


UDC 004.272.45

 10.25209/2079-3316-2022-13-4-25-46



Synchronous execution of group operations in distributed supercomputer components and computer clusters

Gennady Georgievich **Stetsyura**

V. A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Russia

 gstetsura@mail.ru

(learn more about the author on p. 46)

Abstract. This paper proposes decentralized processes for synchronizing the actions of a distributed group of active components (objects) in supercomputers and computer clusters, allowing them to move to specified states or influence the external environment synchronously. The object action depends on the current state of the object and the external environment. The actions should start with the minimum delay after the possibility of their execution is detected. Synchronization is performed by exchanging optical signals over wireless communication channels through an optical signal repeater, combining one group of objects or sequences of groups of objects (layers). Accurate distance measurement performs the compensation of possible changes in distances between objects. Group operations accelerate synchronize and simultaneously receive data from a group of distributed objects. Data processing occurs during their transfer, without increasing the time. The operation time does not depend on the quantity of data processed by the operation. A group operation is performed in a repeater containing no computational means.

Key words and phrases: supercomputers, group operations, decentralized control, multilayer synchronization of object actions, distributed in-network computing

2020 *Mathematics Subject Classification:* 65Y05; 68Q10

For citation: Gennady G. Stetsyura. *Synchronous execution of group operations in distributed supercomputer components and computer clusters* // Program Systems: Theory and Applications, 2022, **13**:4(55), pp. 25–46. http://psta.psiras.ru/read/psta2022_4_25-46.pdf

© Stetsyura G. G.

2022



Эта статья по-русски:

http://psta.psiras.ru/read/psta2022_4_3-24.pdf

Introduction

The article deals with the following problem. There is a group of distributed components (objects) of a supercomputer (SC) or computing clusters. Interaction between objects is performed via wireless optical communication channels. Objects of the group unexpectedly detect the occurrence of an event which requires their joint transition to a new state as quickly as possible. Objects must inform the whole group as quickly as possible about the occurrence of this event and organize the simultaneous transition of the group to the required state.

The requirement for a fast response to an unpredictable event eliminates the often used but slow to implement transition time sign with a clock. Instead, the exchange of signals or messages only makes objects aware of the occurrence of an event that requires their coordinated action.

Transition operations must be performed in a decentralized manner. This is the requirement for obtaining a high transition speed, since the participation of the control center is accompanied by additional time consumption.

Transition comprises many operations. Synchronization of actions of objects, formation of commands by a group of objects, controlling actions of objects, exchange of information on a condition of objects, probably rearrangement of structure of communications between objects and a number of other operations are required. All or part of these operations can be performed or prepared in advance, before the required moment of state change.

The author did not get the specified results using known methods and the author developed new ways and operations that perform them. Some of them have been published earlier and adapted to the proposed problem, and I propose some of them in this article for the first time. The main ones are group operations and a decentralized method of synchronizing object actions that require organizing the structure of links between objects in order to execute the method given below.

A group operation is an elementary operation, which simultaneously receives and processes data from a group of objects of the supercomputer. Objects perform group operations when they receive one such operation — a group command, which is sent to all objects by one or more of them. The group operation transfers data by objects, without increasing this time. The processing time does not depend on the amount of data simultaneously processed by the operation.

The group operations in the article are performed mainly in the network's means that connects the objects, in its simple signal repeater, which does not contain any logical elements. The repeater also performs an important function central to the article. It replaces groups of arbitrarily in space sources and receivers of signals by one source and receiver. Only with this replacement did it become possible to perform group operations involving arbitrarily located relative to each other sources and receivers of messages.

This solution is characterized because here the network facilities, which do not contain computers, simultaneously carry out message transmission, control the transmission process and perform distributed computations. The lack of software processing has also increased the speed of group operations.

The paper discusses single-layer and multilayer group operations. In a single-layer operation, all receiver objects form a single group that receives a group command common to it. In a multilayer group operation, objects are divided into ordered layers.

In the first layer, the layer receivers simultaneously execute sources commands. In the second and subsequent layers, receivers of the previous layer become sources of data and commands for receivers of the next layer. In each such pair of layers, simultaneous execution of commands by receivers is ensured. Several multilayer operations can exist in the system simultaneously, composition of objects in layers and operations is formed in dynamics.

It was noted above that the acceleration of responses to emerging events is most important for operation in hard real time. The first complex computer systems, which the press refers to as supercomputers, appeared in the 1970s and were intended to solve exactly this tasks. These were the RADCAP associative systems, their successors, the STARAN system, and the later MPP. STARAN, for example, regulates aircraft traffic at Kennedy Airport in New York City. Now, in most publications, supercomputers are used in tasks that are less demanding in terms of speed of response to events. Recently, however, it has become possible to use compact supercomputers for such tasks. Practicall desktop supercomputers of Nvidia DGX series, with performance exceeding 1 pflops, can serve as an example.

This article focused primarily on such systems. In the article, the network facilities perform not only message transmission but also manage the use of the network and perform calculations in it.

But such combinations in the network of many functions have already been. In 1997, an article [1] proposed the use of network computer facilities for traffic management, then there appeared to suggest the use of such facilities for network configuration management [2], In-Network Computing direction appeared [3,4,5], in which several types of computational operations are also performed by network computers.

