance features using SVM technique and CBR approach. We have eleven features for which performance is to be predicted. The Performance Feature Vector (PFV) is selected by computing f-measure in comparison with SVM and CBR. We calculated f-measure through CBR, SVM and
other machine learning techniques separately and then compare their results. The experiments using SVM and CBR are explained as below.

4.5.1 Experiments and Results using SVM

A number of experiments are performed on all eleven workload performance features. The performance measures such as precision, recall and f-measure are calculated. Some of the performance features such as Key writes, Workload size and Execution time are discussed below. In Key writes feature, the precision is 49% and recall is 60% whereas the f-measure is 53%. Workload size is another performance feature, the precision value is recorded as 77% and recall is 95% whereas the f-measure value is 81%. The Execution time is also important performance feature. We recorded precision 56%, recall 100% and f-measure 72%. The Figure 4.7 presents precision, recall and f-measure with respect to the Execution time.

![Figure 4.7: Precision, Recall and F-measure using SVM](image)

Figure 4.7: Precision, Recall and F-measure using SVM
4.5.2 Experiments and Results using CBR

We also performed the experiments for the eleven performance features using the proposed CBR approach. We are discussing here a few features such as Key writes, Workload size, Execution time. In the experiments Key writes features produced 90% precision and 100% recall whereas f-measure is 89%.

![Figure 4.8: Precision, Recall and F-measure using CBR](image)

Workload size is another performance feature, the precision is 98%, recall is 95% and f-measure is 87%. The precision, recall and f-measure for the Execution time are recorded as 84%, 98% and 83% respectively. The Figure 4.8 presents precision, recall and f-measure of Key writes, Workload size and Execution time.

The Figure 4.9 represents the comparison of f-measure of SVM and CBR. From the Figure 4.9 it can be observed that overall CBR produced better results as compared to SVM. The performance features such as Byte received, Key read, Query cost produced 100% f-measure which represents their effectiveness. Similarly other features have near 80% and above f-measure.
Figure: 4.9 Comparison of F-measure between SVM and CBR

The accuracy is calculated with CBR Cosine distance providing high similarity measures for prediction and it gives better results as compare to other distance measures. It is observed that CBR with Cosine distance measure is higher than SVM in predicting the performance features accurately as shown in the Figure 4.10. Most of the performance features have 70% or above accuracy in predicting the results. The performance features such as Bytes received has 92% accuracy, Key read has 76%, and Key writes have 88% accuracy.
We have achieved the second research objective by applying machine learning techniques and the case-based reasoning approach that learns from experiences. It can be seen that CBR performs well as compared to SVM and has produced better effectiveness and overall accuracy.

### 4.6 Comparison of CBR with other Machine Learning Approaches

The proposed approach is compared with other machine learning techniques. For machine learning techniques we used Weka implementation (Holmes et al. 1994; Was et al. 1994; Witten et al. 2005). Since finding the optimal parameters for a classifier can be a rather tedious process. We can find optimal parameters through different ways. Weka offers some ways of automating this process. We used cross-validation parameter selection for all the classifiers. Following parameters configuration are used for different classifiers through Weka.
For J48 algorithm, its confidence interval is used as base classifier within cross validation parameter selection. ConfidenceFactor -C 0.25, minNumObj -M 2, where the confidence factor is used for pruning (smaller values incur more pruning) and the minNumObj is the minimum number of instances per leaf. The settings used for Simple Cart algorithm is as follows: seed -S 1, minNumObj-M 2.0, numFoldsPruning-N 5, sizePer -C 1.0, where seed is the random number seed to be used. The minNumObj is the minimal number of observations at the terminal nodes. The numFoldsPruning is the number of folds in the internal cross-validation and sizePer is the percentage of the training set size. The Bayes Net configuration setting is searchAlgorithm K2-P1-SBAYES and estimator-A 0.5, where the searchAlgorithm is the select method used for searching network structures and the estimator selects estimator algorithm for finding the conditional probability tables of the Bayes Network.

We used 5-fold cross validation for training and testing data. The performance of the AWPP is measured in terms of precision, recall, f-measure and accuracy with other approaches. We applied different algorithms and performance of the CBR is observed a good approch. We calculated precision, recall, f-measure and accuracy for all the eleven features. Here we are representing few features such as Key writes, Workload size and Execution time. In Key writes feature the SVM and J48 performed well in terms of f-measure and accuracy, however, CBR outperform producing 89% f-measure and 86% accuracy as shown in the Table 4.5. The f-measure of Workload size through Bayes Net algorithm is 73% with 62% accuracy and f-measure of the SVM is 81% and accuracy 60%. The CBR has produced good f-measure i.e. 89% with 86% accuracy as shown in Table 4.6. In Execution time Naïve Bayes and J48 has 73% f-measure with 57% accuracy.
The SVM has 72% f-measure with 55% accuracy. However overall CBR has 83% f-measure and 71% accuracy as shown in Table 4.7.

Table 4.5: Comparison of Precision, Recall, F-measure & Accuracy for Key writes using different algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Cart</td>
<td>53%</td>
<td>71%</td>
<td>56%</td>
<td>53%</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>26%</td>
<td>35%</td>
<td>28%</td>
<td>47%</td>
</tr>
<tr>
<td>Bayes Net</td>
<td>43%</td>
<td>46%</td>
<td>38%</td>
<td>53%</td>
</tr>
<tr>
<td>J48</td>
<td>59%</td>
<td>70%</td>
<td>61%</td>
<td>57%</td>
</tr>
<tr>
<td>SVM</td>
<td>49%</td>
<td>60%</td>
<td>53%</td>
<td>60%</td>
</tr>
<tr>
<td>CBR</td>
<td><strong>91%</strong></td>
<td><strong>100%</strong></td>
<td><strong>89%</strong></td>
<td><strong>86%</strong></td>
</tr>
</tbody>
</table>

Table 4.6: Comparison of Precision, Recall, F-measure & Accuracy for Workload size using different algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Cart</td>
<td>57%</td>
<td>70%</td>
<td>60%</td>
<td>63%</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>59%</td>
<td>94%</td>
<td>71%</td>
<td>60%</td>
</tr>
<tr>
<td>Bayes Net</td>
<td>75%</td>
<td>79%</td>
<td>73%</td>
<td>62%</td>
</tr>
<tr>
<td>J48</td>
<td>56%</td>
<td>67%</td>
<td>59%</td>
<td>60%</td>
</tr>
<tr>
<td>SVM</td>
<td>77%</td>
<td>95%</td>
<td>81%</td>
<td>69%</td>
</tr>
<tr>
<td>CBR</td>
<td><strong>98%</strong></td>
<td><strong>95%</strong></td>
<td><strong>87%</strong></td>
<td><strong>83%</strong></td>
</tr>
</tbody>
</table>

Table 4.7: Comparison of Precision, Recall, F-measure & Accuracy for Execution time using different algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Cart</td>
<td>57%</td>
<td>59%</td>
<td>54%</td>
<td>58%</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>57%</td>
<td>100%</td>
<td>73%</td>
<td>57%</td>
</tr>
<tr>
<td>Bayes Net</td>
<td>61%</td>
<td>58%</td>
<td>53%</td>
<td>59%</td>
</tr>
<tr>
<td>J48</td>
<td>57%</td>
<td>100%</td>
<td>73%</td>
<td>57%</td>
</tr>
<tr>
<td>SVM</td>
<td>56%</td>
<td>100%</td>
<td>72%</td>
<td>55%</td>
</tr>
<tr>
<td>CBR</td>
<td><strong>84%</strong></td>
<td><strong>98%</strong></td>
<td><strong>83%</strong></td>
<td><strong>71%</strong></td>
</tr>
</tbody>
</table>

In the CBR model different similarity measures have been used to obtain good results. We considered Jaccard distance, Cosine distance, Chebychev distance,
Correlation distance, Euclidean distance, Hamming distance, Minkowski distance and Spearman distance in our experiments. The precision, recall, f-measure and accuracy is calculated for each similarity measure and the Table 4.8 shows performance features such as Key writes, Workload size and Execution time and their comparison.

