The Algorithm for Dynamic Distributing of Processes in Real Time Distributed Computer Systems

E.G.BARDIS*, N.G.BARDIS*, A.V.SIMONENKO**
Department of Computer Science
National Technical University of Ukraine
(Kiev Polytechnic Institute)
*Glyfada-Athens
Tainarou 66
16561
HELLAS
**KPI-20-20, Pr. Peremohy 37,
Kiev-252056,
UKRAINE

Abstract: The approach to a solution of the problem of dynamic distribution of jobs in real time systems is described, in this article. A method for transformation of initial data into the form that makes it possible to use the known algorithms to construct a plan of solution. The offered method allows scheduling of solution efficiency of computing units and the system of priorities -penalties- be fulfilled.

Key words: Optimization algorithms, job scheduling strategy, heterogeneity, parallel systems. Proc.pp..5881-5884

1 Introduction

The real time scheduling is characterized by solution of the following problem: definition of the plan of solution of a set of tasks with a given time of fulfilment and restrictions on time of tasks exit from the system. The system of scheduling should ensure realization of the requirements on minimization of summarized time of deviation of real tasks exit from the system (realization of the terms of solution) at initial temporal restrictions (the time of inflow of a task, time of solution and extreme time of exit of the request from the system is pre-set to the scheduling system) with full observance of the work order. In general setting up this task concerns to the NP-complete class.

In most cases real time scheduling system developers use static algorithms and define beforehand the maximum list of the tasks by supposing the worst case for deriving a static controlling table (plan). This plan is fixed and is used for unconditional fulfilment in a dynamic mode with the following assumptions:

- all temporal restrictions remain constant for the time of plan fulfilment;
- all tasks are can be solved in the limits of their critical time;

Otherwise a static list of priorities is formed by means of static scheduling techniques for use in the dynamic condition during job scheduling.

If the system of real time works only in the dynamic condition, the use of the agreements of static scheduling (where all is known beforehand) is unadvisable. One of the possible algorithms of time-table formation is selected and then carefully analyzed on its applicability in the expected dynamic environment. As a rule, in this case the algorithms which use scheduling by the list or a priority service are appeared.

As a rule, with passage of the tasks through the computing system the multi-level systems of scheduling are applied. The classical scheme of scheduling includes the schedulers of three types:

1. The scheduler of the upper level selects a population of tasks which is authorized to compete for capture of CS resources.
2. The scheduler of an average level selects a tasks which are authorized to compete for capture of time of the processor.

2. The scheduler of an average level selects a tasks which are authorized to compete for capture of time of the processor.

3. The scheduler of the lower level determines for each computing node what task (process) will be activated (solved) in it. Thus, the scheduler of the lower level has N-tasks (processes) subjected to activation for N–computing nodes.

In the given work the method for dynamic distribution of the requests resources of an inhomogeneous distributed computing system in a real time solved by the scheduler of the lower level is offered.

2 Problem Formulation

Let us consider general setting of a problem in a real time.

On the input of HDSD a great number of jobs arrives. Each job is characterized by three temporal parameters:

T^i_{en} - time of inflow of i-th input job in CS.

T^i_{ex} - time of an exit of i-th job from CS.

T^i_{w}- - execution time for i-th job in arbitrary units in a computing node that has maximum efficiency in HDSD.

HDSD has multitude of resources. Each resource is characterized by R_j efficiency.

For each of an i-th job the time of solution of the i-th task on the an j-th resource in arbitrary units may be determinet. For this purpose let us define R_{max} as $R_{max}=\max\{R_1,\dots,R_n\}$

Define relative efficiency of each computing node R_j in relation to R_{max}. To do this compute a set $Z_j=R_j/R_{max}$.

knowing relative efficiency of each node, we can determine the job-resource ratio in view of the time of work execution and efficiency of Computing Device (CD).

To accomplish this let us generate a matrix of coherency (MC), where each element determines the relative time of an i-th job being executed on a j-th resource with respect of its efficiency $MC[i,j]=T^i_w/Z_j$.

