Workload management: a technology perspective with respect to self-* characteristics

Abdul Mateen · Basit Raza · Muhammad Sher ·
M. M. Awais · Norwatti Mustapha

Published online: 3 March 2012

© The Author(s) 2012. This article is published with open access at Springerlink.com

Abstract Rapid growth in data, maximum functionality requirements and changing behavior in the database workload tends the workload management to be more complex. Organizations have complex type of workloads that are very difficult to manage by humans and even in some cases this management becomes impossible. Human experts take long time to get sufficient experience so that they can manage the workload efficiently. The versatility in workload due to huge data size and user requirements leads us towards the new challenges. One of the challenges is the identification of the problematic queries and the decision about these, i.e. whether to continue their execution or stop. The other challenge is how to characterize the workload, as the tasks such as configuration, prediction and adoption are fully dependent on the workload characterization. Correct and timely characterization leads managing the workload in an efficient manner and vice versa. In this scenario, our objective is to produce a workload management strategy or framework that is fully adoptive. The paper provides a summary of the structure and achievements of the database tools that exhibit Autonomic Computing or self-* characteristics in workload management. We have categorized the

A. Mateen (✉)

Department of Computer Science, Federal Urdu University of Arts, Science and Technology,
Islamabad, Pakistan
e-mail: abdulmateen@fuuastisb.edu.pk

B. Raza · M. Sher

Department of Computer Science and Software Engineering, International Islamic University,
Islamabad, Pakistan
e-mail: basitrazasaheed@gmail.com

M. Sher

e-mail: m.sher@iiu.edu.pk

M. M. Awais

School of Science and Engineering, Lahore University of Management Science, Lahore, Pakistan
e-mail: awais@lums.edu.pk

N. Mustapha

Faculty of Computer Science and IT, University Putra Malaysia, Serdang, Malaysia
e-mail: norwati@fsktm.upm.edu.my

database workload tools to these self-* characteristics and identified their limitations. Finally the paper presents the research done in the database workload management tools with respect to the workload type and Autonomic Computing.

Keywords Autonomic Computing · Workload Management · Optimization · Configuration · Inspection · Prediction · Organization · Adoption

Abbreviations

AC	Autonomic Computing
ADDM	Automatic Database Diagnostic Monitor
AWR	Automatic Workload Repository
BI	Business Intelligence
CBMG	Customer Behavior Model Graph
COLT	Continuous On-Line Tuning
DAD	Disk Array Designer
DBA	Database Administrator
DBMS	Database Management System
DSS	Decision Support System
DW	Data Warehouses
DTA	Database Tuning Advisor
MPL	Multiprogramming Level
MV	Materialized View
MWS	Mixed Workload Scheduler
ODRM	Oracle Database Resource Manager
OLTP	On Line Transaction Processing
PAQRS	Priority Adaptation Query Resource Scheduling
PGM	Priority Gradient Multiprogramming
PI	Progress Indicator
PMM	Priority Memory Management
QEP	Query Execution Plan
QP	Query Patroller
QShuffler	Query Shuffler
QUIET	Continuous Query-Driven Index Tuning
REDWAR	RELational Database Workload AnalyzeR
SAA	SQL Access Advisor
SLG	Service Level Goal
SLO	Service Level Objective
TCO	Total Cost of Ownership

1 Introduction and background

The systems, which can execute, adjust and tune themselves in the presence of the huge workload, are called Autonomic Computing (AC) systems. Theme of the AC systems is to focus on what rather than how. The term Autonomic Computing was first time introduced by the IBM in 2001 to describe the systems that can manage themselves without any human interaction. The inspiration of the AC is taken from the human nervous system that performs numerous activities without conscious thought. An AC system controls the functioning of

the computer applications and systems without human intervention, just like the same way human autonomous nervous system regulates the body system without conscious input from the individual. The basic purpose of the AC is to create such systems that have the ability to run themselves with hiding the complexity from the user [Horn \(2001\)](#); [Parashar and Hariri \(2005\)](#). AC is based on a self-managing computing model. The concept of the autonomic or self-management and adoption in computing system is very old. In past, most processes are automatic but afterward it has been realized that the processes should be autonomic [Huebscher and McCann \(2008\)](#); [Beg et al. \(2010\)](#). Automatic means pre-programmed task execution of a system, i.e. system remains in working until something goes wrong and human intervention is necessary for further execution. While autonomic means self-regulation, here system response is also automatic but modulated and system can handle problems itself with no human intervention [Mateen et al. \(2008\)](#). AC is an evolutionary process rather than revolutionary process. The implications for computing are a network of organized computing components that give us what and when we need it without a conscious mental/ physical effort. The basic purpose of the AC is to create such systems that have the ability to manage themselves and hiding complexity from users.

The rest of the paper is organized as follow: Sect. 2 provides an overview of the autonomic workload management. Section 3 provides a discussion of the previous research work that has been done in workload management with respect to the six self-* characteristics, i.e. self-Optimization, self-Configuration, self-Inspection, self-Organization, self-Prediction and self-Adoption. The section also elaborates the shortcomings in the previous work. Section 4 provides the workload types and AC viewpoint of the previous work. Finally Sect. 5 concludes the research and highlights the useful future directions for workload management in Database Management Systems (DBMSs) and Data Warehouses (DWs).

2 Autonomic workload management

The complexity in DBMSs increases due to the various functionality demands from the users, complex data types, diverse workload and huge data volume that are increasing with the passage of time. All these factors cause brittleness and unmanageability in DBMS. To handle this problem organizations hire number of expert Database Administrators (DBAs) and spending lot of money to get expected improvement, throughput and response. Usually DBAs have to take care of all the tasks such as making policy for workload priorities, memory, configuration and other database related tasks as shown in Fig. 1. The cost of the hardware is decreasing but the management cost is increasing day by day. Performing workload management activities manually, by hiring Experts causes increase in Total Cost of Ownership (TCO). Moreover with the advent of distributed systems and DWs, it has become difficult and even some cases impossible for DBA to manually organize, optimize and configure day to day database related tasks. To achieve the better workload management in DBMSs and DWs, various strategies have been introduced. For example, executing queries may be stopped for a while and restarted later on. However, when queries will be stopped, the executed work will be stored in memory otherwise the executed work will be lost even that may be about to complete and will be executed from the scratch when restart.

The workload management consists of three units that are the workload, resources and objectives which are co-related with each other. The workload uses some resources to meet the objectives of the organization. In the other sense resources are allocated through different approaches to the workload to get some management objectives. Workload has evolved through three phases which are capacity planning, resource sharing and performance

Fig. 1 Workload management activities

oriented workload. In the capacity planning cost sharing was the main concept; in resource oriented the idea was maximum resource utilization while in case of performance oriented the focus is on the business objectives. The style of the workload has been changed from the offline analysis to online adaptation. The workload adaptation consists of the workload detection and workload control. Workload is detected through workload characterization and performance monitoring. The workload characterization method is used when performance degrades, by tracking the workload changes proactively while the performance monitoring method is reactive, which take action when performance has been degraded. Three techniques are used to derive the workload control plans which are the Performance Model, Heuristic and Threshold technique.

