Why would I believe your store is better than mine?

Dave Munro[†], Ron Morrison[†], Eliot Moss[¥] and Stephan Scheuerl[§]

[†]School of Mathematical and Computational Sciences,
University of St Andrews, North Haugh, St Andrews, Fife, KY16 9SS, Scotland
Email: {dave, ron}@dcs.st-and.ac.uk

[¥]Department of Computer Science, University of Massachusetts,
Amherst, Massachusetts, MA 01003, U.S.A.
Email: moss@cs.umass.edu

[§]Digital Equipment Corporation,
Livingston, Scotland

Abstract

Comparing the performance of computer systems leads to measurement techniques and experiments that are often questioned or interpreted with prejudice, faith, opinion or sceptical caution. In the main, such results are usually derived either through empirical analysis or simulation. A third technique, analytical modelling, is less widely adopted but can exhibit a number of advantages over the other two approaches. In particular it can be used in the design phase to predict the cost of a proposed system. This paper describes our experiences in designing, calibrating and validating a model to compare recovery costs in database systems called MaStA. The model itself is analytical but the application and validation of the model make use of simulation and direct measurement. So in MaStA the three techniques are complementary and each has a role in the design and development cycle.

Introduction

Quantitative performance analysis to determine behaviour and efficiency is an essential part of any system construction. To ascertain and measure performance a set of assumptions must be defined on which an experimental base can be built. Three different approaches to measurement are commonly used - empirical analysis, simulation and analytical modelling. These three techniques essentially describe different points on the analysis spectrum, trading accuracy of results, cost of construction and prediction capability. At one extreme analytical modelling is the cheapest to produce, giving a measurement of average performance. Because of cost it can be used to sample over a large range. Analytical modelling can also be used to predict the performance of a system under design by describing the proposed system in terms of the model parameters and applying a validated set of experiments to it. At the other end of the spectrum empirical analysis through direct measurement produces precise results but is usually limited to a narrow range because of the cost.

Empirical measurement involves running applications or benchmarks on existing systems and taking measurements using hardware or software monitoring. Arguably this method produces the most accurate results of the three options but it can be expensive to build. More importantly empirical measurement only provides results from one point in the analysis space, essentially showing that under a particular set of parameters on a set configuration the system behaves in a particular way. Empirical analysis therefore has limited potential as a predictor for performance under varying loads and different platform configurations.

Simulation based analysis comprises of a suite of programs that are designed to capture the characteristics of the system under test. The idea is to model the system and act out its dynamic behaviour. By recording suitable measurements from running the test suite the behaviour of the system is approximated and it is thus possible to infer something about the performance of the real system. Unlike direct measurement, simulation can be used to predict the performance of a proposed system or enhancement and is usually easier to construct. Whilst the accuracy of measurements through simulation is dependent on how exact the system is modelled it has some advantages over empirical measurement. For example simulation can eliminate extraneous factors irrelevant to the experimental objectives and the experiment testbed can be run for longer periods than for a real system.

With analytical modelling the behaviour of a system is formed through a set of equations and the performance is derived mathematically. This involves the construction of a number of parameterised functions that approximate the attributes of the system components in terms of workload characteristics. A number of simplifying assumptions are necessary to keep the model tractable and hence this method may only produce average performance figures [Leu88]. It is the efficacy of these assumptions and the extent to which they can be captured into formal equations that determine the usefulness of this approach. However analytical modelling is generally cheaper to build than simulations and can be more easily tailored to measure a wider variety of experiments. Simulations can include more details than analytical models where these details are hard to describe in the model. Simulations may be more expensive to build as they need programmed, debugged and validated where as an analytic model requires only calibration and validation.

In summary, analytical modelling can often be the cheapest way to ascertain reasonable performance measurements for a system. The validity of such a model is dependent on how justifiable the underlying assumptions are and its accuracy comes from the extent to which the mathematical models capture real behaviour. With such a modelling technique it is usually straightforward to re-calibrate for a different platform or workload and as such it can be used to compare and predict system performance with acceptable accuracy and minimum expense. However analytic models produce performance results based on average behaviour and hence are unlikely to predict so-called phase changes [ABJ+92] where a dramatic change in behaviour occurs.

The rest of this paper outlines the design, calibration, validation and experimental testbed of an analytical cost model called MaStA developed to measure performance of recovery mechanisms in DBMSs.

The MaStA Model

The MaStA is an analytical model designed to provide a framework for comparing the costs of recovery mechanisms under a variety of different workloads and configurations, and may be used to guide the choice of mechanism for a particular application.

