

Economic Parallel Models – an Important Step in the Development of Implemented Applications in Grid Computing

Ileana Dumitru

Faculty of Economic Sciences, Petroleum-Gas University of Ploiești, Bd. București 39, 100680, Ploiești, Romania
e-mail: Ilenush2003@yahoo.com

Abstract

As a result of the studies (based on materials of scientific-bibliographic sources) on the notions of parallel algorithms and parallel models (mechanisms) directly applicable in the implementation of applications in Grid networks (Grid Computing), it turns out that by using properly and constantly the principles, patterns, and specific techniques, there are methods that allow a perpetual development in the treatment (solution) of high complexity problems in the current decade and into the following on. This article makes a foray into the fundamental concepts of parallel algorithms and manages to expose and to settle under the form of a mathematical model a classical economic problem – “profit or loss in economic activity”.

Keywords: jobs, tasks, level equations, equations of rhythm, Grid Computing

JEL Classification: C39, O21, O30

Introduction

In order to be able to run an application on a grid network, it has to be broken down into multiple processes. This decomposition assumes partitioning and allocation. The partitioning means to specify a set of tasks (or jobs) that implement the algorithm in a network Grid in the most efficient manner. Allocation is the process of division of tasks between network resources.

The performance of a parallel algorithm corresponding to a Grid Computing application - depends on the application granularity. Granularity refers to the size of a task compared to that extra consumption of resources necessary for implementation. For tasks of large dimensions, the volume of calculations performed is much greater than that required to implement. Therefore, a clear solution of an algorithm with parallel computing is the partitioning in large tasks (coarse granularity). In other words, tasks for large - scale lead to reducing the number of processes, decreasing the degree of parallelism, which makes desirable the partitioning of the problem in a large number of tasks of small size (fine granularity).

In other words, improving the performance of parallel algorithms is done by finding a compromise between the size of the task and the additional consumption of resources in the Grid network. Through the process of *clustering* (when tasks are grouped so that additional

consumption of resources from within the network to be larger than the consumption of resources from outside the network), it outlines a clear solution for improving the performance of parallel algorithms. In practice, the number of resources (processors) is (generally) correlated with the size of the problem so that the duration should be limited to an accepted period of time. In order to achieve this objective, it is very important that the algorithm use all available resources and minimize as far as possible the additional consumption of resources from outside the network.

Grid infrastructure provides computing resources and extensive storage for the solution of problems of high complexity.

From the study of literature and bibliographic resources, we have identified two categories of algorithms and/or techniques that can allow the allocation of resources in Grid Computing:

- techniques oriented on tasks (tasks);
- techniques oriented on workflows activities - orders (jobs).

These techniques aim to make efficient allocation of the jobs of the entire flow and on their basis there can be changed/organized the allocation options of any job based on its component tasks. Among the research directions for obtaining effective solutions to the problems of high complexity, we would like to mention the method of using adaptive allocation strategies. It offers the possibility of implementing the capacity of changing in a dynamic way the decisions of network resources allocation based on past, current and future status of the system.

Defining Ways of Partitioning of an Application under the Form of Jobs

The presentation of programming models for applications that can be solved in Grid Computing, involves the description of algorithms in pseudocode, algorithms that can be easily transposed in applications that can be deployed in such a network. A common feature of the programming of these types of applications lies in the fact that all processors involved in network execute the same program (application), but each of them uses its own data set.

In relation to the organization of memory in Grid Computing configuration, a number of secondary characteristics of the programming model is involved, features that refer mainly to the cooperative mode of the resources involved. In the case of a type of shared memory, communication between resources will be achieved through the common memory, in which case, an ideal architecture of the programming model is the *architecture of the PRAM* (Parallel Random Access Machine).

In the description of algorithms that can be implemented in Grid Computing, there are two important details:

- *execution of instructions in such a network is not synchronous* - that is, each network resource will execute jobs and/or tasks at its own pace;
- *not all resources involved run the same type of job and/or task* - although software used for network resources is the same, there may be differences in the ways of execution of jobs and/or tasks.

