International Journal of Advanced Engineering and Management Research
Vol. 2 Issue 3, 2017

www.ijaemr.com ISSN: 2456-3676

FORECASTING COMPUTER PERFORMANCE USING MATHEMATICAL MODELS – AN INCREMENTAL METHOD FOR ANALYSIS OF BANKING SYSTEMS

Ion LUNGU

Bucharest Academy of Economic Studies

Ion DOBRE

Bucharest Academy of Economic Studies

Florin-Catalin ENACHE

Bucharest Academy of Economic Studies

Adriana-Nicoleta TALPEANU

Bucharest Academy of Economic Studies

ABSTRACT

This paper surveys the contributions and applications of mathematical forecast models in the field of banking data networks.

The most important models (mathematical forecast models, queueing models and benchmark models) will be described will be described, used and confirmed by computer simulations of real queues usually found in the banking computing systems.

The focus of this paper will be the mathematical forecast models. These are based on the probabilities theory and mathematical statistics, and some direct applications in the IT industry will be pointed out.

The goal is to provide sufficient information to computer performance analysts who are interested in using the mathematical forecast to model a network of banking computer systems using the right simulation model applied in real-life scenarios, e.g. overcoming the negative impacts of the European banking regulations while moving towards green computing.

Key Words: computer performance , banking system

Introduction

For the performance of database applications there is a direct dependency on the performance of the included SQL statements as well as the database itself. Anyone who is concerned with the operation of databases could already observe the phenomenon that a database application runs completely satisfactorily and suddenly a performance slump is happening overnight. The batch processes are then no longer executed in a timely manner, and the users complain about overly long reaction times for online queries.

1.Workload statistics

Workload statistics are a prerequisite for creating performance forecasts. Each forecast is based on a baseline. A baseline is a characteristic workload in the considered time window. The statistics also form the basis for the work load monitoring thresholds. Forecast and monitoring require each other. While a forecast delivers the values for the expected workload development to the monitoring, the monitoring sends warnings and alarm signals about deviations from the expected workload to the forecast. The validation process examines the deviations and determines the causes. Deviations can originate, for example, from change management or can be triggered by excessive growth of the database.

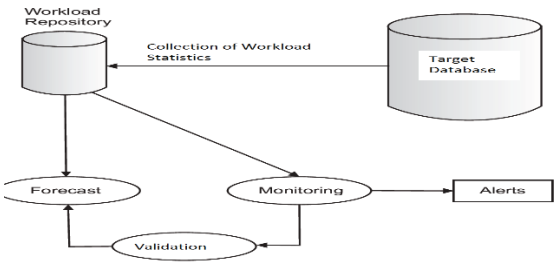

Fig.1 Process structure of Forecating, Monitoring and Workload-Statistics

Before collecting workload statistics, one must determine which data should be collected in which frequency. It is very important to mention that the collection process can lead to a significant performance impact and additional resource consumption on the target database. On the other hand, of course, statistics are needed in sufficient numbers and good to very good quality. Workload statistics are raw data. They must be prepared to form a baseline and provide a basis for predictions, leading to a very complex process.

2. Forecast models:

By using the example in Fig.2 below, workload statistics could be collected and a simple forecast model for dependency of CPU consumption could be set up by the number of active online sessions[3]. The disadvantage of the model lies in the great uncertainty about the accuracy of the

prediction. Assumptions were made and not verified. In this example, the statistics were collected at the main load time of the system.

Based on the scatter plot graph, the assumption that a linear dependency exists can be made. Since the number of sessions is only in the upper area, this assumption contains a certain source of error for the forecast model. The hedging against this inaccuracy was offset by the provision of a buffer reserve.

