Means for Fast Performance of the Distributed Associative Operations in Supercomputers

Gennady Stetsyura✉

Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia
gstetsura@mail.ru

Abstract. This study proposes architectural solutions and operations for the rapid implementation of distributed associative operations in supercomputers. The operations are carried out by means of interactions between supercomputer devices using wireless optical links. Some operations result in improved distributed versions of the local operations of associative memory devices and associative processors. The operations for distributed fast digital calculations are also included. The operations for analog-digital counting are proposed for quick counting the number of records in distributed big data. The structure of the connections between devices can be completely changed in comparable time to the execution time of the processor command.

Keywords: Wireless Optical Network. Retroreflector. Dynamical Reconfiguration. Distributed Synchronization. Barrier Synchronization. Distributed Computing. Fault Tolerance.

1 Introduction

The distributed associative operations (*DAO*) refer to the operations of distributed search and data processing while analyzing data from many records included in supercomputer devices (objects). The associative operations (*AO*) are similar in function to *DAO* but act with the records located within the same device. The associative (or Content Addressable) memory devices (*AM*) and the associative parallel processors (*APP*) were created to quickly perform *AO* operations.

The *AM* device performs a parallel search in the base of records, which have keys equal to those in *AO*. The search also retrieves the records of keys-number values in a given interval, with a maximum value and so on. The *AM* counts the number of records found and resolves conflicts when multiple records meet the search criteria. The APP simultaneously separates the array of records into clusters, performs a limited set of logical and arithmetic operations.

The history of the use of associative operations in computers has many stages. A large number of studies on *AM* and *APP* were done in the ‘60s of the last century.
By the early ‘70s, these studies had led to the creation of several large computers, focused on the implementation of associative operations. Reviews of these areas of work are contained in several books [1-3]. However, ever-increasing demands on the processing of large amounts of data led to the fact that AO, as a rule, is now carried

out by programming. The programmed associative operations are more powerful than the operations in *AM* or *APP* but are much slower. Such operations are used in many algorithms, such as when using associative rules [4, 5].

The hardware implementation of the *DAO* in the supercomputers (*SC*) could accelerate the implementation of these algorithms, but this requires the active cooperation of distributed objects, the exchange of short messages, and distributed quick computations. Standard communications between *SC* devices are not effective at such actions, since they were designed for long message exchanges and do not support calculations directly in communication media. In the article, this disadvantage is eliminated, and objects quickly perform distributed associative operations.

There are opportunities for new types of relations between the SC objects used, as considered by the author at the conference "Supercomputer Days in Russia in 2016" [6, 7] and in [8]. These opportunities allow each facility to operate as a standalone device that performs the *AO* on its local data, but the *DAO* and distributed computations are carried out by means of communications between the objects.

2 Structure of the SC That Supports Distributed Associative Operations

The structure of the optical connections between the objects [6-8] will be used to perform the *DAO* with additions that the *DAO* require (Fig. 1).

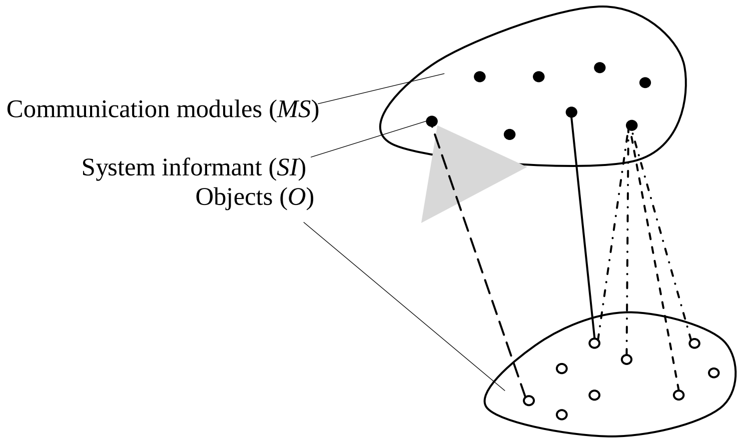

Fig. 1. The structure of the SC connections

The structure contains two sets of nodes - the objects (*O*) and the communication modules (*MSs*). The wireless signals are transmitted between the communication modules and the objects. One *MS*—the systems informant (*SI*)—has special properties.

