THE OPTIMALITY OF SOFTWARE RESOURCES STRUCTURING THROUGH THE PIPELINE DISTRIBUTED PROCESSING OF COMPETITIVE COOPERATIVE PROCESSES

Pavlov P.A.

Polesky state university, Pinsk, Belarus

In different spheres of human activities one has regularly to face important problems the effective solution of which is connected with paralleling of computational processes. The solution of such problems «unites the information from the following areas: architecture of computers and computing systems, system programming and programming languages, various methods of information processing, etc.» [1]. With appearance and usage of scalable systems many problems of parallel computing should be reconsidered. It's also necessary to have a look in a new way at the principles of computing; at the provision of univocal results of program run; at the efficient planning and distribution of concurrent processes [2]. In connection with it, modeling and research of simulators of parallel distributed processes, based on paralleling and pipelining, gain special currency.

1. The simulator of distributed processing of competitive processes. The structural elements for modeling simulators of distributed computing systems are concepts of process and software resource. As in [3], the process will be regarded as a sequence of blocks (commands) $Q_1, Q_2, ..., Q_s$ which are executed with the help of different processors. If all the blocks or part of them are executed by different processors then this process is distributed. To accelerate the execution the processes can be handled across interacting by means of information exchange. Such processes are called cooperative or interacting processes.

The concept of resource is used to identify any objects of computer system which can be used by the processes for the execution. Re-entrant resources are characterized by the opportunity to use several processes simultaneously. As for parallel systems, their characteristic feature is the situation when one and same sequence of blocks should be executed by processors multiply. This sequence will be called a software resource (SR) and a set of appropriate processes will be called competitive ones.

As in the works [3–5] a simulator of distributed processing of competitive processes involves *p* processors of multiprocessing system (MS), *n* competitive processes, *s* blocks $Q_1, Q_2, ..., Q_s$ structured into the blocks of software process, the matrix $T_d = [t_{ij}]$ of time execution software resource blocks by distributed competitive processes. The specified parameters keep changing in the range of $p \geq 2$, $n \geq 2$, $s \geq 2$, $1 \leq i \leq n$, $1 \leq j \leq s$. Let's suppose that all *n* processes use one copy of structured into blocks SR, and a full order of their execution is installed at a set of blocks.

Let's examine the parameter $\tau > 0$ which characterizes the time (system costs) spent by MS to organize parallel execution of SR blocks by a set of distributed competitive processes. In what follows we'll say that the above – listed objects of the simulator form a system of distributed competitive processes.

Definition 1. *The system n of distributed competitive processes is called multivendor if the time execution of software resource blocks* Q_1, Q_2, ..., Q_s *depends on the volume of processing data and/or their structure, i.e. different processes.*

Definition 2. *The system n of distributed competitive processes is called homogeneous if the time execution of software resource blocks by each of competitive processes equals, i.e.* $t_{ij} = t_j$, $i = \overline{1,n}$, $j = \overline{1,s}$.

Let's consider that the interaction of processes, processors and blocks is submitted to the following terms [3–7]: 1) none of the software resource blocks can be handled simultaneously by more than one processor; 2) none of the processors can handle simultaneously more than one block; 3) each block processing is executed without interrupt; 4) the distribution of software resource blocks among the processors for each of the processes is executed in cycles according to the rule: block number $j = kp + i$, $j = \overline{1,s}$, $i = \overline{1,p}$, $k \geq 0$, is distributed to processor number *i*.

In addition, let's introduce some further conditions, which define the modes of interaction of processes, processors and blocks: 5) there's no downtime in the processors on conditions that the blocks are ready as well as non–fulfilment of the blocks when processors are available; 6) for each of *n* processes the terminal time of software resource block at the competitive processor coincides with the beginning of execution of the next software resource block at the next competitive processor, $i = \overline{1, p-1}$, $j = \overline{1, s-1}$; 7) for each block the moment of completion of its execution by process *l* coincides with the beginning of its execution by process *l+1* at the same processor, $l = \overline{1, n+1}$.

Conditions 1–5 define an asynchronous mode of the interaction of processors, processes and blocks, which means the absence of time–out of processors on the condition that the block is ready as well as block failure if processors are available.