Table 4.9 presents a number of different distance measures which have produced good results for Key writes feature and with Cosine distance the f-measure is good i.e. 89% with 86% accuracy. In Workload size, Spearman distance has 60% f-measure and 81% accuracy whereas with Cosine distance the results are better with f-measure 87% and accuracy 83% as shown in the Table 4.9. For the performance feature Execution time, Chebychev distance produced 70% f-measure with 70% accuracy; however, Cosine distance has produced 83% f-measure and 71% accuracy as shown in the Table 4.10.
Table 4.8: Comparison of Precision, Recall, F-measure & Accuracy for Key writes using different similarity measures

<table>
<thead>
<tr>
<th>Similarity Measure</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jaccard Distance</td>
<td>73%</td>
<td>60%</td>
<td>59%</td>
<td>85%</td>
</tr>
<tr>
<td>Cosine Distance</td>
<td>91%</td>
<td>100%</td>
<td>89%</td>
<td>86%</td>
</tr>
<tr>
<td>Chebychev Distance</td>
<td>44%</td>
<td>41%</td>
<td>36%</td>
<td>78%</td>
</tr>
<tr>
<td>Correlation Distance</td>
<td>53%</td>
<td>54%</td>
<td>51%</td>
<td>84%</td>
</tr>
<tr>
<td>Euclidean Distance</td>
<td>53%</td>
<td>53%</td>
<td>51%</td>
<td>85%</td>
</tr>
<tr>
<td>Hamming Distance</td>
<td>62%</td>
<td>66%</td>
<td>60%</td>
<td>83%</td>
</tr>
<tr>
<td>Minkowski Distance</td>
<td>54%</td>
<td>55%</td>
<td>52%</td>
<td>85%</td>
</tr>
<tr>
<td>Spearman Distance</td>
<td>46%</td>
<td>59%</td>
<td>51%</td>
<td>83%</td>
</tr>
</tbody>
</table>

Table 4.9: Comparison of Precision, Recall, F-measure & Accuracy for Workload size using different similarity measures

<table>
<thead>
<tr>
<th>Similarity Measure</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jaccard Distance</td>
<td>76%</td>
<td>56%</td>
<td>58%</td>
<td>80%</td>
</tr>
<tr>
<td>Cosine Distance</td>
<td>98%</td>
<td>95%</td>
<td>87%</td>
<td>83%</td>
</tr>
<tr>
<td>Chebychev Distance</td>
<td>37%</td>
<td>44%</td>
<td>38%</td>
<td>78%</td>
</tr>
<tr>
<td>Correlation Distance</td>
<td>74%</td>
<td>56%</td>
<td>58%</td>
<td>80%</td>
</tr>
<tr>
<td>Euclidean Distance</td>
<td>37%</td>
<td>40%</td>
<td>37%</td>
<td>76%</td>
</tr>
<tr>
<td>Hamming Distance</td>
<td>37%</td>
<td>40%</td>
<td>37%</td>
<td>80%</td>
</tr>
<tr>
<td>Minkowski Distance</td>
<td>52%</td>
<td>40%</td>
<td>39%</td>
<td>79%</td>
</tr>
<tr>
<td>Spearman Distance</td>
<td>76%</td>
<td>58%</td>
<td>60%</td>
<td>81%</td>
</tr>
</tbody>
</table>

Table 4.10: Comparison of Precision, Recall, F-measure & Accuracy for Execution time using different similarity measures

<table>
<thead>
<tr>
<th>Similarity Measure</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jaccard Distance</td>
<td>40%</td>
<td>52%</td>
<td>28%</td>
<td>70%</td>
</tr>
<tr>
<td>Cosine Distance</td>
<td>84%</td>
<td>98%</td>
<td>83%</td>
<td>71%</td>
</tr>
<tr>
<td>Chebychev Distance</td>
<td>58%</td>
<td>88%</td>
<td>70%</td>
<td>70%</td>
</tr>
<tr>
<td>Correlation Distance</td>
<td>42%</td>
<td>49%</td>
<td>41%</td>
<td>71%</td>
</tr>
<tr>
<td>Euclidean Distance</td>
<td>42%</td>
<td>49%</td>
<td>41%</td>
<td>71%</td>
</tr>
<tr>
<td>Hamming Distance</td>
<td>39%</td>
<td>45%</td>
<td>39%</td>
<td>69%</td>
</tr>
<tr>
<td>Minkowski Distance</td>
<td>42%</td>
<td>49%</td>
<td>41%</td>
<td>71%</td>
</tr>
<tr>
<td>Spearman Distance</td>
<td>39%</td>
<td>47%</td>
<td>40%</td>
<td>70%</td>
</tr>
</tbody>
</table>
4.7 Validation of Results; T-Test

We conducted the paired t-test to find the correlation and measure the significance of classification results between CBR and machine learning techniques such as SVM, Simple Cart, Naive Bayes, Bayes Net and J48. The Statistical Package for Social Sciences (SPSS) is used for performing paired t-test to observe the significance of results. It is observed that the significance value is less than 0.05 with the confidence interval of difference 95%. The results are effective and accurate as well as significant. The paired t-test results are shown in the Table 4.11.

Table 4.11: Paired Samples T-Test for performance prediction

<table>
<thead>
<tr>
<th></th>
<th>Paired Differences</th>
<th>95% Confidence Interval of the Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std. Deviation</td>
</tr>
<tr>
<td>CBR - Simple Cart</td>
<td>.27461</td>
<td>.39528</td>
</tr>
<tr>
<td>CBR - Naive Bayes</td>
<td>.28403</td>
<td>.39499</td>
</tr>
<tr>
<td>CBR – Bayes Net</td>
<td>.28947</td>
<td>.38664</td>
</tr>
<tr>
<td>CBR - J48</td>
<td>.30323</td>
<td>.40361</td>
</tr>
</tbody>
</table>

The Table 4.11 represents the significance of results using CBR with SVM, Simple Cart, Naïve Bayes, Bayes Net and J48. The framework produced effective, accurate as well as significant results for performance prediction.
4.8 Summary

This chapter presents the proposed framework for prediction of performance features of the workload. The proposed framework using CBR approach predicts well and performs less number of similarity measures as we already know the type of workload. It finds similarity to the specific workload type making efficient search rather than searching the whole case-base. The workload performance prediction results are tested with other well-known classifier such as J48, Naïve Bayes, and SVM etc. and compared it with the proposed CBR. The results show that proposed CBR approach outperforms in comparison with other algorithms with respect precision, recall, f-measure and accuracy. A number of different similarity measures such Jaccard, Cosine etc. has been applied and compared with respect to precision, recall, f-measure and accuracy. The experiments are performed and results are validated at the end of chapter which represents effectiveness, accuracy and significance of the proposed framework.
CHAPTER 5

ENHANCED WORKLOAD ADAPTIVE FRAMEWORK

Performance prediction of the workload has been discussed in chapter 4. When behavior of workload is changed, system should be capable of handling new workload trends itself without any human intervention. This chapter is dedicated to the third module of the framework. We designed the architecture for the workload adaptation called workload adapter module.

5.1 Workload Adaptation

The adaptation is an important consideration for workload management. A system can have versatile type of workload and the behaviour of the workload may change with the passage of time. Self-adaptation of workload can enhance the performance of the system. Autonomic systems have the ability to adapt the workload with changing behaviour of workload. There is a need to develop autonomic architecture that can adapt new trends of the workload.

The proposed architecture of the workload adapter is based on generic AC architecture (Koehler et al. 2003). AC system consists of five entities; three entities are Negotiation, Observation, and Execution that communicate with the external
environment by communicating its own requirements to the other AC systems it is aware of. The other two, Deliberation and Failure Recovery are internal to the system. The Negotiation component has a two-way interaction, that is, it receives queries from the environment. After processing, it provides output to the environment which is basically the main purpose of this component. Execution is the second component which has one way interaction with the environment. The behavior provided through Negotiation component and determined by Deliberation component is executed by Execution component. The Observation component has also one way interaction (taking status information) with the environment. It actually observes the effect of execution done by Execution component. Deliberation component is responsible for self-optimization and self-adaptation. It evaluates new behavior for the autonomic systems. The Failure Recovery is responsible for self-healing and self-protection which reduces complexity and provides robustness in the system.