The unmanageability in workload management can be handled by making the DBMS to self-manage (Autonomic) that can perform its tasks such as memory, configuration, storage management and resource allocation automatically. The benefit of the AC is to manage complexity itself and works according to the set goals and objectives. AC contributes an important role in managing the systems, networks, DBMSs and DWs. Workload management is a main

Fig. 2 Autonomic workload manager

feature in DBMSs and DWs, so it should be autonomic to improve the efficiency and reduce the complexity. Different techniques, models and tools have been developed by the database vendors, researchers and practitioners to handle the workload. These tools are related with the workload characterization, classification, scheduling, Multiprogramming Level (MPL), optimization, prediction, adoption and resource allocation. The technology of the AC in workload management can be used to manage the workload in an efficient and responsive way. Currently, there is need of workload management framework that can manage the workload without affecting other requests, efficient resources utilization and handle all other database related tasks. The autonomic workload management will perform these tasks by collecting information about the workload type, intensity, resource demand and other related parameters with minimum human intervention as shown in Fig. 2. The AC technology has a high potential to be incorporated in current DBMSs and DWs.

3 Self-management in workload management tools

Autonomic workload management should have the ability to exhibit six basic AC characteristics, i.e. self-optimization, self-configuration, self-inspection, self-prediction, self-organization and self-adoption. Self-optimization in the workload management is characteristic where all the tasks related with workload must be executed in an efficient manner even in

the presence of huge and diverse workload. In order to achieve efficiency in the workload management, configuration of different components should be performed in self-manage and appropriate way. Self-inspection in AC workload management supports better decision making by using the knowledge about its resources, limits, intensity etc. Self-prediction in the workload management helps to forecast the different aspects such as resource demand, workload frequency and memory requirements etc for the future decisions. Self-organization in AC workload management allows reorganizing and restructuring the layout of data and indexes in order to get better results. Self-adoption allows adopting the changes in the database workload according to the available resources and the environment. Now we are discussing the workload management tools in DBMSs and DWs with respect to six AC characteristics.

3.1 Self-optimization

It is the characteristic that is responsible to execute the tasks or utilities in an efficient manner. Self-optimization in database workload provides a way to execute the workload in an efficient and organized way according to the available resources and environment. Much of the research has been done in the context of the workload management with respect to self-optimization. Some of the related key research with workload optimization tools in DBMSs and DWs are discussed as under.

Oracle Database Resource Manager (ODRM) by [Rhee et al. \(2001\)](#) allows DBA to logically divide the workload into distinct units and allocates CPU resources to these units without extra overhead. During the peak hours On Line Transaction Processing (OLTP) workload should be given preference over the Decision Support System (DSS) workload and vice versa. ODRM has scheduling mechanism that is used for fixed time interval and controls the number of active sessions executing at a time. When the available slots for active sessions are filled with new sessions, the remaining sessions will be queued until some slot is available. Through ODRM, administrator can define and set the scheduling policies for the workload based on the predicted execution time of a query. The major components of the ODRM are Resource Consumer Group, Resource Plan and Resource Plan directive.

The research by [Mumtaz et al. \(2008a,b, 2009\)](#) discussed the impact of the query interaction over the workload and introduced a framework named as Query Shuffler (QShuffler). The proposed framework schedules the workload by considering the impact of the queries. The QShuffler classifies the queries according to their type and arranges them in an efficient way as shown in Fig. 3. It adopts the non-preemptive scheduling technique and minimizes the dropping requests using shortest remaining time first scheduling technique. The QShuffler is evaluated by experimenting over the TPC-H workload in DB2 and found to be effective for various workloads.

BI Batch manager proposed by [Mehta et al. \(2008\)](#) is introduced for the enterprise DW that takes queries in the form of batches. It consists of three components which are admission control, scheduler and execution control. Admission control works on the basis of memory requirement in the form of batches. A batch consists of those queries whose memory requirement is equal to the available memory of the system. For scheduling, the query with maximum memory requirement will have the highest priority. The manager uses Priority Gradient Multiprogramming (PGM) to prevent under load and over load problems. Finally some experimental results are discussed to evaluate the performance of the BI batch manager that shows its strength.

In Oracle, the automatic SQL tuning is performed through the query optimizer and SQL Tuning Advisor [Dageville et al. \(2004\)](#). Query optimization has great importance especially

Fig. 3 Query Shuffler (Mumtaz et al. (2009))

in case of complex workload. The SQL Tuning Advisor is an interface between the query optimizer and user. It generates tuning recommendations for the SQL statements or workload. These recommendations are provided to user, who either select or reject. When user selects the recommendations it will be stored in SQL profile that is further utilized by the optimizer for generation of best Query Execution Plan (QEP). SQL Tuning Advisor makes decisions on the basis of information that is provided by the query optimizer, Automatic Database Diagnostic Monitor (ADDM) and Automatic Workload Repository (AWR).

Research presented in Mehta et al. (2009) define the design criteria that make a Mixed Workload Scheduler (MWS) and uses it to design rFEED, i.e. MWS that is fair, effective, efficient, and differentiated. A non-preemptive approach is proposed for scheduling as for them it is expensive to preempt small queries that make the bulk of a BI workload. The approach uses optimizer's estimated execution cost as an approximation as authors thought that approximation is sufficient and no need to calculate execution cost again. Moreover, a single queue for the scheduler and multiple queues for execution are used. The research also assumed that all queries have same normalized service level. rFEED is simulated over the real workloads and compare with models of the current best of breed commercial systems.

Surajit et al. (2007) proposed a framework to stop and restart the queries during their execution to manage the workload efficiently. The restart approach re-executes the stopped query from the position where it was stopped. The technique does not save all the executed information but save only the selected information from the past execution to reduce memory overheads. This method also reduces the running time of re-executed queries. The proposed technique is validated by making the experiments over the real and TPC-H benchmark data.

A technique for query suspension and resumption [Chandramouli et al. \(2007\)](#) is discussed where the authors proposed the induction of the asynchronous checkpoints for each cardinality in the query. An optimized plan is proposed for the suspension which dumps the current state to disc and going back to the previous checkpoint. The optimized plan performs its tasks (suspension or resumption) with less overhead by observing the time constraint during suspension. The technique uses hybrid approach for query suspension where suspend time overheads are negligible and due to this better results can be seen. The memory wastage is higher in previous techniques due to switching points and shows worse results for unexpected suspend. The proposed approach is implemented in PREDATOR tool and proven experimentally for simple and heavy workload. It is observed that it meets suspend time constraint and thereby reducing the overhead.

[Schroeder et al. \(2006\)](#) proposed an external scheduling technique for the OLTP workload. They identified main parameters and used feedback controller to select the appropriate MPL automatically that is based on queuing theoretic model and Markov chain. Priorities are also used for the external scheduling. The technique is validated through experiments and observed that external scheduling can be equally effective to internal scheduling when suitable MPL is selected.

