

## Cluster management of computing resources

V.N. Ruchkin<sup>1</sup>, M.N. Makhmudov<sup>1</sup>, V.A. Romanchuk<sup>1</sup>, V.A. Fulin<sup>1</sup> and B.V. Kostrov<sup>2</sup>

<sup>1</sup>Ryazan State University, Ryazan, Russian Federation

<sup>2</sup>Ryazan State Radio Engineering University, Ryazan, Russian Federation

**Abstract.** The study suggests management of computing resources through set-theoretic clustering. The research proposes production modelling of knowledge — rules of managing explicit and fuzzy structures, according to their technical specifications: system productivity, number of computing modules, capacity of microprocessor memory, etc. The paper describes an algorithm of “evaluation of management criterion value”, according to the additive utility function.

### 1 Introduction

Methods of clustering have found wide use in cases when an a priori definition of data or their use for defining similar objects (samples) is impossible, whether such objects belong to the same group – cluster, or show difference between clusters [1, 2]. The current methods of clustering possess both advantages and disadvantages. That is why a compromise is often necessary between computational complexity, model fitting, and organization of the clusters' structure, depending on the nature of data under analysis and the specific field of their application. Also, the cluster shape depends on the measurement chosen, which influences the visualization and description of a cluster. Due to the uncertain character of the data, the introduction of fuzziness allows modelling smooth change of properties of multi-microprocessor implementations, taking into account qualitative structural relationships, which is especially important for distribution of computing resources [3, 4]. One of the advantages of the fuzzy approach is that the solution of the task of a choice occurs at the same time as the decision-making procedure [8, 9, 13, 14]. Besides, the mathematical tools of fuzzy sets allow for certain optimization, e.g., as max / min operations [14, 15]. As a result, it is possible to state that, to a certain degree, the result is an intellectual optimized system of choice-making [13, 16, 17]. The paper describes multi-criterion fuzzy management of computing resources with the help of linguistic variables, based on cluster analysis proposed by the authors [11-13]. The authors suggest using the set-theoretic approach [8, 9] that allows partitioning the input processing program into non-overlapping sets of equivalence classes, meeting such requirements as reflexivity, symmetry and transitivity, and further to

implement cluster management of computing resources.

The structure of the research is as follows. Section 2 discusses set-theoretic clustering for presenting computing resources. Section 3 focuses on production modelling of knowledge - rules of presenting a management system. Section 4 suggests multi-criterion fuzzy choice of a management structure. Section 5 employs the algorithm of “evaluation of management criterion value”. Section 6 forms a conclusion.

### 2 Set-theoretic clustering for presentation of computing resources

Cluster systems provide sufficient availability of systems at relatively low costs. Clustering is understood as “implementation of joining together machines, viewed as a unity by the OS, system software, applications, and users” [1]. When machines are clustered, if one of them stops working, the other processors can quickly redistribute the computing inside the cluster, because there is a single management mechanism for the cluster, which allows interactive tasking. However, one disadvantage is that it is not possible to fully manage a task after it is sent to execution by the user. Then, the system should enable the user to check the status of the task under execution, the capacity of memory required, and the processing time of the program's implementation.

The suggested approach is based on algorithms  $A^{(j)}, \forall j = \overline{1, N}$  - an organized set of operations, or tuple  $A^{(j)} = \langle O_1, O_q, \dots, O_L, O_2, O_1, \dots, O_r \rangle$  [8, 9], which, in accordance with the chosen element base chipset and transformation  $O_l \rightarrow \{MK_m\}; \forall m = \overline{1, K_m}; \forall l = \overline{1, L}$  is provided by data processing program  $PR^{(j)} = \langle MK_1, MK_2, \dots, MK_l, \dots, MK_M \rangle$ , as a tuple of

commands.

To implement parallelization and partitioning of the processing program into clusters, it is possible to introduce the notion of equality of data processing clusters  $CL_1$  и  $CL_k$ .  $CL_1 = CL_k$ .