In our research we focused on all the three external components i.e. Negotiation, Execution and Observation that provides direct interaction. The Deliberation component is partially considered and we are not dealing with Failure Recovery component.

The adaptation is an important aspect of CBR approach. CBR without adaptation is just a retrieval system. The adaptation is necessary, as the essence of CBR is the reasoning and without this there is no reasoning for producing new solution. A CBR system with adaptation capabilities is called full-fledged CBR system.
5.2 Enhanced Workload Adaptation Architecture

In the AWPP framework, the third important module is the workload adapter. The architecture of the workload adapter module is presented in Figure 5.1.

![Workload Adapter Module](image)

Figure 5.1: Workload Adapter Module

The Adapter module of AWPP handles all the workload changes and behaviour of the workload by adapting all the workload. The incoming workload in the form of case enters into the system. The workload acts as Managed Element and the Sensor and Effector on the Workload Monitoring self-inspect the workload and retrieve the case. The workload is analyzed on the basis of cases stored in the CBR. On finding a match, the case is reused, otherwise it is revised and adapted in the CBR for future prediction. We have training data stored in the CBR case-base, which have workload type and its
performance. While adapting the workload, it self-optimizes the workload and a
deliberation component is responsible for adapting the workload. The Negotiation
component interacts with the external environment for entertaining the new type of
workload. After revise, the case is stored into the CBR. Based on generic architecture the
Workload Adapter has three external components such as Workload Monitoring,
Adaptation, and Execute. These components interact with the external environment. The
two components of the Workload Adapter that are Analyze ad Match Case are internal to
the system.

5.3 Machine Learning and Lazy Learning Techniques

We considered well known machine learning technique such as SVM and our
proposed CBR approach for workload adaptation. The detail of workload adaptation
using SVM and CBR is as follows.

5.3.1 Workload Adaptation using SVM

We performed experiment for workload performance prediction using SVM. The
LibSVM tool is used for SVM experiments. The SVM does not have ability to adapt the
workload.

5.3.2 Workload Adaptation using CBR

The workload adapter architecture has been developed for the adaptation of the
workload. This is an important part of AWPP. When the workload is changed or the
behaviour of the workload is changing, the proposed system detects the change as well as
has the capability to accept the changes and behave according to new changes.
A number of different types of workload are executed and its features are extracted. We categorized the workload OLTP and DSS so that we can predict and adapt the performance in the training and testing workload. For workload adaptation we performed the same procedure for similarity measure as in performance prediction and threshold is set as 80%. When the similarity measure is less than threshold value the case is revised and as a result new case is developed for the solution and is adapted for the future use.

The pseudo-code of Workload Adaptation algorithm in CBR is shown in the Figure 5.2.

Algorithm: Workload Adaptation in CBR

1- Extract the query features of incoming workload
2- Find similarity measures between new workload and existing case-base in the Training data (TD)
3- If exact match found value is predicted
   a. Do step 3 for all features for the feature values to predict
4- Else find the percentage of similarity measure
   a. If high percentage ( > 80%) reuse the case
   b. Else revise the case
5- Adapt the case after revise
6- Extract contents of new case
7- Remove old cases
8- Determine indexes and generalize them
9- Adjust indexes, update case-base
10- Retain the new Case in case-base for next use

Figure 5.2: The pseudo-code of Adaptation algorithm in CBR
5.4 Results and Discussion

We performed number of experiments for workload adaptation using SVM and CBR techniques. LibSVM tool was used for experiments. The SVM is not adaptive. We tested the workload adaptation through case-based reasoning approach. We performed a number of experiments to observe the ability to adapt the workload. We ran training and testing workloads. A number of test workloads were executed and calculated f-measure before workload adaptation and recorded the results for CBR and SVM; we then calculated the f-measure after the workload adaptation for CBR, as SVM is not adaptive. The adaptation results for two iterations are reported. The Figure 5.3 depicts the f-measure before and after workload adaptation. It is observed that SVM have 68% effectiveness before and after adaptation. However CBR produced better results that are 79% before adaptation and 100% effectiveness after adaptation.

![Figure 5.3: F- measure before and after Adaptation using SVM and CBR](image-url)
We calculated the accuracy of workload before adaptation and after adaptation and observed that after adaptation the accuracy is increased for CBR. As the SVM is not adaptive, the results were observed same before and after adaptation. Figure 5.4 represents the accuracy before and after adaptation.

![Figure 5.4: Accuracy before and after Adaptation via SVM and CBR](image)

In CBR, the self-adaptation is performed on the basis of reasoning. Initial training is to be provided in the AWPP framework. After that, it works based on reasoning. As we knew the type of workload, we could search or place the case to the corresponding class, thus reducing the search time by searching the particular class instead of the whole case-base. We performed a number of experiments in order to observe adaptability. We also calculated the accuracy of adaptation in experiments. The SVM gave 58% accuracy before and after adaptation. The CBR produced accuracy 68% before adaptation and 83% after adaptation. We found that, after adaptation, the accuracy was higher than before adaptation, that is, after adaptation 15% increase in accuracy was observed. It can be seen
that the CBR is effective in workload adaptation. As SVM is not adaptive so no adaptation can be performed through SVM.

The CBR is not dependent on example, on the basis of which the accuracy can be achieved. It initially requires the training data then performs on the basis of reasoning. Therefore, based on a number of experiments performed on adaptation, we observed that CBR is effective as well as accurate in adaptability.

5.5 Validation of Results; Non-parametric procedures with post-hoc tests

In an experimental analysis, when a new method is being proposed, performing all pairwise comparison may be useful (Garcia, et. al, 2008, 2010). The purpose is to detect significant pairwise differences among all the selected classifiers. We performed non-parametric procedures for the post-hoc tests for workload performance features vector. The non-parametric procedure such as Friedman and advanced post-hoc tests such as Holm, Nemenyi, Shaffer and Bergman tests are considered for test experiments. As non-parametric procedures do not assume normal distributions or homogeneity of variance, so they are safer than parametric tests. They are also stronger than the other tests studied. The non-parametric test is more often rejects the null-hypothesis than the parametric one. We compared the six classifiers including the proposed CBR based classifier on the selected eleven workload performance features, recording the number of equivalence hypothesis rejected and APVs. We follow a similar method used by Garcia et al. 2008.
The following Table 5.1 presents the workload performance features with a comparison of six classifiers. The KEEL software (Alcala-Fdez et al. 2008) is used for the experiments that allows the input of data in CSV format and obtains result as output a LATEX document.

Table 5.1: Workload Performance 11 Features and 6 Classifiers

<table>
<thead>
<tr>
<th>Features/Classifiers</th>
<th>CBR</th>
<th>SVM</th>
<th>Simple Cart</th>
<th>Naive Bayes</th>
<th>Bayes Net</th>
<th>J48</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte received</td>
<td>1</td>
<td>1</td>
<td>0.3793</td>
<td>0.375</td>
<td>0.378</td>
<td>0.3968</td>
</tr>
<tr>
<td>Byte sent</td>
<td>0.2120</td>
<td>0.35</td>
<td>0.453</td>
<td>0.346</td>
<td>0.5115</td>
<td>0.5413</td>
</tr>
<tr>
<td>Key read request</td>
<td>0.1137</td>
<td>0.24</td>
<td>0.5973</td>
<td>0.4943</td>
<td>0.5677</td>
<td>0.646</td>
</tr>
<tr>
<td>Key read</td>
<td>0.2400</td>
<td>0.28</td>
<td>0.3793</td>
<td>0.375</td>
<td>0.378</td>
<td>0.3968</td>
</tr>
<tr>
<td>Key write request</td>
<td>0.1356</td>
<td>0.31</td>
<td>0.3793</td>
<td>0.375</td>
<td>0.378</td>
<td>0.3968</td>
</tr>
<tr>
<td>Key write</td>
<td>0.5918</td>
<td>0.66</td>
<td>0.6577</td>
<td>0.685</td>
<td>0.3755</td>
<td>0.6053</td>
</tr>
<tr>
<td>Query cost</td>
<td>0.3046</td>
<td>0.36</td>
<td>0.858</td>
<td>0.858</td>
<td>0.858</td>
<td>0.858</td>
</tr>
<tr>
<td>Workload size</td>
<td>0.5789</td>
<td>0.68</td>
<td>0.6038</td>
<td>0.714</td>
<td>0.733</td>
<td>0.5925</td>
</tr>
<tr>
<td>Execution time</td>
<td>0.2064</td>
<td>0.24</td>
<td>0.537</td>
<td>0.726</td>
<td>0.5285</td>
<td>0.726</td>
</tr>
<tr>
<td>Innodb_dblwr_pages_written</td>
<td>0.1714</td>
<td>0.24</td>
<td>0.951</td>
<td>0.951</td>
<td>0.951</td>
<td>0.951</td>
</tr>
<tr>
<td>Innodb_dblwr_writes</td>
<td>0.1714</td>
<td>0.17</td>
<td>0.951</td>
<td>0.951</td>
<td>0.951</td>
<td>0.951</td>
</tr>
</tbody>
</table>