Any object can send signals of three types—f_1, f_2, f_3—to a selected *MS* or simultaneously to groups of *MSs* (including *MSs* not associated with objects). The signals f_1 and f_3 have arbitrary lengths. The objects transmit the messages by means of the f_2

signal. As the *MS* receives signals from f_2 objects, it simultaneously modulates all entering into *MS* f_1 signals without delaying the f_2 signals. This results in the f_1 signals informing the objects about the f_2 signals coming into the *MS*. The f_3 signals prohibit the *MS* to return the f_1 signals to the objects. If the object is associated with the *MS* for receiving signals, it sends the f_1 signal continuously to the MS (solid line in Figure 1). The receiver and its *MS* is the single device consisting of two spaced components.

The source sends f_2 signals into the *MS* modules of the receivers. The *MS* module receives f_2 signals from the source (dashed line), modulates them with continuous f_1 signals coming from all sources (dash-dot line), and returns them to the receiver and to all sources of f_1 signals.

The object receiver acts like the source transmitting f_2 signals for the sources of the f_1 signals. However, it only sends the signals to its *MS*, which the other sources watch. Technically, all operations of the objects in interaction in the system are carried out by the network controller part of the object. The module (*SI*) is different from the *MS*: in obtaining f2 signal, the module creates the non-directional f_{si} signal, which is specific only for *SI*, and sends it for all network objects.

Section 4 of the article requires the next addition in the *SI*. The photodetector (*SI*) will summarize the energy of the f_2 signal from the objects (The VCSEL sources of the objects may have 30 ppm/^{o}C stability [9]). Its analog output is connected to an analog-digital converter (ADC) that digitizes the analog signal from the photodetector and returns it to all objects.

The following characteristics of network resources are used below [6-8]:

1. The fast synchronization of the objects—message sources are obtained. If the group of sources receives the synchronization start signal from the receiver, they send the signals or messages to the receiver so that they arrive at the *MS* receiver simultaneously or sequentially, without pause time between sending sources.

This principle of fast synchronization is as follows: Let the source O_i know the delivery times of the signal T_{ij} to an arbitrary communication modulus MS_j.

For synchronization, the object O_i sends a signal to the MS_j with a delay of $^*T_i$. Relative to the time of arrival from the MS_j, the clock signal $^*T_i = T_{max} - T_{ij}$, where $T_{max} \geq max\ T_{ij}$. Then, the signals of all objects acting in the same way will go to MS_j simultaneously, with the same delay of T_{max}.

If the objects transmit messages at the same time, the same-named bits of the messages will be combined and represented as a single message.

2. If there are conflicts in the message access to the *MS*, they will be quickly detected and eliminated by conflict resolution algorithms using rapid synchronization. There are several ways to resolve the conflict, one of which is discussed in section 3.1 below.

3. There is the quick barrier synchronization operation. The barrier synchronization is widely used in computers. Its main purpose is to allow interacting computers (or programs) to determine the total time of completion of the task with minimum latency in order to ensure that the results obtained are correct. Usually, this is a lengthy operation, but a quick way is given in [6-8].

Let us consider its variant. Let all the sources of the interacting group complete the work and then transmit the messages—the results of their work to the receivers waiting for the message. The sources need different amounts of time to complete the work. After the completion of work by all sources, they must transmit messages to the receivers as a single message without time delays between the individual messages.

To synchronize in a group of sources, one member of the group is allocated. Its module MS^* is known to all sources and receivers that monitor MS^* by sending it a f_1 signal. (A free module that is not associated with the object can be taken as MS^*.) When preparing the message, the sources transmit a continuous f_3 signal to the MS^*, which prohibits the return of the f_1 signals. Having prepared the message, the source removes the prohibiting signal. After all sources are ready, the MS^* will start to return the f_1 signal, which will be a clock signal for the objects. After receiving the signal, the objects will transmit messages synchronously (using fast synchronization) to the MS^*, and all receivers will receive it as a single message.

4. Simultaneous arrival of the messages from the objects in *MS* allow bitwise logical addition and multiplication, as well as finding the max and min values in times that do not depend on the number of participants in the operation. The object connections in the chain allow the logical operations and the arithmetic operations of addition, subtraction and multiplication without delay the calculations [6-8].