A program is executed on a number of n processors of a network; in general the number of processors (n) is less than the total number of processors involved in the network, depending on the complexity of the problem that was proposed to be solved. Processors (resources) are numbered from 0 to $n-1$. This numbering is not in a static way and this is possible only if the job-application is enforced. However, processors (resources) can find out its own address based on a function of the operating system or through the data catalogue corresponding to the network.

In the description of algorithms, it is important to note that the address of a resource will be contained in one variable k , thus confirming that the algorithm is variable and is reported to the processor (resource) P_n .

Description of parallelism

An algorithm model of data applications in Grid Computing describes implicit the parallelism, and facilitate its expression; we assume that all n resources simultaneously start the execution of tasks and/or jobs (which in reality is not always valid). However, this type of presentation has the role of understanding the algorithm as long as jobs and/or tasks are carried out in parallel.

In the descriptions of algorithm parallelism is necessary to introduce an instruction - *instruction in parallel*, instruction that doesn't mean necessarily parallel execution, but certainly reflect its potential.

Creating a processing algorithm for implementation of jobs in Grid Computing

Starting from the model *Six interconnected networks*¹ the algorithm of solving the current problem is analyzed and exposed; this algorithm was described as a mathematical model that can be easily implemented in Grid Computing.

The model can be described as a system of equations which expresses the system states in a given moment of time relative to a previous time.

The *level equations* and *the equations of rhythm* lead to new levels and rhythms of the original system of the model.

An important issue in support of the model is that the time intervals between solutions are relatively short and, without discussion, led by the dynamic characteristics of the real system. Thus, a system of equations may lead to getting the necessary and adequate solutions. Another approach to solve a system is given by Gauss's method for solving systems of equations, the method which succeeds in bringing a complex form of system to the upper triangular form (obtain zeros below the main diagonal) and the solution will be obtained very easily.

The system contains equations that describe the dynamic relationships of a set of variables related to time periods. In these circumstances it is noted that the periodic (time intervals) calculation equations will be required for obtaining new conditions.

The detailed expression of system variables

All system variables are expressed (reported) in (to) time and, more clearly, in (from) time to time $\Delta_t \rightarrow 0$ (time intervals as small) so the approximation curve representative expressing the economic development to be better.

The system has temporary landmarks: t_0 , t_1 and t_2 . Between these moments the time elapsed is guaranteed the same Δ_t .

¹ "Six interconnected networks" (example to describe a dynamic system model) gave by Jay W. Forrester from the Massachusetts Institute of Technology (USA), can create a distributed model to describe a job assignment sites in a Grid Computing. The discussion uses concepts like levels and flow rates as follows: levels - determine the decisions that control the flow rates; flow rates - determine changes in levels. Levels and beats up six interconnection networks, which constitute the industrial activity. Five of these are materials, orders, money, base resources and personal. The sixth, information network - is the one that bounds the others together.

The time t_1 is the present moment (current), once they are aware of the information related to previous period (past) - $[t_0, t_1)$ will be noted for its future expressions t_0t_1 .

The time t_2 is the future time. For these time we can't be able to express relative equations for t_1t_2 because these must be determined (calculated).

As stated at the outset, the system equations are of two categories:

- level equations
- equations of rhythm.

Level Equations are those that have priority in defining the model clearly at every time step for any necessary information.

The results of level equations will be used in the equations of rhythm.

Level equations show how levels are obtained at time t_1 , based on levels at time t_0 and rhythms of the interval t_0t_1 . Obviously at the time t_1 when are rated level equations, all necessary information is available and they will propagate (transmit) in step t_0 (previous time) of time.

The equations of rhythms are valued at the present time t_1 , once the level equations were evaluated. Thus, the equations of rhythm may have as input variables of the present values of the time t_1 .

The values determined from the rate equations (DECISION) define rhythms representing actions that will take place during the next t_1t_2 .

