Knowledge and Information Systems (2019) 61:27–63
https://doi.org/10.1007/s10115-018-1272-0

SURVEY PAPER

Autonomic workload performance tuning in large-scale data repositories

Basit Raza[1] · Asma Sher[1] · Sana Afzal[1] · Ahmad Kamran Malik[1] · Adeel Anjum[1] · Yogan Jaya Kumar[2] · Muhammad Faheem[3]

Received: 14 May 2017 / Revised: 22 May 2018 / Accepted: 14 July 2018 / Published online: 4 September 2018

Abstract
The workload in large-scale data repositories involves concurrent users and contains homogenous and heterogeneous data. The large volume of data, dynamic behavior and versatility of large-scale data repositories is not easy to be managed by humans. This requires computational power for managing the load of current servers. Autonomic technology can support predicting the workload type; decision support system or online transaction processing can help servers to autonomously adapt to the workloads. The intelligent system could be designed by knowing the type of workload in advance and predict the performance of workload that could autonomically adapt the changing behavior of workload. Workload management involves effectively monitoring and controlling the workflow of queries in large-scale data repositories. This work presents a taxonomy through systematic analysis of workload management in large-scale data repositories with respect to autonomic computing (AC) including database management systems and data warehouses. The state-of-the-art practices in large-scale data repositories are reviewed with respect to AC for characterization, performance prediction and adaptation of workload. Current issues are highlighted at the end with future directions.

Keywords Autonomic computing · Workload management · Large-scale data repositories · Classification · Prediction · Adaptation · Online transaction processing (OLTP) · Decision support system (DSS)

Abbreviations

DBMS	Database management system
ADBMS	Autonomic database management system

✉ Basit Raza
basit.raza@comsats.edu.pk

1 Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad, Pakistan

2 Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

3 Department of Computer Engineering, Abdullah Gul University, Kayseri, Turkey

AWPT	Autonomic workload performance tuning
OLAP	Online analytical processing
OLTP	Online transaction processing
KCCA	Kernel canonical correlation analysis
TPC	Transaction Processing Council
DBA	Database administrator
SVM	Support vector machines
QEP	Query execution plan
AC	Autonomic computing
QoS	Quality of service
KNN	K-nearest neighbor
OSN	Online social network
CBMG	Customer behavior model graph
GC	Garbage collection
CRT	Classification and regression tree
BI	Business intelligence
PCA	Principal component analysis
CCA	Canonical correlation analysis
QP	Query patroller
PQR	Predictions of query runtime
SLA	Service level agreement
EQMS	External queue management system
WCF	Workload classification and forecasting
MAPEK	Monitor, Analyze, Plan, Execute, Knowledge
DML	Descartes Modeling Language
ANN	Artificial neural network

1 Introduction

Large-scale data repositories play an important role and are the main source of information. Timely and accurate retrieval of information is critical for organizations to reach success. The workload can be defined as data loads, batch reports or complex requests that include insert, delete and update operations, which may be resource demanding (such as input/output devices, storage and memory). The queries or workload is generated by either a decision support system (DSS) or online transaction processing (OLTP) for a database management system (DBMS) and online analytical processing (OLAP) for a data warehouse (DW). The mix type (mix of OLTP and DSS) of workload is also gaining importance. Therefore, identification of the workload type is deemed important for better workload management. In recent literature, more types of database workload are being studied. Resource allocation is performed based on the workload type. A database administrator (DBA) used to manage the system workload by tuning and configuring it manually. Due to the increase in the volume of data and change in workload behavior, DBAs cannot manage it efficiently and thus DBMS performance decreases. When DBMSs become overloaded, resources cannot be allocated properly. The DBMS performance suffers when the DBA or workload manager is unable to handle it properly; therefore, some requests may not be entertained and are rejected or delayed. Due to low performance, users also lose their interest and it becomes the reason for the wastage of organizational resources like money and time.

The term autonomic computing (AC) was introduced by the IBM in 2001 [151], and since then, research communities, database and data warehouse vendors have been working on the incorporation of autonomic technology. Incorporation of AC is performed in various areas of study, and it is equally important in large-scale data repositories. AC has a number of characteristics, and one or more of them are incorporated into the system to make it autonomic. The study [49] highlighted few shortcomings in the DBMSs with the autonomic point of view. The well-known commercial DBMSs including Oracle, DB2 and SQL Server are investigated in the studies [89, 90, 123] to see how these shortcomings have been achieved in recent DBMSs. Autonomicity is incorporated into different components, utilities and tools, and certain levels are achieved by DBMSs [122]. Database researchers continue to focus on proper management of the database workload due to heterogeneity, growth in data size and complexity in the workload. Autonomic workload management is related to AC characteristics such as self-prediction, self-healing, self-inspection, self-adaptation, self-configuration and self-optimization [66, 151]. These works are being carried out on classification, performance prediction and adaptation of workload. The AC technology has the potential to enhance the performance of large-scale data repositories. However, there are challenges in workload management which include predicting the execution time of workload, resource contention workloads and dealing with problematic workload [150]. A number of tools, models and algorithms have been presented in [121] that are developed for managing the system workload with respect to AC. The literature reveals that two types of workload, i.e., DSS and OLTP, are considered to classify the workload and predictions are made to forecast the performance of workload and adaptation is performed on changing the behavior of workload. The tasks that were performed by a DBA are performed autonomically without or with less human intervention, and DBA has the time to do other useful tasks that will improve the performance, reduce the time and cost of ownership.

In the literature, there exist a few relevant surveys [18, 87, 91, 158]. The survey [18] only presents workload characterization, and it does not describe AC perspective. The survey [87] discusses AC only and is not relevant to workload management. The survey presented in [91] describes the literature review of database workload management with respect to AC for mainly three commercial databases only which are Oracle, DB2 and SQL Server. The recent survey [158] presents a taxonomy of workload management in DBMSs. This survey covers the workload management techniques for characterization, admission control, scheduling and execution control. However, for our survey, we explored all the techniques used for workload management through workload classification, workload performance prediction and workload adaptation. It provides detailed analyses of workload management techniques related to large-scale data repositories such as DBMSs and DWs from the year 2001–2017. We categorized these techniques under three autonomic characteristics, i.e., self-inspection, self-prediction and self-adaptation. The main objective of this study is to conduct a state-of-the-art survey of workload performance tuning to see insights on how the management is done intelligently by incorporating AC characteristics. The paper focuses on management of two types of database workload OLTP and DSS in databases and DWs related to large-scale data repositories. The contributions of this paper include detailed analyses of workload management with respect to AC. Three main AC characteristics (self-inspection, self-prediction and self-adaptation) are selected in this work that are broadly used in workload management by well-known vendors of large-scale repositories. A taxonomy of autonomic workload performance tuning (AWPT) is presented, and the approaches for performance modeling in DWs are explored. State-of-the-art research issues are also described, and research directions are provided.

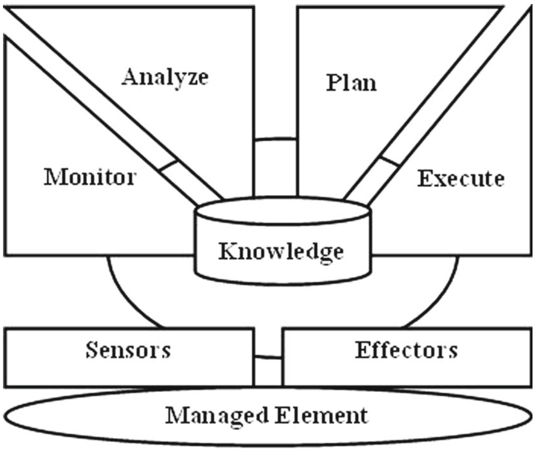

Fig. 1 MAPEK model

The organization of the paper is as follows. Autonomic workload management is presented in Sect. 2. The research methodology followed in this research is provided in Sect. 3. The taxonomy for workload management created in this research is presented in Sect. 4. The issues in large-scale data repositories are discussed in Sect. 5. At the end research is concluded with future work in Sect. 6.

2 Autonomic computing and autonomic workload management

Early unreliable and static systems are now being converted to reliable and dynamic systems, and its dynamic behavior is also evolving. The increase in computer usage has created complexity and due to this manageability has become very difficult for recent and future information technology. These workload manageability issues motivate toward intelligent or AC where the computations will be performed autonomically, and in the meanwhile, DBAs or IT professional can do other useful activities of DBMSs rather than operational tasks with less cost and less human involvement. The systems which have the ability to manage all the activities without or with less human involvement are called autonomic systems [77, 151]. The idea of AC is taken from a human nervous system that is performing required tasks in the human body without conscious mental effort about what is needed and when is needed.

The evolutionary process of AC consists of five basic levels starting from basic to an autonomic level such as managed, predictive, adaptive and autonomic [87, 151]. Figure 1 presents the MAPEK (Monitor, Analyze, Plan, Execute and Knowledge) model of the autonomic system. The workload acts as *managed element*. Managed element, the main component in an autonomic system, is handled through manageability interface which consists of s*ensors* and *effectors*.

From the *managed element*, *Sensors* collect the necessary information and *Effectors* are adjusted according to the intended behavior.

Monitor examines the activities of *Sensor* and stores the data collected by *Sensors* in the *Knowledge-base*. The *Analyze* component performs a comparison of parameters of data with intended parameters and stores the *analyzed* data in the *knowledge-base*. Using the analyzed data, the plan component finds the trends of data by using different methods. The *Execute* component performs adjustments of the parameters of the *managed element* through the *effector* component. The *knowledge-base* is the repository where all collected data and analyzed data are stored. For workload management, the workload can be mapped to the *managed element*. *Sensors* and *effectors* are detecting and identifying workload type or workload performance. The *monitor* and *analyze* components observe and classify the workload, respectively. The *Plan* component examines the trend of incoming data. After executing data through *execute,* the future data are stored into the *knowledge-base*.

For the system to be autonomic, it holds some AC characteristics [151] that include self-configuration, self-inspection, self-prediction, self-adapting, self-healing and self-optimization. An AC system is capable of monitoring the system itself and can perform self-optimization of the resources. Self-optimization is achieved by the AC system by performing different activities efficiently on the basis of environmental setting and available resources. The configuration of AC system is performed itself to meet the desired objectives, and the configurations are predicted and adapted dynamically. All the reconfigurations are done autonomically with less or no human intervention. Self-healing capability of AC system recovers a system from attack using logs statistics and backup files and brings the system to a consistent state after every attack. The AC system has the ability of self-adaptation to recognize the changes occurring in the system and adapt the changes including changes in indexes, data layout, the data structure of the database for improving the system performance. In this study, we have focused on three AC characteristics, i.e., self-prediction, self-inspection and self-adaptation.

DBAs have to face a number of problems while managing the database. The reason behind this is the versatility of data, the growth of data volume, database functionality enhancement, homogenous and heterogeneous data, database maintenance and e-service [70]. In large-scale data repositories, including DBMSs and DWs, to increase the efficiency and reduce the human intervention AC is becoming the need of the systems. The DBMSs that have autonomic capabilities are called autonomic database management systems (ADBMS) and DBMSs are evolving toward ADBMSs [89, 90, 123, 124]. The vendors of well-known DBMSs including SQL Server, DB2 and Oracle are making their DBMS products autonomic, and DB2 is the leading DBMS in this regard. DBMSs are evaluated [89, 90, 123] and optimizers are compared [124] with respect to autonomic perspective along with the degree of human intervention, and the workloads of DBMSs are observed from the autonomic point of view [121]. Workload management with respect to AC (self-inspection, self-prediction, self-adaptation) needs thorough investigation in large-scale data repositories. Self-inspection supports in monitoring and controlling the workflow, self-prediction supports in forecasting the performance aspects of workload, and self-adaptation supports in managing the evolving behavior of the workload.