Further, equality of clusters is regarded as equality of their lengths  $|CL_1| = |CL_k|$  and their coincidence with an accuracy to command  $MK_i(k) = MK_i(l)$ ,  $PR(j) = \langle MK_i, MK_1, \dots, MK_j \rangle$ .

$$CL_1 = CL_k = \begin{cases} |CL_1| = |CL_k| \quad \forall l, k = 1, N \\ MK_i^{(k)} = MK_i^{(l)}, \forall i = 1, |CL_1| \end{cases} \quad (1)$$

Here, Expression (1) serves as a criterion for the evaluation of the partitioning quality for input program clusters, accurate to a command, which serves as the measurement. If there is a change in transformation  $O_l \rightarrow \{MK_m\}; \forall m = \overline{1, K_m}; \forall l = \overline{1, L}$ , or if the element base changes, the clustering measurement changes, too, leading to different clustering results.

Then, in keeping with the design process, we define a set of all possible structures  $S_w \in S$ , that allow a  $j$  processing program  $PR$  to correlate cluster set  $CL_1$   $S_w \in S: PR \Rightarrow \{CL_i\}; \forall i = \overline{1, l}; \forall w = \overline{1, W}$  (2).

Expression (2) defines an already distributed structure of the computing system. The research proves that Expression (2) meets the requirements of reflexivity, symmetry and transitivity.

Thus, structure  $S_w$  of the data processing computing system under design correlates a set of independent and unequal clusters  $CL_1$  to a  $j$  processing program  $PR$ ; here, the number of clusters is equal to the number of equivalence classes  $L$ ,  $CL_1$ ,  $S_w \in S$ , and their repetition factor ( $RO_l$ ) is correlated by the order of their equivalence class ( $a_l$ ):

$$\forall o = \overline{1, N}, PR^{(j)} \Rightarrow \{(CL_l)^q\}, \forall q = 1, a_l; \forall l = 1, L \quad (3)$$

Expression (3) defines an already distributed structure of the computing resource management, is a relationship of the equivalence of partitioning into independent modules, and shows a procedure of explicit parallelism. To preserve the semantics of the input program, we introduce explicit and implicit (fuzzy) parallelism of clusters.

Further, the focus of this research is on possible solutions to task (5): a production model of an expert system.

### 3 Production modelling of knowledge – rules of management system

Solving task (5), we obtain options of structural arrangement for our MPS that define the principal knowledge — rules of management as productions of an expert system.

Associations between facts are shown via heuristic rules – expressions of declarative knowledge about relations between objects: where  $L$  is the number of equivalence classes, and  $|a_l|$  is the order of each class.

According to the model, clusters are used to execute an array of consecutive programs that are not associated with each other; distributed programs (coarse-grain parallelism) of autonomous processes with shared data and parallel programs. Also the distributed programs are the main type of tasks for which clusters are created. Each program of this type consists of a finite number of processes that exchange data via the clusters' communicative environment. For this, the management system acts in accordance with the task and performs operations required to launch and execute a task or a group of tasks (VLIW) during the processing of the computation results.

In some cases, they are package tasks, and their content may be input for other tasks.

As a result, the following options of  $\forall L, |a_l| = 1$ , knowledge — rules are possible:

1. If  $\forall L, |a_l| = 1$ , and data exchange between fragments in equivalence classes is sequential, it is a pipeline structure.
2. If  $L = 1, |a_l| = q$  and data exchange management is required simultaneously by all clusters of classes, it is a vector structure (SIMD according to N. Flynn) [8, 9].
3. If  $L = 1, |a_l| = q$  and management of data exchange between clusters in the classes is sequential, it is a pipeline structure
4. If  $\forall L, |a_l| = q$  is a pipeline structure and data exchange management is required simultaneously by all clusters of classes, it is a vector structure.
5. If  $\forall L, |a_l| = q$  and data exchange management in the classes is sequential, the option is a pipeline structure.
6. If  $\forall L, |a_l| = q$  and data exchange management between clusters in the classes is sequential, with minor exceptions for  $l$  class clusters, it is a near-pipeline structure.
7. If  $\forall L, |a_l| = q$  and data exchange management between clusters in the classes is sequential via a distributed processing program, except for class  $l$  clusters, and class  $l$  requires data simultaneously, it is a near pipeline-vector structure.
8. If  $\forall L, |a_l| = q$  and data exchange management between clusters in the classes is sequential via a distributed processing program, but some classes require data simultaneously, it is a pipeline-vector structure.
9. If  $|a_l| = L$  and does not depend on data transfer, it is an array structure (MIMD).
10. If  $|a_l| = L$  and data exchange management is required simultaneously by clusters of all classes, but some classes require sequential exchange, it is a near vector-pipeline structure.
11. If  $|a_l| = L$  and data exchange management is required simultaneously by clusters of all classes, but class  $l$  clusters exchange data sequentially, it is a multiple vector-pipeline structure.
12. If  $|a_l| = L$  and data exchange management is required simultaneously by clusters of all classes, except for some class  $l$  clusters, it is a near vector structure.
13. If  $|a_l| = L$  and data exchange management is

required simultaneously by all clusters of all classes, it is a vector structure.

The management structures analyzed are explicit ones, as they combine elements of explicit parallelism: pipeline, vector, pipeline-vector, or vector-pipeline. In the design procedures, some tasks are parallelized similarly to a vector structure, and other tasks are parallelized like in a pipeline. The result is a set of structures meeting requirements (5), and each requires classification, e.g., through linguistic variables based on fuzzy sets. Such complex and uncertain structures are fuzzy ones, and they are based on implicit parallelism, e.g., with pipeline or vector elements. The correlations obtained are basic for the training of expert systems.

#### 4 Multi-criterion fuzzy choice of management structure

In practice, the process of a **management structure** design for clusters in a computing system is a complex multi-criterion choice task, because from the various  $S_w \in S$  structures, an expert will make a decision depending on a generalized evaluation of each option, choosing the best one as regards the sum of principal contradictory criteria: 1. performance speed ( $1/T_r$ ), 2. capacity of microprogram memory ( $|MP|$ ), 3. number of microprocessor modules ( $N_o$ ), 4. downtime ( $T_{dt}$ ), as well as additional criteria: 5. power consumption (P), 6. reliability (Q), 7. adaptability (A), 8. cost (C), etc. As a result, we obtain a set of management structures. To specify the choice of a management system, all the above criteria are ranked in a tuple according to their importance in technical design; this is further referred to as the **choice strategy**.

The authors [8, 14] have suggested the method of multi-criterion choice, based on fuzzy sets, where the characteristic function has a definition range  $E = [a, b]$ , and the elements are linguistic variables (LVs). An expert evaluates the structures and criteria, using notions of fuzzy logic, or terms  $t \in T$  in the  $E$  range, each fuzzy variable has a membership function  $m_t(e)$ , i.e.  $t - \{m_t(e), e\}$ .

The suggested approach uses two types of **binary** fuzzy variables: ‘evaluation of management criterion value’  $R_{j|j+1}$  and ‘evaluation of management structure effectiveness’  $Q_{j|j+1}$ , using a **unary** linguistic variable: ‘qualitative evaluation of criteria value’  $m_{z_j}(x)$  and ‘evaluation of structure effectiveness value’  $m_{z_j,si}(x, y)$ , accordingly.

The values of binary linguistic variables ‘evaluation of management criterion value’ и ‘evaluation of management structure effectiveness’ are computed following the method of a fuzzy algorithm for rules of this type: "if A, then B". Two or more rules of fuzzy expressions are connected with "else": "If A, then B, else C"  $\Rightarrow [A][B] + [A]*[C]$ , or  $V$  ("if A, then B, else C")  $= m_A / V(B), 1 - m_A / V(C)$ .