3.1.1 Critical discussion

QShuffler by [Mumtaz et al. \(2008a,b, 2009\)](#) considers the impact of queries over each other as it has vital role in performance and proposed an experimental technique to handle the query interaction problem. It gives optimal solution for scheduling of workload as it is based on linear programming. It gives four times performance over the FCFS scheduler of database systems. However in QShuffler, the large jobs have to wait for a long time even these are generated from the higher priority as it uses SJF algorithm for scheduling. The average execution time is larger as QShuffler uses non-preemptive SJF approach. Moreover this approach can be improved if the Service Level Objectives (SLOs) are incorporated with scheduling algorithm. The QShuffler is tested by experiments over the Business Intelligence (BI) and On Line Analytical Processing (OLAP) workload. BI Batch Manager by [Mehta et al. \(2008\)](#) executes the workload in the form of batches to avoid thrashing and provides an optimal solution for all types of workload. This approach does not require any changes in the internals of DBMS to manage small and heavy workload. The manager gets benefit of the added predictability of queries; stabilize the system and less sensitive to estimation errors. It uses feedforward loop that stabilizes the system (from underload or overload) with maximum ability to tolerant the prediction errors. The Feedback control technique samples the performance metric and controls the incoming requests optimally. When performance metric exceeds from the threshold, rate of admitting requests reduces and vice versa. It gives high throughput for commercial and enterprise class DBMSs; however has no ability to handle interactive and ad-hoc queries. The workload varies from time to time so dynamic approach should be adopted. The other problem in the research is that sub-batches are created on the basis of one parameter, i.e. memory only. As the query with largest memory gets higher priority so the queries with small memory will wait for a long time. Due to this reason throughput will be decrease and chance for the occurrence of starvation. As the PGM executes that query first which requires largest memory so the same problem as with LMP will also occur in PGM. The suggested methodology in [Mehta et al. \(2009\)](#) uses non-preemptive scheduling scheme that gives very inefficient results for time critical systems due to its poor responsiveness and ultimately moves towards starvation and hanging. The approximate values are used in the research that never reflects the actual results. Optimizer's estimated execution cost is used but in real life

approximate values never give the actual results. Single queue is maintained for scheduling, so global optimization cannot be achieved. The proposed methodology assumes the same service level for all requests but in real life all the queries do not have the same service level. The approach set the MPL statically and does not consider the interaction among different queries. SQL Tuning Advisor by [Dageville et al. \(2004\)](#) improves the execution plans through SQL Profiling concept; and on the basis of cost-based access paths and what-if analysis it generates tuning recommendations for the incoming workload. It has a very strong analysis capability where it performs a number of analyses such as estimate, access, parameter setting, statistical and SQL structure analysis. Whenever query optimizer fails due to heavy or complex workload, it assists by stabilizing the system and generating the QEPs. The approach discussed in [Chaudhuri et al. \(2007\)](#) is limited to handle single QEP and does not consider parallel QEPs. There is no dynamic way in proposed technique to maintain the restart plan during the modification of the source records (past executed records). The technique proposed by [Chandramouli et al. \(2007\)](#) solved the optimization problem by using mixed-integer programming; however after query resumption, the technique does not re-optimize the given query. As compared to the previous approaches this technique allows to suspend the whole query. Due to which memory wastage is less as compared to the previous techniques. The research by [Schroeder et al. \(2006\)](#) has provided a mechanism that selects a suitable MPL on the basis of two parameters i.e. disc and CPU. The technique does not consider the impact of queries on each other. The low priority transactions are executed only when there are no high priority transactions. Moreover, the paper does not provide a mechanism to give priorities to different transactions. The technique improves the high priority transactions by the factor of 12.1 while the low priority average suffer is about 16%. Summarized analyses of the database workload management tools with respect to self-optimization are presented in Table 1 (where the last column “Exp/Imp” represents whether the research is validated through the experiments, simulation or implementation).

3.2 Self-configuration

Following section elaborates the research related with the workload configuration tools in DBMSs and DWs:

Database Tuning Advisor (DTA) by [Agrawal et al. \(2004\)](#) is an automated offline physical database design tool of SQL Server 2005. DTA provides physical design for a given workload; and recommends horizontal partitioning, materialized views and indexes. It produces the script for implementation of the recommended physical design. DTA provides the recommendation by performing four steps. First, workload is parsed and compressed. After that each query is selected from a given workload and by using cost based model it provides suitable candidate configuration. Then in merging step best physical design structure is selected among the candidates configuration created in the previous step. In last step Enumeration is done by taking the union of the candidates as produced in last two steps and finally it provides the final physical database design using Greedy (M, K) search scheme.

In Oracle, Index Tuning Wizard is enhanced as SQL Access Advisor (SAA) [Oracle Corporation \(2007\)](#). On the basis of current workload, SAA recommends the indexes (including bitmap, function based and B-tree indexes), MVs and partitioning of tables, indexes, partitions, and materialized views. It has found that the already tuned system can be tuned through SAA in a very short time and effort. It takes the contents from the SQL cache and selects the appropriate indexes and MVs after the analysis. It also performs a quick tune using a single SQL statement and also develops or changes the MVs. Architecture of the SAA is shown in Fig. 4.

Table 1 Analysis of the database tools for workload management with respect to self-optimization

Tool	Purpose	Workload type	Static/dynamic	Tech/model	Human intervention	Exp/Imp
Rhee et al. (2001)	Logically divides the workload into small parts and allocates the resources.	OLTP/TPC-C DSS/TPC-H	–	–	DBA divides workload and allocates resources to these units.	Imp
Mumtaz et al. (2008a; 2008b; 2009)	Schedule the workload by considering the impact of the queries over each other	TPC-H BI/ OLAP	Dynamic	Statistical modeling approach (Linear and Quadratic Models, Regression Trees) Smallest Job First PGM	–	Exp
Mehta et al. (2008)	A workload management technique to execute workload in batches	TPC-H BI/ OLAP	Dynamic	–	DBA can set the MPL	Exp
Dageville et al. (2004)	Tunes the process of QEP and provides the recommendations for tuning	OLTP/TPC-C DSS/TPC-H	Dynamic	What-if Model, Cost Based Model	–	Imp
Mehta et al. (2009)	A design criterion for a MWS is presented with simulation results	OLTP/TPC-C BI/ OLAP	Static (MPL) Dynamic	Non-preemptive Priority scheduling	–	Sim
Chandramouli et al. (2007)	Query suspension and resumption is discussed where asynchronous checkpoints are used for all the cardinalities in a query	–	Dynamic	Mixed-Integer Programming	User can specify the physical plan to be used in executing a query	Exp and Imp
Schroeder et al. (2006)	A technique introduced for external scheduling to handle the OLTP workload	OLTP/TPC-C	Dynamic	Queuing Theoretic and Markov Chain Model	–	Exp
Chaudhuri et al. (2007)	A framework is proposed to stop and restart the queries during their execution	TPC-H	Dynamic	GetNext Model Check-pointing	–	Exp

Fig. 4 SQL access advisor by [Oracle Corporation \(2007\)](#)

DB2 Design Advisor [Zilio et al. \(2004a,b\)](#) is a tool that automatically recommends the physical design features for a workload that consists of set of SQL statements, which may include queries like updates, insert and delete etc. These physical design features may include selection of the indexes, MVs, shared-nothing database partitioning and multi-dimensional clustering of tables. There is no other but DB2 Design Advisor recommends these four features. It provides a set of recommendations for the selected features that reduces the total cost of the workload within given disk space. Design Advisor provides candidate solutions and evaluates these solutions by using of DB2 query optimizer. It has built in compression feature for scalability. Due to this feature even size of the workload grows; execution time does not increase exponentially unlike the Index Advisor. The compression is performed only when heavy workload is encountered and analysis cannot be performed within the finite time.