3 Methodology

We developed a methodology for the survey to find literature for large-scale data repositories such as databases and data warehouses with respect to workload management. The journal papers and conference papers are searched using Google Scholar and DBLP with queries “workload monitoring and classification,” “workload performance prediction,” workload adaptation,” etc. This survey focuses on AC used in large-scale data repositories: to determine AC characteristics that are incorporated, and the techniques that are used. The timeline for this study is set from 2001 to 2017 due to the fact that the term AC was introduced in the year 2001 by the IBM. Workload management was categorized into three types including work-

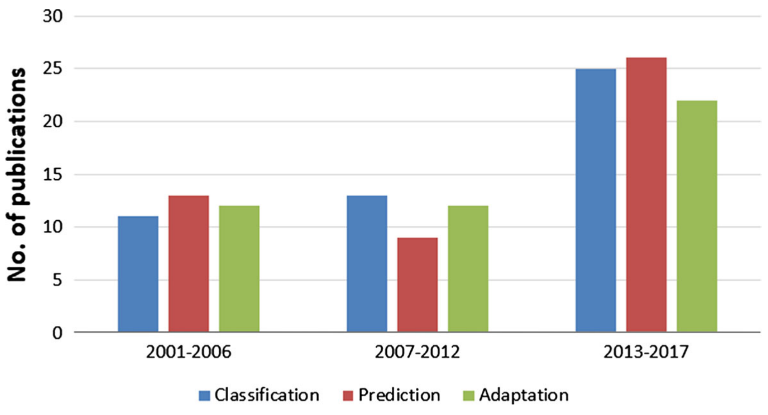

Fig. 2 AWPT publications

load classification, performance prediction and workload adaptation. We created a taxonomy of AWPT that shows these three categories and their subcategories. Our work follows this taxonomy throughout the paper. For inclusion/exclusion of papers, we manually reviewed the literature and a criterion is defined.

The papers found in the literature that present autonomic or self-managing models, frameworks and architectures for large-scale data repositories are included in this survey. To limit the scope, we only considered workload management in databases and data warehouses, especially in large-scale data repositories. All the papers related to workload management are included, which falls under the three selected AC characteristics: self-inspection, self-prediction and self-adaptation; that contributes toward performance tuning. All the papers that present techniques, models, architectures and frameworks developed for workload management and focused on monitoring and controlling of the workload are put into self-inspection category. All the papers related to workload predictions are put into a self-prediction category. Similarly, all the papers related to workload adaptation are put into the self-adaptation category.

The studies related to workload management in big data, cloud and distributed databases are excluded. The papers which discuss workload types other than OLTP and DSS such as workload in mobile, cloud, web, social network are also excluded. The short notes, technical papers, organizations white papers were not included. The papers that are not in English language or cannot be translated well are excluded. All the papers related to workload management that exhibit other autonomic characteristics are excluded. The manually searched 158 papers fall into the three categories and possess autonomic characteristics. In this study, 89 journal papers, 65 conference papers, 1 book chapter and 3 reports are used. The journals and conferences are related to different publishers like IEEE (75), ACM (32), VLDB (13), Springer (13), Wiley (3), Elsevier (14).

The year range-wise publications in three categories classification, prediction and adaptation of workload are presented in Fig. 2.

The work done with respect to autonomic characteristics such as self-inspection, self-prediction and self-adaptation is shown in Fig. 3.

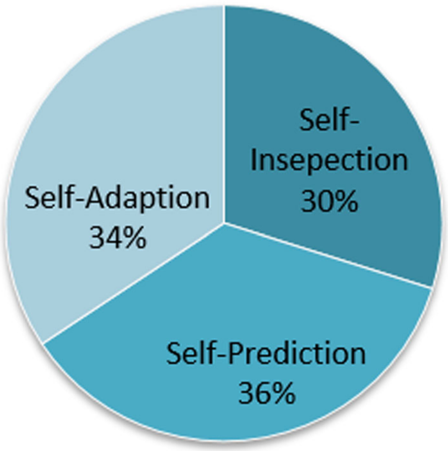

Fig. 3 AWPT publications per subcategory

4 Autonomic workload performance tuning in large-scale data repositories

To describe and classify the autonomic characteristics needed for large-scale data repositories, we have created the taxonomy of autonomic workload performance tuning (AWPT). Three aspects of large-scale data repositories with respect to autonomic characteristics include workload classification, performance prediction and adaptation. The taxonomy is shown in Fig. 4. In the context of large-scale data repositories, the workload is the job (in the form of queries) assigned to a system to be executed in a given time duration, whereas, generally in computer science, it refers to every input or request received by a specified technological infrastructure [106]. Workload management is performed automatically or through the DBAs [80]. After the development of the autonomic technology, the trend is converted from automatic to autonomic [37, 99]. According to this trend, we consider highly relevant to map AC characteristics to workload management. The works of large-scale data repositories related to workload classification, prediction and adaptation are mapped to AC characteristics which are self-inspection, self-prediction and self-adaptation, respectively. For all the works, the techniques and approaches used are discussed, and their limitations are described.

Many studies [6, 13, 18, 81, 116, 125, 134, 138, 147] for workload characterization are found in the literature with applications in various domains such as (see Fig. 5): database management systems, online transaction processing systems, microservices, web browser, video services online social network and mobile devices. This paper focuses only on the workload management of large-scale data repositories through workflow monitoring and performance tuning that includes DBMSs and DWs, excluding other types of workload that are not under the scope of the paper.

The workloads are of different types in different application areas which include DBMS workload [2, 9, 65, 67, 72, 157], Web workload [17, 19, 20, 71, 116, 112, 120, 137], OSN workload [12, 22, 29, 116, 152], Video service workload [28, 52, 134, 129, 138], Mobile app workload [21, 40, 79, 155] and Cloud workload [6, 63, 125, 140, 147]. Every type of workload has its own features and is handled through different ways using a variety of techniques. Data analysis approaches used for different domain/applications workload as shown in Fig. 6 are described as follows. A database workload is a request set upon the DBMS. The database workload can be classified as OLTP, DSS, a mix of OLTP and DSS, commands, sessions and

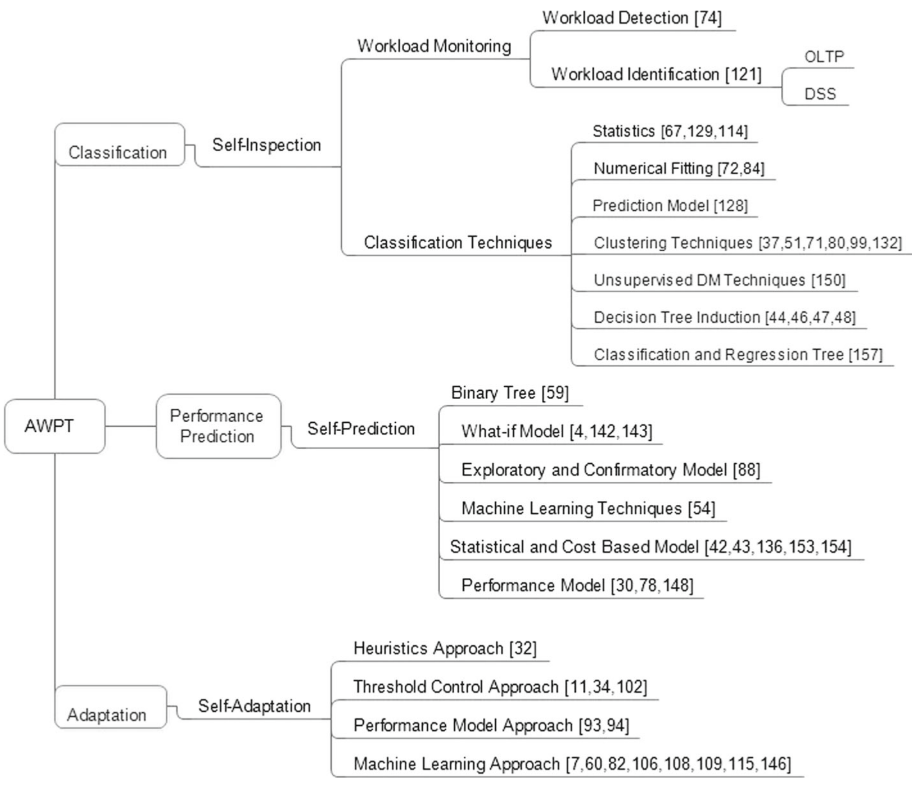
AWPT
Classification
Self-Inspection
Workload Monitoring
Workload Detection [74]
Workload Identification [121]
OLTP
DSS
Classification Techniques
Statistics [67,129,114]
Numerical Fitting [72,84]
Prediction Model [128]
Clustering Techniques [37,51,71,80,99,132]
Unsupervised DM Techniques [150]
Decision Tree Induction [44,46,47,48]
Classification and Regression Tree [157]
Performance Prediction
Self-Prediction
Binary Tree [59]
What-if Model [4,142,143]
Exploratory and Confirmatory Model [88]
Machine Learning Techniques [54]
Statistical and Cost Based Model [42,43,136,153,154]
Performance Model [30,78,148]
Adaptation
Self-Adaptation
Heuristics Approach [32]
Threshold Control Approach [11,34,102]
Performance Model Approach [93,94]
Machine Learning Approach [7,60,82,106,108,109,115,146]

Fig. 4 Autonomic workload performance tuning (AWPT) taxonomy

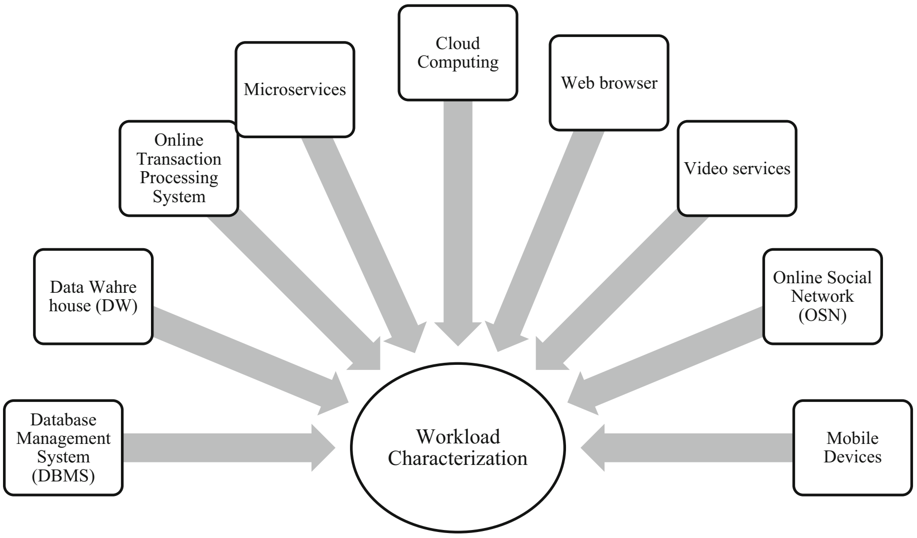
Cloud Computing
Microservices
Web browser
Online Transaction Processing System
Video services
Data Wahre house (DW)
Online Social Network (OSN)
Database Management System (DBMS)
Workload Characterization
Mobile Devices

Fig. 5 Workload characterization in various applications and domains [6, 13, 18, 81, 116, 125, 134, 138, 147]

