SC
scientific conferences

IX international scientific conference
Toronto. Canada
02-03.07.2024

Scientific conferences

QUESTIONS. HYPOTHESES. ANSWERS: SCIENCE XXI CENTURY

Proceedings of the IX international Scientific and Practical Conference

02-03 July 2024

TORONTO. CANADA

2024

UDC 001.1

BBC 1

IX International Scientific and Practical Conference «Questions. hypotheses. answers: science XXI century», July 02-03, 2024, Toronto. Canada. 101 p.

ISBN 978-91-65423-75-6

DOI https://doi.org/10.5281/zenodo.12723817

Publisher: «SC. Scientific conferences»

Main organization: artmedia24

Editor: Hans Muller

Layout: Ellen Schwimmer

The sample of the citation for publication is Gugnin Aleksandr, Lisnievska Yuliia ANTI-ADVERTISING IN THE HOTEL BUSINESS // IX International Scientific and Practical Conference «Questions. hypotheses. answers: science XXI century », July 02-03, 2024, Toronto. Canada. Pp.9-11, URL: https://sconferences.com

Contact information

Website: https://sconferences.com

E-mail: info@sconferences.com

Content

Philosophical sciences

Technical sciences

Mathematical sciences

UDC 004

ANALYSIS OF EXECUTION MODES HOMOGENEOUS DISTRIBUTED COMPETING PROCESSES WITH A LIMITED NUMBER OF COPIES OF THE SOFTWARE RESOURCE

P.A. Pavlov
PhD, Associate professor
Polessky state university, Republic of Belarus, Pinsk

Abstract

In the article, taking into account the limited number of copies of a structured software resource, a comparative analysis of mathematical relationships is carried out to calculate the total execution time of a set of homogeneous distributed competing processes in asynchronous and two synchronous modes in the case of unlimited and limited parallelism by the number of processors of a multiprocessor system.

Keywords: *distributed computing system, process, software resource, structuring, limited parallelism, unlimited parallelism, asynchronous mode, synchronous mode, heterogeneous system, homogeneous system, identically distributed system.*

1. Introduction

Increasing the performance of computing systems has always been and remains a pressing problem. But no computing system can compare in power to the total resources that are concentrated in local and global computer networks. The rapid development of information, communication and network technologies has led to the intensive use of geographically distributed computing resources, and the creation on their basis of dynamically scalable high-performance distributed computing systems (DCS).

There is no canonical definition in the literature of what a "distributed computing system" is. For example, Professor Andrew Stuart Tanenbaum defines a distributed system as "a collection of independent computers that appear to their users as a single unified system" [1]. The book [2] says that "by a distributed system we mean any computing system in which several computers or processors, one way or another, interact". We can also consider that a distributed system is a system whose components are located on different networked computers that exchange data and coordinate their actions by passing messages to each other [3]. A distributed system model can also be a set of software tools, which is a set of interconnected processes running on the same computing device [4].

From the above definitions, important specific features of RCS follow: the absence of shared memory and the need to exchange messages between software components for their interaction and synchronization; lack of a single time that characterizes the territorial distribution of system components; geographical remoteness of system components, which is a distinctive feature of distribution; autonomy and heterogeneity of system nodes allows the system to easily scale; the presence of parallel processes, often sharing system resources; the presence of a connecting or intermediate software layer (middleware) that combines computing resources into a single integrated system.

Today, there are various types of distributed computing systems - these are computing clusters, symmetric multiprocessors (SMP), distributed shared memory (DSM) systems, massively parallel systems (MPP) and multicomputers [5,6].

When creating a DCS, the tasks of constructing and studying mathematical models for organizing the interaction of processes competing for a software resource become especially relevant. In this regard, problems of distributed computing related to obtaining mathematical relationships, which can be both direct and inverse, are of interest. When setting direct problems, the conditions are the values of the parameters of the distributed computing system, and the solution is the minimum total time for implementing the given volumes of calculations. The formulation of inverse problems comes down to calculating the characteristics of distributed systems, searching for criteria for the efficiency and optimality of organizing the execution of many distributed competing interacting processes. In addition, the principles of distributed organization of processes are not only one of the universal ways to achieve *high performance and reliability of computing tools, but also have a fairly general nature and are inherent in processes of various natures, primarily they are characteristic of control systems, operating systems, computer-aided design systems, distributed energy systems, etc. [7,8]. When constructing and studying mathematical models and problems of optimal organization of distributed processes, the apparatus of graph theory, linear Gantt charts, scheduling theory, combinatorial optimization, matrix algebra, etc. are widely used. [9-11].*

The case when there is one copy of a software resource (PR) in the shared memory of a RCS was studied from various points of view in [12-17]. But, unfortunately, there are no works on mathematical modeling of the functioning of distributed systems in which there is not one, but several copies of a software resource in shared memory. Therefore, the study of problems related to the optimal organization of distributed parallel computing becomes particularly relevant in the case when a limited number of copies of a software resource can be simultaneously placed in the shared memory of a DCS. This generalization is of a fundamental nature in view of the fact that it reflects the main features of multi-pipeline processing, and also allows us to compare the effectiveness of pipeline and parallel processing.