After evaluating the levels at the time t_1 and the rhythms for the time t_1t_2 time is advanced one step Δ_t . At the moment t_1 will play the role of t_0 as t_2 will become "present" - so there will be t_1 in the followed reasoning. In the same vein, t_1t_2 become rhythms t_0t_1 and the algorithm is repeated for new states of the system, which is advancing with one moment Δ_t .

The model shows the development of the system in time, given that the levels lead to DECISIONS and ACTIONS (rhythms), which in turn affects the levels. Thus, interactions within the system are forced to follow the algorithm (description) established in model equations.

System variables

Notations used are:

- The number of employees - N_a ;
- Deposit Stock - S_D ;
- The necessary Stock of retail - sale - S_N ;
- The rhythm of production characteristic to the company - R_f .

Starting on the anterior notations there are derived notations which are expressed reported on the level and/or the rhythm used in the equations underlying in the mathematical model.

Therefore:

- Level of employees at time t_i ($i=0,1,2$) will be denoted by $N_a^{t_i}$
- The rhythm of manufacturing in the interval $t_i t_j$ ($i=0,1$ and $j=1,2$) will be denoted by $R_f^{t_i t_j}$
- The constant delay in dispatch of retail - I_e

System Equations

As shown above, the system of equations will contain:

- Level Equations;

- Equations of Rhythms;
- Auxiliary Equations.

Each of these types of equations is given below. Relationships are taken in consideration and the variables are clearly explained.

Level equations

The levels represent the container of tanks (warehouses, inventory, balance sheets) of the system. They exist (for example the stock) even if the system is at rest, and there is no flow. The new values of the levels are calculated for each end of the ranges.

It is assumed that the level is changed into a rhythm between moments and no value is calculated between these times. A specific example of equations can define as follows:

$$S_R^i = S_R^j + \Delta_\tau (T_{Pd}^{jj} - T_{\sigma d}^{jj}) \quad (1)$$

Where:

- S_R^i, S_R^j - represents the real stock of retail outlets (units), “real” being used as a term to distinguish it from the “necessary” (desirable) or other terms of the stock;
- Δ_τ - represents the time interval (for example-weeks), the interval between assessments of the set of equations;
- T_{Pd}^{jj} - represents the disposal transactions received retail (units/week);
- $T_{\sigma d}^{jj}$ - represents transactions sent from retail sale (units/week).

The above given equation expresses the direct calculation relationship by which the present value of actual stock sales at the time i will be determined in relation to the amount of actual stock sales at previous time j plus the difference between the entry rhythm T_{Pd}^{jj} and the output rhythm $T_{\sigma d}^{jj}$ on the interval, the difference multiplied by the length of time Δ_τ that the rhythms are maintained. In other words, what we currently have in stock is what we had at the beginning of specified time plus what we got minus what we shipped.

It appears that the “dimensions” of each term of the equation are the same (well freight units), so that, in the right side of the equation, cargo units are obtained by multiplying the time (split weeks) with the rhythms flow represented in units/week.

Flow rates are always measured in the same units of time as a day, a week or a month, not in terms of the calculation interval Δ_τ . Units of time for the rhythms and time units for Δ_τ should be the same, for example, all in weeks or in months. The equation is valid and independent of time, as long as it does not exceed a maximum value (the maximum value allowed for the calculation is the relationship between the values of what will be tested in the system and the incoming and outgoing flows from these levels). Calculation interval may be modified without the need for any change in expression or equation of any constants that can occur. By mentioning explicit time limits Δ_τ in the equation, one can track the use of the common units of measurement which appear in the real modelled system.

Level equations are independent of one another. Each depends only on the previous time information i . It does not matter the order of solving equations level. At the time of the evaluation i , no level equation use information (outcomes) from another level equation at the same moment of time. A level i time depends on its previous value at time j and flow rates on the interval ji .

Variables that are classified as levels may submit units of measurement such as “units of freight per week”, which at first sight may indicate the existence of a rhythm. In reality, they are average beats which, by their very nature are actually levels not rhythms.