The average ranks are obtained for comparing the classifiers. The Average ranks by themselves provide a fair comparison of the algorithms [47]. The Friedman test [48, 49] is a non-parametric procedure that ranks the algorithms for the data set provided, the best performing algorithm getting the rank of 1, the second best rank 2 and so on, as shown in Table 5.2. The average ranks obtained by applying the Friedman procedure are

Table 5.2: Algorithm and Ranking by Friedman procedure

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBR</td>
<td>1.5909</td>
</tr>
<tr>
<td>SVM</td>
<td>2.7727</td>
</tr>
<tr>
<td>Simple Cart</td>
<td>4.2273</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>3.8182</td>
</tr>
<tr>
<td>Bayes Net</td>
<td>3.8636</td>
</tr>
<tr>
<td>J48</td>
<td>4.7273</td>
</tr>
</tbody>
</table>
From the table the Best, Middle and Worst classifier can be categorized as follows:-

- Best classifier: CBR
- Middle classifiers: SVM and Naive Bayes and Bayes Net
- Worst classifiers: J48 and Simple Cart

We applied Friedman test $n \times n$ and post-hoc methods that incorporate more information about the hypotheses to be tested in $n \times n$ comparisons and there exists a logical relationship among them. We computed the corresponding probability i.e. $p$-values. We obtained the confidence level of alpha ($p$-value for alpha = 0.05 and alpha = 0.01). The $p$-values for alpha = 0.05 for Holm and Shaffer for PFV are shown in Table 5.3 that presents the family of hypotheses with their $p$-value and the adjustment of alpha by Holm’s and Shaffer’s procedures. The $z$ value is used to find the corresponding probability ($p$-value) from the table of normal distribution, which is then compared with an appropriate level of significance alpha. To compensate for multiple comparisons the value of alpha is adjusted. The null-hypothesis developed and is tested that all the classifiers perform the same and observed differences are merely random. The total variability is divided into the variability between the classifiers, variability between the data sets and the error variability. If the between-classifiers variability is significantly larger than the error variability, the null-hypothesis is rejected and there exists some differences between the classifiers and one can proceed with a post-hoc test to find out which classifiers actually differ.

Regarding PFV, for alpha = 0.05 the Holm's procedure rejects those hypotheses that have an unadjusted $p$-value 0.0038. The Shaffer's procedure rejects those hypotheses that have an unadjusted $p$-value 0.0033 and Nemenyi's procedure rejects those
hypotheses that have an unadjusted p-value 0.0033. The Bergmann's procedure rejects these hypotheses (i=12-15) that are CBR vs. Simple Cart, CBR vs. Naive Bayes, CBR vs. Bayes Net, CBR vs. J48. However for alpha = 0.01 the Holm's procedure rejects those hypotheses that have an unadjusted p-value 0.0090. The Shaffer's procedure rejects those hypotheses that have an unadjusted p-value 0.0067. The Bergmann's procedure rejects these hypotheses: CBR vs. Simple Cart, CBR vs. Naive Bayes, CBR vs. Bayes Net, and CBR vs. J48.

Table 5.3: The P-values for alpha = 0.05 for Holm and Shaffer

<table>
<thead>
<tr>
<th></th>
<th>hypothesis</th>
<th>( z = \frac{(R_0 - R_i)}{SE} )</th>
<th>p-value</th>
<th>Holm</th>
<th>Shaffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CBR vs. J48</td>
<td>3.9316</td>
<td>0.0001</td>
<td>0.0033</td>
<td>0.0033</td>
</tr>
<tr>
<td>2</td>
<td>CBR vs. Simple Cart</td>
<td>3.3049</td>
<td>0.0010</td>
<td>0.0036</td>
<td>0.005</td>
</tr>
<tr>
<td>3</td>
<td>CBR vs. Bayes Net</td>
<td>2.8490</td>
<td>0.0044</td>
<td>0.0038</td>
<td>0.005</td>
</tr>
<tr>
<td>4</td>
<td>CBR vs. Naive Bayes</td>
<td>2.7920</td>
<td>0.0052</td>
<td>0.0042</td>
<td>0.005</td>
</tr>
<tr>
<td>5</td>
<td>SVM vs. J48</td>
<td>2.4502</td>
<td>0.0143</td>
<td>0.0045</td>
<td>0.005</td>
</tr>
<tr>
<td>6</td>
<td>SVM vs. Simple Cart</td>
<td>1.8234</td>
<td>0.0682</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>7</td>
<td>CBR vs. SVM</td>
<td>1.4815</td>
<td>0.1385</td>
<td>0.0056</td>
<td>0.0056</td>
</tr>
<tr>
<td>8</td>
<td>SVM vs. Bayes Net</td>
<td>1.3675</td>
<td>0.1715</td>
<td>0.0063</td>
<td>0.0063</td>
</tr>
<tr>
<td>9</td>
<td>SVM vs. Naive Bayes</td>
<td>1.3105</td>
<td>0.1900</td>
<td>0.0071</td>
<td>0.0071</td>
</tr>
<tr>
<td>10</td>
<td>Naive Bayes vs. J41</td>
<td>1.1396</td>
<td>0.2545</td>
<td>0.0083</td>
<td>0.0083</td>
</tr>
<tr>
<td>11</td>
<td>Bayes Net vs. J48</td>
<td>1.0826</td>
<td>0.2790</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>12</td>
<td>Simple Cart vs. J48</td>
<td>0.6268</td>
<td>0.5308</td>
<td>0.0125</td>
<td>0.0125</td>
</tr>
<tr>
<td>13</td>
<td>Simple Cart vs. Naive Bayes</td>
<td>0.5128</td>
<td>0.6081</td>
<td>0.0167</td>
<td>0.0167</td>
</tr>
<tr>
<td>14</td>
<td>Simple Cart vs. Bayes Net</td>
<td>0.4558</td>
<td>0.6485</td>
<td>0.025</td>
<td>0.025</td>
</tr>
<tr>
<td>15</td>
<td>Naive Bayes vs. Bayes Net</td>
<td>0.0569</td>
<td>0.9546</td>
<td>0.05</td>
<td>0.05</td>
</tr>
</tbody>
</table>

The p-value provides information about whether a statistical hypothesis test is significant or not, the smaller the p-value, the stronger the evidence against the null hypothesis. A p-value reflects the probability error of a certain comparison within a multiple comparison leaving the remaining comparisons that belongs to the family. However the Adjusted p-values (APVs) consider that numbers of tests are conducted and the APV can be compared directly with any chosen significance level. Table 4.15 shows
the results in the final form of APVs obtained through Holm, Nemenyi, Shaffer and Bergman tests for the example considered in this section for PFV. The difference of power among the test procedures can be observed in this example. Table 5.4 provides the information about the state of retention or rejection of any hypothesis, by comparing its associated adjusted p-value (APV) with the level of significance previously fixed.

