Clustering, Restorability and Designing Of Embedded Computer Systems Based On Neuroprocessors

Vladimir Ruchkin
Professor of CS and Engineering Chair
Ryazan State University
Ryazan, Russian Federation
v.ruchkin @rsu.edu.ru

VitalyRomanchuk
Associate Professor of CS and Engineering Chair
Ryazan State University
Ryazan, Russian Federation
v.romanchuk@rsu.edu.ru

Roman Sulitsa
Graduate Student of CS and Engineering Chair
Ryazan State University
Ryazan, Russian Federation
r.sulitsa@rsu.edu.ru

***Abstract*—The article is devoted to the possibilities of improving the recoverability of cluster neuroprocessor systems of pipeline, vector, pipeline-vector or vector-pipeline processing data structures on the base of modern Russian microset NM 640X.The work was performed as a part of the RFBRgrant №12-07-97516/12.**

Keywords-neuroprocessor systems, set-theoretic model, equivalence classes, reflexivity, transitivity, clustering, recoverability, reliability, probability of reliable performance

I. INTRODUCTION

The most effective way to achieve the given efficiency level is scalable parallel architectures use [1- 3]. Increasing of reliability is based on the rectifying errors using architectural and additional embedded equipment. The completing of the given tasks is performed by means of software and hardware realization redundancy of different fault-tolerance architectures.The authors suggest a conceptual design model which allows to implement the system as one of the possible structures by means of clustering and to increase cluster neuroprocessor systems recoverability by means of redundancy.

II. MATHEMATICAL METHODS OF CLUSTER ANALYSIS

The term “cluster” is applied to two or more computers connected to a single system with special hardware and software [3]. The clusters can be seen as a heterogeneous computing system with shared memory and distributed control. As a result, high level of parallelization, reliability, availability, and serviceability is provided at relatively low costs.

The term “clustering” means implementation of computer unification representing a single entity for the operating system, system software, applications and users. The computers thus clustered have the following properties: resources sharing, high availability, high throughput, ease of system maintenance and extensibility.

Clustering methodssolve the problems of objects dividing with the given feature space or of objects proximity matrices into equivalence classes [4], and the objects equivalence is based on proximity and likeness measures and so on. Further the term “cluster” would be used for a set of close objects obtained as a result of solving the cluster analysis problem.

The principles according to which the objects are united into one cluster are set in any particular algorithm of clustering. Knowing these principles the user can interpret the results of any specific method.

Different clustering methods can result in decisions having sufficient differences. Thus besides the set of different clustering methods there is a practical interest in the presence of automatic results processing obtained by different algorithms independently [3-5]. There are some ways to solve a cluster analyses problem.

Further the main task of clustering would be considered first and foremost as a task of the search for partition of attribute definitions sample

$$I(S_1), I(S_2), \ldots, I(S_m), I(S) = (x_1(S), x_2(S), \ldots, x_n(S))$$

given by a numerical table $T_{nm.}$

The given problem is considered as a search process of splitting the sample into groups (classes, clusters, taxa) of similar objects. The required splitting itself is considered as a solution of some optimization task, as a result of convergence of some iterative procedure, as the result of applying a deterministic procedure and so on.

In general, let’s consider the task of clustering into *1*clusters. We will assign the sample of attribute descriptions of the objects as

$$X = \{x_1, x_2, \ldots, x_m\}, x_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}.$$

Splitting $K = \{K_1, K_2, \ldots, K_l\}$ the sample into $X = \{x_1, x_2, \ldots, x_m\}$ groups is a random totality of non-overlapping subsets of the set X, covering all objects in the sample

$$K_i \subseteq X, i = 1,2,\ldots,l,$$

$$\bigcup_{i=1}^{l} K_i = X, K_i \cap K_j = \varnothing, i \neq j$$

Assume that some criterion F(K) of K splitting is given. Then the task of clustering means to find the K* splitting delivering extreme value for the criterion

$$F(K): F(K^*) = \underset{K \in \{K\}}{\text{extr}} F(K)$$

For example we can use such criteria as [6]:

1.Amount of intra-class variance or the sum of the squares for the mistakes.

$$F(K) = \sum_{j=1}^{l} \sum_{x_i \in K_j} \rho^2(x_i, y_j)$$

Where

$$y_j = \frac{1}{n_j} \sum_{x_i \in K_j} x_i, n_j = |K_j| \approx$$

is a number of objects in the $K_i$group.

The solution of the cluster analysis problem with the criterion given is such K* splitting which minimizes the functional F(K).

2. Criteria based on scattering matrix. The scattering matrix for the K_i group is defined as

$$S_j = \sum_{x_i \in K_j} (x_i - y_j)(x_i - y_j)^t,$$

and matrix of intragroup scattering is defined as

$$S = \sum_{j=1}^{l} S_j$$

(where*t*means transposition).

