Chapter 7

Conclusion and Future work

7.1 Conclusion

As part of this research, the performance tuning techniques based on machine learning and fuzzy logic have been proposed to dynamically fine-tune the DBMS shared memory area, keeping the overheads to the minimum. The auto-tuning of DBMS and making it truly and completely autonomous is a highly ambitious task. This is for some of the important reasons like, the DBMS software being complex to manage and there are several external factors like workload-types, user-load etc. that affect the performance. The knowledge about sensors and effectors is extremely important for developing a self-tuning system, a series of experiments were carried out to establish the knowledge-base to be used in the proposed tuning methods.

In the first phase of the research, the DBMS tuning parameter characterization was done to understand and establish the knowledge about tuning impact on system performance and also to know the sensors that can be used to identify the workload type, current userload and database size. The experiments helped in gaining deeper insight into the performance
bottlenecks and decide on the range over which tuning is effective. Through these experiments it is established that certain tuning parameters had a greater impact on query response-time than others. For instance, the DB_Cache tuning parameter has the highest impact on the query performance than Shared pool or Large pool sizes. The Large_pool has a greater impact on response time as compared to Shared pool. It is also established through the experimentation, that the tuning interference of the tuning parameter on the response-time is 5-15%. However, this effect is limited only for small values of the tuning parameters. Hence, the inter-tuning interference can be ignored while implementing self-tuning systems. The experiments on finding the effect of tuning parameters on Buffer-hit-ratio under different workload types, helped in identifying the current workload type. For instance, the BHR value in the range 92.11 to 99.98 indicates that the current workload is of type TPC-C. Due to overlapping of BHR values among certain workload types, it was found to be necessary to use the database size also in conjunction with BHR to correctly identify the current workload. Therefore the BHR value in conjunction with the database size has been used as means to identify the workload type in the proposed self-tuning techniques.

As part of this research, a mathematical model to establish a relationship between query response-time, Buffer size and Userload, was attempted. As there are many factors that influence the query response-time, under simplifying assumptions, the model showed good match between experimental result and the model output. However, it was observed that it is extremely difficult to develop a mathematical model that relates the query response-time to all the other parameters like workload-type, scaling factor, shared-pool size etc.

In the second phase, the use of machine learning technique to estimate the values of the tuning parameters was attempted. An expert DBA uses his past experience and in depth understanding of the tuning knowledge, to fine tune the system. This prompted the possibility of using artificial intelligence to mimic the actions of a DBA. Among the machine learning techniques, use of Neural network was the most convenient and the easiest way of implementing the adaptive self-tuning system. Hence, a neural network based adaptive tuning technique was proposed and the training data set was derived from the experimental data obtained in first phase. A three layer ANN with 10 nodes in the middle layer and having three inputs namely, BHR, Userload N and Database size was implemented. The training dataset for each of the workload-type was used as knowledgebase and the other important inputs like learning rate, epoch value and acceptable error were set after trial and error. The system was tested two workload types namely TPC-C(OLTP) and TPC-E(DSS) with user-load varying
from 2-100 users. The results were 27% and 32.8% better as compared to the built-in self-tuning feature of Oracle 10g under TPC-C and TPC-E workload-types respectively.

In the third phase, use of fuzzy logic to implement an adaptive self-tuning architecture was proposed. In this setup inputs were fuzzified using Gaussian membership functions and a set of thirteen rules were formed to decide on the tuning parameters as output. As the system is rule-based, the results were slightly inferior as compared to the neural network approach. The performance of the system showed an improvement of 20.1% and 27.8% as compared to the built-in self-tuning feature of Oracle 10g when tested under TPC-C and TPC-E workload types respectively.

Following are the important findings of this research work:

1. Workload characterization shows that Db_Cache tuning parameter has a highest impact on the response-time as compared to the other two tuning parameters.
2. The tuning interference between the three tuning parameters is negligible.
3. The Buffer-Hit-Ratio and Database size can be used to identify current workload type.
4. Adaptive techniques based on machine learning and fuzzy logic have been successfully implemented with improvement in response-time and efficient usage of DBMS memory.
5. The proposed Neural network based self-tuning method works well in a distributed database environment with an improvement of 18.64% in response-time.

In the last phase, a Neuro-fuzzy based tuning system was proposed in an attempt to combine the best of both the methods. By choosing the output of fuzzy system and neural-network-based tuning system in the ratio 60:40, a performance improvement in the range of 0-28% in response-time was achieved under various workload types and scaling factors. In terms of throughput, no improvement as compared to workload analysis based self-tuning was observed under TPC-C workload. This is due to the fact that transactions in TPC-C which is an OnLine Transaction Processing(OLTP) workload-type will have very few database update operations. On the other hand under TPC-E workload-type which represents a Decision Support System(DSS) type workload, showed an improvement of 6.6% in throughput as compared to the self-tuning approach. This improved performance in throughput is due to the ability of the Neuro-Fuzzy approach to accurately estimate the appropriate values of the tuning parameters and also effectively utilize them for improved performance. Further an
improvement of 22.9% in the memory utilization was observed as compared to the built-in self-tuning feature.

7.2 Scope for future work

Although, the proposed adaptive tuning techniques have been validated under variety of workloads and database platforms, further validation needs to be done with emerging workload type like Web-workload. Further investigation is needed to see the impact on performance when the DBMS is presented with mixed workload types. This is important because, a single database server may host data pertaining to wide variety of applications that generate data of different workload types. As a continuation of this work, application of these adaptive techniques to build self-reconfiguring systems may be explored that provide highly customized configuration settings and also result in improved performance. Following are the scope for extending the knowledge in the self-tuning of DBMS.

1. Devising methods to implement self-tuning at Disk, CPU and Network in conjunction with memory tuning.
2. Translating Service Level Agreements[76] to low level tuning policies.
3. To evolve Self-tuning techniques for distributed databases that are heterogeneous.
4. Validate the results on other Database Management Systems
5. Develop self-tuning methods that adapt to mixed workloads.
7. Evaluating performance at very high scaling factors and userloads.

The self-tuning in DBMS is extremely challenging as several factors affect the query execution and hence, complete automation of tuning is extremely difficult.