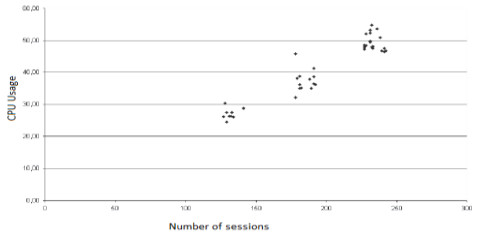

Fig.2. Scatter-Plot of Workload-Statistics

In mathematical statistics and probability calculation, there are several methods that compensate for this shortcoming. A regression analysis can be used to verify the quality of the sample using statistical methods. The linear dependence of the selected variables can also be demonstrated. These methods fall into the field of mathematical forecast models, which are successfully used in many other areas.

Benchmark models deliver results from processes that run on real computer systems and real databases. They thus capture the complexity of an Oracle database much better and deliver real results. The drawback of benchmark models is that they are complex to prepare and perform. It takes much more time to benchmark than to make a prediction with mathematical methods. For large databases it is economically very expensive to provide test systems of the same size. In these cases, it is possible to implement benchmarks on smaller systems and to scale them to a large system. For this purpose, there are proven scaling methods. This approach is very interesting from an economic point of view. In addition, by scaling a real benchmark, a great prediction accuracy can be achieved. For the system implementation of applications into the production, this method not only saves costs, but also time.

The third group of forecast models is based on the queuing theory. It reflects very well the real processes in a computer system. The reliability of this method has been confirmed in many other areas. It is used, for example, for process control in banking systems. In a computer system, processes form queues on the CPU or on the I / O subsystem.

When queuing occurs, response times are increased. Increased response times mean performance problems. Every administrator in the production environment has questions like this:

- How much increase in workload does the current system sustain?
- How many CPUs must the new system have in order to ensure a sufficiently high performance of the database application for the next three years?
- How long can the database still grow without performance problems?

Queuing models are very suitable for answering such or similar questions.

The selection of the model (s) should always be problem-dependent. There is no standard recommendation for the most appropriate model. In the further course of the article, we will present models from the following three categories:

- Mathematical forecast models
- Benchmark models
- Queuing models

We have devoted a separate chapter to each of these models. The following section gives a brief overview.

2.1Mathematical forecast models

Mathematical forecast models are based on mathematical statistics and probability calculations. These models are used successfully in other areas such as sales and marketing, or in stock trading. The statistics are subjected to analysis procedures. This proves that the prerequisites of the model are fulfilled.

In particular, linear regression models have proven themselves in practice. As the name suggests, this is about linear dependencies. They can be used wherever linear dependencies are suspected and the projection to the predicted range does not violate the linearity. The quality of the sample is examined and assessed using statistical methods. It is doubtless whether or not there is a linear dependency.

Consider the example introduced in this chapter and the graph in Fig.2. The chart shows the sample used for the forecast. Considering the graph alone, doubts arise as to whether a linear dependency actually exists. By regression analysis, this sample would smoothly pass through and the predicate "unsuitable" for a linear regression model would be obtained. A statistic that runs over a longer period of time and offers a wider range of independent values provides a better basis for a forecast with the regression method. This requirement is met by the following statistics:

SNAP_TIME	USER_CALLS	ACT_SESS	CPU_USER	CPU_SYS	CPU_TOTAL
16.02.2017_20:07:17	34907	310	55.56	22.56	78.12
16.02.2017_20:07:07	34898	310	55.45	22.84	78.29
16.02.2017_20:06:58	34889	310	49.44	20	69.44
16.02.2017_20:06:49	34880	310	51.05	22.44	73.49
16.02.2017_20:06:38	34865	300	51.23	21.46	72.69
16.02.2017_20:06:27	34856	300	51.23	20.88	72.11
16.02.2017_20:06:16	34847	300	49.54	21.74	71.28
16.02.2017_20:06:03	34838	300	48.54	20.15	68.69
16.02.2017_20:05:52	34823	290	50.52	20.5	71.02
16.02.2017_20:05:42	34814	290	47.97	20.26	68.23
16.02.2017_20:05:32	34805	290	49.57	20.94	70.51
16.02.2017_20:05:22	34796	290	49.42	20.49	69.91
.....					
16.02.2017_19:47:44	33598	20	2.56	0.89	3.45
16.02.2017_19:47:35	33589	20	2.67	0.89	3.56
16.02.2017_19:47:26	33580	20	0.67	0.11	0.78
16.02.2017_19:47:17	33571	20	3.78	2.56	6.34