#### 5 Algorithm of “evaluation of management criterion value”

Evaluation of the importance of each above criterion is performed in keeping with the strategy design according to the composition rule  $Z_j = Z_{j+1} \circ R_{j|j+1}, j = n-1, n-2, \dots, 1$ .

The membership function for the fuzzy set  $Z_j$  will be the result of the max/min operation:

$$m_{z_j}(x) = \bigcup_{y \in Y} m_{z_{j+1}}(y) \vee m_{R_{j|j+1}}(y, x).$$

For weighing the management criteria importance, fuzzy sets are converted to scalars in a suitable manner, e.g., when the choice of such binary variable values provides maximum use of the membership function. Then scalars of management criteria evaluations are associated with a scale and normalized. As a result, each criterion receives a specific weight  $W_i, \forall i = \overline{1, n}; \sum_{i=1}^n W_i = 1$ .

Similarly, from linguistic estimates of the relation of evaluation of management structure effectiveness, a normalized assessment of effectiveness is found for each option  $S_w$ , for each  $j$  criterion  $F_{ij}$ .

An integral characteristic of the suggested multi-criterion fuzzy choice is the additive utility function  $U_i$ , calculated at the final stage of the suggested procedure for evaluation of alternative options of structures of data processing, based on the whole set of criteria. Besides,

options are ranked by their value  $U_i = \sum_{j=1}^n W_j * F_{ij}$ .

Basing on the data obtained, the preferred choice of the CS structure  $S_w \in S$  is defined, so as to maximize its utility function.

Basing on the described dependencies, an algorithm is suggested, broken into these steps.

1. Choice of carriers and initial distribution for the last criterion..
2. Determination of the strategy of design as a ranked set of given criteria.
3. Calling the process of definition of the linguistic variables, ‘evaluation of management criterion value’  $R_{j|j+1}$ , performing computation according to composition rule  $Z_j = Z_{j+1} \circ R_{j|j+1}$ , with choice of criteria beginning with the last one  $j = n-1, n-2, \dots, 1$ .
4. Calling the process of evaluation of quality of criteria value, performing computation according to formula  $m_{z_j}(x) = \bigcup_{y \in Y} m_{z_{j+1}}(y) \cap m_{R_{j|j+1}}(y, x)$  ..
5. Obtaining scalar estimates  $X_j^k$  of the criteria values, i.e. numeric identifications of those carriers in case of which values of criteria reach the value of one.
6. Obtaining evaluations, according to the formula  $H_i = 3(X_j^k + 1) + 1$ .
7. Obtaining normalized scales according to the

formula: 
$$W_j = \frac{H_j}{\sum_{j=1}^n H_j}$$
.

In the general case, a cluster management system should consist of two subsystems: one for the managing

node and the other for the computing node – peripheral. So a cluster should have two communications systems: for interaction of components of a distributed program in the cluster, and for organization of interactions between the subsystems of the cluster management system. In this case, the communications system will ensure a maximum throughput and minimize delays in data transfer.

Otherwise, additional load from the management system will reduce the productivity of the cluster. The management subsystem will interrogate peripheral subsystems of the cluster nodes and accept user tasks, sending them to the computing nodes and receiving information about the current state of each cluster. As a result, it will be possible to avoid traffic volume surges when peripheral systems act in miscoordination.

In the general case, a cluster management system performs on a client-server principle, and the management system is the server, while each peripheral system is a client.

## 6 Conclusion

The paper suggests set-theoretic clustering, based on which, production modelling of knowledge – rules of management is performed; it also describes multi-criterion fuzzy choice of a structure for a computing system, according to a required choice strategy. With a view to this, the authors have developed an expert system to analyze fuzzy structure output according to various technical specifications: performance speed ( $1/T_R$ ), capacity of microprogram memory ( $|MP|$ ), the number of microprocessor modules ( $N_0$ ), etc. To implement the management strategies, the authors have developed an algorithm of evaluation of a management criterion value, in order to compute the additive utility function  $U_i$  of alternatives in management of computing resources.