It is useful to consider the organization of the associative memory for comparison with the organization of the system in Fig. 1. Let us turn to Fig. 2. The associative memory contains the memory of the cells with the records (1), with the separate logical unit connected to each of its cells. The aggregate of these units is the distributed control unit (2) cells. The units from (2) store the result of the search task impacted on each cell in (1). On (1) and (2), a search query (3) is received from the computer containing the *AM*, and each unit from (2) stores the result of the request for this record, allocating the record corresponding to the request. Next, the unit state is used to perform associative operations (section 1). The *AM* has the central control unit (4), which interacts with the computer by exchanging control signals (5). It also receives search results (6) outputted from (1) and (2) to the computer and acts on (1) and (2) together with the signals (3).

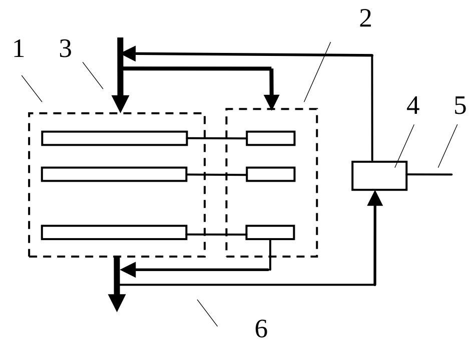

Fig. 2. The structure of the associative device

Thus, the AM is an orderly structure of simple devices, where many records are analyzed simultaneously. The structure of the *APP* is close to the structure of the *AM*, but since the *APP* performs more complex operations, its control unit (2) is more complicated. In the early version of the *APP* [10], the array of records is simultaneously divided into clusters directly in the associative memory of *APP*.

The flexibility of the structure of AM, APP and the structure in Fig. 1 are significantly different. In AM and APP the structure is fixed—all units of devices and their connections are unchanged. The structure in Fig. 1 changes cardinally during the execution of the processor command. In this case, the functions of the devices also change—the search initiator becomes its executor, a group of interacting initiators of the search is quickly created, and so on. As will be shown below, distributed associative devices that perform *DAO* obtain results with significantly expanded capabilities compared to those available in *AM* and *APP*.

3 Distributed Associative Units and Their Actions

In the article, the distributed associative unit (*DAU*) is a collection of objects using wireless optical communications and performs both the *AO*, like the devices *AM* and *APP*, and a number of additional operations, which may be implemented only in the structure of section 2.

3.1 The Simple Search in *DAU*

Let the objects be combined in accordance with Fig. 1 and have the records, among which the operation *DAO* is making the distributed associative search. This search depends on a particular implementation of the object. It runs in the *AM* of the object, in its operative storage device, or directly in the small *AM* in the network controller of the object.

One of the objects O_i is the search initiator and sends the *SI* a command of the simple search for implementation in the *DAU* devices. This command contains the search key (*Q*). In the simple search the value of the key should have an exact match in the records. Objects get this information, conduct a local search in their memory devices (or in the *AM*), and prepare a response for their initiators.

An answer or multiple answers are placed in the network controller of the object for use outside the object. Depending on the tasks involved in the operation, the object sends the reply message to the SI or into the module MS of the initiator.

A conflict will occur in the transfer if the objects are transmitting messages simultaneously, and it must be resolved. We offer the method of using the binary scale to resolve this conflict, which is slightly modified compared to [6-8]. The object does not have information about the conflict, and it sends their messages in *SI*, starting with its name (address). If there is a conflict, the address will be distorted, and the object receives the distorted message from the *SI*.

The object perceives such distortion as a synchrosignal for synchronous transmission of these messages. The physical addresses in these messages, assigned to *N* objects, are divided into *n* groups with *m* objects in the group. Each object knows its affiliation to the group and its serial number between the *m* objects of the group.

We introduce the scale of *A*—the binary string of *n* positions, each of which is one of *n* groups. We will be writing the value one into the position of the string that corresponds to the value of the digit if the object has the answer for *DAO*.

The objects synchronously send their messages—the scales *A* into the module *SI* with superposition of the bits in these scales *A*. The combined scale *A* is returned from *SI* to the objects.

Only the objects, which record the digit one in the scale, send the new scale *B* consisting of m bits, where each bit is allocated to one of the objects of the group. This is similar to scale *A*. The scale *B* comes to *SI*, and the combined scale returns to the objects.