In a view of the fact that $1\geq Z_j>0$ the relative time of realization of each job on resources will be determined by a value of MC[i,j]. Because the task being executed in a real time must be finished up to given T^i_{ex} it is necessary to calculate $\Delta_t^{i,r}$

$$\Delta_t^{i,j} = T_{ex}^{\ i} - T_{en}^{\ i} - MC[i,j]$$

In the view of $(T^i_{output}-T^i_{input})>0$ and $(T^i_{output}-T^i_{input})>MC[i,j]>(T^i_{output}-T^i_{input})$:

$\Delta_t^{i,l}$ can take both positive and negative values. The positive values of $\Delta_t^{i,j}$ mean that the job i nominated on a resource j will be completed before the given term by the time $\Delta_t^{i,j}$, and the negative values mean, that the exit of the request from the system will happen after the given term by the time $\Delta_t^{i,j}$. (Table.1)

At computation of $\Delta_t^{i,j}$ it is necessary to take into account, that on the moment of scheduling all requests have the same base time of the beginning of scheduling irrespective of time of request arrival in the system. Besides, the operating time of the scheduler, i.e. time of scheduling must be taken into account.

Therefore, it is possible to take $T^i_{in}=T_{plan}+\varepsilon$. To account the penalty system $S'_{\Delta t}$, all $\Delta t^{i,j}<0$ must be multiplied by the corresponding values $S'_{\Delta t}$ for all j resources. As a result the initial matrix (Table.1) is obtained.

		RESOURSES					
		1	2	3	4	5	6
T	1	1	3	-2	-3	4	5
A	2	-2	0	-5	-6	-4	-2
S	3	3	6	1	7	9	1
K	4	1	-3	-4	-5	-1	-1
S	5	-1	4	-4	1	-5	-1
	6	8	1	2	4	3	2

Table.1

Table.1: The initial matrix

Corresponding to the requirement of task solution

in a real time the value $\sum_{z=1}^{n} |\Delta_t^{i,j}|$

should be more than or equal to 0. If to obtain the distribution variant with satisfaction of the requirement on an the exit time for all requests and with involvement into distribution of assignings with $\Delta_t^{i,j}<0$ is impossible their sum must be minimal.

3 Problem Solution

In such setting, the solution of the task of assigning the i-th job on the j-th resource can be reduced to the task of searching of the maximum matching in pairs in a weighed bipartite graph is solved by the Hungarian methods that has the temporary complexity of $O(n^3)$. It is necessary to note that searching of the distribution variant is fulfilled for the inhomogeneous computing systems (the degree of a heterogeneity is below 0.7) and the searching of the solution is carried out in a rarefied binary matrix of coherency. In works [3,5] if is proved that with a solution of a task of searching of a maximum matching in it is possible to apply adaptive algorithm which temporal complexity depends on factor of filling of a matrix coherency of a bipartite graph and with factor of filling less than 0.7 temporal complexities do not exceed about $O(n^{1.5}\log n)$.

However, for a problem put by on a comparison with classical needs modification of shaping of input data and base (initial) area of searching .Let's consider procedures of shaping of input data and area of searching on an example of a solution of a task of assigning for the 6 of tasks on the 6 processors. After handling initial temporal restrictions and account of efficiencies of each processor we shall receive an initial matrix $\Delta_t^{i,j}$) (Table.1)

The information in a matrix $\Delta_t^{i,j}$ takes into account only volume of work and relative time of its realization in view of efficiency of computing nodes. However heterogeneity SHDH causes necessity of the account of individual performances of each computing node (presence of sufficient operative memory, presence of necessary data in a node, presence of the necessary programs for realization of the tasks etc.) assumed with a possibility of realization of tasks. In view of these criterions the evaluation of a possibility of realization of each work in each node is necessary. For it the matrix of check conflict of assignings by each i-th of work in j-th node

$$Q_{i,j}=\prod_{x=1}^{p} C_x^{i,j}$$

(Table. 2) is formed.

Where: $C_x^{i,j}$ - degree of realization of x obligatory requirements for assigning i-th of the request on j-th resource;

At the following stage of shaping of an initial information it is necessary to execute a filtration of elements of a matrix $\Delta_t^{i,j}$ in the correspondence with values of the elements of a matrix $Q_{i,j}$. In an outcome is obtained a new matrix $\Delta_t^{i,j}$, where the numeral ∞ designates assignings which cannot be considered as possible. (Table.3)

	1	2	3	4	5	5
1	1	1	0	0	1	0
2	1	1	1	0	1	1
3	0	1	0	1	1	0
4	0	1	1	1	1	0
5	1	0	0	1	1	1
6	1	1	0	1	1	1

Table 2

Table 2 Matrixes of check of disputed assignings.