In contrast to these works, in order to increase the system's reactivity, this paper do not use computers in the network. They have been replaced by non-computing repeaters.

The solutions proposed below accurately determine the time interval required for the transmitted signal to travel to the distance between system components. Such solutions require fast and accurate ways to correct for possible changes in distances between system objects. Small changes in distances, such as those resulting from ambient temperature changes, must be considered for fast-acting computer systems. The basis for such measurements is an industrial network protocol PTP correcting the object clock readings [6–8] and a protocol created in Project White Rabbit for precision physics experiments [9, 10]. Currently developed a standard that combines these two methods [8]. The listed methods use a leading object (leader), which performs the necessary measurements. In PTP, the leader has an accurate clock and step by step synchronizes the clock readings of slave objects, measuring the distance of objects from the leader. In [9, 10], the error in time measurements lies in the sub picosecond range. These results are used unchanged in the article. But the leader replaced by arbitrary objects.

All solutions of this article are oriented to distances between SC objects of tens of meters or less. uLimiting the distances is necessary to get high reactivity of the distributed system. The results of the paper are distributed inside as follows. In section 1, the structure of objects performing group operations is discussed. In section 2, methods for synchronizing receivers of group commands are given. In section 3, synchronization of command sources is described. In section 4, peculiarities of group interaction of objects are considered and a brief description of group commands not included in the article is given. Sections describe a complete set of tools necessary for synchronization of objects' actions.

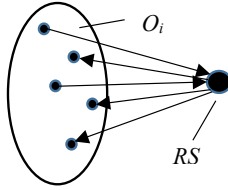


FIGURE 1. Simple network with a single repeater

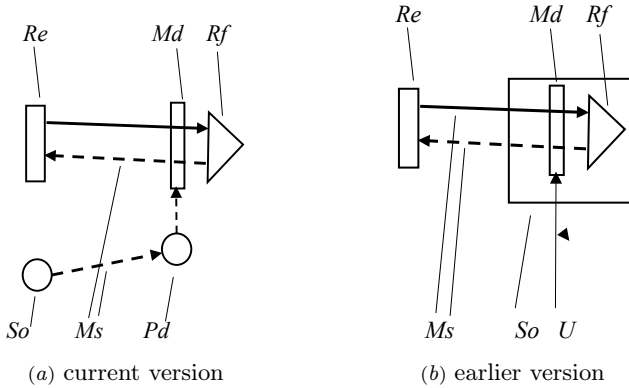


FIGURE 2. two repeaters containing a retroreflector

1. Means for group operations perform

1.1. Structure of connections between objects

The means of performing multilayer group operations in SC is developing the structure of performing single-layer group operations [11] proposed by the author earlier. Its main fragment is shown in Figure 1.

The figure shows objects of two types: objects O_i — (group So) and receivers of signals (group Re) and object RS — repeater of optical signals of O_i . To receive signals from RS objects Re , send continuous optical signals of frequencies $*f_1$ and $*f_0$. A source So sends to an RS signals f_1 and f_0 , carrying messages. An RS modulates with message signals the continuous signals returned to the receiver. Objects can contain a switch to select the necessary RS .

As shown in Figure 2a, the RS object is an optical retroreflector (Rf), which returns the signal coming onto the RS to its source. The RS contains photodetectors (Pd) of f_1 and f_0 signals and filters-modulators (Md) of $*f_1$ and $*f_0$ signals. The filters are typically closed, and the $*f_1$ and

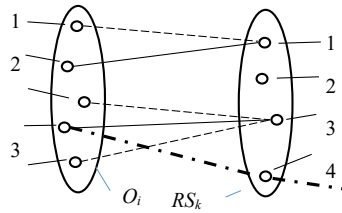


FIGURE 3. Network with Repeater Group

$*f_0$ signals entering the RS do not return to their sources. When a signal f_1 or f_0 arrives at the photodetectors Pd , the frequency-filter $*f_1$ or $*f_0$ opens, respectively. All receivers sending $*f_1$ and $*f_0$ signals will record the arrival of f_1 and f_0 signals in the RS from the source objects. These actions determine the scheme of message exchange signals Ms between objects O_i . This messaging structure elaborates the method developed earlier in [12] to receive data from the source at the expense of the energy of the data-requesting device. It is shown in Figure 2b

The receiver Re data sends a continuous optical signal to the data source retroreflector So . The latter returns the receiver their signal, modulating it with the contents of the requested data. Here there is no separate RS object. The source So has a modulator Md of incoming polling signals from Re . There are no photodetectors. The message signals (Ms) coming into So form the Re return signals in Md . This method was developed for marine applications. It is also proposed to use it for communication between satellites [13]. For communication systems, this is enough, but the SC requires additions Figure 2a and switching of links between many RS .

A particular synchronization of message exchange, discussed in other sections of this article, will also be required. The switching system using the node of Figure 1 is shown in Figure 3 Consider the switching process in the single-layer structure of Figure 3.

Here there are groups of objects O_i and RS_k . The value of k can start from 1 and exceed the number n of O_i objects. Each O_i must have the means to send to any, defined by the current requirements of the task being performed, object RS_k signals f_1 , f_0 , $*f_1$, $*f_0$, and receive from any RS_k signals $*f_1$, $*f_0$, modulated by signals f_1 , f_0 of other O_i .