The objects, which record the digit one in the scale *B*, create the new scale *C*. The scale *C* has slots where the objects point to the number of ready answers. After returning the scale to the objects, they consistently convey their messages without pauses. For small *N*, the types of scales can be reduced and may even have only one scale *C*. The conflict is resolved.

For many years, the *AM* used the paraphase presentation of binary digits with the active zero signal [6-8] for searching. Each bit is encoded by a pair of binary digits: 10 for 1, 01 for 0 and 00 for the mask *M*. The mask in *Q* coincides with any value of the corresponding bit in the records.

The arrival on the object of several responses with the paraphase pair of the bits 11 (*U*) indicates the difference in the responses. The paraphase encoding is also used in the *DAO* for speeding up the distributed logical and arithmetic operations [6-8].

3.2 The *DAO* that Have Keys with Numeric Values

Consider a search of records that have keys with numerical values in the *DAU*. This search includes searching in the records that have keys with maximum (minimum) values, in the records that have keys with values closest to the question, and in the records that have a key with a value in the given interval.

The *p*-ary positional system is used with digits that represent the strings of the bits [6, 11]. We will include digit one in the position of the string, corresponding to the value of the digit. For example, the number 36, with p equal to 10, is 000000100_000100000; for the paraphase bit encoding, it is 010101010101100101_010101100101010101.

The unary representation of the digits greatly speeds up the work when the *DAO* is executed directly on the network, such as when calculating the maximum or minimum value of the number of numbers sent to the network by objects. We will use the result from [6-8].

To determine the maximum or minimum, a group of the objects (performers of the *DAO*) synchronously sends the messages to the *MS* module of the initiator of the *DAO*, and the bits in the same position of the record of the digit are combined together. At first, each source-participant of the *DAO* sends a message with the high-order digit of the compared numbers to the *MS* module. The *MS* module returns the messages received as a result of superimposing the bits of the messages from the sources, and if the source sending the digit to the module detects the presence of a larger number, then it stops attempting to transmit its number. This operation continues for all other digits of the compared numbers. As a result, the maximum value of the numbers sent by the objects will be detected simultaneously for all sources. By inverting the representations of the signals one and zero, the minimum value will be found in a similar way.

The result of the operation is created by the *MS* module without involvement from the object computing facilities, and the execution time of the operation does not depend on the number of the objects participating in the operation. The result is sent in parallel to all objects. Similarly, the value closest to the specified value is found.

Now let us go to the search for the numbers in a given interval. The source sets the search interval and directs it to the distributed objects. It is assumed that the object or its network controller has an *AM* device, and it is required to select the search request form that is directly perceived by the *AM* without the use of the processor. We use a slightly modified solution from [11, 12].

For example, let us say that it is a requirement to find the records with values of the parameter U in the interval $137 \leq U \leq 628$ in the *AM* of objects with the searching rule "bitwise AND ≠ 0". We perform the searches with the specification of the intervals U: 13 ($y \geq 7$); 1 ($y \geq 4$) z; ($2 \leq y \leq 5$)zz; 62 ($y \leq 8$); 6 ($y \leq 1$) z, where y is the value of the digit and z is any value of the digit. For example, the record $2 \leq y \leq 5$ in the paraphase code has the form 000000001010101000. Digits 2–5 are selected. The searching rule "bitwise AND = 0" selects digit “0”.

It is easy to check that these searches select all records with values in the specified interval.

Let the digit "0" is represented by an additional position containing «1», then, for example, the searching "bitwise AND ≠ 0" with the form 0000001010101010101010 selects digits 0÷6 simultaneously.

Now let the *DAO* source require delivery of the numbers from the objects in order to carry out the next steps of the *DAO* on their basis. Objects simultaneously send the *p*-bit strings of the code of the highest digits (or the intervals of digits, given by the chain "1") of numbers corresponding to the requirements of the request to the *DAO* source. The source decides which bits in the string are stored to refine the search, sends the next refined query, etc. The additional controls for the search steps appear if analog-to-digital computation is used (section 4): for each bit one in the string of the digit, the number of records that generate this bit is calculated.

3.3 Distributed Associative Parallel Processor (*DAPP*)

Let us consider the implementation of two *DAO* operations in the *DAPP*, close to the operations in the *APP*. It is the separation of the set of records into clusters and the ordering of records in clusters. The difference from the *APP* arises from the distribution of the records between the objects.