	1	2	3	4	5	6
1	1	3	∞	∞	4	∞
2	-2	0	-5	∞	-4	-2
3	∞	6	∞	7	9	∞
4	∞	-3	-4	-5	-1	∞
5	-1	∞	∞	1	-5	-1
6	8	1	∞	4	3	2

Table 3.

Table 3: Matrixes $\Delta_t^{i,j}$ of assignings in a matrix $\Delta_t^{i,j}$ after a filtration.

For shaping distribution of tasks on processors, in the correspondence with the requirements of Hungarian algorithm and restrictions superimposed our statement of a task requires to generate initial zone of searching. For it is feasible the following operations. As the positive values of the elements of a matrix $\Delta_t^{i,j}$ correspond to assignings which can be certainly included in a solution since any assigning appropriate to i, j coordinate does not contradict conditions of temporal restrictions, all positive elements is appropriated values 0, and negative elements we shall interchange a sign on positive . In an outcome is obtained new matrix $\Delta_t^{i,j}$ (0). The further operations on shaping initial area of searching is fulfilled in the exact correspondence with Hungarian algorithm.

4 Conclusion

The basis for searching of the solution is the Hungarian method which has time complexity $O(n^3)$, where n is the number of graph vertices of the initial matrix. However, we suggest the algorithm AMA [1-3] worked out by the authors instead of the traditional Karp-Hopkroff method which is utilized in the Hungarian method.

For maximal matching in pairs designed time complexity of solution search by means of the modified algorithm is $O(n^{1.5}Logn)$ that is less than $O(n^3)$. Authenticity of the time compexity calculation was verified on a programmed model. The results of statistical inverestigation are presented in Fig.4. Because AMA is an adopted algorithm and searching of the maximal matching in pairs has been carried out in a raveried matrix with the filling coeficient less than 30%, the temporary statistical complexity of ANP is less than $O(n^{1.5}Logn)$ and is close to the linear one.

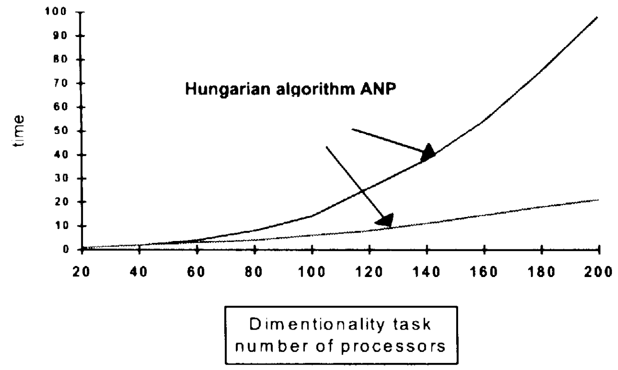

Fig.1

Fig.1: Comperative temporary characteristics of scheduling by the Hungarian method and by ANP.

References:

[1] Simonenko, Objective-Oriented Algorithm for Job Scheduling in Heterogeneous Parallel Systems, Second international conference on parallel processing and applied mathematics - Zacopane, Poland, Part 2, 1997, pp.378-387.

[2] Pham Hong Hang, Valery Simonenko, Adaptation of algorithm for job- resurce assignment in heterogeneous distributed systems, Conference PDPTA "96, Sunnyvale, California USA, 1996. pp.835-846.

[3] Pham Hong Hanh, Valery Simonenko, Objective-Oriented Algorithm for Job Scheduling in Parallel Heterogeneous Systems, 11th International Parallel Processing Symposium Geneve, Sweeden, the 1-5 of April, 1997 pp.127-145.

[4] Loutsky G.M, Samofalov K.G., Simonenko V.P, A method for analyzing the initial information and elaborating the solution strategy for the sceduling problem in a parallel computing system, Engineering Simulation, Vol. 13, 1996, pp. 601-610.