If we add condition 6 to conditions 1–4 we'll get the first synchronous operation which supplies continuous execution of blocks of a software resource inside each process.

The second synchronous operation defined by conditions1–4, 7 provides continuous execution of each block by all the processes.

2. Software execution time of distributed competitive processes. In [3–4] there have been explored basic asynchronous and synchronous operations which appear when distributed processes in terms of competitive environment for common software resources are organized. Within the bounds of these operations there've been received mathematical relations for computing the meaning of minimum general time of execution of heterogeneous distributed competitive processes in case of limited $(s > p)$ and unlimited $(s \le p)$ parallelism according to the amount of processors of multiprocessor.

Let’s consider a homogeneous system of distributed competitive processes. Let t_1^τ, t_2^τ, …, t_s^τ of execution of each block Q_j, $j = \overline{1,s}$, by a SR with an allowance for the parameter $\tau > 0$. If $s \le p$ for computing minimum general time in the asynchronous operation $T_{dh}^{as}(p,n,s,\tau)$ and in the first synchronous operation $T_{dh}^{1}(p,n,s,\tau)$ we’ll have: $T_{dh}^{as,1}(p,n,s,\tau) = \sum_{j=1}^{s} t_j^\tau + (n-1)\max_{1\le j\le s} t_j^\tau$.

Let's consider the case when $s = kp$, $k > 1$, and introduce the following designations (symbols): $t_j^{\tau,l} = t_{(l-1)p+j}^{\tau} + \tau$ – the time of execution of software resource block of group *l* of all *n* processes, $j = \overline{1,p}$, $l = \overline{1,k}$; $T_l = \sum_{j=1}^{p} t_j^{\tau,l} + (n-1)\max_{1\le j\le p} t_j^{\tau,l}$ – the total time of execution of the *l*–group of blocks by all *n* processes at *t* processors, $l = \overline{1,k}$; $E_l^j = \sum_{w=1}^{j} t_w^{\tau,l} + (n-1)\max_{1\le w\le p} t_w^{\tau,l}$ – is the time of finishing execution of $\left[(l-1)p + j\right]$ – software resource block by all *n* processes at *j* processor, $j = \overline{1,p}$, $l = \overline{1,k}$.

The total time of execution of *n* competitive distributed homogeneous processes if $s = kp$, $k > 1$, is defined as a sum of constituents of Gantt chart, with an allowance for maximum permissible overlap along time axis. i.e.

$$T_{dh}^{as,1}(p,n,s = kp,\tau) = \sum_{l=1}^{k} T_l - \sum_{l=1}^{k-1} \min\{\phi_l', \phi_l''\}.$$

Here ϕ_l' – a piece of possible overlap along time axis which is the difference between the beginning of execution of software resource block by the first process for the (*l+1*) group of blocks and the end of execution of software resource block, by the last process for the *l*–group of blocks, and ϕ_l'' is the difference between the beginning of execution of the first block by the *i*–process for (*l+1*) group of blocks and the end of execution of *p* block by *i*–process for the *l* group of blocks, which are computed by the formula:

$$\phi_l' = \min_{1\le j\le p}\left[T_l + \sum_{w=1}^{j-1} t_w^{\tau,l+1} - E_l^j\right] = \min_{1\le j\le p}\left[\sum_{w=j+1}^{p} t_w^{\tau,l} + \sum_{w=1}^{j-1} t_w^{\tau,l+1}\right],$$

$$\phi_l'' = (n-1)\min\left[\max_{1\le j\le p} t_j^{\tau,l}, \max_{1\le j\le p} t_j^{\tau,l+1}\right],\ l=\overline{1,k-1}.$$