In the article, taking into account the limited number of copies of a structured software resource, a comparative analysis of mathematical relationships is carried out to calculate the total execution time of a set of homogeneous distributed competing processes in asynchronous and two synchronous modes in the case of unlimited and limited parallelism by the number of processors of a multiprocessor system.

2. Mathematical model of distributed computing

Constructive elements for building mathematical models that implement distributed computing methods are the concepts of process and software resource. We will consider a process as a sequence of sets of commands (procedures, blocks, subroutines) $I_s = (1,\ 2,\ \dots,\ s)$. Processes that affect each other's behaviour by exchanging information are called cooperative processes. A programme or its part repeatedly executed in a multiprocessor system will be called a software resource, and a set of corresponding processes will be called competing processes.

By solving the problem of software resource allocation between processes, as a consequence, the problems of efficient use of the main computing resources - processors, RAM, I/O channels, etc. - are implicitly solved. From this point of view, a software resource is an integrated means of requesting computational resources. On the other hand, effectively solving the problem of software resource allocation we solve the problem of reducing the time of realisation of given volumes of calculations.

The mathematical model of a distributed processing system of competing cooperative processes under a limited number of copies of a software resource includes:

- $p \ge 2$, *processors of a multiprocessor system that have access to shared memory;*
- $n \ge 2$, *distributed competing processes;*
- $s \ge 2$, *blocks of a software resource structured into blocks;*
- *matrix* $T = [t_{ij}]$, $i = \overline{1,n}$, $j = \overline{1,s}$, *execution times of the software resource blocks by distributed cooperating competing processes;*
- $2 \le c \le p$, *the number of copies of the software resource structured into blocks that can be simultaneously located in the RAM available for all processors;*
- $\theta > 0$ *– is a parameter characterising the time of additional system costs associated with the organisation of the conveyor mode of using blocks of a structured software resource by a set of interacting competing processes during distributed processing.*

We will also assume that the number of processes is a multiple of the number of copies of the structured software resource, i. e. $n = mc$, *where* $m = n/c \ge 2$, *and that the interaction of processes, processors and software resource blocks is subject to the following conditions:*

1) *no processor can process more than one block at a time;*
2) *processes are executed in parallel-conveyor mode in groups, i.e. simultaneous (parallel) execution* c *of copies of each block in combination with pipelining of a group* c *of copies of the* Q_j *–th block by processes and processors, where* $j = \overline{1,s}$;
3) *processing of each block of the software resource is carried out without interruptions;*

4) in the case when the number of software resource blocks $s \le \left[\frac{p}{c}\right]$, *where* $[x]$ *is the integer part of the number, for each* i *-th process, where* $i = c(l-1)+q$, $l = \overline{1,m}$, $q = \overline{1,c}$, *the distribution of software resource blocks among processors is carried out according to the rule: the block with number* j *is distributed to the processor with number* $c(j-1)+q$.

Let us introduce the following basic modes of pipeline implementation of the interaction of processes, processors and blocks, taking into account the presence c *of copies of a structured software resource.*

The asynchronous mode of interaction between processes, processors and blocks of a structured software resource implies that the beginning of execution of a copy of the next block Q_j, $j = \overline{1,s}$, *is determined by the presence* c *of processors and the readiness of the block copy for execution, while a software block is considered ready for execution if it is not executed on any of the processors*

Fig. 1 is a linear Gantt chart illustrating the asynchronous mode of execution $n = 4$ *of distributed competing processes using* $c = 2$ *copies of a software resource structured into* $s = 3$ *blocks in a multiprocessor system with* $p = 7$ *processors and a given matrix of block execution times taking into account additional system costs* $T^\theta = [t_{ij}^\theta]_{4\times3} = [t_{ij} + \theta]_{4\times3}$.