– Selecting the clusters of objects and records. Let one of the above *DAO*s be performed in the *DAPP*, and the objects put the records—the results of the local search—in the *AM* or in the registers of the network controllers. Let there be additional bits in the records of the analyzed array, and the *DAO* specifies the additional bit and requires all objects with the correct answer to write the value equal to one into this bit for all the records found. Thus, a cluster of records distributed among objects will be allocated, and it can be accessed by its name.

– Sorting the records in the clusters. The required order of the objects and the records in them will be created by repeating the scheme for eliminating conflicts with the scales *A*, *B*, and *C*.

The *DAO* states the keys for selecting a group of objects which contain the allocated record clusters. The group objects attempt to send messages to the initiator of the search, and a conflict arises. It is eliminated using the scales *A*, *B* and *C*, after which the objects transmit special messages containing their physical addresses. The distributed cluster is ordered by these addresses, and the source of the DAO can change this order.

Then, the *DAO* conducts an analysis of the data, already taking into account the order of the location of records in the objects. Communication facilities allow the distributed parts of the array of records to quickly form and be ordered into a single array, ensuring interaction with it as a single entity. Such actions are easily supplemented by distributed computations with distributed records, which are performed directly on the network in accordance with [6-8].

This ordering of the records stored in different objects speeds up the analysis of logical, spatial and temporal relationships between the records. The same task was typical for the *APP*.

For example, one of the first developments of the *APP* was intended for grammatical analysis of texts in the information-logical computer [13] with a variable structure. Specialized devices were developed for this computer—the *AM* and *APP* used in this article [10-12].

4 Distributed Analog-Digital Operations

4.1 Distributed Analog-Digital Counting and Summation

Many tasks require counting the number of objects and records corresponding to the condition specified in the *DAO*. Such counts are often iterative. At the beginning of the process, it is enough to have inaccurate but quickly obtained results, and only at the last steps may an exact calculation be required.

If an exact solution is required, ordering records in sub-sets is used (Section 3.3). After the completion of work with the scales *A*, *B* and *C*, the total number of messages sent by the objects is determined, and the quantities of records found by the object are summarized in each message.

The operation of distributed summation from [6-8], in which the objects are connected in a chain, is also applicable. If the records in the object satisfy the condition in the *DAO*, then the object adds their number to the number in the message passing through the chain of objects. The operation is performed without delaying the message to perform the summation.

For a quick approximate calculation, the analog-to-digital method of interaction via *SI* (or *MS*) is proposed below, for which it was required in section 2 to modify *SI* in comparison with [6-8]. The objects perform the following steps for the approximate count.

Step 1.

The initiator of the *DAO* chooses the objects for the counting and conditions of the counting. As objects are programmable, complex conditions are admissible to require

a sequence of searches within the object and interaction with other objects within the same *DAO*.
The time of execution of such operations is unknown, so in general, the operation should be performed in barrier synchronization mode. For its conduct, the initiator of the *DAO* sends the participants of the operation the name of the communication module MS_{br} and the indicator of the moment when the *DAO* be completed.

Step 2.
Each *DAO* executor sends the command in the MS_{br} that prohibits the MS_{br} from returning the f_1 signals to the objects that sent them in MS_{br}. After that, the object conducts the analysis of records specified by the initiator, and after completing it, removes the prohibition of MS_{br} from returning f_1 signals. All *DAO* performers watch the MS_{br}, and returning the f_1 signal to them from MS_{br} is the start of the count in step 3.

Step 3.
– The general provisions for step 3. The source of the DAO sends information about the search condition and the keys *K* in the records to the objects. Each object containing the required records must send to the source of the *DAO* a message consisting of a string of references. Each reference corresponds to one of the keys *K*.

The reference contains the number N_b, which fixes the number of records found by the object with such a key. The number of digits in N_b is given by the source. Each digit is represented by a scale *S* - a string of binary digits in an amount equal to *p* – which is the base of the chosen number system (Section 3.2). The reference has a binary digit R_o, where the object puts “one”, if it finds the corresponding key *K* in the records.
– The object's actions in step 3. Each object sends a message to *SI*. In the references of the messages, the object sends the f_2 signals to R_o and to the strings *S* into the bits represented the digits of the numbers.

The strings are transmitted synchronously, and their same name bits must coincide in time when they enter the *SI*. As a result, the *SI* photodetector will receive a signal from each bit of the string with the total energy sent by all objects.