If $s = kp + r$, $k \ge 1$, $1 \le r < p$, the minimum total time, in the modes considered, is defined by the formula:

$$T_{dh}^{as,1}(p,n,s=kp+r,\tau) = \sum_{l=1}^{k} T_l + T_{k+1} - \sum_{l=1}^{k-1}\min\{\phi_l', \phi_l''\} - \min\{\phi_k', \phi_k''\},$$

where $T_{k+1} = \sum_{j=1}^{r} t_j^{\tau,k+1} + (n-1)\max_{1\le j\le r} t_j^{\tau,k+1}$ – is the time of execution of (*k+1*) group of *r* blocks by *n* processes, $\phi_l' = \min_{1\le j\le r}\left[\sum_{w=j+1}^{p} t_w^{\tau,k} + \sum_{w=1}^{j-1} t_w^{\tau,k+1}\right]$ – is the difference between the beginning of execution of *j* block by the first processor for the (*k+1*) group of blocks and the moment of completion of execution of *p* block by *i* process for the *k* group of blocks, $\phi_k'' = (n-1)\min\left[\max_{1\le j\le p} t_j^{\tau,l}, \max_{1\le j\le r} t_j^{\tau,l+1}\right]$ – difference between the beginning of performance of the first block *i* process for *(k+1)* groups of blocks and the moment of completion of *p* block *i* process for *k* groups of blocks.

If the interaction of processes, processors and blocks is exercised in the second synchronous operation when, for each block of the structured software resource the moment of completion of its execution for *i* process, coincides with the beginning of its execution for the (*i+1*) process at the same processor, $i = \overline{1,n-1}$, then the minimum total time $T_{dh}^{2}(p,n,s,\tau)$ of execution of *n* homogeneous processes at *p* processors is defined by the following formulae:

$$T_{dh}^{2}(p,n,s,\tau) = \sum_{j=1}^{s} t_j^{\tau} + (n-1)\left[t_s^{\tau} + \sum_{j=2}^{s}\max\{t_{j-1}^{\tau} - t_j^{\tau}, 0\}\right],\ s \le p,$$

$$T_{dh}^{2}(p,n,s,\tau)\le$$

$$\le\begin{cases}\sum_{l=1}^{k}T_l-\sum_{l=1}^{k-1}\min\{\psi_l^{'},\psi_l^{''}\},\ s=kp,\ k>1,\\ \sum_{l=1}^{k}T_l+T_{k+1}-\sum_{l=1}^{k-1}\min\{\psi_l^{'},\psi_l^{''}\}-\min\{\psi_k^{'},\psi_k^{''}\},s=kp+r,\ k\ge1,\ 1\le r<p.\end{cases}$$

The meanings T_l, $\psi_l^{'}$, $\psi_l^{''}$, T_{k+1}, $\psi_k^{'}$, $\psi_k^{''}$ are computed according to the formulae: $T_l=\sum_{j=1}^{p}t_j^{\tau,l}+(n-1)\left[t_p^{\tau,l}+\sum_{j=2}^{p}\max\{t_{j-1}^{\tau,l}-t_j^{\tau,l},0\}\right]$ is the total time of execution of the *l p* software resource blocks by all *n* processes at *p* processors, $l=\overline{1,k}$; $\psi_l^{'}$ and $\psi_l^{''}$ are the pieces of possible overlap of two consecutive charts along time axis: $\psi_l^{'}=\min_{1\le j\le p}\{T_l+E_j^{l+1}-nt_j^{\tau,l+1}-E_j^l\}$, $\psi_l^{''}=(n-1)\min\{t_1^{\tau,l+1},t_p^l\}$, $l=\overline{1,k-1}$; $E_j^l=\sum_{w=1}^{j}t_w^{\tau,l}+(n-1)\left[t_j^{\tau,l}+\sum_{w=2}^{j}\max\{_{w-1}^{\tau,l}-t_w^{\tau,l},0\}\right]$, $j=\overline{1,p}$, $l=\overline{1,k}$, is the time of completion of execution $\left[(l-1)p+j\right]$ block by all *n* processes at the *j* processor; $T_{k+1}=\sum_{j=1}^{r}t_j^{\tau,k+1}+(n-1)\left[t_r^{\tau,k+1}+\sum_{j=2}^{r}\max\{t_{j-1}^{\tau,k+1}-t_j^{\tau,k+1},0\}\right]$ is the time of execution of (*k+1*) *r* blocks for all *n* processes; $\min\{\psi_k^{'},\psi_k^{''}\}$ is size of maximum overlap along time axis of *k* and $(k+1)$ charts:

$$\psi_k^{'}=\min_{1\le j\le r}\{T_k+E_j^{k+1}-nt_j^{\tau,k+1}-E_j^k\},\ \psi_k^{''}=(n-1)\min\{t_1^{\tau,k+1},t_p^{\tau,k}\}.$$

3. Mode organization analysis of distributed competitive processes. The problem of comparative analysis of ratio for defining minimum total time of execution of great number of distributed competitive processes is of definite theoretical and practical interest. Let's analyze homogeneous system with an allowance for additional systems costs $\tau>0$.