Continuous Query-Driven Index Tuning (QUIET) [Sattler et al. \(2003, 2007\)](#) is an online and predictive approach to handle the database workload. It selects the effective indexes that are already defined or appear to be beneficial. It predicts the benefit of each candidate index and selects the top beneficial index set. It tracks the most recent queries that can be affected due to the candidate indexes. After the selection of best indexes, if the benefit of new index set is greater than pre-defined threshold then the older indexes are replaced with the newer ones. It reduces overhead of index creation by creating the indexes and executing query at same time.

Continuous On-Line Tuning (COLT) discussed by [Schnaitter et al. \(2006, 2007\)](#) is a framework that continuously examines the workload and proposes the indexes to make the physical design more valuable. COLT performs the index selection in three stages, i.e. first builds a model of current workload; second calculates the estimation of each candidate index;

selects the best one among these. It collects the statistics and allocates the profiling resources where these are required and reduces the overhead of online tuning. It regulates itself on the basis of heuristics. COLT is a separate component, which works parallel to the query optimizer. The proposed framework for COLT is implemented in the PostgreSQL and found to be effective to build the useful indexes.

3.2.1 Critical discussion

Database Tuning Advisor [Agrawal et al. \(2004\)](#) provides the integrated physical design recommendations for the indexes, MVs and horizontal partitioning. DTA is scalable to large databases and can manage heavy workload using workload compression and reduced statistics creation techniques. It also allows DBA to specify his/ her own manageability requirements with the performance. For I/O, it provides maximum scriptability and customization through XML schema. However, DTA sometime generates bad results in some situations, for example when half of the columns in a table are multiple indexes. DTA provides the facility to DBA for iterative tuning and recommendations. So he/ she can modify the previous recommendations and configurations for a certain workload until their satisfaction. DB2 Design Advisor ([Zilio et al. 2004a,b](#)) is used to recommend the indexes, MVs, shared-nothing partitioning and multidimensional clustering of tables. It has a built in module to reduce the current workload thereby enhancing the scalability. When the workload is compressed maximum then there will be performance degradation; however Design advisor reduces the execution time with the medium compression level. It has been proven experimentally that it has enhanced the performance of the database up to 100%. SAA by [Oracle Corporation \(2007\)](#) accepts the database workload and provides the recommendations for the indexes and MVs that are beneficial for data access. All the above mentioned tools are used for the physical database design only; however, such type of improvement and enhancements can also be made for logical design. Unlike DB2 Design Advisor there is no such provision of clustering of the tables in DTA and SAA. COLT [Schnaitter et al. \(2006, 2007\)](#) discussed a predictive workload management approach where the benefit is calculated for the individual indexes. The effectiveness of the index is calculated by subtracting the materialization cost from the predicting benefit. COLT performs the what-if calls to query optimizer during the selection of the best indexes for a given workload. However, COLT reduced this cost by enforcing the limits on what-if calls up to somehow. This limit can be increased or decreased according to the current workload. It claims to be online but it sets many parameters offline. Another problem with COLT is that it is limited to the index selection and has no ability to suggest other physical design features such as selection of MVs, data partitioning etc. Moreover, it suggests single attribute indexes without considering the index correlation but in various cases multiple attribute indexes have much more benefits over single attribute indexes (Table 2).

3.3 Self-inspection

Query Progress Indicators (PI) are used to provides step-by-step status of a query execution. Single Query PI [Chaudhuri et al. \(2005\)](#) is proposed with a graphical user interface for the relational DBMS that keeps the track of work completion and estimates the remaining execution time. The proposed PI starts its working by taking the estimated cost from the query optimizer and calculates the remaining query execution time using statistics. It also monitors the query execution speed continuously and remaining query execution time is estimated by dividing the estimated remaining query cost over the query execution speed. The proposed PI is limited to single query without considering the impact of one query over the other.

Table 2 Analysis of the database tools for workload management with respect to self-configuration

Tool	Purpose	W type	Static/dynamic	Tech/model	Human Intervention	Exp/Imp
Agrawal et al. (2004)	Recommends the physical design (selection of the index, MV's and Partitioning) for given workload	TPC-H	Static	Greedy Algorithm, Cost Based Model	DBA specifies the desired physical design (user specified configuration). DBA may limit the tuning to subsets features., such as in OLTP system, DBA may decide a priori to not include any MV's	Imp
Oracle Corporation (2007)	Recommends indexes and Mvs for a given workload	TPC-H BI/ OLAP	Static	–	–	Imp
Zilio et al. (2004a,b)	Recommends physical design feature of tables for given workload	TPC-H BI/ OLAP	Dynamic	Workload Compression Technique	–	Imp
Schnaitter et al. (2006, 2007)	Examine the workload and propose indexes	TPC-H	Dynamic	Heuristic based	DBA controls max rate of what-if calls to query optimizer	Imp
Sattler et al. (2003, 2007)	An online predictive approach to select the indexes	TPC-H	Dynamic	Cost Model	–	Imp

So during the estimation of the query execution remaining time, the load and query progress is considered in isolation.

Later on, a multi-query PI [Luo et al. \(2006\)](#) has been proposed which represents the progress of running queries and considers the impact of queries on each other as opposite to the previous techniques of single query PIs. The technique of Multi-query PI works by considering the remaining execution time of the concurrent queries with available statistics and predicts the future execution speed of incoming queries. On the basis of estimation, it can also predict the future queries and has the ability to manage the current workload efficiently. The indicator not only provides the visualization of the running queries but helpful to manage the database workload efficiently. This technique takes the workload management problem as an input and provides its solution through the information that is provided by the Multi-query PI. The proposed technique is implemented in POSTGRE SQL and examined with the remaining query execution time.