There are several known criterion definitions of clustering on the base of matrix of intragroup scattering. E.g. it can be the choice of matrix of intragroup scattering determinant $F(K) = |S|$.

A well-known technique is a “k-intragroup averages” technique. This technique implies the creating of splitting consequences $K = \{K_1^i, K_2^i, \ldots, K_l^i\}$ i=1,2,… as a result of the following homogeneous iterations.

Let splitting K is chosen at random. For the K group its center

$$y_1 = \frac{1}{n_1} \sum_{x_j \in K_1^i} x_j$$

is calculated.

Further all the elements of the sample which are closer to y_1 then to similarly obtained $y_2, y_3, \ldots y_l$ are included to the K_1^{i+1} group.

K_l^{i+2} group is formed in a similar way, but according to the set of objects $\frac{X}{K_l^{i+1}}$ and so on..

After the calculating of $K_1^{i+1}, K_2^{i+1}, \ldots, K_l^{i+1}$ the centers of these groups are recalculated and the process of computing repeats.

Another technique is Forel method which implies that the clusters found are not the results of some criterion optimization but by iterative procedures application, when the hyperspheres of the set radius move in the direction of places "condensation" of objects [3,6].

Assume that some positive number R is set. Then the random element is and a hypersphere of R radius with the center in $y_1 = x_t: R_1 = \{x : \rho(x, y_1) \le R\}$ are chosen.

Suppose

$$K_1^1 = \{x_i : x_i \in X \cap R_1\}.$$

Then

$K_1^2 = \{x_i : x_i \in X \cap R_2\}$ is calculated.

A new sphere center is calculated as $y_2 = \frac{1}{|K_1^1|} \sum_{x_j \in K_1^1} x_i$ and a group of $R_2 = \{x : \rho(x, y_2) \le R\}$.

The process is complete when such group of objects $K_1^t = \{x_i : x_i \in X \cap R_t\}$, is calculated for which $K_1^t = K_1^{t+1}$ is fair.

III. THE SET-THEORETIC MODEL

To solve the given problem we can introduce the conception of data $CL_l = CL_k$, processing clusters CL_l and CL_k equality, which presupposes the equality of the lengths of these clusters and match them up to the command $MK_i^{(k)} = MK_i^{(l)}$, thus

$$|CL_l| = |CL_k| \quad \forall l,k = 1,N;$$
$$MK_i^{(k)} = MK_i^{(l)}, \forall i = 1, |CL_l|. \qquad (1)$$

The cluster structure $KS_w \in S$ of data processing introduced is a relation of equivalence and satisfies the conditions of reflexivity, symmetry and transitivity.

Any random cluster of CL_l data processing satisfying (1) is running parallel to itself, so $CL_1 \xrightarrow{KSw} CL_1$.

Thus the condition of reflexivity of any data processing cluster $CL_l \in PR^{(j)}$ is fair.

If CL_l data processing cluster is equal to CL_k data processing cluster and, consequently is parallel to it then CL_kdata processing cluster is equal to CL_l data processing cluster and consequently can run simultaneously, so

$$CL_l, CL_k \in PR^{(j)}: CL_l \; KS_w CL_k \Rightarrow CL_k \; KS_w \; CL_l. \quad (2)$$

In other words, the condition of any two CL_l, $CL_k \in PR^{(j)}$ data processing clusters symmetry is met.

If CL_l data processing cluster is equal to CL_k data processing cluster and, CL_k data processing cluster is in turn equal to CL_q data processing cluster then CL_l data processing cluster is equal and consequently parallel to CL_q data processing cluster so

$$CL_l, CL_k, CL_q \in PR^{(j)} EKS_w: CL_l KS_w CL_k,$$
$$CL_k KS_w CL_q \Rightarrow CL_l KS_w CL_q. \qquad (3)$$

Thus the condition of transitivity of two random data processing clusters CL_l, $CL_q \in PR^{(j)}$ is fair.Then the statement about the relation of KS_w cluster processing structure is a relation of equivalence.

Indeed the relation of KS_w cluster processing structure indicates that any two random clusters CL_l and CL_k satisfying (1) can run simultaneously by different processing

modules under the control of commands belonging to the clusters indicated.

$$CL_l, CL_k \in PR^{(j)} : CL_l KS_w CL_\kappa \quad (4).$$

The introduced relation of KS_w data processing cluster structure puts into correspondence of some *j* program of $PR^{(j)}$ data processing the totality of any independent and unequal CL_1 clusters, number of which is equal to the L number of equivalence classes, and $(CL_l)^q$ multiplication factor is defined by the equivalence class order $|a_1|$:

$$\forall j = 1, N\ PR^{(j)} \xrightarrow{Sw} \{(CL_l)^q\}, \quad \forall q = 1, |a_1|; \forall l = 1, L \quad (5)$$

Any equivalence class has its own CL_l cluster representative, which has macro instruction number equal to the $|CL_l|$ cluster order. The equation (5) is a solution of cluster analyses problem since it specifies the required splitting of the initial program $PR^{(j)}$ into clusters.