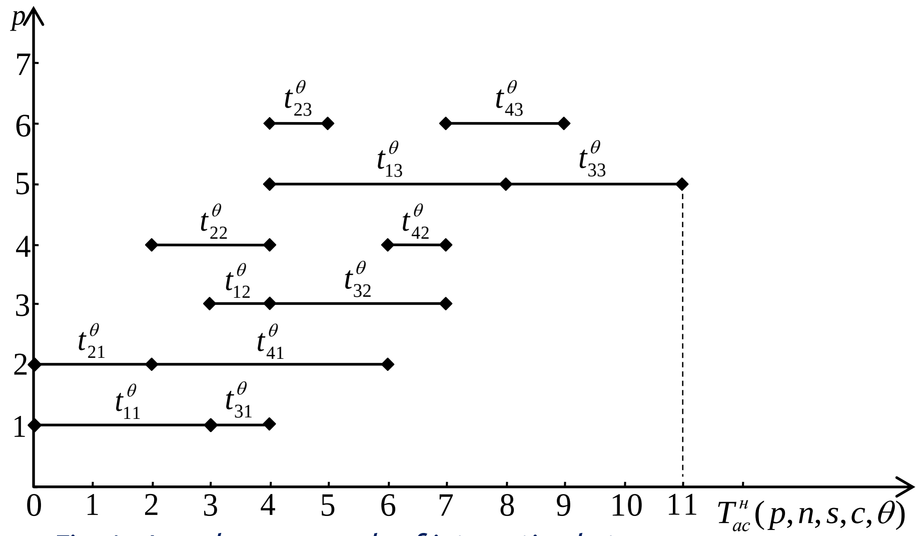

Fig. 1 Asynchronous mode of interaction between processes, processors and blocks of a software resource

The first synchronous mode ensures a linear order of execution of blocks of a software resource within each of the processes without delays, i.e. in the case when $2 \le s \le \left[\frac{p}{c}\right]$, *the moment of completion of the execution of the* Q_j *-th block,* $j = \overline{1,s-1}$, *by the process with number* $i = (l-1)c+q$, $l = \overline{1,m}$, $q = \overline{1,c}$, *on the* $((j-1)c+q)$ *-th processor, coincides with the moment of the start of execution of the next* Q_{j+1} *-th block on the processor with number* $(jc+q)$ *(Fig. 2).*

In the second synchronous mode, in the case when $2 \le s \le \left[\frac{p}{c}\right]$, *the moment of completion of execution by the* i *-th process, where* $i = (l-1)c+q$, $l = \overline{1,m-1}$, $q = \overline{1,c}$, *of the* j *-th block,* $j = \overline{1,s}$, *on the* $((j-1)c+q)$ *-th processor coincides with the moment of the start of execution of the* j *-th block by process number* $(i+c)$ *on the same processor, i.e. continuous execution of each block by all processes is ensured (Fig. 3).*