DB2 Query Patroller (QP) [IBM Corporation \(2003\)](#); [Lightstone et al. \(2002\)](#) is a management tool, which is used to streamline the requests according to the available resources. It is possible to accept the database workload from user and analyze it. On the basis of analysis it prioritizes and schedules the workload according to different query classes. A class is build by considering cost range and Multi-Programming Level (MPL) threshold. Cost range is provided by the query optimizer that calculates the resource demands. MPL threshold is the maximum number of requests in a class that can execute in one time. Remaining queries are placed in a queue when the threshold level is reached and are placed for execution in a class whenever threshold level goes down. It also gives the information to the user about the status of tasks. QP provides sufficient resources for given workload and by using profile (that is created by administrator) saturations for long terms queries can be avoided. QP controls the flow of requests proactively. It provides the information about the completion of request and finds trends of queries, workload of users as well as the frequently used indexes and tables. QP enhances the performance by monitoring the system utilization, canceling or rescheduling the queries and identifying the trends of database usage. Query submitter uses it to monitor the submitted queries, store query result for future perspective and query submission customization. Submitter assigns the higher priorities to some user so that in the class these queries run with less delay. QP suspends high load queries so that they can be cancelled or scheduled to run after peak hours and track the query process. By performing these steps the smaller queries may not stuck and system recourses are used properly. It is based on client-server architecture and consists of three components. These components are Query Patroller server, Query Patroller Center and Query Patroller command line support. DBA uses QP to assign privileges of resources at user and system level.

REDWAR (RElational Database Workload AnalyzeR) by [Yu et al. \(1992\)](#) is an offline tool for DB2 environment, which is used for the characterization of SQL trace, provides structural information and statistics of the query under execution. REDWAR analyzes and classifies the data. During analysis, it uses statistical summaries (correlation, distribution, variation etc) and runtime behavior of the workload. The report generated by REDWAR can be used to plan the physical design and build the benchmark workload. However, it has no functionality to recommend the physical design. It increases the efficiency of the database system by identifying the criteria for a query.

3.3.1 Critical discussion

A single-query PI by [Chaudhuri et al. \(2005\)](#) often provides bad and wrong estimates. As in most of the concurrent queries execution, one query can significantly slow down the

progress of other query. In single query PIs, greedy algorithm is used to speed up the process, where an optimal victim query is selected and the next optimal victim query is chosen. The process continues up to get all the victim queries. The technique in (Luo et al. 2006) is the first proposal of a multi-query PI. As compare to the single query PIs, the Multi-query PI considers the impact of the concurrent queries over each other and predicts the incoming queries with considering the priority and cost. The information provided by these indicators helpful for the workload management tools to take more intelligent decisions. The indicator is able to predict the accurate future queries even initial estimates are wrong by detecting and correcting their estimates. It monitors the system at all the times and manages the workload more dynamically. Multi-query PI is adoptive as it revises its decisions when it found some significant changes as compared to the predicted results. This adaptive behavior of Multi-Query PI shows the consistency with the trends of automatic and AC. Query Patroller, IBM Corporation (2003); Lightstone et al. (2002) monitors the given workload; perform analysis and prioritize it schedules for the incoming requests from the users. It limits the flow of long running queries to avoid saturation and ensures better resource utilization on the basis of profile (created by the administrator). REDWAR by Yu et al. (1992) is a database characterization tool for DB2 environment and assists the physical database design process. However, it does not provide the recommendations for the physical design. Table 3 represent the techniques and models with some other attributes that are related with each research that falls in self-inspection section.

3.4 Self-organization

Self-organization in a DBMS is to reorganize and restructure the layout of data and indexes dynamically. Research in the area of self-organization related with the workload management in DBMSs and DW has been carried out by various researchers. Here, in the following section we are discussing some of these.

A database tool Disk Array Designer (DAD) by Anderson et al. (2005) is used to configure the storage system. It is used to assist DBA in taking the decisions (using device model) about the physical design and design automation process. The algorithm used by DAD selects the best design among possible design choices using best-fit bin packing heuristic with randomization technique. The proposed designer not only defines the array but also performs its configuration and storage for application data. DAD is evaluated by performing experiments over the mix workload and found that it has the ability to handle critical configuration (low or high level) tasks. Moreover, it also produces the near-optimal plan for the design of the storage system with speed and precision.

An adaptive QoS management technique is discussed by Krompass et al. (2008) where the economic model is used to handle an individual request of BI and OLTP workload proactively. A systematic way is provided to arrange different requests by dividing these into classes based on the cost and time limit. A model is also proposed which calculates the cost of a request by differentiating underachieving and marginal gains of a Service Level Objective (SLO) threshold. The effectiveness of the proposed framework is determined by performing experiments on numerous workloads.

A Priority Adaptation Query Resource Scheduling (PAQRS) Algorithm by Pang et al. (1995) is based on the Priority Memory Management (PMM) Algorithm and deals with multi-class query workload. The algorithm reduces the missed deadlines according to the miss distribution defined by the administrator. The algorithm works by allocating memory and assigns priorities by considering resource usage, workload characteristics and the experienced performance. Whenever the workload change, new MPL is calculated, memory

Table 3 Analysis of the database tools for workload management with respect to self-inspection

Tool	Purpose	W type	Static/dynamic	Tech/model	Human intervention	Exp/Imp
Luo et al. (2004)	Display the step by step status of query execution	TPC-R	Dynamic	Greedy Algorithm	–	Imp
Chaudhuri et al. (2005)	Display the step by step status of query execution considering the impact of concurrent queries over each other and manage workload	TPC-R	Dynamic	Greedy Algorithm	–	Imp
Lighstone et al. (2002)	Prioritize and schedule the workload according to query classes	OLTP/TPC-C BI/OLAP	Dynamic	–	Administrator assigns privileges at user and system level	Imp
Yu et al. (1992)	A tool that is used for the characterization of the workload and plan the physical design	–	Static	–	–	Imp

reallocation and priority adjustment is done accordingly. Two techniques a miss ratio projection and resource utilization heuristics are used to calculate the new MPL. In case of miss ratio projection method, previous MPL and miss ratio are used as parameters.

3.4.1 Critical discussion

Disk Array Designer by [Anderson et al. \(2005\)](#) searches the best design by using the best-fit bin packing heuristic with randomization and backtracking techniques. It estimates the performance of storage system through the device models. DBAs can get the optimal solution by comparing their own storage system design to DAD design. They may make changes in some storage part by DAD solution through answering the “what-if” questions. It provides the near optimal solution, as the storage design process is NP-hard problem and provides the automatic adoption. The algorithm used in DAD is limited to the design of storage system and can handle only 1-tier architecture. The framework provided for QoS in workload management by [Krompass et al. \(2008\)](#) is beneficial for the OLTP and BI workload. The framework is scalable as it can implement the new workload management concepts with previously implemented policies. The framework uses economic model with two economic cost functions which are Opportunity Cost and Marginal Gains, where penalty information is added with the queries. The penalty information is used to make the efficient order of pending query execution. The scheduling policy used for OLTP workload in this framework is enhanced by considering the combine effect of the priorities and service level objectives rather than considering merely priority. PAQRS by [Pang et al. \(1995\)](#) is used to schedule the complex type of workload and reduces the number of missed deadline thereby making the efficient use of system resources. It has bias control mechanism, which regulate the distribution of missed deadlines among the query classes. The MPL and memory is allocated on the basis of the regular and reserve group quota. The priority of regular queries is higher than reserve queries. By doing this, it make adjustments between the miss ratio and the target distribution. PAQRS cannot handle transactions and is limited to the workload consisting of mix queries. Its’ performance degrades with the increase of fluctuations in the workload. So the adoption mechanism of PAQRS needs to be improved (Table 4).