IV. Cluster Representation of Neuroprocessor Systems

As a result the following variants of solving the problem (5) are possible.

1. The number of equivalence classes is equal to L and the the order of each class is equal to one $a_l = 1$. Then *j* information algorithm is represented by a procession of L clusters

$$A^{(j)} \xrightarrow{Sw} \{CL_1\} = \langle CL_1, CL_2, ..., CL_l, ..., CL_L \rangle, \quad (6)$$

each of them is a is representative of its equivalence class, and provides input information for the subsequent CL_{l+1} cluster. Taking the number of processing modules equal to L and assigning each l-th unit a corresponding cluster, we obtaina neuroprocessing of the pipeline type on the base of NM 640X neuroprocessor [8] as represented in Figure 1.

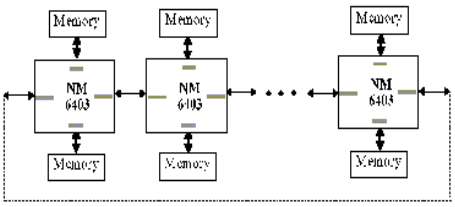

Figure 1. Cluster processing structure

Next, define the basic parameters of the resulting information processing pipeline: the number of processing units of L, the amount of RAM (Memory) |CL|, performance as the time of a pipeline cycle $–T_0$, downtime $–T_{dt}$, and so on [5-7].

2. The number of equivalence classes of L is equal to one and the $|a_l|$ order of the class is equal to q. Then $A^{(j)}$ data processing *j* algorithm is corresponding to the q set of completely identical clusters $A^{(j)} \overset{Sk}{\Rightarrow} \{(CL)^q\}$. Thus, if the output information of each cluster CL_i, $\forall i = 1,q$ is the input information for each subsequent CL_{i+1}, t hen we have the pipeline type cluster structure with the number of clusters equal to q, and each of them functions according to the CL cluster commands (See Pic. 1).

If the input is required simultaneously for all clusters, then, appointing q processor modules for data processing we obtain the neuroprocessor realization of vector or parallel type $A^{(j)} \xrightarrow{Sw} \{CL_1\} = \langle CL_1, CL_2, ..., CL_l, ..., CL_L \rangle$, in which all the q processor modules function according to the same CL_1 cluster (See Figure 2)

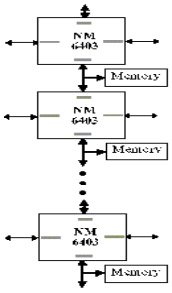

Figure 2. Cluster processing structure

Similarly we define the basic parameters of the resulting vector cluster structure of the information processing: processing module numbers L, RAM (Memory) – |CL|, pipeline productivity $–T_0$, downtime $–T_{dt}$, and so on.

3. The number of equivalence classes is equal to L and the order of l-th each class is $|a_l|$. This case is common and involves the relationship not only between clusters within a class, but also between clusters of different classes.

If the clusters within the l-th class are exchanging data logically, we have pipeline-vector data processing structure. The structures indicated are the particular cases of general matrix cluster structure of data processing shown in Figure 3.

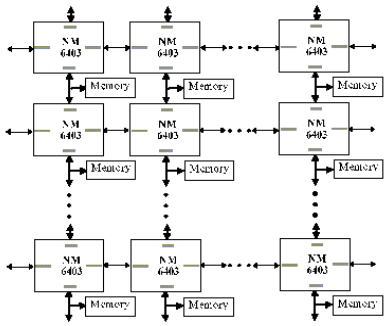

Figure 3. Matrix cluster structure of data processing

Then for each of the obtained data processing structures the basic parameters are defined.

Thus the introducing of clustering gave an opportunity to thread the processing program depending on investigated PR (j) data processing algorithms to obtain cluster processing structures: pipeline, vector [1] and new structures as pipeline-vector and vector-pipeline [5,7]. As a result, the hardware redundancy in the form of separate neuroprocessor modules is introduced.

V. The Recoverability of Neuroprocessor Systems

For calculating the recoverability the time of maintenance is used.Operational measure of maintainability of NPC equipment is the probability of equipment recovery within the specified period of time $p_B(t)$ as a probability of recovery time t_B won't exceed the period of time t specified for the equipment repair: $p_в(t) = P\{t_в<=t\}$.