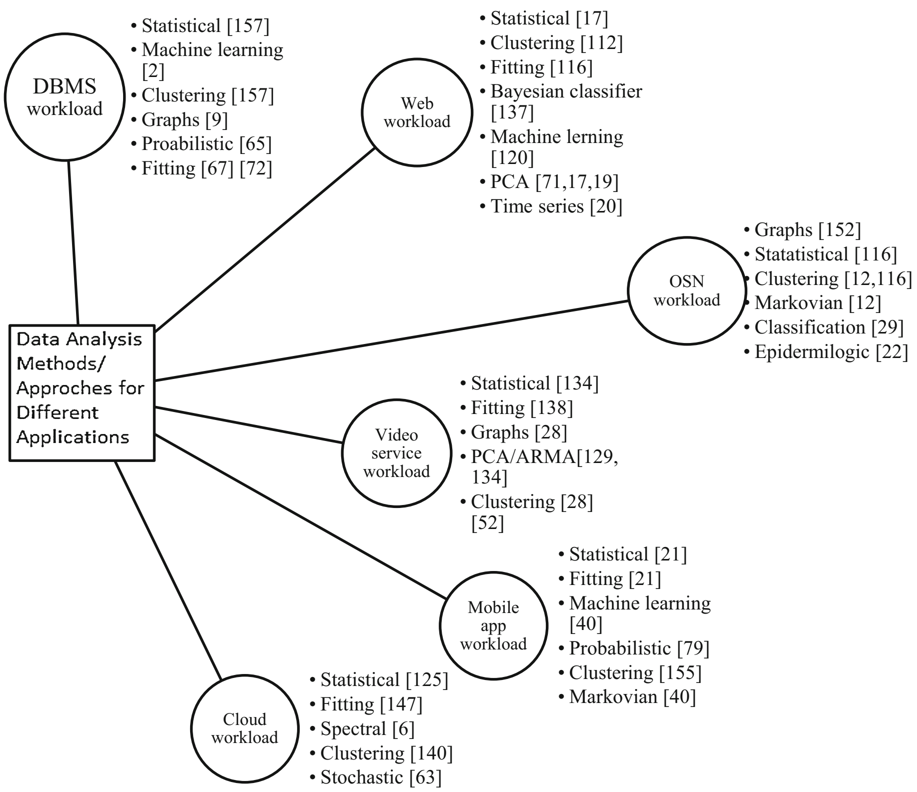

Fig. 6 Data analysis approaches used for different domain/applications workload

further operations that run in a time interval [69]. Several papers [15, 51, 65, 69, 105, 134] characterize database workload using different statistical, data mining and machine learning techniques. In today's enterprises, OLTP systems are popular data processing systems. Few examples of OLTP systems include retail sales, order entry and financial transaction systems. In OLTP systems, insert, update and delete queries are used. The other types of workload can be system logs and variable snapshots [6]. Different techniques like fuzzy logic and neural network are used for the characterization. The nature and the characteristics of web workload have been studied since the web's commencement. Calzarossa in her paper [18] discussed the characteristics of a different impression of conventional web workload in detail. Different techniques like machine learning, fitting, stats, clustering, time series are used for the analysis. Most of the part of conventional web workloads consist of hypertext transfer protocol (HTTP) requests issued by clients. Contrarily, recent technologies allow users to share their own multimedia and textual content to search for information. Different aspects of conventional workload discussed in the literature include web content [89], shopping services [122], web robot traffic [66, 150]. In the literature, these aspects have been analyzed from different perspectives. Characterization of microservices also includes in web workload [121].

The online social network (OSN) workload refers to the load produced by the users which are used to exploit the sophisticated information to provide services offered by these technologies. In the literature, the most relevant research questions considered are referred to user behavior, content propagation and network structure and evolution [106]. Crawling is the method used by most researchers [70, 77, 124] to gather information using application programming interface (API). Video service type of workload refers to users produced load who access and share self-generated content or supplied by media producers. Understanding this content helps in storage and content management, the design of content distribution systems, capacity provisioning and resource allocation [106]. In the literature, different researchers [13, 134, 138] use different techniques like stats, fitting, graphs, clustering for video service workload characterization. Mobile devices like tablets and smartphones deploy many diverse mobile apps that allow users to share and access resources and content. In the literature, two perspectives are considered smartphone and app store for app usage patterns [106]. Several papers [81, 116, 125, 147] characterize the mobile device workload according to above-mentioned perspective using machine learning, Markovian, stats or clustering techniques. A large variety of services and applications deployed on the infrastructure of cloud produce the load which is referred to as cloud workload. Most studies [18, 109, 110, 157] focus on the workload characterization processed by public or private cloud infrastructures by focusing on facts like user behavior, application characterization and virtual machine behavior.

4.1 Workload classification: techniques to support self-inspection

Workload classification plays a key role in performance tuning and is important in performance engineering [18]. This section presents all the techniques that support self-inspection in workload classification. Workload classification is performed on the basis of workflow properties such as resource demands, query cost. It involves two functional components which are performance monitoring and characterization [110]. Monitoring the performance is reactive approach and works when performance degrades, whereas the characterization is a proactive approach which keeps track changes of the workload. Self-inspection is an AC characteristic which is the capability of monitoring the workload. Many techniques, tools and algorithms have been developed for monitoring the workload [101] that are used for controlling the workload. A GATEKEEPER tool was developed for scheduling and workload admission control of E-commerce workload to enhance stability and response time [50]. For managing the workload in an efficient manner through suspension and resumption, a tool PREDATOR was developed [23, 25]. An online tuning framework was proposed for examining changes in the physical design and continuous monitoring of workload [16]. The physical design was altered based on information obtained from the query execution plan (QEP), its cost was calculated, and best QEP is selected. Transaction Processing Performance Council (TPC) [145], a nonprofit organization, recognized the workload characterization which is used to measure system performance for developing benchmarks for experimental purposes. In workload management, classification of workload has become very important and through self-inspection, and the workload can be managed in a better way by knowing the type of workload [33, 113]. In the literature, many studies described the workload classification through adaptation [81, 83, 105, 141]. The study [2] presents workload classification and scheduling in DBMSs, and [63] discussed prediction on time series analysis. Similarly, the work in [9] applied automated techniques using system logs for producing customer behavior model graph (CBMG). The study [74] provides the workload characterization of data centers through fingerprints, and in [31], the TPC-H workload is characterized on Apache Spark for optimization.

In current large-scale data repositories, two classes of workload are mainly used that include DSS and OLTP. DSS consists of complex workloads, business analysis and requires large storage space, whereas OLTP consists of daily transactional activities. The mix type of workflow is also encountered in the system that is a mixture of DSS and OLTP. In the literature, many techniques and approaches have been developed and implemented for workload classification that supports self-inspection, and the details are discussed as follows.

4.1.1 Statistics

For monitoring and characterizing the workload many statistics are used in the literature that includes correlation, mean (harmonic, geometric, arithmetic) or average, mode, variance, range (max or min) and standard deviation. Principal component analysis (PCA) was used in [114, 129] to find principle components by mapping correlated parameters to linearly uncorrelated parameters through an orthogonal transformation for dimensionality reduction and simplification. Visualization methods such as box plots, histograms, scatter plots are used for inspecting the workload parameters. Single-parameter and multi-parameter histograms are used for presenting the correlation among workload parameters. The study [67] characterized the workload through the parameters such as DB transaction and SQL statement using the standard TPC benchmarks. A number of experiments are performed by applying different methods such as nonhomogeneous Poisson process, histograms and summaries (average, distributions) for concurrency control, buffer management and memory management.

4.1.2 Numerical fitting

Numerical fitting techniques have been used for studying the patterns and behaviors of the workload. Several methods such as least squares and maximum likelihood estimations are used for parameter estimation of the function to best fit on the experimental data. The best fitness is evaluated through different statistics such as Anderson–Darling, R-squared, Kolmogorov–Smimov and F tests. For instance, in the study [140], the prediction accuracy is evaluated through the R-squared method, and the shape of experimental data is determined through probabilistic distribution methods such as exponential, lognormal, binomial, Weibull gamma. The work [84] shows that power law distribution is used for presenting the behavior of large distributed values of workload properties. The concept of chains of the sequential request is used for presenting spatial locality of the workload in characterizing the Netflix workload [138]. The DSS and OLTP workloads have different characteristics and have been investigated in [67, 72]. The arrival patterns of the transactions are modeled using a combination of statistical techniques. For nonlinear regression, a numerical fitting technique such as Levenberg–Marquardt is used and for the interactive type of workload, DB transaction arrival time is used.

4.1.3 Prediction model

Prediction models have been used for classification of workload. Predictions are performed for enhancing buffer/hit ratio in database cache, user access behavior and characterization of the access pattern for caching OLAP systems in multi-dimensional information system [128]. Time and sequence attributes are used by applying Markov chain model, and the combination of other statistical techniques is also used for prediction of OLAP queries.

4.1.4 Clustering techniques

Clustering techniques are widely used for classification of workload [51, 71, 132, 157] and can be categorized into two types, hierarchical and nonhierarchical clustering. On the one hand hierarchical clustering techniques group similar classes of the same nature. Clusters are stored in the form of a hierarchy of less similar to most similar parameters through clustering techniques such as hierarchical agglomerative clustering (HAC) and divisive (monothetic or polythetic) clustering. On the other hand, nonhierarchical clustering techniques create nonoverlapping clusters of data; the number of clusters is known in advance, so independent clusters are formed with no hierarchical relationship among them. The approaches of nonhierarchical clustering are nearest neighbor (k-mean) relocation and single pass. The performance measures used for measuring the quality of cluster include purity, confusion matrix, f measure, Fowlkes–Mallows index, Jaccard index and Rand measure.

A model relational workload analyzer database (REDWAR) was developed for characterizing the workload that analyzes SQL statements with respect to WHERE or GROUP BY clauses. For clustering, two types of clusters are developed, i.e., CLUE and HALC. The CLUE characterized OLTP workload, and HALC, a heuristics clustering algorithm, is used to manage a large amount of data. The study [157] presented two types (DSS and OLTP) of the workload in MySQL database for classification of workload using data mining and statistical techniques such as classification and regression tree (CRT) and hierarchical clustering.

4.1.5 Decision tree induction

Literature reveals that many studies were done to classify the database workload of a relational database in IBM's DB2 environment [47, 48] using decision tree technique. The workload management studies [44, 47] presented a classifier that classifies the workload into DSS and OLTP types of workloads using induction tree. A classification model [46] was developed to classify the workload based on workload characteristics and detects the changes in the type of workload. Psychic Skeptic Prediction framework (PSP) was developed to predict the changes in a shift of workload from DSS to OLTP and OLTP to DSS [46].

4.1.6 Classification and regression tree (CRT)

Workload classification for two types of workload (DSS and OLTP) can be performed using classification and regression tree [157]. The study developed a classification model and identified few features that support classification. The identified features which are effective in classification include the number of user logging (Innodb_log_writes), query ratio of Select and Update/Insert/Delete (Com_ratio), the number of a statement executed (Questions) and the number of query cache hits (Qcache_hits). Table 1 presents the summary of workload classification, the techniques used, the attributes selected as input and the type of workload predicted.

4.2 Workload performance prediction: techniques to support self-prediction

In workload management, performance prediction plays an important role in performance tuning. This section presents workload performance prediction techniques that support self-predictions. Self-prediction is the AC characteristics that forecast the future trends based on historical data. Many studies [54, 59] have been carried out for workload performance predictions in DBMSs. The study [98] predicts the performance and resource for cloud database by developing the DBSeer framework. The combination of machine learning and analytical modeling is performed for performance enhancement [38]. Distribution-based predictions are

Table 1 Classification techniques for workload management

Input parameters	Type of workload	Technique used	Purpose of research and limitations
CPU demands Arrival time Number of different pages accessed per transaction Cache miss rate Read/write page accesses Number of transactions completed Deadlocks Number of references to a memory block Memory footprint	Interactive	Statistics [67]	Performed investigation of workload characteristics of production database for top 10 organizations Provided comparison with TPC-C and TPC-D and in some cases the prediction of behavior of production workload becomes out of scope
Arrival time	Interactive	Numerical fitting [127]	Evaluated performance of database on simultaneous multi-threading (SMT) processors Benchmark unable to provide some original characteristics of the workload
Sequence of queries in execution Think time	Interactive	Prediction models [128]	Buffer/hit ratio prediction Enhancing caching and making prefetching in a predictive way
Random I/O operations Response time Sequential I/O throughput rate Average CPU utilization	BI workload	Semidiscrete decomposition (SDD) Singular value decomposition (SVD) [150]	Resource characteristics are used to analyze characterization methods

Table 1 continued

Input parameters	Type of workload	Technique used	Purpose of research and limitations
Pages scanned Queries ratio Hit ratio Pages read Logging Ratio of using indexes Rows selected Average sort time Number of locks held Number of sorts Throughput	OLTP workload and DSS workload	Decision trees induction [47, 48]	The study is limited to only DB2 database and is not applicable to other existing DBMSs Comparison with other techniques such as machine learning is not performed Not suitable in case of trees of big sizes whose pruning could produce wrong results Prediction of workload change is not performed
Questions Query ratio Qcache_hits Innodb_log_writes	OLTP workload and DSS workload	Hierarchical clustering and classification regression tree (CRT) [157]	Characterization attributes used in study are low in importance and are insufficient to characterize the workload in DSS and OLTP Machine learning could be used in improving workload characterization
table_locks key_writes key_reads	OLTP workload and DSS workload	Case-based reasoning fuzzy logic [2]	Limited to characterize into only two types of workload DSS and OLTP Used three attributes only, however, for better characterization other attributes could be investigated
Response time Arrival rate Resource utilization	Access logs of web server	Customer behavior model graph (CBMG) [9]	Automation is not performed to parse and analyze the server logs

Table 1 continued

Input parameters	Type of workload	Technique used	Purpose of research and limitations
Longest_lacache.miss Cycles_lid_pending Cycle_activity	Interactive	Machine learning [74]	A detection model for the workload is developed to characterize workload behavior No mapping between fingerprint data and resource usage
Pattern of requests	Log files of web server	Chains and phases [138]	An algorithm is developed for reduced utilization of hard disk and memory by using workload characteristics Attributes like session and title are not used that could affect characterization
Activities User's profile Publications	Pinterest data	Support vector machine (SVM) [13]	Performed behavior prediction of Pinterest user that could be used for social traffic recommendation

preferred instead of single-point predictions [76]. The focus of the research is on distribution-based prediction, as it works on modeling the workload on the real-valued distribution of features using probability distribution function. It observes the actual values of training data and predicts the probability distribution parameters such as mean, variance over target values, whereas single-point distribution which refers to accurate workload performance prediction and prediction accuracy is challenging and not adequate. In [5], learning-based models are applied for predictions of different workloads. In the literature, the prediction is performed for time ranges [59], throughput and response time [4], multiple performance metrics [54] and other attributes in large-scale data repositories. The details of the techniques and approaches of workload management that support self-prediction are discussed below.