Table 1. Statistics on a larger data base (extract)

The sample has a much broader database in a range of 20 to 310 sessions. The corresponding chart in Fig.3 gives a purely visual impression of a linear relation over a broad range. The goal of the linear regression method is to draw a straight line through the points and thus to express the linear relation by a formula. For a simple linear dependency, the formula looks as follows: $Y = a + bX + e$.

X is the independent variable and represents the number of sessions. On the other hand, Y is the variable dependent on X and represents the CPU utilization in percent. The variable e is the error or the deviation.

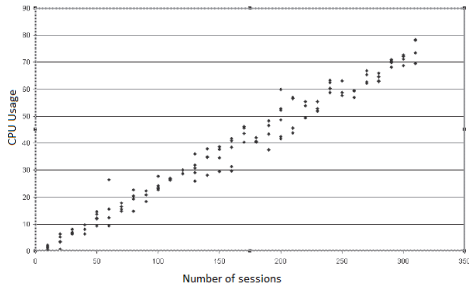

Fig. 3 . Scatterplot graph of improved statistics

Using the found formula, the dependency is projected onto the prediction range. But which of the many possible lines is best? There are different methods to determine the "best" line; The most familiar of them is the method of the *least squares*.

A further advantage of the linear regression method is that the prediction results can be assessed by means of so-called confidence intervals with specific numbers. A confidence interval expresses the probability with which the prediction values lie within a certain range. For example, in the forecast report, we can see that the CPU usage is 95% at a certain interval. If we later deal with the queuing theory, we will see that the dependency is not really linear in this example. However, the deviation is so small that this error is negligible. It can be expressed well by the confidence intervals.

Linear regression models are characterized by a high prediction accuracy.However, they have a sore point exactly when the linear region is exited. A projection of the found formula into this range would be extremely dangerous.

For the present example, it is known that the linear dependency is no longer present if the utilization of the CPU increases by more than 75 percent. This is roughly the limit at which a run queue is formed. The statistics in Fig.4 confirm this.

As we will see, the queuing theory is for this type of forecast because it takes account of the increasing response times as resources are used.

A component of the linear regression model is the *regression analysis*. It checks whether the statistics meet the prerequisites for using the model and, among other things, creates an error graph.

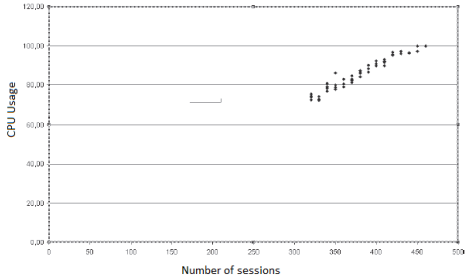

Fig. 4 Nonlinear range of statistics

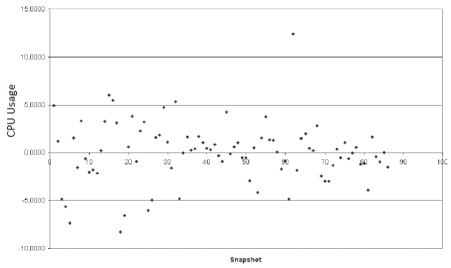

Fig.5 Error graphic of a sample

The distribution of the errors shows whether a linear dependency exists. If the distribution image takes the form of a rectangle as in Fig. 5, a linear relation is usually present. This can be excluded for other distribution images. However, the final confirmation is always providedby the regression analysis.

One method to check the randomness of a sample is error histograms.