The signal will be digitized, and the result is sent simultaneously to the initiator of the search and to all objects. Having received digital values, objects determine the total number of records found and the number of objects that have them. The computation is completed.

If the source only needs the number of objects that meet the request, then the messages are limited to the R_o bit. The counting time does not depend on the number of objects participating in the *DAO*, and consists of a double time interval that includes the time required for the signal to pass between the *SI* and the object most distant from it and the time required to convert the analog signal to a digit.

Another solution is possible. The *DAO* selects certain modules of MS^*, to which the objects will send signals now instead of to *SI*. The module does not create a digital message as *SI does*, but reduces the transparency of the light filter for the f_1 signal in proportion to the energy of all incoming f_2 signals. Each object has a photodetector and an analog-to-digital converter. The object forms the digital value as it does *SI*.

To reduce the number of O^* objects that perform analog-to-digital conversion, we will create a small number of such objects, and we will provide the object O^* in dynamics to different initiators of the *DAO*.

After performing the analog-to-digital conversion, the O^* object will send the result to MS^*, and the result will receive the *DAO* initiator and other objects that watch MS^*.

If the range of energy levels of the total signal arriving at the photodetector *SI* and O^* exceeds the linear region the photodetector, two methods may be applied to reduce the energy of the signals arriving at the photodetector.

The first method: the ADC reduces the throughput of the light filter-modulator receiving the f_2 signals and supplements the message sent to the objects with information about the small precision of the sample.

The second method is the logical method: the request initiator selects groups of objects that must simultaneously send signals to *SI* and O^*. To do this, the initiator details the request in the *DAO*, reducing the number of the sources of the messages, and/or indicates the area of physical addresses of objects that are allowed to send messages.

If different groups of objects are allocated different O^* values, then this reduces the total energy of the signals arriving at O^*.

Additional information is provided using paraphase binary signals. Let us give an example. The group of records, in which the search is performed, usually, has an unknown size. However, to assess the significance of the data found, their share in the total search volume must be known. The paraphase code in the position R_o will be applied. The number of analyzed records may be determined by a count of active signals one and zero in the R_o.

The counting in large numbers of records is required by many applied algorithms, for example, algorithms that work with associative rules [4, 5], and algorithms using a naive Bayesian classifier [14]. The counting operations of keys in various combinations in large sets of records take considerable time in such algorithms.

Our solutions turn the distributed operations of counting records into the summation of the energy of signals within a single message, which is created simultaneously by all objects.

It should be noted that an operation analogous to the summation of the number of records allows the summation of any numbers to be simultaneously transmitted by objects to the communication module. In *MS*, the energy of the signals in the scales representing the digits is summed. From these sums, the objects get the sum of numbers.

Such summation is also performed during the simultaneous sending of messages by objects.

4.2 Associative Operations as a Means of Managing *SC* Objects

The *DAO* is a quick tool for monitoring the state of *SC* objects, but it is also useful for managing their behavior. Let us turn to Fig. 3.

Let the object I_{dao}, initiator of the *DAO*, send a group program (a sequence of the *DAO* commands) via the *SI* module that collects information about the state of objects. The analog-digital calculations and barrier synchronization is used.

Let I_{dao} initially receive a number that exceeds the resolution of its analog-to-digital converter. Upon discovering this, I_{dao} refines the requests (§ 4.1) by dividing the set of responding objects into groups G_1, G_2 ..., G_6. The groups G_1, G_2 and G_3 are assigned an object O_1^* to convert the analog-digit, and the object O_2^* is assigned to groups G_4, G_5 and G_6. The objects O_1^* and O_2^* forward the answers to the I_{dao} after they receive it.

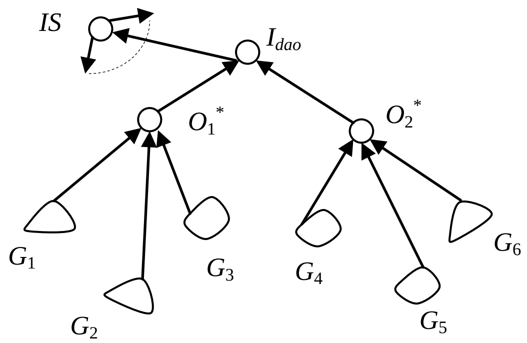

Fig. 3. Managing the state of objects

Now I_{dao} can proceed to the management of the objects, sending the new *DAO* to the group G_i via the *SI* to all objects.