4.2.1 Binary tree

The study [46] proposed a framework called as PSP to predict workload shifts from DSS to OLTP type. PSP has two modules, i.e., online and offline. Psychic is offline, and Skeptic is an online module. Normally, the PSP works in offline fashion. However, when the shift is due, it works online. The offline prediction model is built using polynomial regression technique. The self-healing and self-optimizing characteristics make it autonomic. PSP cannot manage drastic changes of the workload and is used for scheduled tasks. The work [59] provides a prediction model for query execution time in a warehouse environment. It uses binary tree algorithm to build predictions of query runtime (PQR) tree that predict the query execution time using execution plan and system load. PQR is developed in two steps, first is to obtain a PQR tree and second is to predict time range for a new query and is updated periodically.

4.2.2 What-if model

The studies presented in [142, 143] developed a testbed named as "*Ursa Minor*" for workload prediction. It is a cluster-based storage system and has two components, i.e., *Observer* and *Stardust*. These components are developed using what-if model. The system is designed in such a way that it can be used for existing systems as well as for the new systems through modeling infrastructure. The *Observer* is an expectation-based model for performing predictions, and *Stardust* is developed for shared or distributed systems. *Ursa Minor* is scalable due to cluster-based technique and provides adaptive behavior through online choice. The system performance is much higher; however, re-encoding process takes some extra time. To predict the response time and throughput dynamically, DB resource advisor [4, 101] has been developed in SQL Server. It predicts the status of resources using what-if model. The Resource Advisor is experimented on OLTP workload, and changes in the workload are predicted accurately. Resource allocation by the advisor is done through continuous monitoring. It has a limitation that when the size of the buffer pool is low, the advisor has more overhead per transaction.

4.2.3 Exploratory model and confirmatory model

The study [88] presents the requirement of workload models for ADBMSs that includes exploratory and confirmatory models. Workload monitoring is done in the exploratory model, and for workload analysis, the confirmatory model is used. Many data mining and machine learning techniques are applied to the concept of the requirement of exploratory and confirmatory models for the ADBMS. Some issues of the proposed work are presented such as efficient monitoring, storage of workload model and maintenance of the models. The proposed study has no generic solution and is performed in a test scenario.

4.2.4 Machine learning

A number of machine learning techniques have been applied to perform prediction. Principal component analysis (PCA) [54] is applied for the prediction that is used for singular value decomposition or eigenvalue of a data matrix and presents the variance of data in the best way. PCA performs well when it is implemented independently for two datasets and it has limitations, i.e., it cannot identify the relationships. Another approach known as canonical correlation analysis (CCA) can find the correlation between the sets using Euclidean vector spaces. Similar queries are found based on the high value of Euclidean dot product. It has a limitation whereby the queries are textually the same but may have different performances. The study [39] applied fuzzy logic for developing fuzzy rules by assuming the parameter values on the TPC-C dataset, and the study [10] applied Kernel canonical correlation analysis (KCCA) that is based on CCA.

4.2.5 Statistical and cost-based models

The study [136] used a modular approach for estimation of the execution time of SQL query, and its execution plan is mapped for planning the elementary steps. The execution time of many DB access, IO access is evaluated through this approach. Real applications are used to evaluate through TPC-H benchmark for transactional and analytical workloads. Cost-based analyzer [154] elements are used to predict the execution time of query in Postgres through query execution plan, and associated queries are used to estimate the optimizer's cost of query parameters. The study [153] predicts the number of CPU-disk visits and CPU-benefit time for large data size, and the study [43] measures the idle time of concurrent queries by using the I/O access time spent. The query execution time was predicted for simultaneous workload using query fingerprints in [42].

4.2.6 Performance model

The study [30] presents the performance modeling of complex queries of various sizes using three benchmarks TPC-W, RUBBoS and RUBiS. The work describes that execution time of all queries is not important to be predicted and only about 20% of the queries are performance sensitive. The study [78] describes the performance modeling for nine different applications by developing an automatic framework COMPASS for performance prediction of CPU. Analytical performance modeling was performed by PEMOGEN [15] in [139] for relieving the burden and performance modeling during the program execution. For predicting query response time, a neural system-based tool was developed. Queries predictions are done on benchmarks dataset TPC-DS for evaluating long and short running queries [156]. In OLTP system, the association between system throughput and query response time is analyzed by proposing performance prediction approach for concurrent database workload using machine learning approach for query scheduling, resource allocation and QoS management [95].

The study [69] presented an approach through a feature space which uses kernel properties to predict response time using machine learning in a grid environment. The work [131] described a performance prediction systems taxonomy and provided many configuration

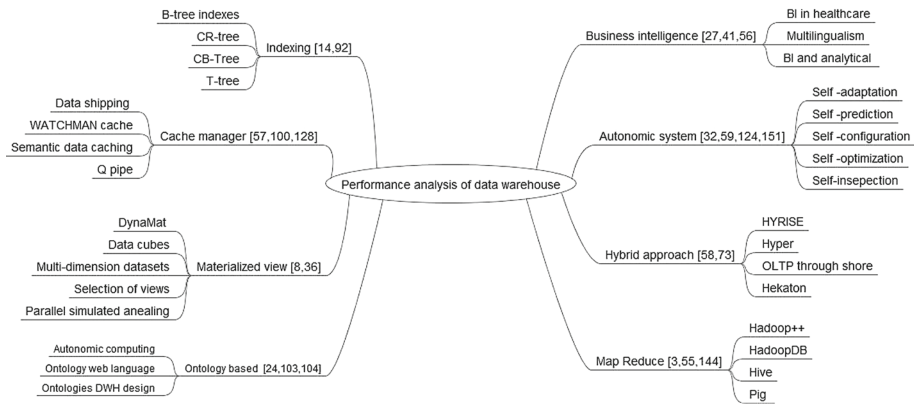

Fig. 7 Approaches to performance modeling in data warehouses

parameters that affect and control the performance of DBMS and DW. A layered AC framework was provided for performance prediction [75]. The resource allocation issue is handled by providing a framework that does performance analysis and prediction of OLTP workloads using performance parameters like resource consumption, resource bottlenecks and throughput [97]. The reputation-based scheduling was provided for distributed systems to predict the future behavior [35]. The performance prediction of SPARQL query was presented in [61] that links data in a decentralized manner where execution time is obtained through machine learning with 84% accuracy. Similarly, run-time behavior [141] was predicted through the regression modeling that removes manual observation of runtime information which helps in developing complex structures. The study [83] reduced the query response time through scheduling the queries by proposing efficient scheduler technique.

Many studies are available in the literature regarding performance modeling in data warehouses. Performance modeling is done in various ways, which include business intelligence [41], map-reduce [144], materialized view [8], indexing [92], ontology-based [104], autonomic system [110], cache managers [128] and hybrid approaches [73]. Figure 7 shows a variety of approaches that have been used for modeling the performance in DWs. There exist two types of indexing which are performed on DWs. First is main memory indexes such as B+-Trees and T-Trees, and the other is cache-conscious indexes such as CSS-Trees and CSB+-Trees [14, 92]. Literature reveals that cache-conscious indexes performed better compared to main memory indexes. DWs performance was improved by developing cache manager called WATCHMAN [57, 100] by using well-known techniques including least recently used (LRU), most recently used (MRU), Q-Pipe and CLOCK for cache replacement.

In a DW, selection of materialized view is important and a number of studies were performed to examine the evolutionary algorithms for materialized view selection which includes data cubes, simulated annealing and DynaMat [8, 36]. The semantic web and AC technology are used for DWs cache allocation via decision support system [104], and use of ontologies in designing DWs is discussed in [24, 103]. Business intelligence (BI) is incorporated in data warehousing [41] and is also used for visualization, data exploration [27] and legal context [56]. Many systems have incorporated AC characteristics for improving the performance which includes self-configuration [151], self-optimization [124], self-inspection [16, 23, 25, 50, 101], self-prediction [59] and self-adaptation [32]. For handling OLTP and OLAP, a number of hybrid approaches are proposed which includes Hekaton, HYRISE and Hyper [58, 73] for concurrent processing in DW environment. The map-reduce [144] is the best approach for data processing and analysis of large-scale data repositories and has been implementing in Hive, Hadoop DB and Pig [3, 55]. The study [86] provided customer behavior model for E-commerce that uses state transition graph to manage customer activities. Hill climbing algorithm is used to achieve suboptimal solution at peak time. The study [76] presents the runtime of workload prediction of Hive query through the mixture density networks model.

Table 2 shows the techniques of performance prediction and the predicted attributes for managing the workload.

4.3 Workload adaptation: techniques to support self-adaptation

This section presents the techniques that support self-adaptation in workload management. The changes in the workload are dynamically handled through the self-adaptation ability of the system. In a real environment, the workload behavior can evolve at any time due to the workload versatility in large-scale data repositories. The behavior of the workflow entering into the system is nondeterministic, and the trend of behavior may change with time. Data repositories workflow can be OLTP or DSS, and it experiences a mix of OLTP and DSS. Many studies have been done to create adaptive systems. Many areas of computer science are explored for the applications of workload adaptation such as web services [93, 94] and DBMSs [66, 118, 119, 130, 153]. Legacy systems can be enhanced for improved efficiency using self-adaptation that has the ability to retain the modifications in the system according to changes in the environment. Many adaptation approaches have been proposed in workload management such as threshold approach, heuristic approach and performance model approach. The performance modeling approach is proved as the best adaptation approach. The details of these approaches are discussed below.

4.3.1 Heuristic approach

The application of stochastic search techniques was applied in earlier research that incorporated the concept of randomness [32]. The limitation of heuristics approach is reduced through the use of randomness of stochastic search and is used for the self-adaptation system. A prototype is developed for stochastics search, and search-based software engineering is performed to verify the claim.

4.3.2 Threshold control approach

DB2 query patroller (QP) [34] is used to control the requests dynamically according to available resources. The execution of small and large priority queries is done with no delay, and the completion of queries and trends is provided by QP. Different system and user-level privileges are assigned using QP by the DBA. QP acts as a gatekeeper and monitors the workload for optimal resource utilization. Based on control theory principles, query prioritization is performed in [102]. The study [11] presented Teradata's active system management (ASM) that manages the workload through resource allocation to different groups using preventive approach.