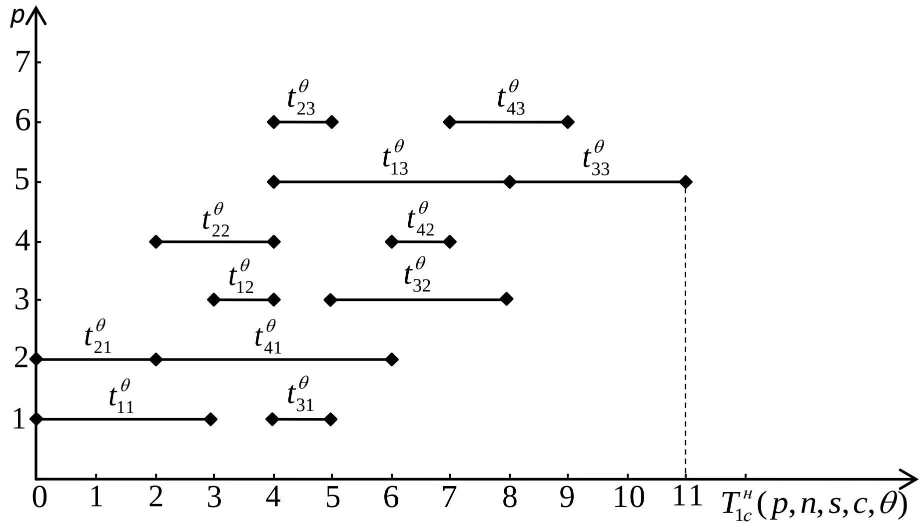

Fig. 2 First synchronous interaction mode processes, processors and blocks

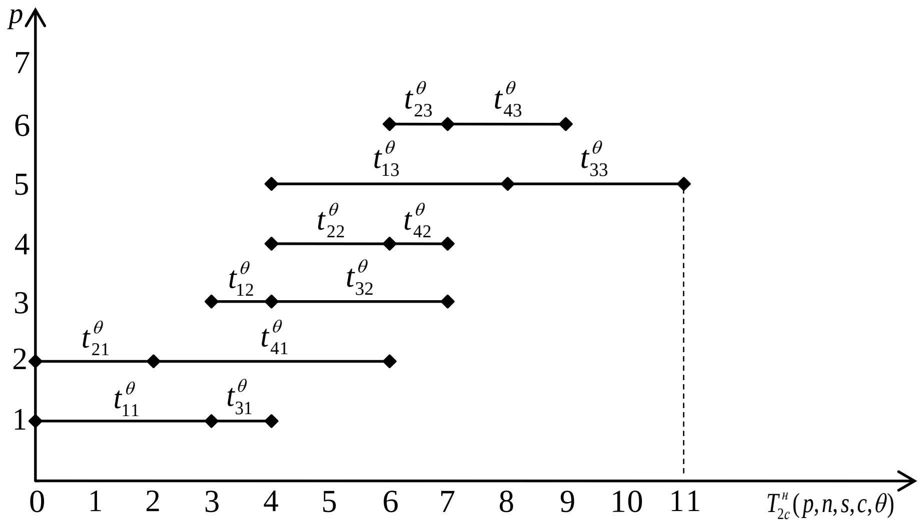

Fig. 3 Second synchronous interaction mode processes, processors and blocks

Definition 1. A system n of distributed competing processes is called heterogeneous if the execution times of software resource blocks Q_1, Q_2, ..., Q_s depend on the volumes of processed data and/or their structure, i. e. different for different processes.

Definition 2. We will call a system of distributed competing processes homogeneous if the execution times of the Q_j-th block by each of the i processes are equal, i. e. $t^{\theta}_{ij} = t^{\theta}_{j}$, $i = \overline{1,n}$, $j = \overline{1,s}$.

Definition 3. We will call a system of distributed competing processes identically distributed if the execution times of all blocks of a software resource by each of the processes coincide and are equal t^{θ}_{i}, i. e. $t^{\theta}_{i1} = t^{\theta}_{i2} = ... = t^{\theta}_{is} = t^{\theta}_{i}$, the chain of equalities is valid for all $i = \overline{1,n}$.

3. Анализ режимов выполнения однородных распределенных конкурирующих процессов

The works [18-20] studied in detail the basic asynchronous and two synchronous modes of interaction of distributed processes in conditions of competition for a limited number of copies of a software resource. Within the framework of the outlined modes, various mathematical relationships have been obtained to calculate the total execution time of a set of heterogeneous, homogeneous and identically distributed processes, taking into account the presence of a limited number of copies of a software resource. The problems of comparative analysis of the obtained relationships are of certain *theoretical and practical interest. Let us carry out such an analysis for a class of homogeneous systems, taking into account the parameter* $\theta > 0$ *characterizing the time of additional system costs associated with the organization of a conveyor mode for the use of blocks of a structured software resource by many competing processes during distributed processing.*

For a class of homogeneous distributed computing systems in the case of unlimited parallelism in the number of processors of a multiprocessor system, to calculate the minimum total execution time of n *processes on* p *processors, taking into account additional system costs* $\theta > 0$ *in asynchronous mode (Fig. 1) and the first synchronous mode (Fig. 2), which ensures linear order execution of software resource blocks within each process without delay, the following formula is obtained:*

$$T_{as}^{o}(p,n,s,c,\theta) = T_{1s}^{o}(p,n,s,c,\theta) = T_s^{\theta} + \left(\frac{n}{c} - 1\right)\max_{1 \le j \le s} t_j^{\theta},$$

where $T_s^{\theta} = \sum_{j=1}^{s} t_j^{\theta}$ *is the duration of execution of the software resource by each process, and* $t_j^{\theta} = t_j + \theta,\ j = \overline{1,s}$.

For the second synchronous mode, which ensures continuous execution of each block of a software resource by all processes, to calculate the execution time of homogeneous distributed n *processes competing for the use of* c *copies of a structured software resource, taking into account the parameter* $\theta > 0$, *the following formula applies:*

$$T_{2s}^{o}(p,n,s,c,\theta) = T_s^{\theta} + \left(\frac{n}{c} - 1\right)\left(t_s^{\theta} + \sum_{j=2}^{s} \max\{t_{j-1}^{\theta} - t_j^{\theta}, 0\}\right). \qquad (1)$$

Definition 1. We will call the set of parameters $(t_1^{\theta},\ t_2^{\theta},\ \ldots,\ t_s^{\theta},\ T_s^{\theta})$ *of a homogeneous distributed system of competing processes characteristic.*