Let's consider a homogeneous system of distributed competitive processes with the time execution of blocks of a structured software process $t_1^\tau,\ t_2^\tau,\ ...,\ t_s^\tau$. Let $T_p^\tau=\sum_{j=1}^{s}t_j^\tau$ be the total time of execution of software resource by each process with an allowance for systems costs and a set of parameters $(t_1^\tau,\ t_2^\tau,\ ...,\ t_s^\tau,\ T_p^\tau)$ of this system will be characteristic.

Let $\beta = \left\{ (t_1^\tau,\ t_2^\tau,\ ...,\ t_n^\tau,\ T_p^\tau)\ \middle|\ T_d^\tau = \sum_{j=1}^{s} t_j^\tau,\ t_j^\tau = t_j + \tau > 0\ \ j = \overline{1,s} \right\}$ be a number of all legal characteristic processes. Let's highlight characteristic subset out of sets β:

$$H(T_d^\tau) = \{(t_1^\tau,\ t_2^\tau,\ ...,\ t_s^\tau,\ T_p^\tau) \in \beta \ \big|\ t_1^\tau \le t_2^\tau \le ... \le t_l^\tau \ge t_{l+1}^\tau \ge ... \ge t_s^\tau,\ l = \overline{1,s}\}\ .$$

Then for this subset the following theorem is fair theorem.

Theorem 1. *Let* $\delta \in H(T_d^\tau)$ *be a characteristic set of any homogeneous system with the parameters* $p,\ n,\ s \ge 2$ *and systems costs* $\tau > 0$. *Then in case of unlimited parallelism minimum total times* T_{dh}^{as}, T_{dh}^{1} *and* T_{dh}^{2} *of execution of a number of homogeneous distributed competitive processes in asynchronous and basic synchronous operations coincide.*

Proof. Let $t_l^\tau = \max\limits_{1 \le j \le s} t_j^\tau$. Then for both asynchronous and first synchronous operation with contiguous transition from one block to another for any characteristic legal set of homogeneous system including any characteristic set $\delta \in H(T_d^\tau)$ when $2 \le s \le p$, there occur equalities:

$$T_{dh}^{ac}(p,n,s,\tau) = T_{dh}^{1}(p,n,s,\tau) = T_d^\tau + (n-1)t_l^\tau\ ,$$

where $T_d^\tau = \sum\limits_{j=1}^{s} t_j^\tau$, $t_j^\tau = t_j + \tau$, $j = \overline{1,s}$.

Let the interaction of processes, processors and blocks be exercised in the second synchronous operation with a contiguous transition along the processes. In this mode for any characteristic set out of β if $2 \le s \le p$ the equality is performed:

$$T_{dh}^{2}(p,n,s,\tau) = \sum_{j=1}^{s} t_j^\tau + (n-1)\left[t_s^\tau + \sum_{j=2}^{s} \max\{t_{j-1}^\tau - t_j^\tau, 0\} \right]. \qquad (1)$$

Thus, for any characteristic set $\delta \in H\left(T_d^\tau\right)$ the equality $t_s^\tau + \sum\limits_{j=2}^{s} \max\{t_{j-1}^\tau - t_j^\tau, 0\} = t_l^\tau$ is performed, so the theorem is proved.

Taking into account that $t_l^\tau = \max\limits_{1 \le j \le s} t_j^\tau$, for all the numbers $j \le l$ there's an equality $\sum\limits_{j=2}^{l} \max\{t_{j-1}^\tau - t_j^\tau, 0\} = 0$, and for $j > l$ there's an equality $\sum\limits_{j=l+1}^{s} \max\{t_{j-1}^\tau - t_j^\tau, 0\} = t_l^\tau - t_s^\tau$.

Therefore, $t_s^\tau + \sum_{j=2}^{s} \max\{t_{j-1}^\tau - t_j^\tau, 0\} = t_s^\tau + t_1^\tau - t_s^\tau = t_1^\tau$ which was to be proved (Q.E.D. quod eras demonstrandum).