Since only RS_k is the only intermediary between message source and receiver, which does not introduce delay into the transmission process, switching is straightforward. The source and receivers of the message are

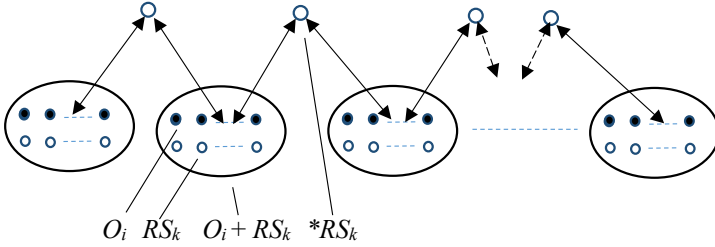


FIGURE 4. Multilayer network

connected in the above way to the RS_k known to them in advance, and a general message transmission to the entire group of receivers is performed. At the same time, using other RS_k will perform the transfer of different sources. In Figure 3, one of the RS_k group repeaters (repeater number 4) will be needed as $*RS_k$ to communicate between the groups of objects in Figure 4. In order to transmit the message simultaneously to all O_i , it is reasonable to introduce a broadcast RS , which sends the message received by it to all objects via a non-directional optical or radio channel.

The organization of the multilayer operation is illustrated in Figure 4. Here the O_i objects are divided into groups assigned to different layers. Each such group is assigned a group RS_k . Within this group, the repeater of the $*RS_k$ layer is distinguished. It differs from the rest RS_k layer only in the way it is used in layer interaction.

The object $*RS_k$ is accessed simultaneously by O_i of two layers — the current and the preceding. As indicated in the introduction, the O_i objects of the preceding layer work with $*RS_k$ as sources and the current layer's O_i as receivers. Switching links between objects within a layer needs to be done more often and faster than O_i links with $*RS_k$. Therefore, these types of switching can be performed by technical means that differ significantly in speed.

Above, the repeaters were outside the object O_i . The repeater can be used in the object with slight modifications. Such a repeater $**RS_k$, shown in Figure 5, consists of a receiver B_1 and a source B_3 of optical signals, which has a switch to select the external RS_k . The computer of the object interacts with devices B_1, B_2, B_3 via channels U_1, U_2, U_3 . It has an object-controlled switch B_2 that connects the receiver's output to the input of the optical signal source. When the key is open, it works as an additional conventional source/receiver pair of optical signals. However, the incoming optical signal will be transmitted to the desired external RS_k

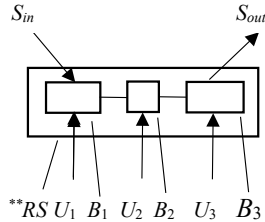


FIGURE 5. Repeater included in the object

without delay when the key is closed. The object can affect the relayed signal by participating in group operations.

The presence of $**RS_k$ makes it possible to transmit a message through a chain of objects with minimal delay and at the same time to perform a sequence of group operations. If the switch can keep a group of RS_k connections on simultaneously, sending one message to several objects is possible.

Remark. *The linkage structure of subsection 1.1, when the SC is included in a control system with mobile objects, prohibits the latter from leaving the field of signal reception by the retroreflector. This limitation is eliminated in the article [14].*

Let us note the basic capabilities of subsection 1.1 structures.

- (1) A group of objects may use a separate channel for each paired connection (switch mode) or a common channel for connecting sources to a single receiver (bus mode). Both types of structures may exist simultaneously. The structure of links is changed with high speed by sending a joint command to objects without reducing data processing speed.
- (2) Between sources and receivers, there is a single intermediary RS , which does not reduce the speed of interaction between objects, and allows accurate synchronization of transmitted data when they come to the RS . Alignment of arrivals to RS of the same name bits of messages of a group of sources serves as a basis for all group control, and computational operations applied further in the paper.
- (3) The structure allows replacing the failed communication channel quickly. It is enough to have one redundant RS to replace any failed RS . In addition, as a rule, channels have a reserve of bandwidth, and the load of the failed channel may be redistributed between them.

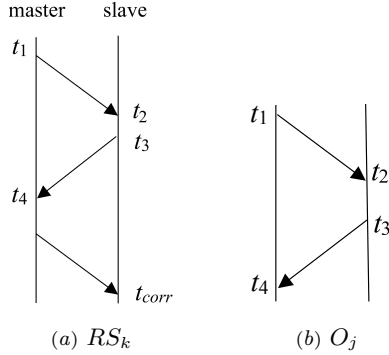


FIGURE 6. Time measurement in PTP and a variant of its use in the article

1.2. Means of measuring distances between objects

Here we summarize the basic principles of measuring the time intervals required for an optical signal to cover the distance between two devices in the PTP method given in the introduction. Two objects, master and slave, interact, as shown in Figure 6a.

Master sends to slave in moment t_1 the signal of the beginning of synchronization and its clock. This information at the moment t_2 comes to slave. The slave corrects their clock on the master clock. At the moment t_3 , the slave sends to master a response signal and his clock. Master at moment t_4 receives this reply, determines the time of signal transfer between master and slave $\tau = (t_4 - t_3)/2$ and reports the value τ to slave the corrected clock reading. The slave corrects the clock. Without several details, this is the basis of time correction in PTP.