Table 2 Performance prediction techniques and the predicted attributes in workload management

Input	Attributes predicted	Techniques used	Purpose of research and limitations
Code for tracking the usage of resource like memory, CPU and I/O	Throughput Response time	What-if model [101]	A resource advisor is developed that monitors performance continuously; however, it affects the system performance The transaction overhead becomes higher upon reducing the buffer pool
Load feature vector Query plan Query cost estimate by the optimizer	Time ranges	Binary tree [59]	Time of a day is not considered in the study Abrupt change in behavior of workload is not handled due to data skewness
Number of aggregation columns Number of equijoin predicates Number of nested subqueries Number of sort columns Number of selection predicates Number of join predicates Number of nonequijoin predicates Number of nonequality selection predicates Number of equality selection predicates	Message bytes Elapsed time Message count Disk I/O Records used Records accessed	Kernel canonical correlation analysis (KCCA) [54]	When comparing each data point with all other points the training process takes much time, and in K-nearest neighbor (KNN), it is difficult to decide the number of K Computing cost is directly proportional to number of data points and takes cubic time for computing dimension of correlation Evolving workload behavior is not handled which is expected in workload Degrades the performance on continuous retraining due to distance computation for each workload

Table 2 continued

Input	Attributes predicted	Techniques used	Purpose of research and limitations
TPC-C workload	Response time Hit_Ratio DB_Cache_Size Large_Pool_Size Buffer-Cache	Fuzzy logic [39]	The study assumes the attributes values for the fuzzy rules
Sort-merge join operators TPC-H workload Sort Nested loop join Hash join Aggregate	Execution time	Cost-based optimization model [136]	The study does not support both OLTP and analytical type of workload
RUBiS RUBBoS TPC-W	Execution time	Performance prediction model [30]	For cloud applications OLAP technology could be more investigated
Amazon EC2	Performance prediction	Performance prediction framework, Ernest [148]	The study provides integration of algorithm attributes with performance features of clustering metric
9 different benchmarks	CPU runtime	Aspen model, COMPASS prototype system [78]	Analytical performance model is generated and used for appropriate online decision of performance question in advance intelligent systems
TPC-DS workload	Response time	Neural network-based methods [156]	The proposed system is not tested on multiple databases
TPC-W workload	QoS attributes	Hill climbing approach [86]	The study predicts QoS features and provides workload configuration
TPC-H	Runtime and CPU prediction	MDN model [76]	Using MDN model better results are produced; however, additional resources are required

4.3.3 Performance model approach

The QoS controller is developed for managing the workload for e-commerce applications [93, 94]. The performance such as the probability of rejection, average throughput and average response time is achieved through QoS controller by parameter adjustments. The QoS controller predicts the performance using queuing model, and adaptation is performed through performance modeling. The study [112] presents the architecture of cluster-based web services, and for each gateway, the workload is divided, in-service classes. The services opt resources with respect to MPLs, and the service classes are maintained in the performance model through a feedback control component. The study [130] proposed a framework for scheduling the concurrent requests to achieve QoS through response time as service level agreement (SLA). On concurrent requests, the external queue management system (EQMS) is used to limit the multi-programming level (MPL) that provides an adaptive response to the MPL advisor and scheduler for dynamically managing the workload. It can handle all types of workload by imposing a limit on MPL by reducing the contention. The study [64] presented a self-tuning system called starfish and is used for big data analytics. Another self-tuning system is proposed in [26] that has the capabilities of auto-tuning and self-assessment. The study [63] discussed the characteristics of workload intensity behavior (WIB) by presenting a self-adaptive approach that presents the autonomic classification and prediction methods. Workload classification and forecasting (WCF) is developed to connect the WIB groups with existing prediction methods. Similarly, the studies [62, 96, 117] provide runtime models that provide adaptation in a self-managed way. Self-adaptive approach has also been developed to maintain performance requirement and resource efficiency automatically by using Descartes Modeling Language (DML) to model runtime adaptation. The study [53] provides self-adaptation approaches for the distributed systems, and [107] provides a solution for performance tuning through self-adaptation using context-aware model.

4.3.4 Machine learning approach

Machine learning approaches are widely used for performance tuning. The study [66] presents the comparison of artificial neural networks (ANNs) with natural neural networks (NNNs). The ANNs perform better than NNNs, in two steps of the adaptation process; first is the analysis and training data and second is NN topology development and computations of weights. The study [82] presents AdaptDB that adaptively partition the distributed joins and provides good performance. Hyper-joins algorithm is developed where data shuffling is avoided on the join attribute. The adaptation is performed well for the experiments performed on TPC-H and CMT datasets. Similarly, the study [60] presented AdPart that is used for data partitioning through hashing. The study [106] presents the context-aware system with a self-adaptation ability that performs well for unknown context. The KNN is used in [7] to develop AQWA for an adaptive workload-aware approach for partitioning huge data. When the workloads are executed, the partitions are updated in AQWA and no prior knowledge is needed about the workload. The fuzzy inference system (FIS) is also used for self-tuning in databases management system [126]. The performance parameters include database size, number of users and buffer/hit ratio. Fuzzy rules are developed and evaluated through TPC-E and TPC-C benchmark data. The study [133] developed a technique for increasing robustness and accuracy for prediction algorithm. The study [81] presented an adaptive approach for workload prediction that categorizes the workload into classes based on workload features. The evaluations are performed on Google clusters and compared with machine learning techniques such as support vector machine, ARIMA and linear regression.

As the workload varies dynamically with time, if the change is not properly managed or controlled by the system, it causes overload. The study [111] provides the mechanism for controlling the admission of database workload by proposing a predictive and reactive model. The study [127] provides a methodology based on data partitions for improving the performance of large data and automation of partition process [146]. The study [85] provides a framework WiSeDB for managing the workload. A decision tree is used for scheduling, resource provisioning and query replacement decision. This approach performs well with little re-training and achieves performance goals by adapting offline models which provides better cost trade-offs.

Recently, deep learning is used in different areas of research and produces better results as compared to flat learning. A few studies [1, 106, 115, 135, 149] that are found in the literature have investigated and applied deep learning in large-scale data repositories. The architecture of Peloton [115], the first self-driving DBMS and autonomous operation, is performed in the Peloton. Deep learning framework is used for workload forecasting. Due to advancement in deep learning, adaptiveness can also be achieved. The paper [149] describes that database applications can take advantage of deep learning and system performance can be improved. They discussed possible improvements by jointly applying database techniques and deep learning techniques. Table 3 presents the different workload adaptation techniques used in DBMS workload management.

4.4 Comparative analysis

Workload management in large-scale repositories is related to effective monitoring of the workflow and controlling queries in an efficient way to achieve the performance. The literature reveals that a number of researchers in industry and academia have designed and developed advanced technologies for workload management and these developments have also been incorporated in commercial products of large-scale data repositories. As autonomic workload characterization is a requirement for automatic configuration and performance tuning, the literature reveals that several studies have been conducted to autonomically characterize the DBMS workload [2, 44, 46–48, 157]. These studies have characterized workload/queries autonomically for OLTP and DSS type. However, DBAs need to manage mix type of workload (mix of OLTP and DSS) experienced by the DBMSs [2, 35, 45, 147]. The autonomic detection and classification of mix type of workload are significant to be performed. As the behavior of workload is nondeterministic [2], while tuning the DBMS, the same configuration setting cannot be used for mix type of workload to achieve optimal performance [144].

Adaptation is also an important aspect of workload management that has been investigated by many researchers. They have developed some adaptive frameworks and models using queuing theory, machine learning approaches. The adaptive frameworks and models are good; however, these could be further investigated to improve their accuracy, reliability and robustness. Variety of techniques are applied to achieve the AC capabilities for workload management. Most of the workload management problems are solved using the techniques as discussed in Tables 1, 2 and 3. Based on the literature, little work was found on deep learning in large-scale data repositories. Deep learning can enhance the learning abilities of the data repositories, making them more intelligent, and performance tuning can be achieved. The combination of deep learning techniques and large-scale data repositories techniques can result in applications that are more autonomous with predictive and adaptive capabilities.

The guidelines for the future research for system workload classification, prediction and adaptation in large-scale data repositories are as follows. There is a need to develop frame-

Table 3 Workload adaptation techniques used in workload management

Input used	Adaptation goal	Techniques used	Purpose of research and limitations
MPLs Query costs	Workload admission control	Query patroller (QP), DB2 [34, 80]	Performance objectives are not used as guide in DB2 QP
MPLs Number of users	Workload management using rules based upon workload threshold	Active system management (ASM), Teradata [11]	The resource requirements as per each performance class are not predicted by ASM, so its resource allocations do not reflect the actual requirement of each class
Training dataset	Investigation of training dataset, NN modeling and calculations of weights	ANN [66]	ANN modeling is very difficult
Response time as an SLA	To schedule MPL setting, performance monitoring and self-optimization adaptively	Queuing theory [130]	Produced much delay in managing the workload
SLO	User satisfaction	Layered queuing network modeling [119]	Client distribution is not managed Low-performance prediction accuracy
SLO	To achieve SLOs by enhancing accuracy of performance prediction	Kalman filter [108–110]	The conflict in resource optimization and user satisfaction creates problems in workload adaptation
DSS workload OLTP workload	To achieve adaptation of concurrent workload execution when an unknown context enters in the context-aware system	Context-aware model [107]	Context-aware database performance tuning is not performed
Application services	Provide runtime workload management model for managing heterogeneity	MAPEK model [117]	The proposed framework is not evaluated
Resources and customer workload	Automatic efficient performance management of heterogeneous resources	Holistic model-based approach [68]	Cost factor is not considered

Table 3 continued

Input used	Adaptation goal	Techniques used	Purpose of research and limitations
TPC-H queries	To improve the performance of the query	Hyper-join [82]	Proposed AdaptDB repartitions, on query execution, the parts of tables based on attributes of query
Twitter data KNN queries	Adaptation of variations in data	Adaptive and workload-aware approach, AQWA [7]	The study handles a number of performances issues of Hadoop AQWA extension to spatial join queries needs to be investigated
Database size Buffer/hit ratio Number of users	Performance improvement for achieving business goals and cost reduction	Fuzzy-based approach [126]	The study analyzes the performance measures of scalability perspective
TPC-H queries	Performance improvement and cost minimization for resource allocation and scheduling	Decision tree model, WiSeDB [85]	Handle the changes in the workload caused by dynamic constraint

works, models and architectures that have the ability to solve all these workload management problems. The existing solutions can handle the changing behavior of workload to some extent; however, the future solutions should cater sudden changes that occur in the workload which are experienced in the real applications. Adaptation of changes in small-scale repositories is manageable; however, large-scale repositories are difficult to manage. Therefore, solutions are required for large-scale repositories when the data size and space complexity grow up till certain limit. The calibration of the learnt models can play important role in improving the proficiency of the system. The calibration could be performed by applying machine learning techniques that can learn from the existing solutions and provide more realistic solutions to the workload management problems. Due to the large volume of data repositories, deep learning can be applied to analyze the data in more depth. Similarly, existing machine learning techniques need to be further investigated for better workload management and performance tuning.

This study has focused on three autonomic characteristics, i.e., self-inspection, self-prediction and self-adaptation of the workload management. Further studies could be performed for other autonomic characteristics such as self-protection, self-configuration, self-healing and self-optimization to find their corresponding solutions. Moreover, studies can be conducted to get insights on other autonomic characteristics that could be incorporated to enhance the workload management. The accuracy and efficiency are the important aspects of the workload management. Algorithms need to be designed that could improve data storage and retrieval from the data repositories. More solutions are required in workload characterization for OLTP, DSS and mix type of workload, through data mining techniques for efficient retrieval. The predictive and adaptive solutions will be helpful for resource utilization in large-scale data repositories. The scheduling, resource allocation and de-allocation algorithms could improve the performance of DBMSs and DWs by using the predictions. Workload management solutions are also required for the Internet of things (IoT) by providing adaptive frameworks for the dynamic and heterogeneous environment.

A number of DBMSs such as Oracle, SQL Server, MySQL, DB2 and DWs need to be examined to see the insight on the overall functionality and components-wise functionality that supports workload management. For example, storage optimization could be done for efficient retrieval. The vendors of the large-scale data repositories could incorporate autonomic workload management to improve the performance of their products. NoSQL databases are gaining attention and providing solutions for structured, unstructured and semistructured databases. Further work could be done by investigating the workload management perspective in complex and unstructured databases. In a distributed database environment, workload classification, prediction and adaptation could be of further interest by considering a number of factors like complex workloads, number of users, network congestion. Many proposed studies for workload management are performed through testing and simulations; however, these studies need to be tested in large-scale real environments. For workload classification, the techniques that support self-inspection include statistics, numerical fitting, prediction model, clustering, decision tree induction and CRT. For workload performance prediction, the techniques that support self-prediction are a binary tree, what-if model, exploratory and confirmatory model, machine learning, statistical and cost-based models, and performance model. Similarly, for workload adaptation, the techniques that support self-adaptation are heuristic, threshold control, performance model and machine learning.