Theorem 2. *For any homogeneous distributed system with the parameters p, n, s and systems costs* $\tau > 0$*, a legal characteristic set of which is* $\delta \notin \mathrm{H}\left(\mathrm{T}_p^\tau\right)$ *if* $2 \le s \le p$*, the ratio is:*

$$\mathrm{T}_{dh}^{2}(p,n,s,\tau) > \mathrm{T}_{dh}^{as}(p,n,s,\tau) = \mathrm{T}_{dh}^{1}(p,n,s,\tau). \qquad (2)$$

Proof. The terms of theorem 2 equals to the inequality $t_s^\varepsilon + \sum_{j=2}^{s} \max\{t_{j-1}^\tau - t_j^\tau, 0\} - \max_{1 \le j \le s} t_j^\tau > 0$. The proof of the indicated above is carried out by blocks *s*, $s \ge 2$, induction.

If $s = 2$, a number of all legal characteristic sets of homogeneous systems of competitive processes $\beta = (t_1^\tau,\ t_2^\tau)$ will belong to class $\mathrm{H}(\mathrm{T}_d^\tau)$.

If $s = 3$, inequality validity (2) for $\delta \notin \mathrm{H}(\mathrm{T}_p^\tau)$ is easily defined by direct check–out.

Let, then, inequality (2) be performed if $s = i$, i.e. $t_i^\tau + \sum_{j=2}^{i} \max\{t_{j-1}^\tau - t_j^\tau, 0\} - \max_{1 \le j \le i} t_j^\tau > 0$. Let's show that it is valid if $s = i+1$.

Indeed, if $s = i+1$ we have:

$$t_{i+1}^\tau + \sum_{j=2}^{i+1} \max\{t_{j-1}^\tau - t_j^\tau, 0\} - \max_{1 \le j \le i+1} t_j^\tau =$$

$$= t_{i+1}^\tau + \sum_{j=2}^{i} \max\{t_{j-1}^\tau - t_j^\tau, 0\} + \max\{t_i^\tau - t_{i+1}^\tau, 0\} - \max_{1 \le j \le i+1} t_j^\tau .$$

Let's examine two cases.

1) The maximum meaning t_j^τ, $1 \le j \le i+1$, equals t_{i+1}^τ, then we have:

$$t_{i+1}^\tau + \sum_{j=2}^{i} \max\left\{t_{j-1}^\tau - t_j^\tau, 0\right\} + \max\left\{t_i^\tau - t_{i+1}^\tau, 0\right\} - t_{i+1}^\tau =$$

$$\sum_{j=2}^{i} \max\left\{t_{j-1}^\tau - t_j^\tau, 0\right\} + \max\left\{t_i^\tau - t_{i+1}^\tau, 0\right\} > 0 .$$

Here the second composed equally to zero, and the first composed is more than zero for otherwise $\delta \in \mathrm{H}(\mathrm{T}_p^\tau)$ that contradicts a condition of the theorem 2.

2) The meaning $\max_{1 \le j \le i+1} t_j^\tau$, $1 \le j \le i$, then we have:

$$t_{i+1}^{\tau}+\sum_{j=2}^{i}\max\{t_{j-1}^{\tau}-t_{j}^{\tau},0\}+\max\{t_{i}^{\tau}-t_{i+1}^{\tau},0\}-\max_{1\le j\le i+1}t_{j}^{\tau}=$$

$$=t_{i+1}^{\tau}-t_{i}^{\tau}+t_{i}^{\tau}+\sum_{j=2}^{i}\max\{t_{j-1}^{\tau}-t_{j}^{\tau},0\}-\max_{1\le j\le i+1}t_{j}^{\tau}+\max\{t_{i}^{\tau}-t_{i+1}^{\tau},0\}.$$

Here $t_{i}^{\tau}+\sum_{j=2}^{i}\max\{t_{j-1}^{\tau}-t_{j}^{\tau},0\}-\max_{1\le j\le i+1}t_{j}^{\tau}>0$ according to the induction hypothesis and because $\max_{1\le j\le i+1}t_{j}^{\tau}=\max_{1\le j\le i}t_{j}^{\tau}$. Let's show then, that $t_{i+1}^{\tau}-t_{i}^{\tau}+\max\{t_{i}^{\tau}-t_{i+1}^{\tau},0\}\ge 0$. Indeed for $t_{i}^{\tau}=t_{i+1}^{\tau}$ equality to zero is obvious.