Normality is given when the histogram takes the form of a bell curve. It is therefore also called normal distribution. Again, the mathematical proof in the form of the regression analysis must also be used. Other distribution images in any case allow the reverse conclusion that the sample is not suitable for the regression model.

Regression models are not limited to an independent variable. The Multiple Linear Regression Model describes the relation of a dependent one of several independent variables. This makes the simple regression model the special case of the multiple regression model. The prerequisites for both models are similar and differ only in their different dimensions. The formula for the multiple model is:

$$Y_i = b_0 + b_1 X_{1,i} + b_2 X_{2,i} + ... + b_k X_{1,k} + e_i \text{ , for i=1,2,…,n.}$$

While for a simple regression model the forecast values can be calculated using an equation, a linear equation system must be solved for the multiple model, which is not a problem in the computer age.

2.2 Queuing models

The example in the previous section has shown that a linear dependency no longer exists starting a given certain degree of utilization of resources. This situation can not be covered by linear regression models. Queuing theory chooses a different approach and reflects this behaviour. Queuing problems occur in many places in everyday life and in the economy. Thus, it is not surprising that it has found many fields of application outside information technology. The classic example is the call center. Customers call completely at any time and at different times, and must be served with as few waiting times as possible.

It has been found that the real sequences in a computer can be modelled very well by queuing models. Also, the CPU must work requests and build a queue when more request arrives than can be processed.

The running time of a request is also called the response time and is composed of the service time and the waiting time in the queue. The formula for this is:

TA = TS + TW.

This is exactly the reason why the curve in the previous example slides into the nonlinear range. The higher the system's utilization rate, the longer the queue and the greater the response time. Fig.6 shows a typical curve[1].

The curve shows again graphically what has already been described. From a certain degree of utilization, the linear regression method is no longer able to reflect the real processes in the model. The queuing theory, on the other hand, makes the reality in the nonlinear range very good. It does not matter, for example, whether you are viewing a CPU or an I / O subsystem. The curve is always similar.

The queuing model is therefore very interesting for many tasks. This is the answer to the question of the size of a database that can grow to a significant increase in response times. Surely you know the so-called Kipp effect. A database runs smoothly and relatively unobserved over a long period of time. Virtually overnight, performance problems are created that are so great that users make the hell hot. This effect can be explained by the curve in Fig.6. It runs over a large area very flat, almost linear, and then suddenly rises steeply. The queuing model can determine this time in advance and to avert disaster.

Or when thinking about planning a new server for an existing database that threatens to tip. How many CPUs should the new hardware have? What is the throughput of the I / O subsystem?

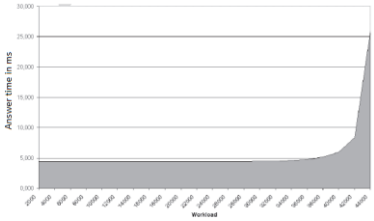

Fig.6 Distribution of service and waiting time according to utilization rate

As a rule, an estimate is then made with a large security buffer. As a result, the hardware is oversized, and an additional IT budget has been spent, which is urgently needed elsewhere. Here, too, a forecast with the queuing model can provide reliable values that allow you to provide sufficient resources for a high-performance database operation and, on the other hand, do not have to allocate any redundant resources.

With the help of queuing theory, another problem can be solved. Many database servers are poorly balanced. That is, they provide either too much CPU capacity in comparison to I / O throughput or vice versa. The chart in Fig.7 describes such a situation.

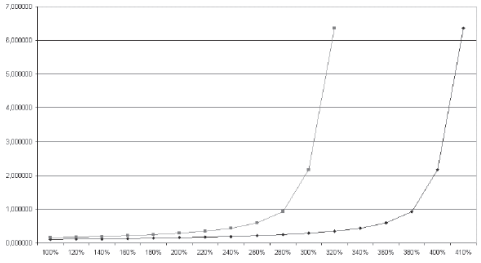

Fig.7 Auslastung von CPU- und I/O-Subsystem

While the response times of the CPU subsystem are already significantly increased with a workload of 250% and go into the critical range, the I / O subsystem can handle a much larger workload. The reverse is often the case. It has been invested in hardware that is not needed.