Table 5.4 APVs obtained in example by Holm, Nemenyi, Shaffer and Bergman

<table>
<thead>
<tr>
<th>i</th>
<th>hypothesis</th>
<th>p-value</th>
<th>APV-Holm</th>
<th>APV-Nemenyi</th>
<th>APV-Shaffer</th>
<th>APV-Bergman</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CBR vs .J48</td>
<td>0.0001</td>
<td>0.0013</td>
<td>0.0013</td>
<td>0.0013</td>
<td>0.0013</td>
</tr>
<tr>
<td>2</td>
<td>CBR vs .Simple Cart</td>
<td>0.0010</td>
<td>0.0133</td>
<td>0.0143</td>
<td>0.0095</td>
<td>0.0095</td>
</tr>
<tr>
<td>3</td>
<td>CBR vs .Bayes Net</td>
<td>0.0044</td>
<td>0.0570</td>
<td>0.0658</td>
<td>0.0439</td>
<td>0.0307</td>
</tr>
<tr>
<td>4</td>
<td>CBR vs .Naive Bayes</td>
<td>0.0052</td>
<td>0.0629</td>
<td>0.0786</td>
<td>0.0524</td>
<td>0.0367</td>
</tr>
<tr>
<td>5</td>
<td>SVM vs .J48</td>
<td>0.0143</td>
<td>0.1571</td>
<td>0.2142</td>
<td>0.1428</td>
<td>0.1428</td>
</tr>
<tr>
<td>6</td>
<td>SVM vs .Simple Cart</td>
<td>0.0682</td>
<td>0.6825</td>
<td>1.0237</td>
<td>0.6825</td>
<td>0.4095</td>
</tr>
<tr>
<td>7</td>
<td>CBR vs .SVM</td>
<td>0.1385</td>
<td>1.2463</td>
<td>2.0772</td>
<td>0.9693</td>
<td>0.9693</td>
</tr>
<tr>
<td>8</td>
<td>SVM vs .Bayes Net</td>
<td>0.1715</td>
<td>1.3717</td>
<td>2.5719</td>
<td>1.2002</td>
<td>0.9693</td>
</tr>
<tr>
<td>9</td>
<td>SVM vs .Naive Bayes</td>
<td>0.1900</td>
<td>1.3717</td>
<td>2.8502</td>
<td>1.3301</td>
<td>0.9693</td>
</tr>
<tr>
<td>10</td>
<td>Naive Bayes vs .J48</td>
<td>0.2545</td>
<td>1.5267</td>
<td>3.8168</td>
<td>1.5267</td>
<td>1.5267</td>
</tr>
<tr>
<td>11</td>
<td>Bayes Net vs .J48</td>
<td>0.2790</td>
<td>1.5267</td>
<td>4.1846</td>
<td>1.5267</td>
<td>1.5267</td>
</tr>
<tr>
<td>12</td>
<td>Simple Cart vs .J48</td>
<td>0.5308</td>
<td>2.1232</td>
<td>7.9620</td>
<td>2.1232</td>
<td>1.5267</td>
</tr>
<tr>
<td>13</td>
<td>Simple Cart vs .Naive Bayes</td>
<td>0.6081</td>
<td>2.1232</td>
<td>9.1211</td>
<td>2.1232</td>
<td>1.8242</td>
</tr>
<tr>
<td>14</td>
<td>Simple Cart vs .Bayes Net</td>
<td>0.6485</td>
<td>2.1232</td>
<td>9.7276</td>
<td>2.1232</td>
<td>1.8242</td>
</tr>
<tr>
<td>15</td>
<td>Naive Bayes vs .Bayes Net</td>
<td>0.9546</td>
<td>2.1232</td>
<td>14.3184</td>
<td>2.1232</td>
<td>1.8242</td>
</tr>
</tbody>
</table>
5.6 Summary

This chapter presents the enhanced workload adaptive framework. The adaptiveness is an important property of autonomic systems. SVM and CBR are considered for the workload adaptation. The SVM is not adaptive. The CBR based adaptation architecture is developed and tested. The adaptive nature of this model enables the case-base to be revised without human intervention. An increase of 15% is observed in accuracy after adaptation of workload. The results produced through CBR model are effective and accurate. The results are validated by performance advanced non-parametric procedures with post-hoc tests.
CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter begins with the summary of investigations done in chapter 4 and 5. It follows with the discussion on the findings in the context of design and development considerations and performance evaluation. Finally, this chapter concludes the research outcomes and provides some detail about the limitations of the study. Future work highlights the improvements that can be made to enhance the performance of the conducted research.

6.1 Summary of Research

In this research work, we have discussed various important aspects of autonomic workload management. A framework is developed for autonomic workload management that has the ability to handle all the tasks proactively and autonomically. The CBR approach and machine learning techniques are mainly used for developing the Autonomic Workload Performance Predictor (AWPP) framework. This research work differs from other works and builds the AWPP for DBMS that provides a framework consisting of workload prediction and adaptation for performance tuning. This research work is
designed and developed using the CBR based solution (single approach solution) for the three modules (Workload Feature Extraction, Workload Performance Prediction, Workload Adaptation) of AWPP to achieve the multiple objectives of the research. Our first contribution is to develop a framework to predict performance metrics that make DBMS aware of itself, making it autonomic. The predicted performance features are helpful for workload management, resource allocation, system sizing, and capacity planning. Our second contribution is to make the proposed framework adaptive to the new trends in the workload. When the workload evolves and changes its behaviour then AWPP has the capability of handling and adapting for new changes in the DBMS. Our third contribution is incorporation of four autonomic characteristics which includes self-inspection, self-configuration, self-prediction and self-adaptation.

CBR is selected because it learns from experience based on reasoning and no machine learning is required. It works on direct experience. In CBR data elicitation is not required. We compared the results of CBR with other machine learning techniques such as Support Vector Machine and others and evaluated its performance.

CBR produced good results when incorporated in our proposed solution. Our proposed solution for AWPP performed all the work autonomically and less human intervention is required. However, training data is required in the initial stage by the Database Administrator (DBA). We are able to make AWPP autonomic by achieving some of the autonomic characteristics.

We make the CBR approach autonomic with no human involvement. Cases are retrieved, reused, revised and adapted without any human intervention. Optimizer and DBMS algorithms can take advantage of the predicted features of AWPP and configure
DBMS accordingly to achieve better performance. Optimizer can also generate query execution plan (QEP) that can be more optimal.

Figure 6.1 summarizes the design and development phase that leads to the proposed AWPP framework. The objective of first phase is to extract features of workload. The second phase is dealt with the design and development of performance prediction of the OLTP and OLAP type of workload. It predicts the performance of the workload as we already know its type. The objective of third phase is to make the system adaptive. When there is change in the trend of workload, it should have the ability to adapt it accordingly. All the three phases will work autonomically as our objective is to make DBMS aware of itself.

Figure 6.1 represents the phase wise sequence of design and development of the proposed AWPP framework.

![Figure 6.1: The sequence of design and development phases leading to the proposed framework](image)

6.2 Research Findings

Research findings are discussed here from two viewpoints; one is design and development point of view and the other is performance of the proposed framework.
6.2.1 Design and Development Consideration

Proposed Framework

The proposed framework for autonomic workload performance predictor is shown in Figure 6.2. AWPP framework consists of three phases (workload feature extraction, workload performance prediction, workload adaptation). Phase I is workload feature extraction phase; workload is detected and features are extracted. Phase II is workload performance prediction phase in which similarity measure of Workload Feature Vector (WFV) and Performance Feature Vector (PFV) is performed and performance is predicted. The phase III is adaptation phase in which workload is adapted when workload evolves or new trend(s) occurs.
Figure 6.2: Autonomic Workload Performance Predictor (AWPP)

The workload is first detected by the workload detector component. The identified features of workload and the performance features of workload are extracted here. It is assumed that the workload is classified into two types such as DSS or OLTP. The classified workload is further used for workload performance prediction phase. This is the important phase of the AWPP module. The similarity measure is performed between the training workload data and testing workload data for predicting the performance. The new case is compared; if the case is similar to the cases stored in case-base, the match is found, and we have the predicted result. When cases are similar to the input case; the results are obtained from the repository. In other situations when there are different cases
that are not in repository, a new case is built and it is revised and adapted in the repository for next use.