In Fig. 3, the object I_{dao} is somehow allocated in advance. However, unlike *AM* and *APP*, the objects participating in the *DAO* are active and each takes into account its state and the state of the objects it observes. They can each receive the rights of the initiator of the *DAO*. Therefore, within the framework of Fig. 3, there is a place for a common control center of the system, but there is an additional control capability. Any object is allowed to promptly intervene in the behavior of the system, leaving slow work for the center to manage access to the shared resources.

5 Conclusions

The considered organization of the *DAO* with the use of wireless optical connections makes it possible to obtain the following main results.

1. This study proposes the extension of local operations of associative memory and associative processors up to the distributed associative operations that operate between the groups of interacting *SC* objects.

2. Data exchange with distributed computations requires the same amount of time as data exchange without computation.

3. Analog-to-digital approximate calculations are developed that accelerate the distributed summation of numbers, each of which is located in a separate device of the *SC*. The time of the summation does not depend on the number of participants in the addition operation and is performed during the time of simultaneous message transmission by devices with overlapping messages in time (section 4). The options to increase the accuracy of calculations are shown. Such a calculation method greatly speeds up the analysis of large data sets.

4. Proposed DAOs do not require sophisticated technical aids in addition to the technical aids proposed in [6 - 8].
5. The structure of the device connections and their functions are quickly changed by sending a broadcast message to the *SC* devices.

New *SC* functions are obtained through the use of optical wireless connections, using retroreflectors.

References

1. Krazmer, L.P.: Associative memory devices. Energy, Leningrad. (1967). (in Russian)
2. Kohonen, T.: Content-Addressable Memories. Springer, Berlin (1980).
3. Foster, C.: Content Addressable Parallel Processors. Van Nostrand, New York (1976).
4. Agrawal, R., Shafer, J.C.: Parallel Mining of Association Rules. IEEE Transactions on Knowledge and Data Engineering 8, (6), 962-969 (1996).
5. Vasoya, A., Koli, N.: Mining of association rules on large database using distributed and parallel computing. Procedia Computer Science. 79, 221-230 (2016).
6. Stetsyura, G.G.: Addition for Supercomputer Functionality. Reports of the International supercomputing conference "Russian Supercomputing Days 2016". Moscow, MSU, 314-324 (2016). (in Russian). URL: http://russianscdays.org/files/pdf16/314.pdf
7. Stetsyura, G.G.: Addition for Supercomputer Functionality. Supercomputing. CCIS. 687, 251-263 (2017).
8. Stetsyura, G.G.: The Computer Network with Distributed Fast Changing of Structure, and with Data Processing during Transmission. Control Sciences. (1), 47-56 (2017). (in Russian). URL: http://pu.mtas.ru/archive/Stetsyura_117.pdf
9. Downing J., Babićb D., Hibbs-Brennerc M. *An ultra-stable VCSEL light source//* SPIE Photonics West 2013 and 2015 Conferences on VCSELs. USL Technologies, 22 p. (2015). URL: http://www.ultrastablelight.com/papers/SPIEPapers.pdf (2012).
10. Stetsyura, G.G.: Decisive device for translation tasks. Reports at the Conference on Information Processing, Machine Translation and Automatic Reading of Text. Acad. Sciences of the USSR. Institute of Scientific. Information. Issue. 8. 9 p. (1961). (in Russian).
11. Stetsyura, G.G.: A New Principle of Building Memory Devices. Reports of the Academy of Sciences of the USSR. 132, (6), 1291-1294 (1960). (in Russian).
12. Stetsyura, G.G.: Automatic search device with feedback. Reports at the Conference on Information Processing, Machine Translation and Automatic Reading of Text. Acad. Sciences of the USSR. Institute of Scientific. Information. Issue.4. 10 p. (1961). (in Russian).
13. Kleinerman, G.I., Kosarev, A.A., Nisnevich, L.B., Stetsyura, G.G.: On the question of constructing universal digital machines with many solving devices. ib. Issue. 9. 5 p. (1961). (in Russian).
14. Naive Bayes classifier (Wikipedia) https://en.wikipedia.org/wiki/Naive_Bayes_classifier.