In Figure 8, we will see the results of a forecast using the queuing method to find the best hardware for a database server. The vertical axis shows the response times as a function of the CPU number.

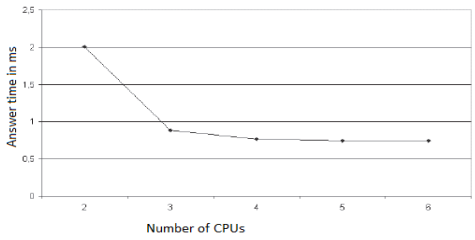

Fig.8 Detection of needed hardware

With a configuration of 3 CPUs, the response time is less than one second. On the other hand, the acceptance of additional CPUs leads only to a minimal reduction of the response times. The optimal configuration is found. A more than necessary oversized equipment can be avoided with this prediction.

The examples have clarified the results achieved with queuing models. They are particularly suitable when it comes to predicting response times or the degree of component utilization. The queuing method allows you to determine the optimal hardware configuration to provide neither too few nor too many resources.

2.3Benchmark models

Benchmark models deliver results from processes that run on real computer systems. They differ fundamentally from the other forecast models. They are more complex in their deployment and implementation than the mathematical and queuing models.

Their superiority is reflected in the accuracy and reliability of the predictions. Finally, a real system reflects the complexity of the Oracle database much better than any other model.

Benchmarks have a wide range of applications. They can be fed with real workload or workload simulations. On test systems that are identical in their equipment and size, the workload can be tested under certain conditions before transferring to the productive system. In this way, the effects of an increase in the workload on the test system can be analysed.

If the test system is smaller than the production system, the benchmark results can be scaled to the larger hardware. Scaling is a process supported by a mathematical method that transfers the found results to production. This method saves expensive and expensive testing in rented data centers and is very well suited for database software manufacturers to make predictions for the necessary hardware equipment in the customer environment.

A commercial feature of the Oracle Database, Oracle Database Replay supports these processes. The workload is collected on the production system, and the workload files are transferred to the test system and played there. The real workload from production can be used for the benchmark, and workload simulations need not be used. Database Replay is supported by the Oracle Enterprise Manager and can be conveniently operated via a graphical user interface. Benchmark models are a useful addition when used in combination with other forecast models.

They help to improve the quality of the workload statistics or can be used to test special cases. In order to scale to a larger system, the sample is first scrutinized using statistical methods. Like the linear regression models, it is subjected to an analysis to prove that the prerequisites of the model are met.

In the subsequent scaling, for example, the results are projected from 4 to 16 CPUs. A common scaling method is Amdahl's law, which can be well applied to the parallelization of computer processes. In his approach, Amdahl assumes that the speedup of the parallelization is limited by the serial components in the process.

The graph in Fig.10 shows the relationship between physical CPUs and effectively effective CPUs with different serial components from "zero" to 0.5. If the serial components are equal to "zero", the number of effective CPUs is equal to the number of real CPUs. As the serial portion increases, the curve becomes correspondingly flatter.

This corresponds to the behaviour, as is known from practice. Because we know that 10 CPUs do not provide ten times the performance of a CPU.

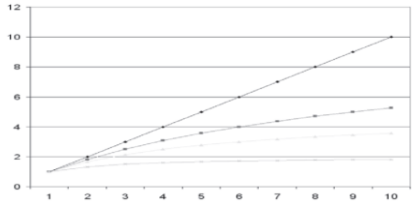

Fig.9. The Amdahl's law with different serial parts

Although the Amdahl's law - after all, it is from 1967 - is controversially discussed, it has retained its validity. An Amdahlian criticism is the fact that an increase of the degree of parallelization thanks to the serial parts in the process no longer makes sense, since the speedup increases only minimally.

