The statistical models of parallel applications performance

Justyna Berlińska[*]

*Technical University of Szczecin, Institute of Manufacturing Engineering,
Piastów 19, 70-310 Szczecin, Poland*

Abstract

In the above raport the usage of the statistical methods to predict the efficiency of the parallel application was proposed. Statistical models can be helpful in choosing the suitable working environment and also parameters to put in motion the applications. With the help of statistical models it can be stated if the inclusion of parallel computers is justifiable. According to the execution time, speed up with regard to sequential program and efficiency of exploiting the processors.

1. Introduction

The parallel applications have been quite willingly used for solving problems which are complicated for calculation. However, it is hard to estimate the time execution and efficiency. Literature shows that the maximum processing speed equals to the product of the number of processors and the speed of one processor. Moreover, it is impossible to parallelize completely the code and the hardware characteristics in practice.

The real time processing speed is assessed in an experimental way. The number of operations performed in a program is divided into the measured time. This kind of the procedure is applied to machines already in use and the performance assessment depends, to a large extent, on the software used in the experiment [1]. Therefore, in order to achieve complete characteristics, benchmark programs are created, each of which tests the efficiency of typical operations processor [2].

It seems advisable to create mathematical performance models that characterize execution time, efficiency and speed up of a parallel program. A performace model gives information about one aspect of parallel aplication [3].

---

[*]*E-mail address*: majaab@ps.pl

Such a model allows one to determine what algorithms will be effective depending on the hardware profile and the software operating parameters.

In order to create a model, one needs a ready application as well as additional information provided by the software manufacturers.

The person creating a model will treat the application as a 'black box', so the parallel algorithms and the compiler will not have to be determined as their choice should have been made at the stage of designing the system.

The model will also ignore these factors which cannot be uniquely presented as numbers in the model.

The model is supposed to be an approximation of the most important properties of the investigated object [4]. It will be the basis on which conclusions of the application performance will be drawn. It will make it possible to make a decision whether it is more effective to conduct a program on a parallel computer.

2. Methodology

The methodology of creating models describing application performance answers the question what factors should be taken into account when creating the model, how the measurements should be conducted, what model should be chosen, how its quality should be tested and finally how to use the model.

Before setting out to build a model, the program environment should be thoroughly investigated. All the variables which may affect the quality of modeling should be singled out [5].

A statistical model is built on the basis of experimental data. If the information is appropriately gathered, it can later be processed and further computer analyzed. The more experiments have been carried out the more precisely the real data can be fitted into the model. This kind of information should be delivered along with the application by its manufacturer.

Having both this data and the available hardware parameters, one can embark on building a model which will make it possible to describe how different measurable input values affect the investigated factor [6].

Having first determined the output variable (which can, for example, be time, speed up, efficiency), independent variables are to be identified. It is advisable to choose such variables which significantly affect the output quantity and, at the same time, are measurable.

As far as parallel programs are concerned, the main independent factors can be as fallows:
 – the number of processors or computers. It is well known that an appropriate allocation of processes to a respective processor can substantially speed up the time of executing the program. However, the number of processors should be adequate to how much time-consuming and parallel the program is and how it is divided. According to Amdahl's

law even an unlimited number of processors will not speed up calculations more than the inversion/converse of the sequence part in the program run on a single processor [7].

– The number and size of threads. The main principle of parallel calculation is to divide it into parts – threads, each of which can be computed on a separate processor. After completing a series of operations, constituting each of the threads, there comes a synchronization stage where the results of all respective threads join in to create a final result.

– The method of access to the common resources. Parallel computers are expensive and the advantages brought by the increased speed of computation do not always outweigh the initial costs connected with the purchase. Distributed computing seems to be an alternative for parallel machines, an alternative which is both readily accessible and inexpensive.

– Hardware platform. The efficiency of computation is influenced not only by the choice of a model and parameters of a given processor, but also by the kind and capacity of memory and the data transmission speed. An appropriate hardware architecture, guaranteeing a quick access to common resources significantly improves the efficiency of running an application.

– The features of tasks executed by an application.

Some variables may not be used for building a model, as there is no access to them at the level on which the application is run. The following belong to such a class of information: a method of distribution of iteration space for loop parallelization, parallel algorithms, different kinds of compilers, operation environment.

It is to be decided, which of the above pointed out factors are, in every given case, most significant and have the greatest influence on the investigated process.

Depending on the distribution of variables in a model, the kind of regression should be determined. In order to make it easier a trend line can be plotted. It will show the general data structure as well as the initial form of approximating function.

The models of multiple regression should take into account all or only chosen combinations of independent variables. The regression line should have the highest value of R-square statistics or the lowest value of Mallows Cp statistics [8].

When all the data has already been collected and an approximating function has initially been selected, computer tools can be applied so as to verify the correctness of initial assumptions.

Estimation makes it possible, with initially assumed probability, to create an evaluation called confidence interval, which defines the probable range of calculations' deviation from the real values [8].

A verification of statistical hypotheses aims at examining earlier formulated statistical hypotheses, so it affects the decision making process.

After the initial stages of creating a model, verifying it, and eliminating its errors have been completed, one can start predicting how the application will behave.