Let $\alpha = \left\{(t_1^{\theta},\ t_2^{\theta},\ \ldots,\ t_s^{\theta},\ T_s^{\theta}) \ \middle|\ T_s^{\theta} = \sum_{j=1}^{s} t_j^{\theta},\ t_j^{\theta} = t_j + \theta,\ \ j = \overline{1,s}\right\}$ *be the set of all admissible characteristic sets of a system of homogeneous distributed competing processes. Let us select from the set a subset of characteristic sets of the form:*

$$S(\alpha) = \{(t_1^{\theta},\ t_2^{\theta},\ \ldots,\ t_s^{\theta},\ T_s^{\theta}) \in \alpha \ \big|\ t_1^{\theta} \le t_2^{\theta} \le \ldots \le t_l^{\theta} \ge t_{l+1}^{\theta} \ge \ldots \ge t_s^{\theta},\ \ l = \overline{1,s}\}.$$

Theorem 1. *Let* $\eta \in S(\alpha)$ *be the characteristic set of any homogeneous distributed system with parameters* $p \ge 2$, $n \ge 2$, $s \ge 2$, $2 \le c \le p$, $\theta > 0$. *Then in the case* $2 \le s \le \left[\frac{p}{c}\right]$ *of the minimum total execution times of a set of homogeneous distributed competing processes in asynchronous and synchronous modes coincide (Fig. 4).*

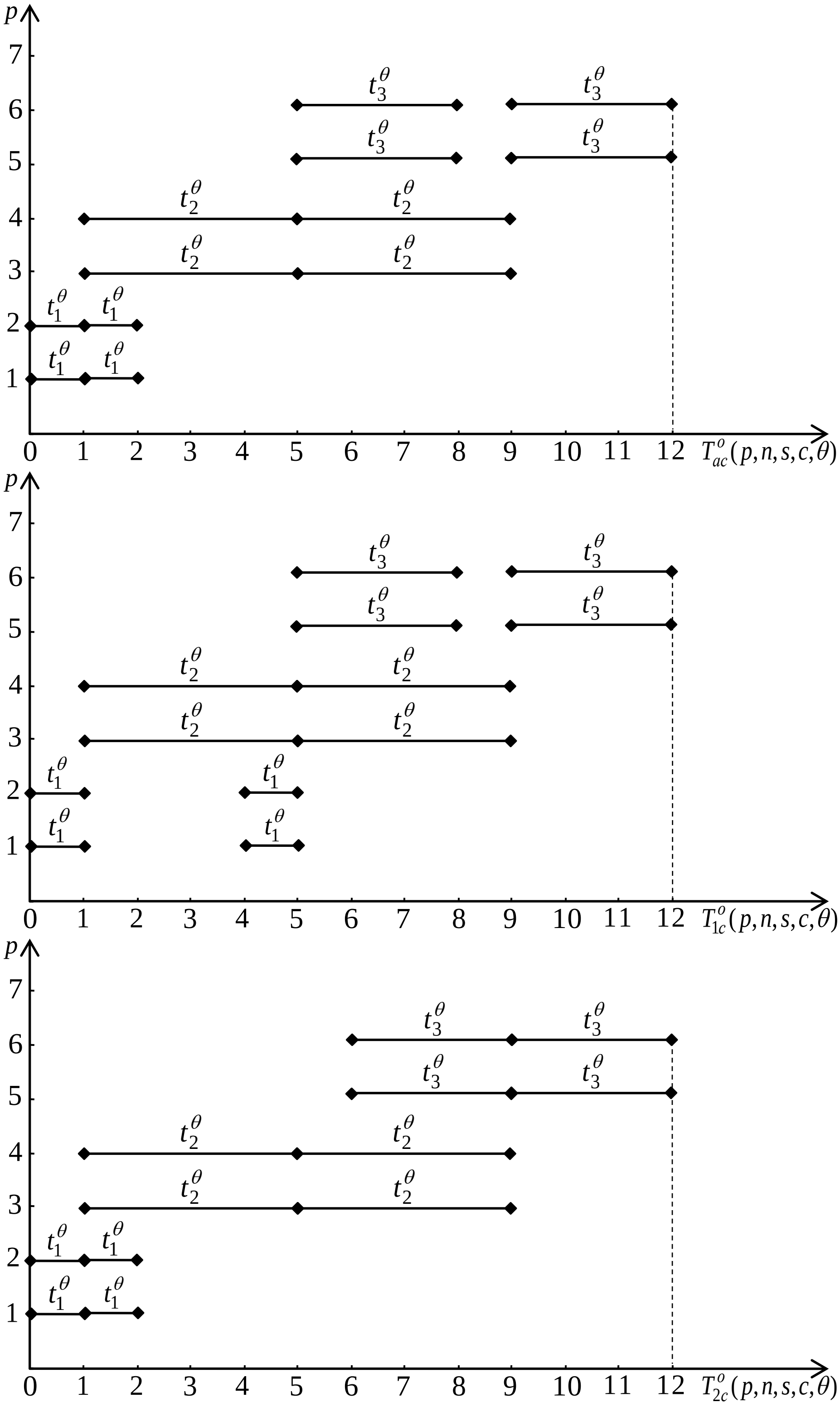

Fig. 4 Modes of execution of homogeneous distributed processes with
$T^o_{as}(p,n,s,c,\theta) = T^o_{1s}(p,n,s,c,\theta) = T^o_{2s}(p,n,s,c,\theta)$