If $t_{i}^{\tau}>t_{i+1}^{\tau}$ we'll get $t_{i+1}^{\tau}-t_{i}^{\tau}+\max\{t_{i}^{\tau}-t_{i+1}^{\tau},0\}=t_{i+1}^{\tau}-t_{i}^{\tau}+t_{i}^{\tau}-t_{i+1}^{\tau}=0$, and if $t_{i}^{\tau}<t_{i+1}^{\tau}$ we'll have $t_{i+1}^{\tau}-t_{i}^{\tau}+\max\{t_{i}^{\tau}-t_{i+1}^{\tau},0\}=t_{i+1}^{\tau}-t_{i}^{\tau}>0$, which was to be proved. (Q.E.D.)

4. The efficiency of the system of homogeneous competitive processes in terms of unlimited parallelism. Let's introduce the following definition which singles out in the class of homogeneous system of competitive processes a specific subclass of, the so–called, even system.

Definition 3. *Let a homogeneous distributed system of competitive processes call even, if* $t_{1}^{\tau}=t_{2}^{\tau}=\ldots=t_{s}^{\tau}=t^{\tau}$.

Theorem 1 proves that for homogeneous systems of competitive processes the minimum total time, with an allowance for systems costs $\tau>0$ for all the three basic modes indicated in point 1, if $s\le p$ formula evaluated:

$$T_{dh}^{as,1,2}(p,n,s,\tau)=T_{p}^{\tau}+(n-1)t_{max}^{\tau}, \tag{3}$$

where $T_{d}^{\tau}=\sum_{j=1}^{s}t_{j}^{\tau}$, $t_{j}^{\tau}=t_{j}+\tau$, $j=\overline{1,s}$, $t_{max}^{\tau}=\max_{1\le j\le s}t_{j}^{\tau}$.

In case of even homogeneous system of competitive processes, the minimum total time of their execution is defined by the equality:

$$\overline{T}(p,n,s,\tau)=(n+s-1)t^{\tau}, \tag{4}$$

where $t^{\tau}=T^{s}/s+\tau$, $T^{s}=st$.

Definition 4. *Let's a homogeneous system of distributed competitive processes name efficient if* $p,\ n\ge 2$, *nis fixed and the ratio* $\Delta_{\tau}(s)=nT^{s}-\overline{T}(p,n,s,\tau)\ge 0$ *is performed, where* nT^{s} *– is the time of execution of n processes in a contiguous mode, and* $T^{s}=\sum_{j=1}^{s}t_{j}$.

If we have two efficient homogeneous systems of competitive processes, let's suppose that the first one is more efficient than the second, if the quantity

$\Delta_\tau(s)$ of the first system isn't less than the second corresponding quantity. The following statement is legal for the introduced subset of homogeneous systems.

Theorem 3. *For any efficient homogeneous systems of competitive processes if* $s \le p$ *and* $\tau > 0$ *there exists a more efficient even homogeneous distributed system.*

Proof. Let's examine any efficient homogeneous distributed pipeline system. According to definition 4, the condition of its efficiency with an allowance for (3) is written down as the next inequality:

$$\Delta_\tau(s \le p) = (n-1)(T^s - t^s_{\max}) - (n+s-1)\tau \ge 0\,, \tag{5}$$

where $T^s = \sum_{j=1}^{s} t_j$, $t^s_{\max} = \max_{1\le j\le s} t_j$.

For any even homogeneous distributed system with an allowance for (4) we have, that

$$\overline{\Delta}_\tau(s \le p) = (n-1)(T^s - t) - (n+s-1)\tau \ge 0\,, \text{ where } \; t = T^s / s\,. \tag{6}$$

To be convinced of the theorem validity 3 it's enough to prove the inequality $\overline{\Delta}_\tau \ge \Delta_\tau$ for introduced efficient systems. Substituting the left right parts of the recent inequality (5), (6) for $\overline{\Delta}_\tau(s \le p)$ and $\Delta_\tau(s \le p)$ corresponding quantities and carrying out some simple transformations, we get unequal inequality $T^s - t^s_{\max} \le (s-1)t$.