The version of measurement used in the article without using the clock is shown in Figure 6b. The signal transfer time between the object and the RS signal repeater is measured. For optical signals, a passive retroreflector can be used as RS . An arbitrary object O_i has a timer, turns it on, and at time t_1 sends a signal to RS . The signal arrives at RS at time t_2 . With the delay of RS at time t_3 , the signal will be sent to RS , and at time t_4 object O_i determines the signal transfer time between O_i and RS as $T_{O_i,RS} = ((t_4 - t_1) - (t_3 - t_2))/2$. The value of $T_{O_i,RS}$ is small and easily controlled.

Protocol WR has achieved a more accurate way of measuring time intervals. Simple means achieved accuracy in the femtosecond range [10]. Protocol uses a digital version of the beat method.

The protocol uses a digital version of the beat method. The master has a generator of digital signals sent to the slave with frequency f and a generator of auxiliary digital signals with frequency $*f$. The ratio $nf = (n + 1)*f$ is observed. The master generates a beat U signal with a frequency $f - *f$. Increasing n decreases the frequency U of the signal. The signal returned from the slave object also has a frequency f . Similarly, the master object generates a signal $*U$ and from the time shift between these signals determines the timing of the frequency f signal between it and the slave object. This more time-consuming procedure allowed to obtain high accuracy measurements on simple equipment [8–10]. As a result, the PTP (HA) more accurately corrects the clock readings.

In the article, this method is most helpful in synchronizing objects performing high-speed cooperative distributed computations. Objects measure distances one by one in PTP and WR, so the repeater RS should not change the input signals.

2. Synchronization of actions of command receivers

This section describes the synchronization of distributed objects of the SC to perform required actions simultaneously or with additional time delays that can be set for each object. In subsection 2.1, a single-layer synchronization of command receivers sent by a single source is considered. In subsection 2.2, we present multilayer synchronization with a single command source.

2.1. Single-layer synchronization of command receivers

Let there be a group of command sources of which only one, arbitrarily located O_i , sends a command to object receivers via RS_k . Let the command receivers know their distance from RS_k and the signal transfer time between these objects. In order for the group to act simultaneously, each member of the group, upon receiving the command, must delay their action in order to equalize their distance from RS_k compared to the distances of other objects.

To compensate, object O_i performs a delay $O_i = T - T_i$ after receiving a command. Here T is the time interval, not less than the signal transfer time between RS_k and the farthest from RS_k receiver, and T_i is time interval of signal transfer between RS_k and O_i . It is easy to check; with this O_i , the actions of the receivers' commands will be executed simultaneously. If we add delay a_i to O_i for O_i , then a_i will shift the action of O_i relative to other receivers. This action completes the one-layer synchronization.

The execution of synchronous actions becomes considerably more complicated when there are changes in the dynamics of T_i times. There are two main reasons for such changes: changes in the distances between RS_k and O_i due to external influences, e.g., temperature changes. If the duration of signals transmitted between RS_k and O_i exceeds T , the changes can be neglected. Nevertheless, in the SC, when working with signals of picosecond duration, changes in T_i must be taken into account. Let us outline the necessary actions for this.

Since it will be necessary to have several independently acting groups of objects simultaneously, let us require that the correction of T_i changes does not require a particular central device (leader). All O_i groups will initiate the correction by performing the following actions.

The receiver sends RS_k a synchro signal S of duration not less than T if it has not detected before that moment the presence in RS_k of signal S sent by other sources. The superposition of signals S creates a joint signal S of variable duration. The moment of its termination $*S$ serves as a synchronization start signal. Let the receivers have ordinal numbers. The receiver with the lowest number measures the distance from RS_k . Then the other receivers measure it one by one. The next receiver starts the measurement after the moment of measurement completion by the previous receiver, known to the objects.

When all the objects in the group have been completed, the measurement process is repeated. It is reasonable to perform measurements following the solutions [6–10] given in the introduction. These solutions are based on the measurement of signal transfer time between two objects, which is used to synchronize the objects' clocks. From these solutions, only measuring the mutual distance of objects is taken.

After performing the time measurements, the synchronization is completed as described at the beginning of subsection 2.1.

Supplement. *Let us use signals with frequencies different from those of the message signals to measure distances. Also, assume that the modulator in the RS passes the measuring signals in any of its states. In this case, measurements are accelerated because all objects can measure distances simultaneously.*

This result follows the retroreflector property to return the signal passed through the directional channel to its source.

In subsection 2.1, a time interval T is allocated to each object to measure the distance from RS_k . A more complex measurement method,

which allows performing measurements for all receivers in time T , described in the paper [16], is possible.

Object in this section corrects the distance from RS only after all objects have completed the distance measurement. The new method for eliminating the accumulated error uses the channel allocated only for it. The action of the method is as follows. Let the objects first perform the above described alternate measurement of distances to the RS . After that, the objects send a signal S . Receipt by an object of the signal $*S$ serves as the beginning of accelerated synchronization. On receiving a signal $*S$ with a delay O_i , each object O_i transmits the scale to the RS .