The workload management problems also exist in other emerging domains such as web browser, mobile devices, online social networks, microservices, online transaction processing systems, video services and cloud environment. The nature of workload is different in various domains; therefore, its management should also be performed accordingly. These domains need to be explored by investigating their workload management issues and challenges. There is a need for solutions for autonomic workload management with respect to classification, prediction, adaptation and other AC characteristics in the corresponding domains. Other research areas such as distributed databases, cloud computing, IoT and big data analytics could be explored with regard to AC characteristics. Moreover, applications of machine learning, including deep learning techniques, need to be examined for workload management problems in large-scale data repositories.

5 Research issues in large-scale data repositories

The literature reveals that in large-scale data repositories, various models, architectures and frameworks have been developed for the classification of workload, its performance prediction and its adaptation. AC technology supports the large-scale data repositories to be autonomic and has potential to be incorporated. Literature shows that systems are made partially autonomic by incorporating one or more self-* properties (self-inspection, self-configuration, self-healing, self-prediction and self-adaptation etc.). It is challenging to incorporate self-* properties to make it fully autonomic. In this section, the issues and limitations of the existing techniques are discussed. Workload classification in the literature characterizes the workload with less accuracy due to missing important attributes which ultimately leads to less effective and inaccurate classification. While managing the workload, it is important to know what type of workload is entering into the system. The type of workload can be predicted on the basis of workload characteristics; thus, existing classification approaches can be further improved.

A number of questions arise about workload types in workload management. For instance, which workload types need characterization? How should the workload be classified based on its type? What are potential techniques to improve workload classification and how to optimize the results? As mix workload type is essential to handle, what is the percentage of different workload types in a mix-type workload? If we know this in advance, the distribution of execution can be handled efficiently. The performance of queries depends on a number of metrics. Literature reveals some of the performance metrics (see Table 2). In large-scale data repositories, performance metrics such as memory, communication cost and locks need to be predicted for performance optimization, scheduling, resource allocation and adaptation. Before workload execution, many questions arise for managing the workload in large-scale data repositories, such as when should a query be executed? How much can we delay the query execution? In case of a problematic query, should we kill the workload? How will the query perform?

For performance tuning, workload prediction and adaptation are required for capacity planning, system sizing and resource allocation. For system sizing problem with a time constraint, a number of questions arise such as how much memory, network bandwidth and processing power are required to execute the workload? Another problem is the capacity planning for the workload, for which questions arise related to up-gradation or down-gradation of the system? A number of questions related to resource allocation arise. For instance, what is the workload execution time for resource contention queries? How much is the total workload and how many concurrent requests exist? What is the number of bytes sent and received for workload execution? What is the number of disk input/output requests and the cost of communication? How much buffer size is required and how many disk writes are performed for workload execution? As the behavior of workload is nondeterministic, therefore the adaptation of workload can enhance the system performance. Workload adaptation effectiveness and accuracy have also become the challenges in large-scale data repositories. As the workload is nondeterministic and the behavior of the workload may change with time, the following questions arise with respect to workload adaptation. How can the changes be adapted to the changing behavior of workload? How and when the change occurs in the workload behavior? What are the techniques to achieve the optimal results in workload adaptation?

6 Conclusion and future work

This study has provided the state-of-the-art literature survey on autonomic workload management in large-scale data repositories. The survey organized the literature in a way that provides guidelines for researchers working on large-scale data repositories for workload management with respect to classification, performance prediction and adaptation. Limitations found from the literature are highlighted in different domains of workload management. Early studies applied traditional techniques for classification, performance prediction and adaptation of workload in large-scale data repositories. Current studies focus on autonomic approach to manage the workload intelligently without human intervention. However, the literature reveals that only limited solutions exist. The survey also highlights the needs for developing and building new autonomic models, frameworks and architectures. Machine learning techniques can be used for developing new frameworks and models. It was also found that deep learning, which is an emerging learning technology, has been applied in workload management.

Our future work includes developing a framework that characterizes the workload, predicts the performance and adapts the workload autonomically. We will explore machine learning techniques that can best solve the workload management problems. Autonomic characteristics will be investigated to make the system more autonomic to alleviate the burden of DBA.

Acknowledgements The study is funded by COMSATS University Islamabad (CUI), Islamabad, Pakistan, under CIIT/ORIC-PD/17. We appreciate the suggestions and comments of esteemed reviewers that helped in improving the quality of paper.

References

1. Abadi M et al (2016). TensorFlow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467
2. Abdul M, Muhammad AM, Mustapha N, Muhammad S, Ahmad N (2014) Database workload management through CBR and fuzzy based characterization. Appl Soft Comput 22:605–621
3. Abouzeid A, Bajda-Pawlikowski K, Abadi D, Silberschatz A, Rasin A (2009) HadoopDB: an architectural hybrid of MapReduce and DBMS technologies for analytical workloads. Proc VLDB Endow 2(1):922–933
4. Agrawal S, Chaudhuri S, Kollar L, Marathe A, Narasayya, V, Syamala M (2005) Database tuning advisor for microsoft SQL server, In: The proceeding of the 30th VLDB conference, pp 1110–1121
5. Akdere M, Cetintemel U, Riondato M, Upfal E, Zdonik SB (2012) Learning-based query performance modeling and prediction. In: IEEE 28th international conference on data engineering (ICDE), pp 390–401
6. Alvarez GP, Chau WJ (2016) Scenario-aware workload characterization based on a max-plus linear representation. In: International conference on formal modeling and analysis of timed systems, Springer International Publishing, Berlin, pp 177–194
7. Aly AM, Mahmood AR, Hassan MS, Aref WG, Ouzzani M, Elmeleegy H, Qadah T (2015) Aqwa: adaptive query workload aware partitioning of big spatial data. Proc VLDB Endow 8(13):2062–2073

8. Aouiche K, Darmont J (2017) Index and materialized view selection in data warehouses. arXiv preprint arXiv:1701.08029
9. Awad M, Menasc DA (2015) Automatic workload characterization using system log analysis. In: Computer measurement group conference on performance and capacity, San Antonio, TX, USA
10. Bach FR, Jordan MI (2002) Kernel independent component analysis. J Mach Learn Res 3(Jul):1–48
11. Ballinger C (2002) Introduction to teradata's priority scheduler, http://www.teradatalibrary.com/pdf/eb3092.pdf. Accessed 16 May 2018
12. Benevenuto F, Rodrigues T, Cha M, Almeida V (2012) Characterizing user navigation and interactions in online social networks. Inf Sci 195:1–24
13. Bernardini C, Silverston T, Festor O (2014) A pin is worth a thousand words: characterization of publications in pinterest. In: IEEE international conference on wireless communications and mobile computing (IWCMC), pp 322–327
14. Bernstein PA, Das S, Ding B, Pilman M (2015) Optimizing optimistic concurrency control for tree-structured, log-structured databases. In: Proceedings of the ACM SIGMOD international conference on management of data, pp 1295–1309
15. Bhattacharyya A, Hoefler T (2014) Pemogen: automatic adaptive performance modeling during program runtime. In: 23rd international conference on parallel architecture and compilation techniques (PACT), pp 393–404
16. Bruno N, Chaudhuri S (2007) An online approach to physical design tuning. In: IEEE 23rd international conference on data engineering (ICDE), pp 826–835
17. Calzarossa MC, Massari L (2011) Analysis of web logs: challenges and findings. In: Performance evaluation of computer and communication systems. Milestones and future challenges, Springer, Berlin, pp 227–239
18. Calzarossa MC, Massari L, Tessera D (2016) Workload characterization: a survey revisited. ACM Comput Surv (CSUR) 48(3):48
19. Calzarossa MC, Tessera D (2014) Multivariate analysis of web content changes. In: IEEE/ACS 11th international conference on computer systems and applications (AICCSA), pp 699–706
20. Calzarossa MC, Tessera D (2015) Modeling and predicting temporal patterns of web content changes. J Netw Comput Appl 56:115–123
21. Carbunar B, Potharaju R (2015) A longitudinal study of the Google app market. In: IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), pp 242–249
22. Cha M, Benevenuto F, Ahn YY, Gummadi KP (2012) Delayed information cascades in Flickr: measurement, analysis, and modeling. Comput Netw 56(3):1066–1076
23. Chandramouli B, Bond CN, Babu S, Yang J (2007) Query suspend and resume. In: ACM proceedings of the 2007 ACM SIGMOD international conference on management of data, pp 557–568
24. Chang X, Terpenny J (2009) Ontology-based data integration and decision support for product e-design. Robot Comput Integr Manuf 25(6):863–870
25. Chaudhuri S, Kaushik R, Pol A, Ramamurthy R (2007) Stop-and-restart style execution for long running decision support queries. In: Proceedings of the 33rd international conference on very large data bases, VLDB endowment, pp 735–745
26. Chaudhuri S, Weikum G (2000) Rethinking database system architecture: towards a self-tuning RISC-style database system. In: VLDB, pp 1–10
27. Chen H, Chiang RH, Storey VC (2012) Business intelligence and analytics: from big data to big impact. MIS Q 36(4):1165–1188
28. Cheng X, Liu J, Dale C (2013) Understanding the characteristics of internet short video sharing: a YouTube-based measurement study. IEEE Trans Multimed 15(5):1184–1194
29. Chetsa T, Landry G, Lefevrem L, Stolf P (2014) A three step blind approach for improving high performance computing systems' energy performance. Concurr Comput Pract Exp 26(15):2612–2629
30. Chi C, Zhou Y, Ye X (2013) Performance prediction for performance-sensitive queries based on algorithmic complexity. Tsinghua Sci Technol 18(6):618–628
31. Chiba T, Onodera T (2016) Workload characterization and optimization of TPC-H queries on Apache Spark. In: IEEE international symposium on performance analysis of systems and software (ISPASS), pp 112–121
32. Coker Z, Garlan D, Le Goues C (2015) SASS: self-adaptation using stochastic search. In: IEEE/ACM 10th international symposium on software engineering for adaptive and self-managing systems (SEAMS), pp 168–174
33. Cyran M, Green CD (2001) Oracle 9i database performance guide and reference. Release 1(9.0): 1
34. DB2 Query Patroller Guide: Installation, Administration and Usage (2003) IBM Corporation
35. de Carvalho Costa RL, Furtado P (2015) Elections and reputation for high dependability and performance in distributed workload execution. IEEE Trans Parallel Distrib Syst 26(8):2233–2246