### 6.2.2 Performance Evaluation of the proposed framework

The performance of the proposed framework is evaluated through effectiveness, accuracy, significance and adaptiveness. For measuring the effectiveness we calculated precision, recall and f-measure. The evaluation metrics for the framework are as follows:

- **Effectiveness**
- **Accuracy**
- **Significance**
- **Adaptiveness**

### 6.3 Conclusion

In this research work, autonomic framework AWPP has been developed for performance prediction and its adaptation for two types of workload. A CBR based approach is used for this framework and other machine learning techniques such as SVM Bayes Net, Naïve Bayes, J48 and Simple Cart are also applied for comparison purpose. The TPC-H benchmark dataset is used for the experiments. The TPC-H and TPC-C like workload are executed for training and testing workload.

The experiments and results for AWPP presented in the previous section using CBR and SVM with other machine learning techniques give us an insight into the performance of these supervised machine learning techniques. From our experiments conducted on the second phase; SVM produced good results. However, overall CBR produced better results for performance prediction. The results show that the proposed
framework provides effectiveness of different performance features between 80% - 100% and the accuracy between 70% - 86%. For Adaptation phase, CBR is adaptable, that is 79% effectiveness before adaptation and 100% effectiveness after adaptation is observed. However SVM is not adaptive as the effectiveness is 68% before and after adaptation. The accuracy of CBR is 68% before adaptation and 83% after adaptation. An improvement of 15% is observed in accuracy after adaptation. The study reveals that Cosine distance measure produced better results as compared to other distance measures.

Conventional CBR based systems present deficiency in context of self-managing or autonomic system. A new CBR based AWPP framework has been introduced in this research work. This approach helps in performance tuning and is measured in terms of precision, recall, f-measure, accuracy and significance. In AWPP framework, cases for OLTP and OLAP are stored separately in the case-base. As we know about the type of workload, therefore the retrieval is performed in the relevant type of workload. Hence, it reduces the number of comparisons for searching the case.

To validate our obtained results we also conducted the paired t-test to find the correlation and measure the significance of performance prediction results between CBR and SVM and other machine learning techniques used. We observed the significance of features of performance prediction. The results are significant with significance difference less than 0.05 with 95% confidence interval of difference. The paired t-test results represents that the results using CBR approach for workload performance prediction are significant.

We conducted the test experiments for the performance features vector of the proposed AWPP framework. We performed paired T-Test between the proposed CBR
based model and other machine learning techniques in our experiments. The advanced non-parametric testing procedures are also considered for conducting all pairwise comparisons of classifiers in a multiple comparisons analysis such as Holm, Nemenyi, Shaffer and Bergman procedures. The non-parametric Friedman test $n \times n$ and the post-hoc methods are applied. The classifier ranks are obtained from Friedman test and it is observed that among all classifiers CBR is the best classifier.

We have incorporated four characteristics of Autonomic Computing that includes self-inspection, self-configuration, self-prediction and self-adaptation. When workload enters into the system it is self-inspected and features are extracted. Self-configuration of the workload either OLTP or DSS is performed for the performance phase and other performance features that are predicted are configurable to DBMS algorithms. Through self-prediction the performance features are predicted. The behaviour of the workload may change any time so self-adaptation is performed to adapt new changes in the workload. With adaptation the system self-optimizes itself and tunes the performance.

The ability of CBR to be superior than SVM and other machine learning approaches for AWPP could closely be related to nature of its learning method itself, i.e. lazy learning. As opposed to eager learning methods which need to generalize the training data to classify new cases, lazy learning is a learning method which performs classification based on the similarity of that problem with already known problems. On the whole, our proposed CBR approach for AWPP has able to outperform the popular SVM technique and other classification and prediction techniques applied. On top of that, CBR best fit our problem since it is capable to adapt new cases into its case-base. This
will not require retraining of data as opposed to SVM and other machine learning techniques.

6.4 Limitation of the Study

a) The proposed framework can handle similar cases stored in the case-base. It can not handle the sudden changes in the workload behavior.

b) AWPP framework works well when case-base size is small but it degrades it performance when the size of case-base is grown up to certain limit due to storage of number of new cases in the case-base. The AWPP framework has no ability to calibrate the learnt model.

c) The study is limited to prediction of performance of DSS and OLTP type of workload. Performance of mixed type of workload is to be predicted.

d) A few autonomic characteristics are incorporated in the framework. To make the framework fully autonomic other AC characteristics are to be incorporated.

e) The research is limited to DBMS only. Performance prediction and adaptation for multiple databases in Data warehouse can be a challenge.

f) Other factors such as number of users, complex workloads, network congestion, a distributed database is not considered in this research.

g) The proposed approach is tested on training and testing data, the same is not tested in a real environment.
6.5 Future Work

In this work we developed a framework for autonomic workload performance predictor. We tested and validated the framework. We are able to predict the performance of workload and workload adaptation on evolution. Our future work includes:

a) Based on the framework our future work is to build a tool using AWPP. In future we will enhance the functionality of the CBR framework by incorporating other autonomic characteristics.

b) In this research a CBR based framework is presented. A new direction of research can be to maintain the knowledge-base autonomically. The framework is adaptive, that is a new case is adapted for future use, if it is different from other cases stored in the case-base. Due to this the size of the case-base can grow up to certain limit and space complexity can increase. Other machine learning techniques can be applied to calibrate the learnt model.

c) Similarity measure matching is done for the all the cases stored in the case-base for retrieving the case. This retrieval time is reduced as we already know the class, so comparison is performed for that class to which it belongs, not for the whole case-base for retrieving the case. The research is conducted for DSS and OLTP type of workload it can be extended for other type of workload. For retrieval of the case time complexity can increase. In future the clustering approach can be applied for efficient retrieval of the case, so the case comparison is performed to the specific cluster for efficient retrieval.
d) Based on these predictions resources can be allocated and DBMS performance can be improved. The resource allocation and scheduling algorithms can be improved through these predictions.

e) During experiments it is observed that effectiveness of some features is very high using CBR as compared to SVM. In future we will further investigate the learning method of each technique used.

f) We performed the experiments on MySQL database; the same can be performed on other DBMS such as SQL Server and, Oracle to see insight how other databases perform and to differentiate their results.

g) The Autonomic Workload Performance Predictor would be helpful in a Data Warehouse (DW) environment, so this can be applied on DW for better workload management.

h) In future other factors including number of users, complex workloads, network congestion, and a distributed database can be considered.

i) The current research is tested and validated on TPC-H and TPC-C like workload on benchmark data set on small scale. The case-base size is also limited. Testing the framework on large scale in real environment is also a challenging issue.
6.6 Closing Note

The research problem was addressed holistically through synergy of multiple components covering workload features extraction, performance prediction and adaptation. The performance prediction framework is proposed and enhanced through adaptability. The adaptability produced better results and the effectiveness & accuracy has also been improved. This research has contributed a new theoretical framework, an improved adaptive framework that can adapt the new trends producing accurate results after adaptation. Contribution from this research is a step forward in realization of truly autonomous adaptive framework for DBMS workload. This research has opened up new research opportunities with respect to the enhancement of the proposed framework.
References


References


References


Ganapathi A, Harumi A. Kuno, Umeshwar Dayal, Janet L. Wiener, Armando Fox, Michael I. Jordan, and David A. Patterson (2009), Predicting Multiple Metrics for
References


Hsu C W, Chang C C, and Lin C J (2010), A Practical Guide to Support Vector Classification, Department of Computer Science, National Taiwan University, Taipei 106, Taiwan.


References


Mateen A, Raza B, Hussain T, Awais M M (2009). Autonomicity in Universal Database DB2. 8th International Conference on Computer and Information Science (ICIS 09), June 1-3, Shanghai, China.


References


References


Raza B, Mateen A, Sher M, Awais M M, Hussain T (2010b) .Autonomicity in Oracle Databases Management System. The 2010 International Conference on Data Storage and Data Engineering (DSDE 2010), February 9-10, India.


References


APPENDIX A1

DATABASE ENTITIES, RELATIONSHIPS, AND CHARACTERISTICS

The components of the TPC-H database are defined to consist of eight separate and individual tables (the Base Tables). The relationships between columns of these tables are illustrated in the TPC-H Schema below.