The scale is a sequence of binary positions. Each bit of the scale is assigned to one of the O_i participating in the synchronization. The object O_i places in its scale position with the order number i the signal St . This signal has a shorter duration than the allocated for the transmission of the scale position. It is placed in the center of the discharge. The objects scales with the alignment of the same-numbered discharges go to RS and are returned to the objects. When the distances between O_i and RS change, the St signal shifts within the discharge, allowing all objects to account for the change in distance from RS during the scale transfer time.

Comment. *Let signals with frequencies different from the frequencies of message signals be used to measure distances. Let also an additional retroreflector without a modulator be used in RS (or the modulator always passes the measurement signals). In this case, measurements are accelerated, since all objects can measure distances at the same time.*

This result follows from the property of a retroreflector to return a signal that has passed through a directional channel to its source. An acceleration has been obtained, without which each object will need to allocate a time interval T to measure the distance to RS_k .

2.2. Multilayer synchronization the receivers of command

Let us divide the single group of receivers into several groups (layers), as shown in Figure 4.

Here, the group of objects is divided into layers (subgroups). Within a layer, the objects act as shown above. Signals exchanged between objects of a layer are not available in other layers. Repeaters $*RS_k$ of neighboring layers are available to objects of these layers and can be combined into a single device. The first layer of receivers receives source commands from its layer's repeater. Then one of the first layer receivers acts via the

repeater as a source for the receivers of the second layer. The receivers of the following layers will act similarly. In the simplest case, it is required that the layer's receivers act synchronously to external objects. Now, an external object is an object of the external environment or an SC object that is not included in a specific synchronization process. In a more complex case, computers of layer objects will perform joint actions, exchanging messages through their *RS*. This section will consider only actions between layer objects necessary to synchronize actions of layer receivers receiving commands from a single command source for a layer. For exchanging messages between objects within a layer and passing messages to the next layer from many sources of the previous layer will be used the results of section 3. Consider synchronization of actions of objects on external objects (process 1).

Process 1. There are k layers of objects, $k = 0, \dots, n$. The objects-receivers of layer k for interaction with receivers of layer $k + 1$ allocate a single source that synchronizes the receivers of layer $k + 1$. The process reaches the last layer, and its receivers affect the external objects simultaneously or with specified additional shifts in time. Objects of intermediate layers do not act on external objects, including objects of SC. The following process two is intended for their action.

Process 2. Let the time interval C be required for the completion of any layer. After receiving a command, receivers of arbitrary layer $k \leq n$ repeatedly calculate the value $F = n - k$, increasing k by 1. Each next 'calculation F is performed after time delay C . When $F = 0$, the layer receivers will perform an external action. Thus, the first calculation for layer $k = 0$ yields $F = n$, for layer $k = 1$ yields $F = n - 1$, and so on. As a result, when receiving a command in the last layer, the receivers in all layers receive $F = 0$ and simultaneously perform the required actions.

If objects are acting *asynchronously*, the best time of action execution is achieved by using a synchronization process consisting of two steps.

Step 1: Preparation for synchronization. In this step, the first layer sources prepare data asynchronously to transmit them to the first layer's receivers. All actions are performed using barrier synchronization (see section 3). Receivers of the first layer generally asynchronously prepare additional data to prepare actions as sources of the second layer. The transition to actions of the second layer is performed by a barrier synchronization signal informing the objects that they have all completed preparation for synchronization.

In this way, all layers are executed in turn. As long as all layers do not need to perform external actions simultaneously, they will be performed on the next step.

Step 2: Synchronization. The process is similar to process 2, but all layers are renumbered in reverse order. The last layer n now has the number 0. To synchronize, objects of the layer with new numbering $k = 0$ act as sources and start synchronizing receivers of all layers.

After receiving a command, the receivers of arbitrary layer $k \leq n$ repeatedly calculate the value $*F = n - k$, increasing k by 1. Each next 'calculation $*F$ is performed after time delay $*C$. When $F = 0$ is received in all layers, the receivers simultaneously perform the external action.

Synchronization is faster because objects here do not perform any action other than sending a command to the preceding layer.

Thus, without using a common control center, the receivers will synchronize their actions and execute them simultaneously in all layers of the considered multilayer object structure.

3. Synchronization of command sources actions

3.1. Synchronization of messaging processes

The synchronous actions of a group of sources are discussed in detail in [16], here we will give a brief and more appropriate than section 2 summaries of the synchronous actions of a group of sources.

The sources act in the following way, similar to receivers' actions in section 2. The sources So_j from source group So , ordered by j , send to RS a signal S . In response, So_j receives from RS a signal Srs and a signal $*Srs$ — a sign of completion of S . After that, So_j alternately determine the distance from RS and can calculate delays $D_j = 2(C - t_j) + a_j$, similar to O_i , C , T_i , and a_i from section 2.

Now, the sources can send synchronous messages (and commands to receivers) to RS without using a dedicated center.

The sources then use the LS logic scale, a sequence of binary positions equal to the number of sources in So . The source So_j , which must send a message to RS , puts in its respective bit j of LS scale one and transmitted to RS signal with a carrier frequency f_1 . The remaining bits of LS may contain no signals, or So_j contributes zero, which is transmitted by the signal f_0 .