Proof. *Let* $t_{\max}^{s+} = \max\limits_{1 \le j \le s} t_j^\theta$. *Then for the asynchronous and first synchronous modes for any characteristic set of a homogeneous distributed system, including for any characteristic* $\eta \in S(\alpha)$ *set* $2 \le s \le \left[\frac{p}{c}\right]$ *at we have:*

$$T_{as}^o(p,n,s,c,\theta) = T_{1s}^o(p,n,s,c,\theta) = T_s^\theta + \left(\frac{n}{c} - 1\right) t_{\max}^{s+}. \qquad (2)$$

If the interaction of processes, processors and blocks is carried out in the second synchronous mode, then for any characteristic set α *from the set at* $2 \le s \le \left[\frac{p}{c}\right]$, *formula (1) applies to calculate the minimum total execution time of* n *homogeneous processes on* p *processors, taking into account the parameter* $\theta > 0$.

Let us show that for any characteristic set $\eta \in S(\alpha)$ *equality holds* $t_s^\theta + \sum\limits_{j=2}^{s} \max\{t_{j-1}^\theta - t_j^\theta, 0\} = t_{\max}^{s+}$, *this will prove the theorem.*

Since $t_{\max}^{s+} = \max\limits_{1 \le j \le s} t_j^\theta$, *then for all numbers* $1 \le j \le l \le s$ *the equality holds* $\sum\limits_{j=2}^{l} \max\{t_{j-1}^\theta - t_j^\theta, 0\} = 0$, *and for* $1 \le l \le j \le s$ *we get that* $\sum\limits_{j=l+1}^{s} \max\{t_{j-1}^\theta - t_j^\theta, 0\} = t_l^\theta - t_s^\theta$.

Hence,

$$t_s^\theta + \sum_{j=2}^{s} \max\{t_{j-1}^\theta - t_j^\theta, 0\} = t_s^\theta + t_l^\theta - t_s^\theta = t_l^\theta = t_{\max}^{s+}.$$

The theorem is proven.

Theorem 2. *If the characteristic set of any homogeneous distributed system with parameters* $p \ge 2$, $n \ge 2$, $s \ge 2$, $2 \le c \le p$ *and system costs* $\theta > 0$ *does not belong to the subset* $S(\alpha)$, *then in the case of unlimited parallelism the following relations hold:*

$$T_{2c}^o(p,n,s,c,\theta) > T_{ac}^o(p,n,s,c,\theta) = T_{1c}^o(p,n,s,c,\theta) \text{ (Fig. 5).} \qquad (3)$$

Proof. *Taking into account formulas (1) and (2), the conditions of Theorem (3) are equivalent to the inequality* $t_s^\theta + \sum\limits_{j=2}^{s} \max\{t_{j-1}^\theta - t_j^\theta, 0\} - \max\limits_{1 \le j \le s} t_j^\theta > 0$. *We will prove the latter by induction on the number of blocks of a software resource.*

When $s = 2$ *the set of all admissible characteristic sets of a system of homogeneous distributed competing processes* $\alpha = (t_1^\theta,\ t_2^\theta)$ *will belong to the subset* $S(\alpha)$.

Let, further, inequality (3) be satisfied for $s = l$, *i. e.* $t_l^\theta + \sum\limits_{j=2}^{l} \max\{t_{j-1}^\theta - t_j^\theta, 0\} - \max\limits_{1 \le j \le l} t_j^\theta > 0$, *let us show that it is valid for* $s = l + 1$.