The TPC-H Schema
APPENDIX A2

TABLE LAYOUTS

**PART Table Layout**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype Requirements</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_PARTKEY</td>
<td>identifier</td>
<td>SF*200,000 are populated</td>
</tr>
<tr>
<td>P_NAME</td>
<td>variable text, size 55</td>
<td></td>
</tr>
<tr>
<td>P_MFGR</td>
<td>fixed text, size 25</td>
<td></td>
</tr>
<tr>
<td>P_BRAND</td>
<td>fixed text, size 10</td>
<td></td>
</tr>
<tr>
<td>P_TYPE</td>
<td>variable text, size 25</td>
<td></td>
</tr>
<tr>
<td>P_SIZE</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>P_CONTAINER</td>
<td>fixed text, size 10</td>
<td></td>
</tr>
<tr>
<td>P_RETAILPRICE</td>
<td>decimal</td>
<td></td>
</tr>
<tr>
<td>P_COMMENT</td>
<td>variable text, size 23</td>
<td></td>
</tr>
</tbody>
</table>

Primary Key: P_PARTKEY

**SUPPLIER Table Layout**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype Requirements</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_SUPPKEY</td>
<td>identifier</td>
<td>SF*10,000 are populated</td>
</tr>
<tr>
<td>S_NAME</td>
<td>fixed text, size 25</td>
<td></td>
</tr>
<tr>
<td>S_ADDRESS</td>
<td>variable text, size 40</td>
<td></td>
</tr>
<tr>
<td>S_NATIONKEY</td>
<td>Identifier</td>
<td>Foreign key to N_NATIONKEY</td>
</tr>
<tr>
<td>S_PHONE</td>
<td>fixed text, size 15</td>
<td></td>
</tr>
<tr>
<td>S_ACCTBAL</td>
<td>decimal</td>
<td></td>
</tr>
<tr>
<td>S_COMMENT</td>
<td>variable text, size 101</td>
<td>Primary Key: S_SUPPKEY</td>
</tr>
</tbody>
</table>
PARTSUPP Table Layout

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS_PARTKEY</td>
<td>Identifier</td>
<td>Foreign key to P_PARTKEY</td>
</tr>
<tr>
<td>PS_SUPPKEY</td>
<td>Identifier</td>
<td>Foreign key to S_SUPPKEY</td>
</tr>
<tr>
<td>PS_AVAILQTY</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>PS_SUPPLYCOST</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>PS_COMMENT</td>
<td>variable text, size 199</td>
<td></td>
</tr>
</tbody>
</table>

Primary Key: PS_PARTKEY, PS_SUPPKEY

CUSTOMER Table Layout

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_CUSTKEY</td>
<td>Identifier</td>
<td>SF*150,000 are populated</td>
</tr>
<tr>
<td>C_NAME</td>
<td>variable text, size 25</td>
<td></td>
</tr>
<tr>
<td>C_ADDRESS</td>
<td>variable text, size 40</td>
<td></td>
</tr>
<tr>
<td>C_NATIONKEY</td>
<td>Identifier</td>
<td>Foreign key to N_NATIONKEY</td>
</tr>
<tr>
<td>C_PHONE</td>
<td>fixed text, size 15</td>
<td></td>
</tr>
<tr>
<td>C_ACCTBAL</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>C_MKTSEGMENT</td>
<td>fixed text, size 10</td>
<td></td>
</tr>
<tr>
<td>C_COMMENT</td>
<td>variable text, size 117</td>
<td></td>
</tr>
</tbody>
</table>

Primary Key: C_CUSTKEY

ORDERS Table Layout

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_ORDERKEY</td>
<td>Identifier</td>
<td>SF*1,500,000 are sparsely populated</td>
</tr>
<tr>
<td>O_CUSTKEY</td>
<td>Identifier</td>
<td>Foreign key to C_CUSTKEY</td>
</tr>
</tbody>
</table>
### O_ORDERSTATUS
fixed text, size 1

### O_TOTALPRICE
Decimal

### O_ORDERDATE
Date

### O_ORDERPRIORITY
fixed text, size 15

### O_CLERK
fixed text, size 15

### O_SHIPPRIORITY
Integer

### O_COMMENT
variable text, size 79

Primary Key: O_ORDERKEY

---

### LINEITEM Table Layout

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_ORDERKEY</td>
<td>identifier</td>
<td>Foreign key to O_ORDERKEY</td>
</tr>
<tr>
<td>L_PARTKEY</td>
<td>identifier</td>
<td>Foreign key to P_PARTKEY, Foreign Key to (PS_PARTKEY, PS_SUPPKEY) with L_SUPPKEY</td>
</tr>
<tr>
<td>L_SUPPKEY</td>
<td>Identifier</td>
<td>Foreign key to S_SUPPKEY, Foreign key to (PS_PARTKEY, PS_SUPPKEY) with L_PARTKEY</td>
</tr>
<tr>
<td>L_LINENUMBER</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>L_QUANTITY</td>
<td>decimal</td>
<td></td>
</tr>
<tr>
<td>L_EXTENDEDPRICE</td>
<td>decimal</td>
<td></td>
</tr>
<tr>
<td>L_DISCOUNT</td>
<td>decimal</td>
<td></td>
</tr>
<tr>
<td>L_TAX</td>
<td>decimal</td>
<td></td>
</tr>
<tr>
<td>L_RETURNFLAG</td>
<td>fixed text, size 1</td>
<td></td>
</tr>
<tr>
<td>L_LINESTATUS</td>
<td>fixed text, size 1</td>
<td></td>
</tr>
<tr>
<td>L_SHIPDATE</td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>L_COMMITDATE</td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>L_RECEIPTDATE</td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>L_SHIPINSTRUCT</td>
<td>fixed text, size 25</td>
<td></td>
</tr>
</tbody>
</table>
L_SHIPMODE fixed text, size 10
L_COMMENT variable text size 44

Primary Key: L_ORDERKEY, L_LINENUMBER

### NATION Table

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_NATIONKEY</td>
<td>identifier</td>
<td>25 nations are populated</td>
</tr>
<tr>
<td>N_NAME</td>
<td>fixed text, size 25</td>
<td></td>
</tr>
<tr>
<td>N_REGIONKEY</td>
<td>identifier</td>
<td>Foreign key to R_REGIONKEY</td>
</tr>
<tr>
<td>N_COMMENT</td>
<td>variable text, size 152</td>
<td></td>
</tr>
</tbody>
</table>

Primary Key: N_NATIONKEY

### REGION Table

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_REGIONKEY</td>
<td>identifier</td>
<td>5 regions are populated</td>
</tr>
<tr>
<td>R_NAME</td>
<td>fixed text, size 25</td>
<td></td>
</tr>
<tr>
<td>R_COMMENT</td>
<td>variable text, size 152</td>
<td></td>
</tr>
</tbody>
</table>

Primary Key: R_REGIONKEY
# APPENDIX A3

## DATABASE SIZE

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Cardinality (in rows)</th>
<th>Length (in bytes) of Typical Row</th>
<th>Typical Table Size (in MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUPPLIER</td>
<td>10,000</td>
<td>159</td>
<td>2</td>
</tr>
<tr>
<td>PART</td>
<td>200,000</td>
<td>155</td>
<td>30</td>
</tr>
<tr>
<td>PARTSUPP</td>
<td>800,000</td>
<td>144</td>
<td>110</td>
</tr>
<tr>
<td>CUSTOMER</td>
<td>150,000</td>
<td>179</td>
<td>26</td>
</tr>
<tr>
<td>ORDERS</td>
<td>1,500,000</td>
<td>104</td>
<td>149</td>
</tr>
<tr>
<td>LINEITEM</td>
<td>6,001,215</td>
<td>112</td>
<td>641</td>
</tr>
<tr>
<td>NATION</td>
<td>25</td>
<td>128</td>
<td>&lt; 1</td>
</tr>
<tr>
<td>REGION</td>
<td>5</td>
<td>124</td>
<td>&lt; 1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>8,661,245</strong></td>
<td><strong>956</strong></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX A4