MaStA focuses only on the I/O costs of recovery mechanisms. It categorises I/O operations performed by recovery mechanisms by the manner in which they operate. For example, a mechanism may perform data reads on the database and data writes to the log, both of which are categorised for that recovery mechanism. These categories are termed I/O cost categories and the overall cost of a mechanism is the sum of the costs of its constituent I/O cost categories.

$$\text{Total Cost} = \sum_{i} \text{CatCost}(i), \ (i \in \text{Categories})$$

Each category is assigned one or more I/O access patterns according to the properties of the I/O operations performed by the mechanism within the category. For example, log writes may be assigned sequential write costs in a log-based mechanism. The number of accesses incurred in a category of a particular access pattern is derived from a workload function composed of workload variables such as the number of reads and writes performed by the application and locality. The cost of an I/O cost category is a product of the number of accesses of a given pattern and the cost of the pattern, or the sum of a number of such products.

$$\text{CatCost}(i) = \sum_{j,k} n_{i,k} \times A_k, \; (j \in \text{Occurrences}, \; k \in \text{Access Patterns}, \; i \in \text{Categories})$$

The derivation of a cost estimate for a particular combination of mechanism, configuration and workload is derived by analysing:

- The workload: measuring and choosing values to predict the workload. This can be done either by tracing real applications or from benchmark suites or alternatively from simulation.
- The mechanism: identifying cost categories and assigning access patters to these categories. The costs to each cost category is achieved by calculating the number of accesses from the workload abstraction.
- The configuration: determining the cost of each access pattern for each platform. This calibration may be done experimentally, analytically or by simulation.

Validation

The MaStA model is based on four critical underlying assumptions.

I/O Assumption:: In applications where variations in total costs of using different recovery mechanisms are significant, the variations in the CPU costs incurred are insignificant compared to the variations in the I/O costs.

Cost Category Interaction Assumption: The interaction between the different categories of I/O accesses is not significant; that is, the cost of running the I/O stream generated by a given recovery mechanism is not significantly different from the sum of the costs of running the streams of each I/O cost category separately.

Access Pattern Cost Assumption: To make predictions of the relative costs of recovery mechanisms for all workloads, it is sufficient to assign a predicted average cost to each I/O access pattern.

Workload Assumption : The cost of running the I/O stream generated by an application is approximately the same as running the I/O stream generated by the workload abstraction.

To validate the assumptions of MaStA [SCM+95a, MCM+95] a variety of workload traces produced by a synthetic workload generator and by the OO1 [CS92] and OO7 [CDN93] benchmarks are recorded. Each workload trace records the database accesses performed by a particular benchmark query and allows the same workload to be executed multiple times on different recovery mechanisms and platforms.

The workload traces are executed over three different recovery mechanisms and on two platforms - a Sun SPARCStation and a DEC Alpha configured with different devices and operating systems. The I/O and CPU costs of executing each workload trace are measured and traces of the I/O accesses performed are recorded. In addition, the workload traces are analysed in terms of the MaStA categorisation to provide I/O cost predictions of the workloads. These predicted and real I/O costs are then compared to validate the assumptions of MaStA. A strength of this strategy is that by validating each assumption for more than one platform, operating system and device, it illustrates the independence of the MaStA assumptions from these components.

Experiments

In total over 2000 separate experiments were performed and recorded in the validation process and each experiment was repeated a number of times so that any fluctuations in the costs measured could be factored out.

To investigate whether the I/O costs measured in the validation procedures are accurate, the costs are recorded using two methods . The first measures the cost of individual I/O operations using the standard library functions. The second measurement method calculates I/O costs by subtracting the CPU costs from the total costs. An average variation of 1.8% was observed between the two methods of measuring I/O costs.

To avoid platform interference the experiments were run on raw partitions under a single-user system. To ensure that the results of the experiments performed in each validation procedure are comparable, the same area of disk is used for each experiment. Finally to further eliminate platform dependence the set of experiments were also run on the same drive configured on both systems.

References

[CDN93] Carey, M.J., DeWitt, D.J. & Naughton, J.F. “The OO7 Benchmark”. In SIGMOD Conference on the Management of Data, Washington, DC, May 1993.

[CS92] Cattell, R.G.G. & Skeen, J. “Object Operations Benchmark”. ACM Transactions on Database Systems 17,1 (1992) pp 1-31.

[Leu88] Leung, C.H.C. “Quantitative Analysis of Computer Systems”. John Wiley & Sons Ltd. 1988.

[MCM+95] Munro, D.S., Connor, R.C.H., Morrison, R., Moss, J.E.B. & Scheuerl, S.J.G. “Validating the MaStA I/O Cost Model for Database Crash Recovery Mechanisms”. In Proc. OOPSLA'95 Workshop on Object Database Behaviour, Benchmarks and Performance, Austin Texas (October 1995).

[SCM+95a] Scheuerl, S.J.G., Connor, R.C.H., Morrison, R., Moss, J.E.B. & Munro, D.S. “The MaStA I/O Cost Model and its Validation Strategy”. In Proc. Second International Workshop on Advances in Databases and Information Systems (ADBIS'95), Moscow, June 27-30 1995, Volume 1 pp 165-175.

[SCM+95b] Scheuerl, S.J.G., Connor, R.C.H., Morrison, R., Munro, D.S. & Moss, J.E.B. “The MaStA I/O Trace Format”. Technical Report CS/95/4 (1995), University of St Andrews.