36. Derakhshan R, Stantic B, Korn O, Dehne F (2008) Parallel simulated annealing for materialized view selection in data warehousing environments. Lect Notes Comput Sci 5022:121–132
37. Diao Y, Hellerstein JL, Parekh S, Griffith R, Kaiser G, Phung D (2005) Self-managing systems: a control theory foundation. In: Proceedings of the 12th IEEE international conference and workshop on the engineering of computer-based systems, pp 441–448
38. Didona D, Quaglia F, Romano P, Torre E (2015) Enhancing performance prediction robustness by combining analytical modeling and machine learning. In: Proceedings of the 6th ACM/SPEC international conference on performance engineering, pp 45–156
39. Ding Z, Wei Z, Chen H (2017) A software cybernetics approach to self-tuning performance of on-line transaction processing systems. J Syst Softw 124:247–259
40. Do TMT, Gatica-Perez D (2014) Where and what: using smartphones to predict next locations and applications in daily life. Pervasive Mob Comput 12:79–91
41. Dona J, Ortega A, Holgado M (2016) Business intelligence strategy for data warehouse in andalusian health service. InImpact J Innov Impact 6(1):121
42. Duggan J, Chi Y, Hacigumus H, Zhu S, Cetintemel U (2013) Packing light: portable workload performance prediction for the cloud. In: IEEE 29th international conference on data engineering workshops (ICDEW), pp 258–265
43. Duggan J, Papaemmanouil O, Cetintemel U, Upfal E (2014) Contender: a resource modeling approach for concurrent query performance prediction. In: EDBT, pp 109–120
44. Elnaffar S (2002) A methodology for auto-recognizing DBMS workloads. In: Proceedings of the conference of the centre for advanced studies on collaborative research, IBM Press, p 2
45. Elnaffar S, Martin P (2004) An intelligent framework for predicting shifts in the workloads of autonomic database management systems. In: Proceedings of IEEE international conference on advances in intelligent systems–theory and applications
46. Elnaffar S, Martin P (2009) The psychic-skeptic prediction framework for effective monitoring of DBMS workloads. Data Knowl Eng 68(4):393–414
47. Elnaffar S, Martin P, Horman R (2002) Automatically classifying database workloads. In: Proceeding of the ACM conference on Information and Knowledge management, pp 622–624
48. Elnaffar S, Martin P, Schiefer B, Lightstone S (2008) Is it DSS or OLTP: automatically identifying DBMS workloads. J Intell Inf Syst 30(3):249–271
49. Elnaffar S, Powley W, Benoit D, Martin P (2003) Today's DBMSs: How autonomic are they? In: Proceedings of the 14th international workshop on database and expert systems applications, IEEE Computer Society, pp 651–655
50. Elnikety S, Nahum E, Tracey J, Zwaenepoel W (2004) A method for transparent admission control and request scheduling in e-commerce web sites. In: ACM proceedings of the 13th international conference on World Wide Web, pp 276–286
51. Fenacci D, Franke B, Thomson J (2010) Workload characterization supporting the development of domain-specific compiler optimizations using decision trees for data mining. In: Proceedings of the 13th ACM international workshop on software and compilers for embedded systems, p 5
52. Figueiredo F, Almeida JM, Gonçalves MA, Benevenuto F (2014) On the dynamics of social media popularity: a YouTube case study. ACM Trans Internet Technol (TOIT) 14(4):24
53. Florio L (2017) Design and management of distributed self-adaptive systems. Dissertation, Politecnico di Milano
54. Ganapathi A, Kuno H, Dayal U, Wiener JL, Fox A, Jordan M, Patterson D (2009) Predicting multiple metrics for queries: better decisions enabled by machine learning. In: IEEE 25th international conference on data engineering (ICDE), pp 592–603
55. Gates AF, Natkovich O, Chopra S, Kamath P, Narayanamurthy SM, Olston C, Reed B, Srinivasan S, Srivastava U (2009) Building a high-level dataflow system on top of Map-Reduce: the Pig experience. Proc VLDB Endow 2(2):1414–1425
56. George J, Kumar V, Kumar S (2015) Data warehouse design considerations for a healthcare business intelligence system. In: World congress on engineering
57. Gour V, Sarangdevot SS, Tanwar GS (2010) Performance tuning mechanisms for data warehouse: query cache. Int J Comput Appl 2(2):70–75
58. Grund M, Krüger J, Plattner H, Zeier A, Cudre-Mauroux P, Madden S (2010) HYRISE: a main memory hybrid storage engine. Proc VLDB Endow 4(2):105–116
59. Gupta C, Mehta A, Dayal U (2008) PQR: predicting query execution times for autonomous workload management. In: International conference on autonomic computing (ICAC), pp 13–22
60. Harbi R, Abdelaziz I, Kalnis P, Mamoulis N, Ebrahim Y, Sahli M (2016) Accelerating SPARQL queries by exploiting hash-based locality and adaptive partitioning. VLDB J 25(3):355–380

61. Hasan R (2014) Predicting SPARQL query performance and explaining linked data. In: European semantic web conference, Springer, Cham, pp 795–805
62. Heinrich R, Jung R, Schmieders E, Metzger A, Hasselbring W, Reussner R, Pohl K (2015) Architectural run-time models for operator-in-the-loop adaptation of cloud applications. In: IEEE 9th international symposium on the maintenance and evolution of service-oriented and cloud-based environments (MESOCA), pp 36–40
63. Herbst NR, Huber N, Kounev S, Amrehn E (2014) Self-adaptive workload classification and forecasting for proactive resource provisioning. Concurr Comput Pract Exp Wiley 26(12):2053–2078
64. Herodotou H, Lim H, Luo G, Borisov N, Dong L, Cetin FB, Babu S (2011) Starfish: a self-tuning system for big data analytics. CIDR 11(2011):261–272
65. Holze M, Ritter N (2008) Autonomic databases: detection of workload shifts with n-Gram-models. In: ADBIS, vol 8, pp 127–142
66. Horzyk A, Dudek-Dyduch E (2005) Effectiveness of artificial neural networks adaptation according to time period of training data acquisition. In: Intelligent systems design and applications (ISDA), pp130–135
67. Hsu WW, Smith AJ, Young HC (2001) Characteristics of production database workloads and the TPC benchmarks. IBM Syst J 40(3):781–802
68. Huber N, Walter J, Bähr M, Kounev S (2015) Model-based autonomic and performance-aware system adaptation in heterogeneous resource environments: a case study. In: IEEE 2015 international conference on cloud and autonomic computing (ICCAC), pp 181–191
69. Hurault A, Baek K, Casanova H (2015) Selecting linear algebra kernel composition using response time prediction. Softw Pract Exp 45(12):1659–1676
70. IBM (2000) DB2 universal database version 7 administration guide: performance. IBM Corporation, New York
71. Jia Z, Zhan J, Wang L, Han R, McKee SA, Yang Q, Luo C, Li J (2014) Characterizing and subsetting big data workloads. In: IEEE international symposium on workload characterization (IISWC), pp. 191–201
72. Keeton K, Patterson DA (2000) Towards a simplified database workload for computer architecture evaluations. In: Workload characterization for computer system design, Springer, USA, pp 49–71
73. Kemper A, Neumann T (2011) HyPer: A hybrid OLTP&OLAP main memory database system based on virtual memory snapshots. In: IEEE 27th international conference on data engineering (ICDE), pp 195–206
74. Khanna R, Ganguli M, Narayan A, Abhiram R, Gupta P (2014) Autonomic characterization of workloads using workload fingerprinting. In: 2014 IEEE international conference on cloud computing in emerging markets (CCEM), pp 1–8
75. Khattab A, Algergawy A, Sarhan A (2015) MAG: a performance evaluation framework for database systems. Knowl Based Syst 85:245–255
76. Khoshkbarforoushha A, Ranjan R (2016) Resource and performance distribution prediction for large-scale analytics queries. In: Proceedings of the 7th ACM/SPEC on international conference on performance engineering, pp 49–54
77. Koehler J, Giblin C, Gantenbein D, Hauser R (2003) On autonomic computing architectures. Research report, IBM Zurich Research Laboratory, Switzerland
78. Lee S, Meredith JS, Vetter JS, (2015) Compass: a framework for automated performance modeling and prediction. In: Proceedings of the 29th ACM on international conference on supercomputing, pp 405–414
79. Liao ZX, Pan YC, Peng WC, Lei PR (2013) On mining mobile apps usage behavior for predicting apps usage in smartphones. In: Proceedings of the 22nd ACM international conference on information and knowledge management, pp 609–618
80. Lightstone SS, Lohman G, Zilio D (2002) Toward autonomic computing with DB2 universal database. SIGMOD Rec 31(3):55–61
81. Liu C, Liu C, Shang Y, Chen S, Cheng B, Chen J (2017) An adaptive prediction approach based on workload pattern discrimination in the cloud. J Netw Comput Appl 80:35–44
82. Lu Y, Shanbhag A, Jindal A, Madden S (2017) AdaptDB: adaptive partitioning for distributed joins. Proc VLDB Endow 10(5):589–600
83. Maghawry EA, Ismail RM, Badr NL, Tolba MF (2014) An enhanced queries scheduler for query processing over a cloud environment. In: IEEE 9th international conference on computer engineering and systems (ICCES), pp 409–414
84. Mahanti A, Carlsson N, Mahanti A, Arlitt M, Williamson C (2013) A tale of the tails: power-laws in internet measurements. IEEE Netw 27(1):59–64
85. Marcus R, Papaemmanouil O (2016) WiSeDB: a learning-based workload management advisor for cloud databases. Proc VLDB Endow 9(10):780–791

86. Marcus R, Papaemmanouil O (2016) Workload management for cloud databases via machine learning. In: IEEE 32nd international conference on data engineering workshops (ICDEW), pp 27–30
87. Huebscher MC, McCann JA (2008) A survey of autonomic computing—degrees, models, and applications. ACM Comput Surv 40(3):1–28
88. Martin P, Elnaffar S, Wasserman T (2006) Workload models for autonomic database management systems. In: IEEE international conference on autonomic and autonomous systems (ICAS), p 10
89. Mateen A, Raza B, Hussain T, Awais MM (2008) Autonomic computing in SQL server. In: IEEE/ACIS 7th international conference on computer and information science (ICIS), pp 113–118
90. Mateen A, Raza B, Hussain T, Awais MM (2009) Autonomicity in universal database DB2. In: IEEE/ACIS international conference on computer and information science (ICIS), pp 445–450
91. Mateen A, Raza B, Sher M et al (2014) Workload management: a technology perspective with respect to self-characteristics. Artif Intell Rev 41(4):463–489
92. Medina JM, Barranco CD, Pons O (2017) Indexing techniques to improve the performance of necessity-based fuzzy queries using classical indexing of RDBMS. Fuzzy Sets Syst. https://doi.org/10.1016/j.fss.2017.09.008. 28 Sep 2017
93. Menasce DA, Barbará D, Dodge R (2001) Preserving QoS of E-commerce sites through self-tuning: a performance model approach. In: Proceedings of the 3rd ACM conference on electronic commerce, Tampa, Florida, USA, pp 224–234
94. Menasce DA, Bennani MN (2003) On the use of performance models to design self-managing computer systems. In: Proceedings of computer measurement group conference, December 7–12, Dallas, TX, USA, pp 1–9
95. Milicevic M, Baranovic M, Zubrinic K (2015) Application of machine learning algorithms for the query performance prediction. Adv Electr Comput Eng 15(3):33–44
96. Moreno GA, Cámara J, Garlan D, Schmerl B (2015) Proactive self-adaptation under uncertainty: a probabilistic model checking approach. In: ACM proceedings of the 10th joint meeting on foundations of software engineering, pp 1–12
97. Mozafari B, Curino C, Jindal A, Madden S (2013) Performance and resource modeling in highly-concurrent OLTP workloads. In: Proceedings of the 2013 ACM sigmod international conference on management of data, pp 301–312
98. Mozafari B, Curino C, Madden S (2013) DBSeer: resource and performance prediction for building a next generation database cloud. In: CIDR
99. Muller H, Klein M, Wood W, O'Brien W(2006) Autonomic computing (CMU/SEI-2006-TN-006) software engineering institute, Carnegie Mellon University http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=7855, Accessed 19 May 2018
100. Müller S, Nica A, Butzmann L, Klauck S, Plattner H (2015) Using object-awareness to optimize join processing in the SAP HANA aggregate cache. In; EDBT, pp 557–568
101. Narayanan D, Thereska E, Ailamaki A (2005) Continuous resource monitoring for self-predicting DBMS. In: International symposium on modeling, analysis, and simulation of computer and telecommunication systems (MASCOTS), pp 239–248
102. Narayanan S, Waas F, (2011) Dynamic prioritization of database queries. In: IEEE 27th international conference on data engineering (ICDE), pp 1232–124
103. Nebot V, Berlanga R, Pérez J, Aramburu M, Pedersen T (2009) Multidimensional integrated ontologies: a framework for designing semantic data warehouses. J Data Semant XIII:1–36
104. Nicolicin-Georgescu V, Benatier V, Lehn R, Briand H (2009) An ontology-based autonomic system for improving data warehouse performances. Int Conf Knowl Based Intell Inf Eng Syst. Springer, Berlin, pp 261–268
105. Nikravesh AY, Ajila SA, Lung CH (2017) An autonomic prediction suite for cloud resource provisioning. J Cloud Comput 6(1):3
106. Nimalasena A, Getov V (2013) System evolution for unknown context through multi-action evaluation. In: IEEE 37th annual computer software and applications conference workshops (COMPSACW), pp 271–276
107. Nimalasena A, Getov V (2015) Context-aware framework for performance tuning via multi-action evaluation. In: IEEE 39th annual computer software and applications conference (COMPSAC), pp 318–323
108. Niu B, Martin P, Powley W (2011) Towards autonomic workload management in DBMSs. In: Theoretical and practical advances in information systems development: emerging trends and approaches, IGI Global, pp 154–173
109. Niu B, Martin P, Powley W, Bird P, Horman R (2007) Poster session: adapting mixed workloads to meet SLOS in autonomic DBMSs. In: IEEE 23rd international conference on data engineering workshop, pp 478–484