3.5 Self-prediction

Self-prediction in workload management is used to forecast the various aspects in DBMSs and DWs. The research related with the self-prediction in workload management is highlighted as under:

DB Resource Advisor by [Narayanan et al. \(2005\)](#) is used to predict the response time and throughput dynamically. It predicts the workload using what-if model without using configuration description. This will help the advisor to guess about the status of the resources. The components are identified that are required for the self-prediction which includes the low level instrumentation, end to end transaction tracing and parameterized models of hardware resources. It provides accurate trends for the response time in transactional tracing. Experiments are performed on the OLTP workload and it has observed that Resource Advisor accurately predicts the workload changes.

The research by [Menasce et al. \(1999, 2001\)](#), [Menasce and Bennani \(2003\)](#) devised the QoS controller for E-commerce applications that has ability to manage the workload by adjusting different configuration parameters. The adjustments are done through the QoS Controller by considering three performance goals that are average response time, average throughput and probability of rejection. The paper [Menasce et al. \(1999\)](#) introduces a technique for the

Table 4 Analysis of the database tools for workload management with respect to self-organization

Tool	Purpose	W Type	Static/dynamic	Tech/model	Human intervention	Exp/Imp
Anderson et al. (2005)	Assist the DBA to design storage process and has the capability to design itself	TPC-C TPC-D	Dynamic	Best-Fit Bin Packing Heuristic, Randomization and Backtracking	–	Imp
Krompass et al. (2008)	An adaptive QoS management technique to handle the BI and OLTP workload	OLTP/TPC-C BI/OLAP	Dynamic	–	–	Exp
Pang et al. (1995)	The proposed algorithm reduces the missed deadlines, allocate memory and assign privileges	–	Dynamic	PMM Algorithm	Administrator defines missed distribution deadlines in different classes and performance objectives	Exp

workload characterization and generates the workload models for the E-commerce environment. The Customer Behavior Model Graph (CBMG) or state transition graph is introduced that represents the similar navigational pattern for group of customers who perform same activities. The workload model and its parameters are identified and presented the clustering algorithm for workload characterization. At the end this technique is evaluated with different experiments. The research by [Menasce et al. \(2001\)](#) improved the QoS level in E-commerce application by dynamically monitoring and tuning. This technique identifies best configuration parameters by combining Hill Climbing Technique with Analytical Queuing Model. Experiments are performed to evaluate their technique by making comparison of QoS levels. In the paper [Menasce and Bennani \(2003\)](#), authors have designed the controllers that used the Analytic Performance models with Combinatorial Search technique. The modeling technique is used to identify the best configuration for the given workload. Their model is used to predict the QoS parameters of the workload. They show effectiveness of their technique through simulation and experiments.

3.5.1 Critical discussion

The Resource Advisor by [Narayanan et al. \(2005\)](#) is presented with a modular architecture in which CPU, buffer and storage models are integrated to predict the response time and throughput by identifying the required key components. Authors have taken the advantage of end-to-end tracing technique in visualization and understanding performance of the system. Resources are properly allocated on the basis of continuous monitoring. As compare to the resource advisor, current DBMSs lack of CPU, buffer and disk models. By using these models, Resource Advisor provides an accurate prediction and best performance results. Resource Advisor has high overheads per transaction when size of the buffer pool is small. Due to continuous monitoring, the CPU overhead is 6.2% for online and 1.2% for offline execution. This overhead can be reduced by using some other appropriate techniques. Finally, the tool is evaluated through a prototype implementation in SQL Server however it has the ability to incorporate with some other DBMSs. Maximum session drops will be occur in the characterization technique as proposed by [Menasce et al. \(1999\)](#) during the large number of sessions or maximum load and the proposed technique has no provided any mechanism to manage or recover these session drops. The technique for QoS of E-commerce [Menasce et al. \(2001\)](#) workload can handle dynamic workload and short-term fluctuations. The technique uses Heuristic optimization with Predictive Queuing Model and provides better results. It uses reactive approach rather than proactive. The techniques use Hill Climbing technique for searching but when it stuck; the sub-optimal solution will be achieved. The QoS Controller maximizes the throughput up to 88% on average. When control interval level is less than or equal to 11, QoS controller do not exhibit any performance; however when control interval exceeds 11, performance increases up to 95% (Table 5).

3.6 Self-adoption

Self-Adoption or adaptation in DBMS is used to adopt the new changes according to the available resources and environment. In case of workload, self-adaptation is the way to adopt the workload in an efficient and responsive way according to available resources and environment. There are number of studies available by database researchers for the workload adoption in DBMSs and DWs that are discussed as under:

Resource Governor by [Microsoft Corporation \(2007\)](#) is introduced for SQL Server 2008 that has the ability to manage the workload and resources with BI feature. It provides a

Table 5 Analysis of the database tools for workload management with respect to self-prediction

Tool	Purpose	W type	Static/dynamic	Tech/model	Human intervention	Exp/Imp
Narayanan et al. (2005)	Technique to predict the response time and throughput Manage workload in E-Commerce application	OLTP/ TPC-C	Dynamic	What-if model	–	Pro Exp
Menasce et al. (1999)	Dynamic monitoring and tuning of E-commerce application	TPC-W	Dynamic	Queueing model	–	Imp
Menasce et al. (2001)	Provide the configuration for given workload and predicts QoS parameters	TPC-W	Dynamic	Hill Climbing Technique, Analytical Queueing Model	–	Imp
Menasce and Bennani (2003)		TPC-W	Dynamic	Hill Climbing Technique, Analytical Performance Model, Combinatorial Search Technique	–	Sim and Exp

consistent, balanced and predictable response to users by imposing limits on the resource consumption. It provides performance for concurrent workload by using a profile created by the DBA that contains the information about the resource limits and priorities for different workload groups. Resource Governor allows setting timeout for certain queries and suggesting priorities without changing server settings. Resources are allocated by the resource governor as per requirements and priorities of different sub-departments within the organization. So sub-department with high priority and requirements will get the large share of resources. The basic steps performed by the Resource Governor are to create resource pools, workload groups, classifier function, monitoring and adoption.

A framework is proposed by [Bruno and Chaudhuri \(2007\)](#) for online tuning that examines current workload at all the time and then changes the physical design accordingly. When a workload is processed the proposed framework collects the information of the QEP, calculates its associated cost and selects the best QEP to alter the physical design. During the process, it uses the query optimizer for acquiring the QEP. This framework also considers the index correlation during the process and by doing this it avoids the physical design oscillation.