Equations of rhythm (features)

The equations of rhythms define the flow rates of streams system levels. Rate equations are “decision - making functions” which will be discussed later. An equation of rhythm is assessed in the present values of the levels of the system, often including the level and pace of that in which it enters. In turn rhythms cause changes in levels. Rhythm can be equations or equations of explicit decision or implied decision type without the use of a structural or formal distinction between them.

Rate equations, the equations of the broader decision are best suited for control of what was going to happen within the system. An equation that evaluates to the rhythm i to determine what decision will direct the flow rate on the interval of time immediately following the ik . Rate equations depend mainly on the restriction values only from the time i .

Rate equations are evaluated independently of each other within any period of time, as it does for the equations. The interaction occurs through their effect on the levels which in turn influences other rates in the times ahead. An equation rhythm determines immediate action to follow. After this action is quite “immediate” (well, the time interval Δ_t is small enough), it appears evident that the decision is not likely to be affected by the decisions taken at the same time in other parts of the system (a short enough span of time stops intercommunications of information between decisions within a step of time and decisions can depend on so many incoming information as are considered to be significant at one time). Therefore, the equations are independent of one another and can be assessed by the equations. A specific example of equations can define as follows:

$$R_i^{ik} = \frac{V_{a,g}}{d_{m,j}} \quad (2)$$

Where:

- R_i^{ik} - represents the output rate (units/week) within the time ik ;
- $V_{a,g}$ - the current volume is stored in the delay (units);
- $d_{m,j}$ - is a constant, the average length of delay (weeks).

This equation defines the output rate R_i^{ik} with its value for the next period of time ik . During the evaluation of this equation numerical values of $V_{a,g}$ and $d_{m,j}$ are necessary.

Auxiliary equations

In most situations, an equation rhythm gets very complex if it is couched only in terms of levels, as outlined so far. A better formulation (a definition) of an equation of rhythm can be done by using several independent meanings with concepts that are derived from the levels of the system. So it is convenient to have the idea of separation of an equation of rhythm in more component equations called auxiliary equations.

These are a great help in maintaining the defining model in close correspondence with the real system, as long as the separate may be most involved in the decision-making process.

Auxiliary equations are those that accompany the other equations and appear only occasionally, they may be replaced by further one to another (where there are several levels of auxiliary equation) and then to the equations of rhythm.

By substituting algebraic, the auxiliary variables can disappear on account of the increase in complexity and form of the equations of pace and, why not, easy grade understanding of the model equations.

Auxiliary equations are evaluated at the time i only once the level equations for the time i have been assessed, because as the rhythms of which they belong to, they need to present values of the levels (they need to be evaluated before the rhythms because their values will be substituted into equations of rhythm).

Unlike the equations of level or rhythm, auxiliary equations cannot be assessed in any order - some may be components of others. Two or more auxiliary equations can form a “chain” that must be evaluated in a specific order, so that an equation to be used in the next. A succession of substitutions can be possible only if the equations are formulated correctly. A closed loop of auxiliary equations (where two variables are dependent on one another) leading to a set of simultaneous equations (which is prohibited and unnecessary wording). An example of a “chain” of two auxiliary equations between the two levels and a rhythm can be shown as follows:

$$S_{Rd}^i = A_{Sr} \cdot N_{pd}^i \quad (3)$$

Where:

- S_{Rd}^i - is an entry which brings another level in the system and is a retail stock at the time i ;
- A_{Sr} - is a constant (a subsidiary) and represents the auxiliary stock of retail outlets;
- N_{pd}^i - is a level (retail receipt) at the time i used as auxiliary input S_{Rd}^i .

$$A_{cd}^i = A_{ad} + A_{dd} \cdot \frac{S_{Rd}^i}{S_R} \quad (4)$$

Where:

- A_{cd}^i - auxiliary (constant) current (at the time i) of transactions received from the sale of retail, used together with another level of equation of rhythm;
- A_{ad} - auxiliary (constant) of transactions received from retail sale;
- A_{dd} - auxiliary (constant) - dynamic transactions received from retail sale.