QUERIES FOR CREATING DATABASE, TABLES, LOAD DATA FROM DBGEN

CREATE DATABASE tpch;
use tpch;

create table nation (  
n_nationkey  decimal(3,0) not null,
n_name       char(25) not null,
n_regionkey  decimal(2,0) not null,
n_comment    varchar(152)
) TYPE=MyISAM;

create table region (  
r_regionkey  decimal(2,0) not null,
r_name       char(25) not null,
r_comment    varchar(152)
) TYPE=MyISAM;

create table part (  
p_partkey     decimal(10,0) not null,
p_name        varchar(55) not null,
p_mfgr        char(25) not null,
p_brand       char(10) not null,
p_type        varchar(25) not null,
p_size        decimal(2,0) not null,
p_container   char(10) not null,
p_retailprice decimal(6,2) not null,
p_comment     varchar(23) not null
) TYPE=MyISAM;

create table supplier (  
s_suppkey     decimal(8,0) not null,
s_name        char(25) not null,
s_address     varchar(40) not null,
s_nationkey   decimal(3,0) not null,
s_phone       char(15) not null,
s_acctbal     decimal(7,2) not null,
s_comment     varchar(101) not null
) TYPE=MyISAM;
create table partsupp (  
ps_partkey     decimal(10,0) not null,
ps_suppkey     decimal(8,0) not null,
ps_availqty    decimal(5,0) not null,
ps_supplycost  decimal(6,2) not null,
ps_comment     varchar(199) not null
) TYPE=MyISAM;

create table customer (  
c_custkey     decimal(9,0) not null,
c_name        varchar(25) not null,
c_address     varchar(40) not null,
c_nationkey   decimal(3,0) not null,
c_phone       char(15) not null,
c_acctbal     decimal(7,2) not null,
c_mktsegment  char(10) not null,
c_comment     varchar(117) not null
) TYPE=MyISAM;

create table orders (  
o_orderkey       decimal(12,0) not null,
o_custkey        decimal(9,0) not null,
o_orderstatus    char(1) not null,
o_totalprice     decimal(8,2) not null,
o_orderdate      date not null,
o_orderpriority  char(15) not null,
o_clerk          char(15) not null,
o_shippriority   decimal(1,0) not null,
o_comment        varchar(79) not null
) TYPE=MyISAM;

create table lineitem (  
l_orderkey    decimal(12,0) not null,
l_partkey     decimal(10,0) not null,
l_suppkey     decimal(8,0) not null,
l_linenumber  decimal(1,0) not null,
l_quantity    decimal(2,0) not null,
l_extendedprice  decimal(8,2) not null,
l_discount    decimal(3,2) not null,
l_tax         decimal(3,2) not null,
l_returnflag  char(1) not null,
l_linenumber  char(1) not null,
l_shipdate    date not null,
l_commitdate  date not null,
l_receiptdate date not null,
l_shipinstruct char(25) not null,
l_shipmode     char(10) not null,
l_comment     varchar(44) not null
) TYPE=MyISAM;
Load Data Queries

LOAD DATA INFILE "D://implementation//DBGen//region.tbl" INTO TABLE region FIELDS TERMINATED BY "|" LINES TERMINATED BY "\r\n" (r_regionkey, r_name, r_comment);

LOAD DATA INFILE "D://implementation//DBGen//nation.tbl" INTO TABLE nation FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';

LOAD DATA INFILE "D://implementation//DBGen//supplier.tbl" INTO TABLE supplier FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';

LOAD DATA INFILE "D://implementation//DBGen//part.tbl" INTO TABLE part FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';

LOAD DATA INFILE "D://implementation//DBGen//partsupp.tbl" INTO TABLE partsupp FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';

LOAD DATA INFILE "D://implementation//DBGen//customer.tbl" INTO TABLE customer FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';

LOAD DATA INFILE "D://implementation//DBGen//orders.tbl" INTO TABLE orders FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';

LOAD DATA INFILE "D://implementation//DBGen//lineitem.tbl" INTO TABLE lineitem FIELDS TERMINATED BY '|' LINES TERMINATED BY '\r\n';

ALTER TABLE region ADD CONSTRAINT pkey_region PRIMARY KEY(r_regionkey);
ALTER TABLE nation ADD CONSTRAINT pkey_nation PRIMARY KEY(n_nationkey);
ALTER TABLE part ADD CONSTRAINT pkey_part PRIMARY KEY(p_partkey);
ALTER TABLE supplier ADD CONSTRAINT pkey_supplier PRIMARY KEY(s_suppkey);
ALTER TABLE partsupp ADD CONSTRAINT pkey_partsupp PRIMARY KEY(ps_partkey,ps_suppkey);
ALTER TABLE customer ADD CONSTRAINT pkey_customer PRIMARY KEY(c_custkey);
ALTER TABLE lineitem ADD CONSTRAINT pkey_lineitem PRIMARY KEY(l_orderkey,l_linenumber);
ALTER TABLE orders ADD CONSTRAINT pkey_orders PRIMARY KEY(o_orderkey);

create index fkey_nation_1 on nation(n_regionkey);
create index fkey_supplier_1 on supplier(s_nationkey);
create index fkey_customer_1 on customer(c_nationkey);
create index fkey_partsupp_1 on partsupp(ps_suppkey);
create index fkey_partsupp_2 on partsupp(ps_partkey);
create index fkey_orders_1 on orders(o_custkey);
create index fkey_lineitem_1 on lineitem(l_orderkey);
create index fkey_lineitem_2 on lineitem(l_partkey,l_suppkey);
create index fkey_lineitem_3 on lineitem(l_suppkey);
APPENDIX A5

DEFINITION OF TERMS

i). ADBMS (Autonomic Database Management System)
ADBMS is a DBMS that have few or all autonomic computing characteristics such as self-inspection, self-configuration, self-prediction self-adaptation and self-optimization.

ii). AC (Autonomic Computing)
Autonomic Computing is first time introduced by IBM in 2001. It has number of characteristics that a system contain to be autonomic. The purpose of autonomic computing is to develop self-managing system with less or no human involvement and have the ability to solve the complex problems which is beyond the human capability by decision making itself.

iii). Workload Classifier
Workload classifier is first component of AWPP that built through training it from TPC-C and TPC-H like workloads. It predicts the type of workload either DSS or OLTP.

iv). Workload Predictor
Workload predictor is a component of AWPP framework that predicts the performance features of the workload.

v). Workload Adapter
Workload Adapter is the component of AWPP framework; it adapts the changing behavior of the workload of the system.

vi). DBA (Database Administrator)
   
   DBA is a person who is responsible for managing the database activities such as configuration, performance, recovery from failure etc.

vii). DBMS (Database Management System)
   
   DBMS is a computer used for managing database that is in the form of tables consisting of row and columns. Different operations can be performed on data by users.

viii). Training data and testing data
   
   In machine learning technique we prepare the training and testing data. We build rule or model and train it with training data and generalize it. Through testing data we test the model and can predict the value.

ix). DSS (Decision Support System) Workload
   
   DSS workload consists of decision-support queries with complex queries and low volume of transaction.

x). OLTP (On-Line Transaction Processing) Workload
   
   OLTP workload consists of some simple queries and high volume of transactions.

xi). TPC (Transaction Processing Performance Council)
   
   TPC is an organization that produces industry-standard benchmarks for the DBMSs.

xii). TPC-C Workload
   
   TPC-C is related with OLTP type of workload that simulates an environment where terminal operators execute transactions against a database.

xiii). TPC-H Workload
TPC-H is related with DSS workload that examines large volume of data, complex queries and answer to the critical business questions.

xiv). Features Selection

For machine leaning application subset of features are selected for the data and the feature selection is performed. Feature Selection is a process of selecting the subset of all features that best represent the all. Computationally it is not feasible to use all the features even for small data sample.

xv). Workload

DBMS queries or requests that need resources to perform execution.
APPENDIX A6

LIST OF RESEARCH PUBLICATIONS

JOURNALS PAPERS


CONFERENCES PAPERS


4. Raza B, Mateen A, Hussain T, Awais M.M, Autonomic Success in Databases Management Systems, 8th International Conference on Computer and Information Science (ICIS 09), Shanghai, China, June 1-3 2009.


OTHER JOURNAL PUBLICATIONS (Co-authorship)


OTHER CONFERENCES PUBLICATION (Co-authorship)