3. An example of building a performance

On the basis of the above presented methodology, a model has been created. It measures the executing time and the efficiency of the application which is used for solving a system of linear equations by means of Gaussian elimination method.

Such a model allows one, without actually the need to run the application, to determine the time of completing the task, the speed up in relation to the sequencing program, and to assess how effectively processors have been used in the system. These values depend, to a large extent, on the following factors:

– the number of equation systems, which is equal to the number of variables to be calculated. The tested application was run for the systems ranging from 1000 to 2000 equations at the step set at every 100. The time results received in the tests for the given file sizes are shown in Fig. 1.
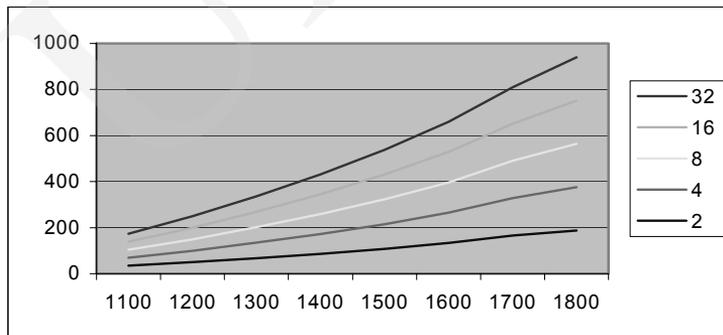

Fig. 1. The time of running the program with a given size of a problem depending
on the number of threads

– a number of threads – the program was run with 2,4,8,16 and 32 thread divisions.

The obtained results show that it is possible to use the linear form of the regression line.

For the investigated group of results, a regression model was created by means of the STATISTICA program. The model defines the functional relationship of the expected time variable value $Y = TIME$ (the time of running a parallel program) from non-random variables $X_1 = SIZE$ (the size of the matrix) and $X_2 = THREAD$ (the number of threads). Due to the insignificance of

coefficients describing the number of threads for the regression line, the model general form $E(Y/x) = b_0 + b_1X_1 + b_2X_2$ was simplified.

Only one function was chosen for the regression model. Only the coefficient for the variable SIZE of the estimated equation: $Y = -215,241 + 0,22*SIZE$ is significant. It is evident from the value of p level, which is lower than the adopted significance level α (α=0.05), the value of t-student distribution and from the F − Snedecor's statistics substantially exceeding the critical values ($t_{0.005/2.45}$=2.014, $F_{0.05,3.45}$=2.85). The square root of the multiple correlation coefficient R^2=0.9868 confirms a very good fitting of the regression lines to the experimental data.

The model describing the speed up of running a program defined as the quotient of the sequencing time and the parallel running of the program takes the following form:

$$Y = 8.1161125 - 0.006862*SIZE + 0.000002*SIZE^2 .$$

For the efficiency of paralleling defined as the quotient of speed up and the number of processors, the model general form $E(Y/x) = b_0 + b_1X_1 + b_2X_2 + b_3X_1^2 + b_4X_2^2$ for the random variable Y = E (the efficiency of running a parallel application), X_1 = SIZE (the size of matrix) and X_2 = THREAD (the number of threads), was reduced to the following form: $Y = 2.029031 - 0.001715*SIZE + 0.000001*SIZE^2 .$

Conclusions

The created model may be useful in making decisions about the conditions and parameters of running an application. It may also be conducive in choosing an optimal variant of running the application bearing in mind the time, speed up, efficiency or another investigated factor.

A model makes it possible, without the necessity of actually running the program, to select variables most or least significantly affecting the object.

Using a model may help reduce the time and costs and it may also provide us with the information whether it is advisable to apply expensive parallel computer architecture or it is better to use much more economical distributed systems.

References

[1] Bielecki W., *Essentials of Parallel and Distributed Computing*, Informa, Szczecin, (2002)
[2] Cuthbertson K., Hall S.G., Taylor M.P., *Applied Econometric Techniques*, Ann Arbor The University of Michigan Press, (1992).
[3] Dongarra, J.J., Gentzsch W. *Computer Benchmarks*, Advances in Parallel Computing, North-Holland, (1993) 8.
[4] Foster I., *Designing and Building Parallel Programs*, Addison Wesley 1994; www.mes.anl.gov
[5] Nowak E., *Zarys metod ekonometrii*, PWN, (1998), in Polish.

[6] Naishlos D., Nuzman J., Tseng C-W., Vishkin U., *Evaluating the XMT Parallel Programming Model*, To appear in the Proc. of the 6th Workshop on High-Level Parallel Programming Models and Supportive Environments (HIPS-6), April (2001).
[7] Karbowki A., Niewiadomska-Szynkiewicz E., *Obliczenia równoległe i rozproszone*, Oficyna Wydawnicza Politechniki Warszawskiej, in Polish.
[8] Kozielski S., Szczerbiński Z., *Komputery równoległe*, WNT, Warszawa, (1993), in Polish.
[9] Welfe A., *Ekonometria – metody i ich zastosowanie*, PWE, Warszawa, (1995), in Polish.
[10] STATISTICA PL dla Windows (Tom III), Statystykq II, Statsoft, (1997), in Polish.