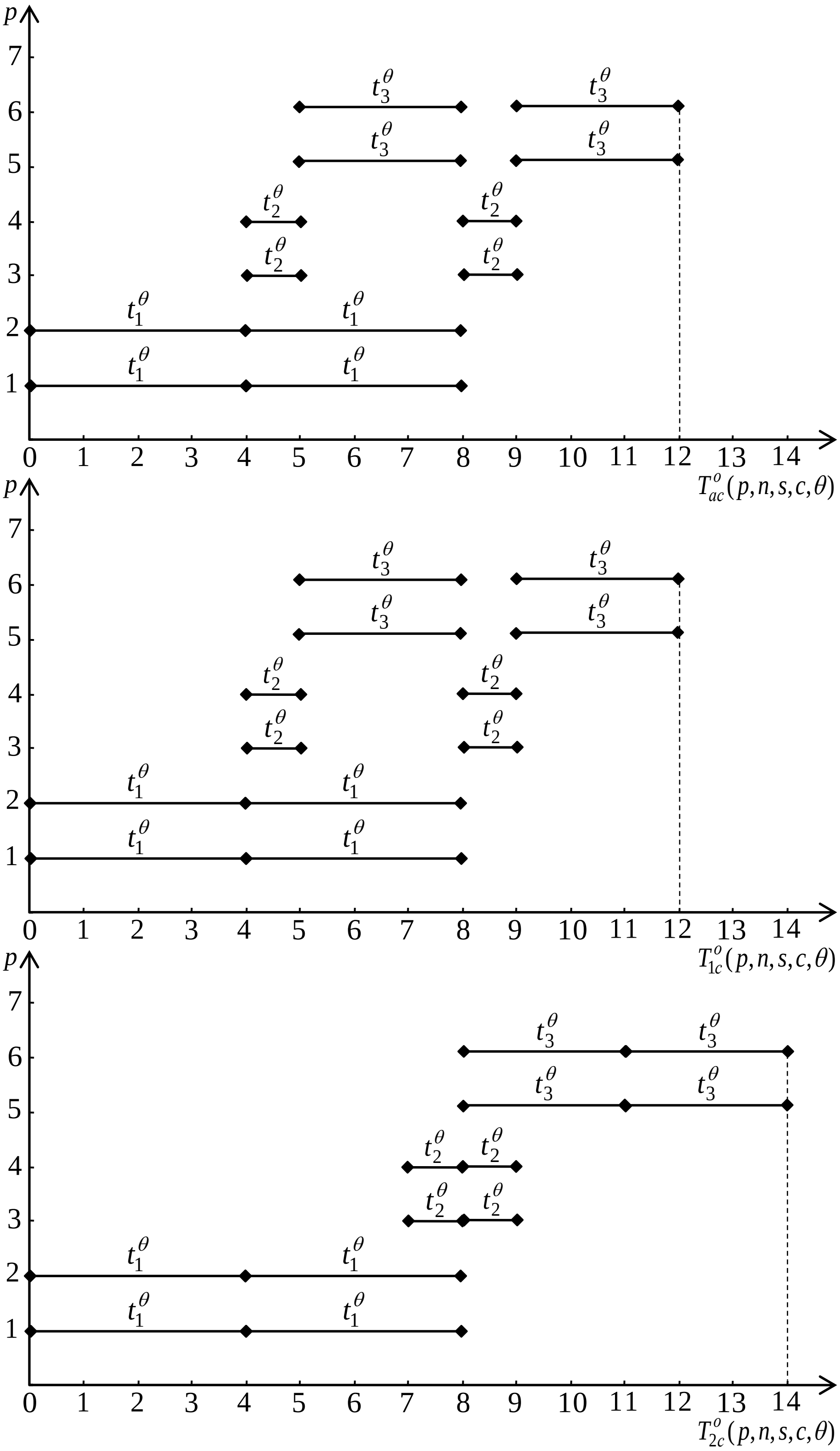

Fig. 4 Modes of execution of homogeneous distributed processes with
$T_{2s}^o(p,n,s,c,\theta) > T_{as}^o(p,n,s,c,\theta) = T_{1s}^o(p,n,s,c,\theta)$

When $s = l+1$ *we have:*

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

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

Let $\max_{1\le j\le l+1}t_{j}^{\theta}=t_{l+1}^{\theta}$ *then:*

$$t_{l+1}^{\theta}+\sum_{j=2}^{l}\max\{t_{j-1}^{\theta}-t_{j}^{\theta},0\}+\max\{t_{l}^{\theta}-t_{l+1}^{\theta},0\}-t_{l+1}^{\theta}=$$

$$=\sum_{j=2}^{l}\max\{t_{j-1}^{\theta}-t_{j}^{\theta},0\}+\max\{t_{l}^{\theta}-t_{l+1}^{\theta},0\}>0.$$

Here the second term is equal to zero, since $t_{l+1}^{\theta}\ge t_{l}^{\theta}$, *and the first term is greater than zero, because otherwise* $\eta\in S(\alpha)$, *which contradicts the conditions of the theorem.*

If the value $\max_{1\le j\le l+1}t_{j}^{\theta}$ *is in the interval* $1\le j\le l$, *then we have:*

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

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

Here, $t_{l}^{\theta}+\sum_{j=2}^{l}\max\{t_{j-1}^{\theta}-t_{j}^{\theta},0\}-\max_{1\le j\le l+1}t_{j}^{\theta}>0$ *by inductive hypothesis and due to the fact that* $\max_{1\le j\le l+1}t_{j}^{\theta}=\max_{1\le j\le l}t_{j}^{\theta}$.