The sources to start work with RS using the scales, send a signal S to RS and receive So_j and $*Srs$ in response. Then, using delays, the sources send their LS scales to RS to get an $*LS$ scale, which combines the same-named discharges of all scales received in RS . So_j can now send their messages to RS orderly without delaying on sources that have not requested a message transmission.

RS now acts as a single source, sending messages or commands from RS to receivers.

With asynchronous objects, there is no possibility to specify the time of occupation of the RS object. Need to apply barrier synchronization in the form [3]. One or more objects performing a joint operation at its execution sends a signal B to the repeater RSb available to all objects. When several objects work, and some object completes its work, this object stops transmitting signal B . Transmission of signal B stops when all participants complete the operation. Its absence allows other objects to start the next operation. If objects do not apply these actions, the value of C will have to choose unreasonably large.

Thus, objects must perform the following steps to transmit their messages. Objects decentralized send signals S to RS , receive from RS the signal $*Srs$, take turns to determine the distance from RS with use delays D_j , send to RS their scales LS , receive from RS a joint scale, and get the right to orderly order transfer messages.

3.2. Changing of sources message transmission orders

In subsection 3.1, the scale is a fixed construction in which the number of binary positions corresponds to the number of sources. Each bit of the scale is permanently assigned to a particular source, which sequence number coincides with the sequence number of the bit in the scale. However, in SC, it may be necessary to change this scale-fixed order as quickly as possible in solving a problem. Below is a possible variant of such a change.

First, each source requiring urgent transmission must form a service priority code. It consists of the source sequence number preceded by a group of binary priority digits. The binary number represented by this group evaluates the priority in serving the message. The higher this number, the higher the priority of service. The sources then send a notification command to start forming a new scale with a changed messaging order. After these actions, the sources synchronously perform

the following process 1¹ to determine the object with the current highest priority.

Process 1.

Step 1: The source transmits to *RS* the value of the highest binary digit of its priority code (the highest of the digits not transmitted in this process earlier). Value “1” is transmitted by a signal of frequency f_1 , value “0” — by a signal of frequency f_0 .

Step 2: If the source in step 1 sent signal f_0 and received signal f_1 from other sources, it completes process 1. The remaining sources proceed to step 3.

Step 3: The source checks if any binary number bits are not transmitted in step 1. If there are, the source returns to step 1. Otherwise, process 1 is complete.

This process separates a single source from the applicants for a change sequence of service. If several sources happen to have the same priorities, the order numbers of the sources will allocate only one of them. Next, process two is performed to form a priority-aware *RS* access scale.

Process 2.

- The process of forming a new scale of access to the *RS* begins. At the beginning of the scale, we introduce an additional priority zone consisting of one bit. Initially, the zone contains zero and is ignored by the sources. The source must enter bit one in the zone and enter bit zero in its scale bit for transmitting a message before all other sources. The Priority Transmission Request process is complete.
- The source in the zone now transmits the message, followed by the other sources who placed a one in their scale position. After the urgent message transfer is canceled, the source must remove the one in the zone.

It is essential that in section 2, receivers act as sources in several cases. Therefore, all of the above applies to them. Thus, an operative accounting of requirements in the change of rights of objects to transmit messages is obtained.

¹The initial variant of process 1 is the DPU method (decentralized priority control), in which the object with the highest current priority gets the right to transmit a message. For the wire bus, the DPU was developed in the IAT (later IPU) of the USSR Academy of Sciences in 1970 [17]. The DPU was used in industrial control systems. Its wireless version was taken from the article [16]. Later, a similar solution was proposed in Philips and is widely used to resolve service conflicts as an I2C interface.

A more flexible but more complex way of accounting source priorities is possible. In them, a zone contains several binary positions. Sources execute process 1 many times in the presence of zeros in the zone. At each step of process 1, the winner occupies the highest of the zone's zeros. The objects in the zone over the allowed time interval must be removed.

4. Features of group interaction of objects

This section highlights the main actions performed in the article by group operations and give some examples of other group operations extending distributed systems' capabilities. The following operations have been developed in the Institute of Control Problems of the Russian Academy of Sciences; some were used in industrial control systems.

4.1. Summary of group operations used in the article

The article uses group operations to control object actions. In this operation, the samenamed bits of objects messages arrive in RS simultaneously and are processed without delay and without using computational means. Operations perform the following actions.

1. Initial transfer of asynchronously working objects into synchronous state using signals S and $*S$ in subsection 2.1. The operation is performed with and without using RS (see below).
2. Formation of delays in message transmission by objects to RS for simultaneous delivery to RS .
3. Correction by receivers of times of use of messages received from RS at different times, depending on distances of receivers from RS . The correction allows receivers to perform their actions simultaneously or with prescribed additional time delays.
4. Variants of multilayer synchronization of objects. It is provided synchronous execution of actions by objects of the last layer, intermediate layers, iterative correction of actions of objects of previous layers taking into account the results obtained in the following layers.
5. Application of scales for accelerating RS distance measurement of objects; simultaneous resolution of RS access conflict for a group of objects; change of RS access order taking into account current object access priority.
6. Apply barrier synchronization to synchronize a group of objects, each of which performs a common task for the group, acting asynchronously.