## References

1. A.I. Avetisyan, D.A. Grushin, A.G. Ryzhov. Systems of cluster management. // Proceedings of the Institute for System Programming of Russian Academy of Sciences. – 2002. – L. 3.
2. Ruchkin V.N., Romanchuk V.A., Development of algorithms determining the type of structure neuroprocessor system, based on the description of relations of its elements // Informatics and Applied Mathematics: Interuniversity collection of proceedings. Ryazan – 2011. – p. 99.
3. L. A. Zadeh, “Fuzzy logic and its application to approximate reasoning”. In: Information Processing 74, Proc. IFIP Congr. 1974 (3), pp. 591–594.
4. L. A. Zadeh, “Computing With Words. Principal Concepts and Ideas”. Berlin: Springer, 2012.
5. R. Jordans, L. Jozwiak, H. Corporaal, “Instruction Set Architecture Exploration of VLIW ASIPs Using a Genetic Algorithm”, 3<sup>rd</sup> Mediterranean Conference on Embedded Computing MECO – 2013 Budva, Montenegro, pp.32-35
6. Ruchkin V.N., Romanchuk V.A., Evaluation of the results of modelling of computing systems based on neuroprocessors // Izvestiya TulGU. Tekhnicheskiye Nauki. – Tula: Tula State University, 2013. Issue 9, part 2.
7. E. Diken, R. Jordans, R. Corvino, L. Jozwiak, H. Corporaal, and F. A. Chies, “Construction and exploitation of VLIW ASIPs with heterogeneous vector-widths”, Microprocessors and Microsystems, vol. 38, no. 8-B, pp. 947–959, November 2014.
8. Romanchuk V.A., Ruchkin V.N. Algorithms of analysis of computing structures based on microprocessors // Vestnik RGRTU. 2012.№2. Issue 40. Pp. 61-67.
9. Ruchkin, V. Romanchuk, R. Sulitsa. Clustering, Restorability and Designing of Embedded Computer Systems Based on Microprocessors“, 2nd Mediterranean Conference on Embedded Computing MECO – 2013 Budva, Montenegro, pp. 58-61.
10. Yury Bekhtin, Victor Gurov, Maria Guryeva Algorithmic Supply of IR Sensors with FPN Using Texture Homogeneity Levels 3rd Mediterranean Conference on Embedded Computing MECO – 2014. Budva, Montenegro. pp. 252-255.
11. Grigorenko D.V., Ruchkin V.N. Improving restorability of cluster data processing computing systems. // Tsifrovaya obrabotka informatsii № 1, 2013
12. Ivutin A.N., Larkin E.V. Estimation of Latency in Embedded Real-Time Systems 3rd Mediterranean Conference on Embedded Computing MECO – 2014. Budva, Montenegro. pp. 236-239.
13. V.N. Ruchkin, B.V. Kostrov, A.A. Kolesenkov, E.V. Ruchkina. Anthropogenic Situation Express Monitoring on the Basis of Fuzzy Neural Networks // Proceedings of the 3rd Mediterranean Conference on Embedded Computing MECO, 2014. Budva, Montenegro. pp. 166-169.
14. Ruchkin V.N., Romanchuk V.A., Fulin V.A. Design of computing systems based on fuzzy clustering // Vestnik RGRTU. 2014. №4. Issue 50-1. Pp. 87-93.
15. Romanchuk V.A., Ruchkin V.N. Design of program means of analysis of microprocessor systems // Vestnik RGRTU. 2010. №32. Issue 40. Pp. 61–67.
16. Ruchkin V.N., Fulin B.A. Architecture of computer networks. Moscow: Dialog-MIFI, 2008.
17. Ruchkin V.N., Fulin V.A. Development of a model of a complex neuroprocessor structure // Tsifrovaya obrabotka signalov, 2012, № 4, pp. 70-74.