In Teradata's ASM [Ballinger \(2002\)](#), [Brown et al. \(2008\)](#) workload is defined as a group that further consists of the classification rules, MPL, exception and service level goals. Classification rules are defined as attributes of the request that qualify the query to run under the workload definition. MPL is the number of queries that can be run in parallel under the workload definition. When limit exceeds, incoming requests are placed in the delay queue. Exception is used to control the actions and produced when some abnormal behavior is found in the workload. The objectives of the workload are described in the Service Level Goals (SLGs). The exceptions defined in the workload definition react with the admission control of ASM. It cannot predict the resource demand required by each performance class; however maps the performance class with allocation group. The architecture of the Teradata ASM (shown in Fig. 5) consists of four phases, which are:

- i. Queries are divided into classes by analyzing log and suggestions are made for workload definition.
- ii. Workload definitions are maintained by adjusting criteria, goals, and performance mappings.
- iii. Workload flow and priorities are regulated on the basis of workload definition.
- iv. Examines the workload execution and improve the performance with respect to the goals. When sub-optimal performance is observed during the workload execution four methods are used to improve the performance. These methods are workload management, performance tuning, capacity planning and performance monitoring.

A framework for workload adaptation by [Niu et al. \(2006\)](#) is proposed that has two components, i.e. workload detection and workload control. The workload detection finds the changes and provides information of the workload. The framework has also four functional components, namely workload characterization, performance modeling, system monitoring and workload control. The effectiveness of the framework is proven by implementing the query scheduler through a large number of experiments. Query scheduler can directly handle OLAP workload where as OLTP workload cannot be handled directly. They proposed a technique by using indirect control of the OLTP through directly controlling OLAP workload. So due to this enhancement in query scheduler both workload can be handled to achieve the performance goals. Performance is improved by prediction process using Kalman filter as performance prediction plays vital role in workload adaptation. Kalman filter is very powerful tool that is used to make the estimation of past, present and future states. The filter provides an optimal solution for linear processes and sub-optimal solutions for the non-linear processes.

Fig. 5 ASM elements (Ballinger 2002)

Through experiments, more accurate prediction results are obtained with less unpredicted SLO violations. In short, the research contribute by designing a general framework for performance oriented workload adaptation, prototype implementation of framework (Query Scheduler), a cost-based performance model for the workload control plans and improves the accuracy of the prediction through Kalman filter.

3.6.1 Critical discussion

Resource Governor by [Microsoft Corporation \(2007\)](#) with BI features is used to manage the workload in SQL Server 2008. It manages the given workload according to the profile created by the DBA. The profile contains information about the users such as their requirements, priorities etc. ASM by [Ballinger \(2002\)](#) and [Brown et al. \(2008\)](#) automate the workload up to a significant level and focuses over the ease of use with the help of monitoring and analysis. ASM is workload-centric as compares to other database tools that are system-centric mostly. ASM changes the order of the workload by using a preventive approach. ASM maps the performance class with allocation group and has not ability to predict the resource demand for each performance class. The experiment in Query Scheduler [Niu et al. \(2006\)](#) is performed on the stable workload, which is not suitable for dynamic environment where the workload changes rapidly. During the experiments, total cost of a query instead of detailed cost is used as a parameter that may generates wrong results. Moreover, the scheduler is confined to the linear workload however, in a real environment most of the time workload behavior remains non-linear (Table 6).

4 Discussion

In previous section, we have discussed the various database tools that are used to handle the workload in different DBMSs and DWHs. Among these tools some are internal while

Table 6 Analysis of the database tools for workload management with respect to self-adoption

Tool	Purpose	W type	Static/dynamic	Tech/model	Human intervention	Exp/Imp
Microsoft Corporation (2007)	Management of the workload and resources with BI feature	OLTP/TPC-C	Dynamic	–	DBA create profile for concurrent workload	Imp
Ballinger (2002); Brown et al. (2008)	Automate and advise for workload management and performance monitoring	DSS/TPC-H OLTP/TPC-C BI/OLAP	Dynamic	–	DBA make assignments based on understanding about the workload. DBA defines multiple allocation groups	Imp
Niu et al. (2006)	Performance oriented workload adaptation framework	OLTP/TPC-C BI/OLAP	Dynamic	Kalman Filter Queuing Model	–	Proto Imp

Table 7 Research work done with respect to workload type

S.no	Workload benchmark/workload type	Work done	Percentage
1	OLTP/ TPC-C	24	33.3
2	DSS/ TPC-H	18	25.0
3	BI/ OLAP	16	22.2
4	TPC-D	2	2.8
5	TPC-R	3	4.2
6	TPC-W	9	12.5

Fig. 6 Percentage of the research work with respect to the workload benchmark/ type

the others are external that do not change the internals of the DBMSs and DWHs. We have divided the entire work into two perspectives, i.e. workload type and autonomic computing perspective. The workload type perspective reveals that how much research work has been done over different workload types while the autonomic computing perspective shows the research that has been done in each autonomic characteristic.

4.1 Workload type perspective

Table 7 summarizes the research work done related with the workload management in DBMSs and DWs by different database researchers and vendors that are evaluated through experiments which are performed on various workload benchmarks and workload types. The columns “Work Done” and “Percentage” of the Table 7 is calculated by taking the summation of the research that is discussed in the Sect. 3 and our previous work [Raza et al. \(2011\)](#). Figure 6 presents the percentage of the research work with respect to Workload Benchmark/Type.

4.2 Autonomic computing perspective

Table 8 summarizes the research work done on the workload management tools in DBMSs and DWs with respect to the six AC characteristics that has been done. Again, the column “Work Done” and “Percentage” of Table 8 are calculated by taking the summation and average of the research that is discussed in the Sect. 3 and our previous work [Raza et al. \(2011\)](#). Figure 7 shows the percentage of the research work that shows how much work is done in each AC characteristics.

Table 8 Research work done with respect to self-* characteristics

AC characteristic	Work done	Percentage
Self-optimization	13	21.7
Self-configuration	13	21.7
Self-inspection	10	16.7
Self-organization	6	10.0
Self-prediction	11	18.3
Self-adoption	7	11.7

Fig. 7 Percentage of the research Work with respect to self-* characteristics

5 Conclusion and future work

The research presents the various aspects of the autonomic computing such as its architecture, elements, characteristics, levels, Autonomic DBMS, motivation towards ADBMSs and Autonomic Workload Management. The available literature of workload management tools are observed to find the current AC level in the workload management. Tables 1, 2, 3, 4, 5 and 6 summarizes the research related with the workload management tools with respect to the AC characteristics. The analyses show the effectiveness of various available tools for the workload management. Some advances on workload management in the context of Autonomic Computing have been done. However, more efforts and improvements are essential on the current as well as new workload management techniques and tools. In future, we are planning to design and develop a framework for the Autonomic Workload Management for DBMSs and DWs that has the ability to handle all database tasks proactively and efficiently.