4.2. Computational group operations performed on the repeater

The following operations include distributed computational operations, which speed up system control when searching for objects with a given set of properties and evaluating the system as a whole. These are the bitwise **AND/OR** operations, finding **max** and **min** numbers, and analog-digital arithmetic operations. Such operations are not used in this paper. However, they allow us to evaluate the state of all objects and the data in them for the time simultaneously that does not depend on the number of operation participants.

The bitwise **AND** and **OR** operations make it possible to quickly evaluate the state of all objects in the system. For this purpose, the object state is described by a scale – a sequence of binary positions. Each of them is equal to one in the presence of the corresponding attribute and equal to zero in its absence and transmitted accordingly by frequency signals f_1 and f_0 . Estimation of state of all objects is performed at the simultaneous transmission of scales in *RS* with combination in *RS* of identical positions of object sequences.

During the execution of **AND** operation, presence in *RS* of f_0 scale position means the absence of the corresponding feature in at least one object. Otherwise, the feature is present in all objects. For **OR** operation in the same conditions, the presence of f_1 means that at least one object has the feature. Otherwise, all objects do not have the corresponding feature.

A logical scale is used when determining the **max** of numbers stored in objects with digits represented in an arbitrary base calculus. All positions in the scale contain zero, except for the position corresponding to the digit value. The objects send numbers to *RS* to determine operation **max**. The numbers should arrive in *RS* synchronously with the coincidence of the samenumber scales positions.

In the first step, the objects transmit the highest digit of the number. The transmission of the next digit involved only the objects that have transmitted the largest of the digits before. The transmission of the next digit involves only the objects that have transmitted the largest of the digits before, and the process repeats. Remains the maximum of the transmitted numbers. With the replacement of ones by zeros, **min** is determined.

As an example, we use the decimal system. With the scales, the digits 7, 3, and 1 be written as 001000000, 000000100, 000000001. When combining the same-named positions of these scales in *RS*, all objects

will get the result of combined new scale 001000101. It follows that **max** equals 7 and **min** equals 1. Increasing the base of the number system for distributed systems is helpful because it reduces the number of transmissions over communication channels. The result will be obtained in time independent of the number of objects participating in the group operation. The use of active signals zero and one with frequencies f_1 and f_0 gives an additional acceleration. If all digits of the numbers in RS have not binary digits with signals f_1 and f_0 simultaneously, then the search is complete because all the maximal numbers of objects are the same.

Consider performing analogue-digital operations. RS must contain an analogue-digital converter (ADC) to perform analogue-to-digital operations. Consider the execution of the number addition operation. The energy of the optical signals coming to RS with time coincidence is measured by a photodetector, which transmits the measurement result to the ADC. The latter converts the result into a number and sends it to all objects.

As in the previous example, the numbers transmitted by the objects in RS are represented in decimal notation using scales. Let three objects transmit to RS the combination of the digits of the scales of three numbers 789, 988, 786, respectively. When scales are applied, these numbers will be written as

$$\begin{aligned} & [001000000; 010000000; 100000000], \\ & [100000000; 010000000; 010000000], \\ & [001000000; 010000000; 000100000]. \end{aligned}$$

When the scale digits are combined in RS , the objects will get three scales from RS

$$[102000000; 030000000; 110100000].$$

Here, the numbers 3 and 2 in the first and second scales show the ADC digital readout summing the energy of the three and two signals. As a result, each object, using its computing facility, locally performs the summation and obtains the result

$$(9 + 2 \times 7) \times 100 + 3 \times 8 \times 10 + 9 + 8 + 6 = 2300 + 240 + 23 = 2563.$$

Thus, calculating the sum does not depend on the number of objects involved in the addition operation. For subtraction, it is sufficient to get two sums in RS for the number to be reduced and the number to be subtracted and to complete the operation locally in each object.

In particular, addition can be used for counting. For example, when searching for **max** by the energy of the signals coming in, the number of identical **max** numbers in the objects is determined. To obtain in the SC

an accurate digital value of analogue-to-digital operations while summing up several thousands of signals need sources with a stable value of the energy of the optical signal. In [17], a simple LED source with output power stability better than 50 ppm/°C is given.

4.3. Three principal mechanisms of objects synchronization acceleration

It is clear from the previous sections that three mechanisms play a central role in synchronizing object actions. These operations include initial synchronization of asynchronous objects by S and $*S$ signals, replacement of a signal source group by an RS signal repeater, and logic scales representing zero and one by active signals.





Although in subsection 2.1, objects use RS at initial startup, its use is only necessary to obtain high rates of object interaction. For example, the objects may not use RS to communicate with the periphery. The objects need directly receive signals S and detect $*S$ without RS . The objects received $*S$ will alternately perform the required operations.

However, the operations will be performed significantly slower than with RS . After occurrence in the system of a signal $*S$, the first object will detect $*S$ and start message transmission no later than the moment of time T . The next object will start transmission no later than the moment of time T after detecting the transmission of the predecessor, et cetera. An unsupervised pause of duration $\leq T$ appears between messages.