We will show further that $t_{l+1}^{\theta}-t_{l}^{\theta}+\max\{t_{l}^{\theta}-t_{l+1}^{\theta},0\}\ge 0$.

When $t_{l}^{\theta}=t_{l+1}^{\theta}$ *equal to zero, it is obvious.*

If $t_{l}^{\theta}>t_{l+1}^{\theta}$ *we receive* $t_{l+1}^{\theta}-t_{l}^{\theta}+\max\{t_{l}^{\theta}-t_{l+1}^{\theta},0\}=t_{l+1}^{\theta}-t_{l}^{\theta}+t_{l}^{\theta}-t_{l+1}^{\theta}=0$.

If $t_{l}^{\theta}<t_{l+1}^{\theta}$ *we have* $t_{l+1}^{\theta}-t_{l}^{\theta}+\max\{t_{l}^{\theta}-t_{l+1}^{\theta},0\}=t_{l+1}^{\theta}-t_{l}^{\theta}>0$.

The theorem is proven.

4. Conclusion

The results obtained have numerous areas of application, in particular, they can be used in the design of system and application software aimed at multiprocessor distributed computing systems and complexes, as well as in solving problems of optimal use of computing resources.

References

1. Andrew Stuart Tanenbaum, Maarten Van Steen. Distributed Systems. Amazon Digital Services LLC, 2023. 684 p.
2. Robey R., Zamora Y. Parallel and High Performance Computing. Manning, 2021. 800 p.
3. Babichev S., Konkov K. Distributed Systems. Moscow, Juright, 2019, 507 p.

4. Kosyakov M. Introduction to Distributed Computing. SPb., NRU ITMO, 2014, 155 p.

5. Toporkov V., Emelyanov D. Models, methods and algorithms for planning in grid and cloud computing. Bulletin of the Moscow Energy Institute. 2018, №6. pp. 75-86.

6. Antonov A., Afanasyev I., Voevodin Vl.V. High-performance computing platforms: current status and development trends. Computational methods and programming. 2021, Volume 22. pp. 135-177.

7. Zaiets N., Shtepa V., Pavlov P., Elperin I., Hachkovska M. Development of a resource-process approach to increasing the efficiency of electrical equipment for food production. Eastern-European Journal of Enterprise Nechnologies. 2019, №8 (101). pp. 59-65.

8. Kaplun V., Pavlov P., Shtepa V., Prokopenya O. Resource-process model of energy management of a local facility with several energy sources. Bulletin of Brest State Technical University. 2019, №4 (117). pp. 86-91.

9. Emelichev V., Kovalev M., Kravtsov M. Polyhedra. Graphs. Optimization. Moscow, Nauka, 1981, 344 p.

10. Tanaev V., Sotskov Yu., Strusevich V. Scheduling theory. Multistage systems. Moscow, Nauka, 1989, 327 p.

11. Lazarev A. Scheduling theory. Methods and algorithms. Moscow, ICS RAS, 2019, 408 p.

12. Pavlov P. Organization of homogeneous competing processes in distributed pipeline processing. Control Sciences. 2010, №3. pp. 66-75.

13. Pavlov P. Optimal structuring of software resources during pipeline distributed processing. Software products and systems. 2010, №3. pp. 79-85.

14. Pavlov P. The problem of optimizing the number of processors in distributed processing. Bulletin of the State Samara Aerospace University named after Academician S.P. Koroleva. 2011, №4. pp. 230-240.

15. Pavlov P., Kovalenko N. Mathematical modeling of parallel processes. Lambert Academic Publishing, 2011. 246 p.

16. Pavlov P. The optimality of software resources structuring through the pipeline distributed processing of competitive cooperative processes. International Journal of Multimedia Technology (IJMT). 2012, Vol.2, №1. pp. 5-10.

17. Kovalenko N., Pavlov P. Optimal Grouping Algorithm of Identically Distributed Systems. Programming and Computer Software. 2012, №3. pp. 143-150.

18. Pavlov P., Kovalenko N. Distributed computing with a limited number of copies of a software resource. Software products and systems. 2011, №4. pp. 155-163.

19. Kovalenko N., Pavlov P., Ovseec M. Asynchronous distributed computations with a limited number of copies of a structured program resource. Cybernetics and systems analysis. 2012, №1. pp. 86-98.

20. Pavlov P., Kovalenko N. Synchronous mode of distributed computing with continuous execution of blocks of a limited number of copies of a software resource. Software products and systems. 2024, №1. pp. 43-53.