Acknowledgments We are grateful to the Higher Education Commission (HEC) of Pakistan and International Islamic University, Pakistan who are supporting this research work.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Agrawal S, Chaudhuri S, Kollar L, Marathe A, Narasayya, Syamala M (2004) Database tuning advisor for microsoft SQL server 2005. In: Proceedings of the 30th international conference on very large data bases, pp 1110–1121
- Anderson E, Spence S, Swaminathan R, Kallahalla M, Wang Q (2005) Quickly finding near-optimal storage designs. *ACM Trans Comput Syst* 23(4):337–374
- Ballinger C (2002) Introduction to Teradata's priority scheduler. <http://www.teradatalibrary.com/pdf/eb3092.pdf>
- Beg S, Naru U, Ashraf M, Mohsin S (2010) Feasibility of intrusion detection system with high performance computing: a survey. *Int J Adv Comput Sci* 1(1):26–35
- Brown DP, Richards A, Zeehandelaar R, Galeazzi D (2008) Teradata active system management. <http://www.teradata.com/t/page/145613/index.html>
- Bruno N, Chaudhuri S (2007) An online approach to physical design tuning. In: Proceedings of the 23rd international conference on data engineering, pp 826–835
- Chandramouli B, Bond CN, Babu S, Yang J (2007) Query suspend and resume. In: Proceedings of the ACM SIGMOD international conference on management of data, pp 557–568
- Chaudhuri S, Kaushik R, Ramamurthy R (2005) When can we trust progress estimators for SQL queries? In: Proceedings of the ACM SIGMOD international conference on management of data, pp 575–586
- Chaudhuri S, Kaushik R, Ramamurthy R, Pol A (2007) Stop-and-restart style execution for long running decision support queries. *VLDB*, pp 735–745
- Dageville B, Das D, Dias K, Yagoub K, Zait M, Ziauddin M (2004) Automatic SQL tuning in Oracle 10g. In: Proceedings of the 30th international conference on VLDB, pp 1098–1109
- Horn (2001) Autonomic computing: IBM's perspective on the state of information technology. IBM J. Paper
- Huebscher MC, McCann JA (2008) A survey of autonomic computing—degrees, models, and applications. *ACM 0360-0300/2008/08-ART7*
- IBM Corporation (2003) DB2 Query Patroller Guide: Installation, administration, and usage
- Krompass S, Scholz A, Albutiu MC, Kuno H, Wiener J, Dayal U, Kemper A (2008) Quality of service enabled management of database workload. *IEEE Database Eng Bull-DEBU* 31(1):20–27
- Lightstone SS, Lohman G, Zilio D (2002) Toward autonomic computing with DB2 universal database. *SIGMOD* 31(3):55–61
- Luo G, Naughton JF, Ellmann CJ (2004) Toward a progress indicator for database queries. *SIGMOD*, pp 791–802
- Luo G, Naughton JF, Yu PS (2006) Multi-query SQL Progress indicators. In: Proceeding of the 10th international conference on extending database technology (EDBT), pp 921–941
- Mateen A, Raza B, Hussain T, Awais MM (2008) Autonomic computing in SQL server. In: 7th international conference on computer and information science, pp 113–118
- Mehta A, Gupta C, Dayal U (2008) BI batch manager: a system for managing batch workloads on enterprise data warehouses, *EDBT*, pp 640–651
- Mehta A, Gupta C, Wang S, Dayal U (2009) rFEED: a mixed workload scheduler for enterprise data warehouses. *ICDE*, pp 1455–1458
- Menasce DA, Bennani MN (2003) On the use of performance models to design self-managing computer systems. In: Proceedings of computer measurement group conference, pp 1–9
- Menasce DA, Almeida VAF, Fonseca R, Mendes MA (1999) A methodology for workload characterization of E-commerce sites. In: Proceedings of the first ACM conference on electronic commerce, pp 119–128
- Menasce DA, Barbara 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, pp 224–234
- Microsoft Corporation (2007) Microsoft SQL server 2005 books online. <http://msdn2.microsoft.com/en-us/library/ms190419.aspx>
- Mumtaz A, Ashraf A, Shivnath B (2008a) Modeling and exploiting query interactions in database systems. *CIKM*, pp 183–192
- Mumtaz A, Ashraf A, Shivnath B, Munagala K (2008b) QShuffler: getting the query mix right. *ICDE*, pp 1415–1417
- Mumtaz A, Ashraf A, Shivnath B (2009) Query interactions in database workloads, *DBTest*
- 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, pp 239–248

- Niu B, Martin P, Powwley W, Horman R, Bird P (2006) Workload adaptation in autonomic DBMSs. In: Proceedings of the 2006 conference of the center for advanced studies on collaborative research (CAS-CON'06). ACM Press, New York, p 13
- Oracle Corporation (2007) Oracle database performance tuning guide 11g. Release 1 (11.1)
- Pang HH, Carey MJ, Livny M (1995) Multiclass query scheduling in real-time database systems. *IEEE Trans Knowl Data Eng* 7(4):533–551
- Parashar M, Hariri S (2005) Autonomic computing: an overview. Springer, Berlin pp 247–259
- Raza B, Mateen A, Awais M, Sher M (2011) Survey on autonomic workload management: algorithms, techniques and models. *J Comput* 3(7):29–38
- Rhee A, Chatterjee S, Lahiri T (2001) The Oracle database resource manager: Scheduling CPU resources at the application. High performance transaction Sys Workshop.
- Sattler K, Geist I, Schallehn E (2003) QUIET: continuous query-driven index tuning. Proceedings of the 29th very large database, pp 1129–1132
- Sattler K, Lühring M, Schmidt K, Schallehn E (2007) Autonomous management of soft indexes. In: Proceedings of the international workshop on self-managing database systems, pp 450–458
- Schnaitter K, Abiteboul S, Milo T, Polyzotis N (2006) COLT: continuous on-line database tuning. In: Proceedings of ACM SIGMOD, pp 793–795
- Schnaitter K, Abiteboul S, Milo T, Polyzotis N (2007) On-line index selection for shifting workloads. In: Proceedings of the ICDE workshops: 459–468
- Schroeder B, Harchol-Balter M, Iyengar A, Nahum EM, Wierman A (2006) How to determine a good multi-programming level for external scheduling, ICDE, pp 60–71
- Weikum G, Hasse C, Mönkeberg A, Zaback P (1994) The COMFORT automatic tuning project. *Inform Syst J* 19(5):381–432
- Yu PS, Chen M, Heiss H, Lee S (1992) On workload characterization of relational database environments. *Softw Eng* 18(4):347–355
- Zilio DC, Zuzarte C, Lightstone S, Ma W, Lohman GM, Cochrane R, Piraresh H, Colby LS, Gryz J, Alton E, Liang D, Valentin G (2004a) Recommending materialized views and indexes with IBM DB2 design advisor. In: Proceedings of international conference on autonomic computing, pp 180–188
- Zilio DC, Rao J, Lightstone S, Fadden S (2004b) DB2 design advisor: integrated automatic physical database design. In: Proceedings of 30th international conference on very large data bases, pp 1087–1097