References

- [1] D. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, G. J. Minden. “A survey of active network research”, *IEEE Communications Magazine*, **35**:1 (1997), pp. 80–86. [doi](#)↑₂₈
- [2] N. Zilberman, P. M. Watts, C. Rotsos, A. W. Moore. “Reconfigurable network systems and software defined networking”, *Proc. of the IEEE*, **103**:7 (2015), pp. 1102–1124. [doi](#)↑₂₈
- [3] Y. Tokusashi, Tu Dang H., F. Pedone, R. Soulé, N. Zilberman. “The case for in-network computing on demand”, *EuroSys '19: Proceedings of the Fourteenth EuroSys Conference 2019* (March 25–28, 2019, Dresden, Germany), ACM, New York, 2019, ISBN 978-1-4503-6281-8, 16 pp. [doi](#)↑₃₉

- [4] A. Sapio, I. Abdelaziz, A. Aldilajjan, M. Canini, P. Kalnis. “In-network computation is a dumb idea whose time has come”, *HotNets-XVI: Proceedings of the 16th ACM Workshop on Hot Topics in Networks* (30 November–1 December, 2017, Palo Alto, CA, USA), ACM, New York, 2017, ISBN 978-1-4503-5569-8, pp. 150–156. [doi](#)↑
- [5] D. Kim. *Towards elastic and resilient in-network computing*, CMU-CS-21-143, Carnegie Mellon University, Pittsburgh, 2021, 150 pp. [URL](#)↑
- [6] *IEEE standard for a precision clock synchronization protocol for networked measurement and control systems*, IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002), 2008, 269 pp. [doi](#)↑_{28, 35}
- [7] *IEEE standard for a precision clock synchronization protocol for networked measurement and control systems*, IEEE 1588-2019, IEEE Instrumentation and Measurement Society, 2020. [URL](#)↑_{28, 35}
- [8] F. Girela-López, J. López-Jiménez, M. Jiménez-López, R. Rodríguez, E. Ros, J. Díaz. “IEEE 1588 high accuracy default profile: Applications and challenges”, *IEEE Access*, **8** (2020), pp. 45211–45220. [doi](#)↑_{28, 34, 35}
- [9] Sliwczynski Ł., P. Krehlik, Buczek Ł., Schnatz H.. “Picoseconds-accurate fiber-optic time transfer with relative stabilization of lasers wavelengths”, *Journal of Lightwave Technology*, **38**:18 (2020), pp. 5056–5063. [doi](#)↑_{28, 34, 35}
- [10] P. Moreira, I. Darwazeh. *Digital femtosecond time difference circuit for CERN’s timing system*, 4 pp. [URL](#)↑_{28, 33, 34, 35}
- [11] G. Stetsyura. “Addition for supercomputer functionality”, *RuSC-Days 2016: Supercomputing*, Communications in Computer and Information Science, vol. **687**, eds. Voevodin V., Sobolev S., Springer, Cham, 2017, ISBN 978-3-319-55668-0, pp. 251–263. [doi](#)↑₂₉
- [12] W. S. Rabinovich, P. G. Goetz, R. Mahon, L. A. Swingen, J. L. Murphy, M. Ferraro, H. R. Burris, Ch. I. Moore, M. R. Suite, G. Ch. Gilbreath, S. C. Binari, D. J. Klotzkin. “45-Mbit/s cat’s-eye modulating retroreflectors”, *Optical Engineering*, **46**:10 (2007), 104001, 8 pp. [doi](#)↑₃₀
- [13] Y. Zhu, G. Wang. “Research on retro-reflecting modulation in space optical communication system”, IOP Conference Series Earth and Environmental Science, vol. **108**, 2018. [doi](#)↑₃₀

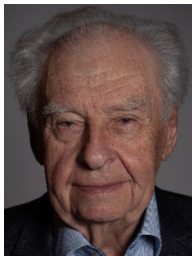
- [14] G. G. Stecyura. “Network information-computing support of automatic mobile objects interaction”, *Problemy upravleniya*, 2018, no. 5, pp. 56–65 (in Russian).  [doi](#)^{↑32}
- [15] Stecyura G. G.. “Decentralized autonomic synchronization of interaction processes of mobile objects”, *Problemy upravleniya*, 2020, no. 6, pp. 47–56 (in Russian).  [doi](#)[↑]
- [16] G. G. Stetsura. “Comments on “Peripheral interface standards for microprocessors””, *Proceedings of the IEEE*, **65**:11 (1977), pp. 1920.  [doi](#)^{↑36, 38, 40}
- [17] M. Bosiljevac, D. Babić, Z. Sipus. “Temperature-stable LED-based light source without temperature control”, Photonic Instrumentation Engineering III (15 March 2016, San Francisco, CA, USA), Proc. SPIE, vol. **9754**, 2016, 6 pp.  [doi](#)^{↑40, 44}

Received 01.07.2022;
approved after reviewing 18.09.2022;
accepted for publication 19.09.2022.

Recommended by

д.ф.-м.н. С. М. Абрамов

Information about the author:



Gennady Georgievich Stetsyura

Doctor of technical sciences, professor, chief researcher at IPU RAS. Area of interest: informatics, computing. The main works in the field of distributed multicomponent digital systems. More than 150 publications and more than 20 patents and copyright certificates for inventions.



0000-0003-4606-4424

e-mail: gstetsura@mail.ru

The author declare no conflicts of interests.