This fact is still true, but was criticized by Gustafson, who criticizes the fact that Amdahl keeps the problem constant and distributes it across the processors.

Gustafson, on the other hand, assumes that in practice the problem size with the processor number also grows.

In other words, as long as the workload is guaranteed to keep up with the performance and parallelization level, further parallelization can still provide a gain in performance. For databases, the Super Serial method developed by Gunther is a very suitable scaling mechanism. Gunther[2] refined the formula with an additional approach. He says that as concurrency increases, the concurrency problems intensify and introduces a second factor into the formula. This is exactly the problem that we often find in databases. Thus, as the degree of parallelization increases, the probability of the occurrence of wait events due to parallel access to the same resource increases, e.g. Buffer Gets or Latches and Mutexes.

For Oracle databases, there are default or experience values for the two parameters of the super-serial method. The corresponding graphic is shown in Figure 10. This corresponds to the practical experience that 16 real CPUs have an effect in the range of 10 to 12. The factor can be adapted for special application profiles, which already have a high proportion of concurrency. The curve then runs correspondingly flatter.

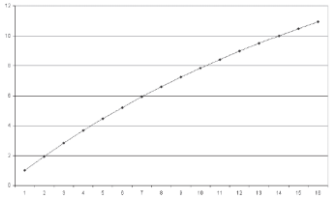

Fig.10 Scaling according to Gunther

Application area. This means that the tests can be carried out on smaller systems before the hardware purchase and the results can be scaled to the production environment. By combining with queuing models, the prediction precision can be increased and the optimal configuration can be determined. This approach not only saves hardware costs, but also shortens the time to start production.

Manufacturers of database applications often do not have the test environment of the required scale to test the performance of their products in production size.

Through benchmarks on small test systems they can still find the character of the application and recognize possible scaling problems. Problems in the area of concurrency can thus be discovered and eliminated before delivery.

Transferring the problems to production can lead to a multiplication and the performance of the application into a critical area.

By the scaling, itself, the expected behaviour in the production environment of the individual customer can be determined.

Conclusions

The search for the causes and the elimination of the problems is often a time- and cost-intensive process after the occurrence of the event, which is accompanied by disturbances in the operating sequence and a loss of confidence. It is often attempted to solve the problem by means of immediate measures such as an expensive hardware upgrade. The effects are, however, crushing, since the actual problem causes are not eliminated. In an increasing number of areas, the dependency on the right-of-time provision and transmission of data is meanwhile so great that the associated disturbances cause considerable economic damage. Banks and insurance companies are already predicting the failure or delay of a few hours for business-critical applications with multi-digit millions. Service providers work on the basis of Service Level Agreements (SLAs) and must expect compensation claims from their customers.

Many companies are therefore trying to integrate performance into their risk management. However, due to the complexity of the topic and the lack of available methods and tools, this is often only possible with moderate success. Added to this is the fact that performance monitoring is costly with conventional methods.

But how can you solve the problem? Partially, the productive introduction of databases and applications is a great effort to ensure functionality and performance of the product. Nevertheless, the database tilts after a few months, and the performance losses lead to operational disturbances.

By combining mathematical methods with a pragmatic approach, it demonstrates how permanent monitoring of business-critical databases can take place and cost-effective, reliable predictions can be made. The predictions do not only serve to improve performance stability, IT budgets can be planned more reliably and more efficiently.

References

[1] R. B. Cooper, Introduction *to Queueing Theory, Second Edition. New York: North Holland, New York,1981*

[2] N. Gunther, *Guerrilla Capacity Planning - A Tactical Approach to Planning for Highly Scalable Applications and Services*. Springer, 2007.

[3] F.-C., Enache,*Stochastic Processes and Queueing Theory for Cloud Computer Performance Analysis*, In: Conference Proceedings of the 14th International Conference on Informatics in Economy, pp13-19, 2015