Let's prove this inequality. Let's examine an even homogeneous distributed system, in which $t = \max_{1\le j\le s} t_j = t^s_{\max}$. Let $t^s_{\max} = t_l$, then a chain of relations

$$T^s - t^s_{\max} = \sum_{j=1}^{l-1} t_j + \sum_{j=l+1}^{s} t_j \le (s-1)t^s_{\max} = (s-1)t\,,$$

is true, which was to be proved.

The next statement establishes a sufficient condition of the efficiency of a homogeneous system in case of unlimited parallelism.

Theorem 4. *The homogeneous system of competitive processes with the parameters* $p,\ n,\ s,\ \tau$ *satisfying the relations* $3 \le s \le p$, $s = n \ne 3$, $ns \ge 2(n+s-1)$ *and* $0 < \tau \le \min_{1\le j\le s} t_j$ *is efficient.*

Proof. According to (5) the condition of efficiency is equal to the inequality

$$\frac{T^s - t^s_{\max}}{\tau} \ge \frac{n+s-1}{n-1}\,. \tag{7}$$

Therefore, to prove theorem 4 is enough to be convinced of inequality (7) validity. Direct checking out shows that the consequence of relations $0 < \tau \le \min\limits_{1 \le j \le s} t_j = t^s_{\min}$ is a chain of inequalities

$$\frac{T^s - t^s_{\max}}{\tau} \ge \frac{(s-1)t^s_{\min}}{\tau} \ge s-1, \tag{8}$$

because of the choice of τ the inequality $t^s_{\min} / \tau \ge 1$ is performed.

From $ns \ge 2(n+s-1)$ results the inequality validity

$$s-1 \ge \frac{n+s-1}{n-1}. \tag{9}$$

Verification shows that inequality (7) is the consequence of inequalities (8) and (9). Thus theorem 4 is proved.

The criterion of the existence of efficient homogeneous system of distributed competitive processes under sufficient number of processes against the burden rate τ is formulated and examined further.

Theorem 5. *For the existence of efficient structuring of a software resource under set–up parameters* $3 \le s \le p,\ T^s,\ \tau > 0$ *it's necessary and sufficient for the following conditions to be fulfilled:*

$$\tau \le \begin{cases} \phi(1+\sqrt{n}), & \text{if } \sqrt{n} \ - \ \text{int egral}, \\ \max\{\phi(1+[\sqrt{n}]),\ \phi(2+[\sqrt{n}])\}, & \text{if } \sqrt{n} \ - \ \text{non int egral}, \end{cases} \tag{10}$$

where $\phi(x) = \dfrac{(n-1)T^n(x-1)}{x(n+x-1)}$, $[x]$ *is the biggest integral, not bigger than x.*

Proof. According to (6) the condition of efficiency of any homogeneous distributed system of competitive processes is equal to the inequality

$$\varepsilon \le \frac{(n-1)T^s(s-1)}{s(n+s-1)}. \tag{11}$$

Let's examine the function $\phi(x) = \dfrac{(n-1)T^s(x-1)}{x(n+x-1)}$.

It isn't difficult to check that it is at its maximum at the point $x = 1+\sqrt{n}$ when *x>0*. Choosing as an efficient one structuring for *s* blocks, when $s = x = 1+\sqrt{n}$ if $\sqrt{n}$ is integral, or $s = x \in \{1+[\sqrt{n}], 2+[\sqrt{n}]\}$ if $\sqrt{n}$ is non–integral, we prove necessity.

Sufficiency follows from (11), as $\phi(x)$ is at its maximum when $x = 1+\sqrt{n}$, if $\sqrt{n}$ is integral or $x \in \{1+[\sqrt{n}], 2+[\sqrt{n}]\}$, $\sqrt{n}$ is non integral.

5. Optimality of homogeneous systems of competitive processes

Definition 5. An efficient equally distributed system is called optimal, if the quantity Δ_τ is at its maximum.