$$T_{pd}^i = \frac{N_{pd}^i}{A_{cd}^i} \quad (5)$$

Where:

- N_{pd}^i - represents a level for transactions in retail marketing at the time i ;

By substitution of equation (3) into equation (4) to obtain a result which can in turn be substituted into equation (5) and lead to a single equation of the form:

$$T_{pd}^i = \frac{N_{pd}^i}{A_{ad} + A_{dd} \cdot \frac{A_{Sr} \cdot N_{pd}^i}{S_R}} \quad (6)$$

As it can easily be observed, auxiliary equations are gone leaving instead only the equation of rhythm T_{pd}^i that depends only on constants. Each of the auxiliary equations defines the auxiliary variable that has an understood conceptual importance, by substitution, these concepts are lost without being able to be recovered during the evaluation system of equations.

An auxiliary variable depends on, in principle, only the levels already known and other auxiliary variables that can be calculated before being used.

The equations of initial values

They are used to define the initial values of all levels (and some of the rhythms) that must be known before the first step in the calculation of the model equations.

They are also used in the beginning for calculating the values of some constants of other constants. These equations are evaluated only once, before the start of any trial runs of the model.

Additional equations

This type of equations are used to define some variables that are not actual parts of the model, but that appear in the drawing of any significant reaction model values-if you want information that is not used by any of the model (for example, the sum of all stocks in the whole system).²

Share effects in decision functions

The dynamic behaviour of systems with information reaction is determined by the way in which a variable changes produce changes over another (other) variables.

This can lead to the idea that you get a high sensitivity of the system to the exact values of the parameters of decision functions (which is not always true). If the model is set up properly to represent the actual structure of our social systems information, he will present the same adaptability autocorrect that exists in real situations. In the formulation of the chosen model all the parameters are estimated for the decision. They act through the levels to get the rhythms controlled by decisions. Levels, in turn, are adjusted by the resulted decisions. An incorrect parameter specified in a decision function may lead to the disappearance of the modification operations (or transformation) of the model levels so that the rhythms appear tied up interlinked.

Conclusions

Specifically, what interested most is what results from the model in relation to the factors that cause changes in rhythms and levels, than the precision of the determination of the average quantities of rhythms and levels. A properly built model is often surprisingly unaffected by the possible changes in the majority of values of the parameters - even sometimes extreme changes. In a model, sensitivity to the chosen values for the parameters should not be greater than that of the real system at relevant factors.

The essence of the use and/or implementation of applications in Grid Computing are given by their ability parallelization and finding techniques and patterns of expression of the components obtained workloads, jobs and/or tasks.

References

1. Caprita, A. D., Current Developments in Grid Computing, *Informatică Economică*, Vol. 40, No.4, 2006.
2. Foster, I., Kesselman, C., *Grid 2: Blueprint for a New Computing Infrastructure*, Second Edition, Morgan Kaufmann Publishers, Elsevier, US.A, 2004.
3. Foster, I., Kesselman, C., Tuecke, S., *The Anatomy of the Grid*, International Journal Supercomputer Applications, 2001.

² Forrester, J. W., *Dinamică industrială*, Technical Publishing House, Bucharest (trad. after 8th Edition), 1980

4. Forrester, J. W., *Dinamică industrială*, Technical Publishing House, Bucharest (trad. after 8th Edition), 1980.
5. Grandinetti, L., *High Performance Computing and Grids in Action (Advances in Parallel Computing)*, IOS Press, USA, 2008.
6. Ladd, S. - *Guide to Parallel Programming*, Springer - Verlag, 2004.
7. Plaszczyk, P., Wellner, R. Jr., *Grid Computing*, MORGAN Kaufmann Publishers, Elsevier, USA., 2006.
8. Udoh, E., Wang, F. Z., *Grid Technologies and Utility Computing: Concepts for Managing Large- Scale Applications*, Information Science Publishing, 2009.