Project Proposal
Automatic Tuning of DBMS using Machine Learning

CMU 10-701: Machine Learning (Fall 2014)

Ram Raghunathan (Andrew ID: rraghuna)
Joy Arulraj (Andrew ID: jarulraj)

Goal:

Modern database systems are highly configurable and must support a wide variety of workloads. However, *tuning* these systems is challenging. Configuration assistants like Microsoft's AutoAdmin [1] allow administrators to use statistics and suggestions from the database system to guide the physical design of the database. However, these methods still require experienced administrators and prior knowledge about the workload. We seek to apply machine learning methods to automate database system tuning with minimal user input and knowledge.

Milestones:

First, we plan to *map* a given workload comprised of SQL transactions to a standard database benchmark workload. This will allow us to use prior knowledge about the standard benchmark gained from previous DBMS deployments. We plan to collect features like workload characteristics and DBMS performance metrics. We may need to make use of a feature extraction algorithm like *principal component analysis* (PCA) as a preprocessing step. It will be interesting to see if feature extraction yields insights about the most influential features. We will then use unsupervised techniques like *clustering* for mapping workloads. Performance analysis of the resulting classifier will be done via cross-validation.

Second, we plan to *estimate* the DBMS performance given the DBMS configuration, workload and hardware setup. This will be done with supervised techniques like *Gaussian process regression*. As we expect a lot of features to be present in the input, we may need to make use of a feature extraction algorithm like principal component analysis (PCA) first. Performance analysis of the estimator will be done using cross-validation.

Dataset:

For the first part, we will use SQL workloads of standard benchmarks in OLTPBench[2], a well-known testbed for benchmarking relational databases, as training data. We plan to generate more synthetic variants of these workloads for training and testing purposes. While this synthetic data will not be representative of real-world workloads, we feel it is a good starting point for evaluating viability of the approach outlined above. We anticipate some preprocessing steps in order to extract features from the workload such as types of database queries, distribution of query types, table access patterns, etc. This will involve making a fairly sophisticated workload analyzer. We will try to hook this into an existing database to collect the features. For the second part, we plan to run different workloads in OLTPBench on the DBMS under different configurations and obtain performance metrics. This will provide both training and testing data.

Timeline:

We estimate that it will take 2 weeks for creating the workload analyzer that extracts the required features. Then, we will spend 2 weeks solving the first problem as well as generating data for the second problem. We plan to solve the first problem by the midway report deadline. After that, we plan to solve the second problem in another 2 weeks. We reserve the final 2 weeks for wrapping up the project report and poster.

Minimum goals:

The minimum goals are : (a) to create a workload mapper that maps an arbitrary SQL workload to a well-known standard benchmark, and (b) to estimate the performance of a DBMS given a workload and

configuration pair.

Stretch goals:

As a real-world workload can exhibit some aspects of different standard benchmarks or can exhibit different characteristics across time (e.g. read-mostly during day and write-mostly during night), mapping the entire workload onto a single benchmark may not be an entirely accurate characterization. To help characterize the workload more accurately, we will consider mapping parts of a workload onto different benchmarks using techniques like *multi-label prediction*.

References

[1] S. Agrawal, S.Chaudhuri, A. Das, and V. Narasayya. Automating layout of relational databases. In ICDE, pages 607-618, 2003.

[2] D. E. Difallah, A. Pavlo, C. Curino, and P. Cudre-Mauroux. OLTP-Bench: An extensible testbed for benchmarking relational databases. In VLDB 2014. http://oltpbenchmark.com