In 4 it's shown that an optimal homogeneous distributed system should be searched for among efficient homogeneous distributed systems. Moreover, according to theorem 3 an optimal homogeneous distributed system should be searched for among even homogeneous distributed systems. Then with an allowance for (6) we have: $\overline{\Delta}_\tau(s) = (n-1)T^s(1-1/s)-(n+s-1)\tau$.

Let's introduce the function of actual argument *x*:

$$\overline{\Delta}_\tau(x) = (n-1)T^s(1-\frac{1}{x})-(x+s-1)\tau,\ \ x \geq 1 .$$

Solving the problem of optimality of even structuring of software resource for *s* blocks for a sufficient number of processors, including all the three basic modes, results from the theorem.

Theorem 6. *So as an efficient structuring of software resource for s blocks, when* $s \leq p$ *, to be optimal, under given* $s \geq 2$ *,* T^s *,* $\tau > 0$ *, it's necessary and sufficient for it to be even and a number of blocks* s_0 *is equal to one of the figures* $\left[\left[\sqrt{\frac{(n-1)T^s}{\tau}}\right], \left[\sqrt{\frac{(n-1)T^s}{\tau}}\right]+1\right] \cap [2,p]$*, in which function* $\overline{\Delta}_\tau(x)$ *is at its maximum. Here* $[x]$ *means the biggest integral not more than x.*

Proof. *Necessity*. Let's examine the function:

$$\overline{\Delta}_\tau(x) = (n-1)T^s\left(1-\frac{1}{x}\right)-(x+s-1)\tau,\ \ x \geq 1.$$

According to definition 5 a homogeneous distributed system will be optimal at that point *x*. where the function $\overline{\Delta}_\tau(x)$ is at its maximum. The function $\overline{\Delta}_\tau(x)$ is at its maximum at the point $x = \sqrt{\frac{(n-1)T^s}{\tau}}$. Actually,

$$\overline{\Delta}'_\tau(x) = \frac{(n-1)T^s}{x^2} - \tau,\ \overline{\Delta}''_\tau(x) = -\frac{2T^s(n-1)}{x^3} < 0, \text{as } n \geq 2,\ \ x > 0.$$

Consequently the function $\overline{\Delta}_\tau(x)$ is at its maximum at the point, where its first derivative is transformed into naught $\overline{\Delta}'_\tau(x) = 0$, i.e. $x^* = \sqrt{\frac{(n-1)T^s}{\tau}}$.

Integer–valued points, in which the function $\overline{\Delta}_\tau(x)$ is at its maximum, will be $s_0 = [x^*]$ or $s_0 = [x^*]+1$. Consequently, it's possible to choose one of the figures $\left[\sqrt{\frac{(n-1)T^s}{\tau}}\right], \left[\sqrt{\frac{(n-1)T^s}{\tau}}\right]+1$, in which the function $\overline{\Delta}_\tau(x)$ is at its maximum, doesn't belong to $[2,p]$, then we choose $s_0 = p$ as an optimal structuring by the number of blocks.

Under negativeness of the second derivative, the function under study is convex. Consequently, a maximum point always exists, which means the existence of an efficient homogeneous distributed system of competitive processes in the case $s \to \infty$.

Sufficiency results from the convexity of the function $\overline{\Delta}_\tau(x)$ when $s \le p$ on the range $[2,p]$.

6. Conclusion. The received criteria of efficiency and optimality of software resources structuring can be used while designing system and applied software of multiprocessor and computer complex. The received formulas can also serve as a basis for solving problems of optimality of the number of processors under given computation and (or) directive terms of process realization.

REFERENCES

1. Voevodin V.V., Voevodin Vl.V. Parallel calculations. – St. Petersburg, 2002.
2. Toporkov V.V. Models of the distributed calculations. – Moskwa, 2004.
3. Ivannikov V.P., Kovalenko N.S., Metelski V.M. About the minimum time of realisation of the distributed competing processes in synchronous modes // Programming and Computer Software. – 2000. – N5. – P. 44–52.
4. Kapitonova Yu.V., Kovalenko N.S., Pavlov P.A. Optimality of systems of identically distributed competing processes // Cybernetics and Systems Analysis. – Springer, 2006. – P. 793–799.
5. Pavlov P.A. The analysis of modes of the organisation of equally distributed competing processes // Bulletin of the Belarus State University. – 2006. – N 1. – P. 116–120.