110. Niu B, Martin P, Powley, W, Horman R, Bird P (2006) Workload adaptation in autonomic DBMSs. In: ACM proceedings of the conference of the center for advanced studies on collaborative research (CASCON), USA, pp 161–173
111. Oh J, Kang KD (2013) A predictive-reactive method for improving the robustness of real-time data services. IEEE Trans Knowl Data Eng 25(5):974–986
112. Pacifici G, Spreitzer M, Tantawi AN, Youssef A (2005) Performance management for cluster-based web services. IEEE J Sel Areas Commun 23(12):2333–2343
113. Packer AN (2001) Configuring and tuning databases on the solaris platform. Prentice Hall, Upper saddle River
114. Panda R, John LK (2014) Data analytics workloads: characterization and similarity analysis. In: IEEE international performance computing and communications conference (IPCCC), pp 1–9
115. Pavlo A, Angulo G, Arulraj J, Lin H, Lin J, Ma L, Menon P, Mowry TC, Perron M, Quah I, Santurkar S (2017) Self-driving database management systems. In: CIDR 17,Chaminade, California, USA
116. Peters N, Park S, Chakraborty S, Meurer B, Payer H, Clifford D (2016) Web browser workload characterization for power management on HMP platforms. In:IEEE international conference on hardware/software codesign and system synthesis (CODES + ISSS), pp 1–10
117. Poggi F, Rossi D, Ciancarini P, Bompani L (2016) An application of semantic technologies to self adaptations. In: IEEE 2nd international forum on research and technologies for society and industry leveraging a better tomorrow (RTSI), pp 1–6
118. Qian S, Wang S (2010) Research on workload adaptation architecture for DBMS. In: International symposium on intelligence information processing and trusted computing, pp 382–385
119. Qiang Y, Li Y, Chen J (2009) The workload adaptation in autonomic DBMSs based on layered queuing network model. In: Second IEEE international workshop on knowledge discovery and data mining (WKDD), pp 781–785
120. Radinsky K, Bennett PN (2013) Predicting content change on the web. In: Proceedings of the sixth ACM international conference on Web search and data mining, pp 415–424
121. Raza B, Mateen A, Awais MM, Sher M (2011) Survey on autonomic workload management: algorithms, techniques, and models. J Comput 3(7):29–38
122. Raza B, Mateen A, Hussain T, Awais MM (2009) Autonomic success in databases management systems. In: 8th international conference on computer and information science (ICIS), Shanghai, China, pp 439–444
123. Raza B, Mateen A, Sher M, Awais MM, Hussain T (2010) Autonomicity in Oracle database management system. In: IEEE international conference on data storage and data engineering (DSDE), pp 296–300
124. Raza B, Mateen A, Sher M, Awais MM, Hussain (2010) Autonomic view of query optimizers in database management systems. In: IEEE 8th ACIS international conference on software engineering research, management and applications (SERA). pp 3–8
125. Ren Z, Dong J, Ren Y, Zhou R, You X (2016) Workload characterization on a cloud platform: an early experience. Int J Grid Distrib Comput 9(6):259–268
126. Rodd SF, Kulkarni UP (2015) Adaptive self-tuning techniques for performance tuning of database systems: a fuzzy-based approach with tuning moderation. Soft Comput 19(7):2039–2045
127. Rosas C, Sikora A, Jorba J, Moreno A, César E (2014) Improving performance on data-intensive applications using a load balancing methodology based on divisible load theory. Int J Parallel Prog 42(1):94–118
128. Sapia C (2000) PROMISE: predicting query behavior to enable predictive caching strategies for OLAP systems. In: Proceeding of the second international conference on data warehousing and knowledge discovery (DAWAK), pp 224–233
129. Sarkar J, Saha S, Agrawal S (2014) An efficient use of principal component analysis in workload characterization—a study. AASRI Proced 8:68–74
130. Schroeder B, Harchol-Balter M, Iyengar A, Nahum E (2006) Achieving class-based QoS for transactional workloads. In: IEEE proceedings of the 22nd international conference on data engineering (ICDE) pp 153–153
131. Seneviratne S, Levy DC, Buyya R (2013) A taxonomy of performance prediction systems in the parallel and distributed computing grids. arXiv preprint arXiv:1307.2380
132. Seo B, Kang S, Choi J, Cha J, Won Y, Yoon S (2014) IO workload characterization revisited: a data-mining approach. IEEE Trans Comput 63(12):3026–3038
133. Shetty J, Shobha G (2016) An ensemble of automatic algorithms for forecasting resource utilization in cloud. In: IEEE future technologies conference (FTC), pp 301–306
134. Silva T, Almeida JM, Guedes D (2011) Live streaming of user generated videos: workload characterization and content delivery architectures. Comput Netw 55(18):4055–4068
135. Silver D et al (2016) Mastering the game of go with deep neural networks and tree search. Nature 529:484–503

136. Singhal R, Nambiar M, (2016) Predicting SQL query execution time for large data volume. In: ACM proceedings of the 20th international database engineering and applications symposium, pp 378–385
137. Stassopoulou A, Dikaiakos MD (2009) Web robot detection: a probabilistic reasoning approach. Comput Netw 53(3):265–278
138. Summers J, Brecht, Eager D, Gutarin, A (2016) Characterizing the workload of a Netflix streaming video server. In: IEEE international symposium on workload characterization (IISWC), pp 1–12
139. Tallent NR, Hoisie A (2014) Palm: easing the burden of analytical performance modeling. In: Proceedings of the 28th ACM international conference on supercomputing, pp 221–230
140. Tesfatsion SK, Wadbro E, Tordsson J (2016) Autonomic resource management for optimized power and performance in multi-tenant clouds. In: IEEE international conference on autonomic computing (ICAC), pp 85–94
141. Tetzlaff D, Glesner S (2013) Intelligent prediction of execution times. In: IEEE second international conference on informatics and applications (ICIA), pp 234–239
142. Thereska E, Narayanan D, Ailamaki A, Ganger GR, (2007) Observer: keeping system models from becoming obsolete. In: Workshop on hot topics in autonomic computing (HotAC), vol 11
143. Thereska E, Narayanan D, Ganger GR (2006) Towards self-predicting systems: What if you could ask 'what-if'? Knowl Eng Rev 21(3):261–267
144. Thusoo A, Sarma JS, Jain N, Shao Z, Chakka P, Anthony S, Liu H, Wyckoff P, Murthy R (2009) Hive: a warehousing solution over a map-reduce framework. Proc VLDB Endow 2(2):1626–1629
145. Transaction Processing Council (TPC). http://www.tpc.org Accessed 14 May 2018
146. Turcu A, Palmieri R, Ravindran B, Hirve S (2016) Automated data partitioning for highly scalable and strongly consistent transactions. IEEE Trans Parallel Distrib Syst 27(1):106–118
147. Ueda T, Nakaike T, Ohara M (2016) Workload characterization for microservices. In: IEEE international symposium on workload characterization (IISWC), pp 1–10
148. Venkataraman S, Yang Z, Franklin MJ, Recht B, Stoica I (2016) Ernest: efficient performance prediction for large-scale advanced analytics. In: NSDI, pp 363–378
149. Wang W, Zhang M, Chen G, Jagadish HV, Ooi BC, Tan KL (2016) Database meets deep learning: challenges and opportunities. In: ACM SIGMOD record, ACM New York, NY, USA, vol 45, no 2, pp 17–22
150. Wasserman T, Martin P, Skillicorn DB, Rizvi H (2004) Developing a characterization of business intelligence workloads for sizing new database systems. In: Proceedings of the 7th ACM international workshop on data warehousing and OLAP, pp 7–13
151. White SR, Hanson JE, Whalley I, Chess DM, Kephart JO (2004) An architectural approach to autonomic computing. In: Proceedings of the IEEE international conference on autonomic computing (ICAC'04), pp 2–9
152. Wilson C, Sala A, Puttaswamy KP, Zhao BY (2012) Beyond social graphs: user interactions in online social networks and their implications. ACM Trans Web (TWEB) 6(4):17
153. Wu W, Chi Y, Hacígümüş H, Naughton JF (2013) Towards predicting query execution time for concurrent and dynamic database workloads. Proc VLDB Endow 6(10):925–936
154. Wu W, Chi Y, Zhu S, Tatemura J, Hacigümüs H, Naughton JF (2013) Predicting query execution time: Are optimizer cost models really unusable? In: IEEE 29th international conference on data engineering (ICDE), pp 1081–1092
155. Yang J, Qiao Y, Zhang X, He H, Liu F, Cheng G (2015) Characterizing user behavior in mobile internet. IEEE Trans Emerg Top Comput 3(1):95–106
156. Yusufoglu EE, Ayyildiz M, Gul E (2014) Neural network-based approaches for predicting query response times. In: IEEE international conference on data science and advanced analytics (DSAA), pp 491–497
157. Zewdu Z, Denko MK, Libsie M (2009) Workload characterization of autonomic DBMSs using statistical and data mining techniques. AINA workshops, pp 244–249
158. Zhang M, Martin P, Powley W, Chen J (2017) Workload management in database management systems: a taxonomy. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2017.2767044

Basit Raza is working as an Assistant Professor in the Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad, Pakistan. He received his Ph.D. degree in computer science in 2014. He has published a number of conference and journal papers of internal repute. His research interests are database management system, security and privacy, data mining, data warehousing, machine learning and artificial intelligence.

Asma Sher received the master's degree in computer science from the COMSATS University Islamabad (CUI), Islamabad, Pakistan, in 2017. Her area of specialization is computer science, and research interests include databases, data warehouses and machine learning.

Sana Afzal received her master's student from COMSATS University Islamabad (CUI), Islamabad, Pakistan. She received her bachelor's degree in computer science from Arid Agriculture University, Rawalpindi. Her research interest includes machine learning and data mining, databases and data warehouse.

Ahmad Kamran Malik received his Ph.D. from the Vienna University of Technology (TU-Wien), Austria. He has been teaching at Quaid-I-Azam University and now working as an Assistant Professor at COMSATS University Islamabad (CUI), Islamabad, Pakistan. He studied MS in computer science at Muhammad Ali Jinnah University, Islamabad. Currently, his research interest is focused on social network analysis, access control and collaborative systems.

Adeel Anjum is an Assistant Professor in the Department of Computer Sciences at COMSATS University Islamabad (CUI), Islamabad, Pakistan. He completed his Ph.D. with distinction in the year 2013. His area of research is data privacy using artificial intelligence techniques. He has several publications in international conferences. He is also the author of a book on data privacy. He serves in the technical program committees of various international conferences.

Yogan Jaya Kumar is a Senior Lecturer at the Department of Intelligent Computing and Analytic in the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM). He earned both his bachelor degree and master degree from Universiti Sains Malaysia (USM), in the field of mathematical science in the year 2003 and 2005. He completed his Ph.D. studies at Universiti Teknologi Malaysia, in 2014 in the field of computer science. Currently, his research involves in the field of text mining, information extraction and AI applications.

Muhammad Faheem received the B.Sc. computer engineering degree in 2010 from the Department of Computer Engineering at the University College of Engineering & Technology, Bahauddin Zakariya University Multan, Pakistan. In 2012, he received an MS degree in computer science from the Faculty of Computer Science and Information System at Universiti Teknologi Malaysia. Currently, he is a Ph.D. student at Abdullah Gul University, Kayseri, Turkey. His research interest includes the areas of smart grid communications, underwater acoustic communications, wireless ad hoc, sensor networks and cognitive radio networks and information storage and retrieval architecture from sensor memory. Mr. Faheem has authored several papers in refereed journals and has been serving as a reviewer for numerous journals, such as journal of network and computer applications, ad hoc networks, computer communications, computer standards and interfaces and IEEE Access, in above-mentioned areas.