If all the elements in the series operate independently, the probability of failure-free operation of the NPS circuit consisting of N elements for t period of time is calculated on the following formula:

$$P(t) = p_1(t)p_2(t)..p_i(t)..p_N(t) = \prod_{i=1}^{N} p_i(t)$$

In the given circuit connected of k elements along with the basic elements there are (k-l) reserve elements. According to it, the probability of parallel circuit failure is equal to

$$Q(t) = q_1(t)q_2(t)..q_i(t)..q_k(t) = \prod_{i=1}^{k}[1 - p_i(t)],$$

where $q_i(t)$ is a probability of i-thneuroprocessor of the parallel circuit failure.Then the probability of the faultless functioning of the parallel neuroprocessor circuit is equal to

$$P(t) = 1 - \prod_{i=1}^{k}[1 - p_i(t)]. \quad (7)$$

The ratio of the number of reserve circuits to the number of major circuits is called themultiplicityof reservation. The reliable functioning probability is found assuming that the failure of the entire system, including one primary and m backup circuits will occur after all (m+1) parallel circuits denied independently. Then the probability of $Q_{gen}(t)$ system failure is equal to:

$$Q_{gen}(t) = Q_1(t)Q_2(t)..Q_j(t)..Q_{m+1}(t) = \prod_{j=1}^{m+1} Q_j(t),$$

And a probability to faultless functioning P_{gen} (t) of the system consists:

$$P_{gen}(t) = 1 - Q_{gen}(t) = 1 - \prod_{j=1}^{m+1} Q_j(t) = 1 - \prod_{j=1}^{m+1}\left[1 - \prod_{i=1}^{N} p_i(t)\right] \quad (8)$$

where$Q_j(t)$ is a failure probability for the time t of *j-th* reserve circuit; pi(t) is a probability of faultless functioning for the time t of *j-th* element of the circuit (major or reserve).

If all (m+1) circuits in parallel connection are equally reliable then the faultless work probability is defined as

$$P_{gen}(t) = 1 - \left[1 - \prod_{i=1}^{N} p_i(t)\right]^{m+1}. \quad (9)$$

The faultless functioning probability with separate redundancy is defined assuming the NPS consists of N consists of Nserially connected elements, and the element failures in the section are independent events, so the probability is defined as

$$P_{sep}(t) = \prod_{i=1}^{N} p_{ich}(t) = \prod_{i=1}^{N}[1 - q_{ich}(t)] = \prod_{i=1}^{N}\left[1 - \prod_{j=1}^{m[1} q_{ij}(t)\right] = \prod_{i=1}^{N}\left\{1 - \prod_{j=1}^{m+1}[1 - p_{ij}(t)]\right\}, \quad (10)$$

where $p_{ich}(t)$, $q_{ich}(t)$ are a faultless functioning probability and i-th section in the connection failure correspondingly; and $p_{ij}(t)$, $q_{ij}(t)$ are a faultless functioning probability and the j-th element in i-th section failure correspondingly.If all (m+1) elements in the section are equally reliable than the probability of separate redundancy is defined as

$$P_{sep}(t) = \prod_{i=1}^{N}\{1 - [1 - p_i(t)]^{m+1}\} \quad (11)$$

The comparison of formulae (9) and (11) allows to ascertain that for all $p_i(t)$, N и m (with the exception of trivial case when $p_i(t) = 0$ и $N = 1$) the values $P_{gen}(t) < P_{sep}(t)$.

The multifunctional software-based complex "NeuroCS" based upon the given results of scientific researchwas designedfor modeling, clustering and analyzing compound, distributed and cloud computing systems based on neuroprocessors[5,9].

VI. Conclusion

The technique suggested permits to represent the system designed as a totality of clusters and implement it as one of structures obtained: vector, pipeline, vector-pipeline or pipeline-vector on the base of modern Russian microsetNM 640X or K1879 BMX. With general redundancy the failure of any element of the functioning circuit recalls the necessity of one element switching that promotes the cluster neuroprocessor systems recoverability increase.

References

[1] T.P. Baranovskaya, V.I. Loiko, M.I. Semenov, A.I. Trubilin, "Computer Systems Architecture", Moscow: Finances and Statistics, 2003, 256 p.

[2] O.P. Kuznetsov, G.M. Adelson-Velsky, "Discrete Mathematics for Engineers", Moscow: Energy, 1980, 361 p.

[3] V.K. Zlobin, V.N. Ruchkin, "Neural Networks and Neurocomputers", St. Petersburg: BHV-Petersburg, 2011, 256 p.

[4] D.V. Grigorenko, V.N. Ruchkin, "Clustering of Special Data Processing Systems" in Methods and Tools for Information Processing and Storage, Ryazan: Ryazan State Radio Engineering University EPC, 2012, 118 p.

[5] V.N. Ruchkin, "The Design and Selection of Specialized Information Processing", Moscow: Moscow State Open University, 1997, 120 p.

[6] P.E. Viksne, D.V. Fomin, V.M. Chernikov, "Single-Chip Digital Neural Variable-bit Operands" in Proceedings of Higher Education, Instrument Making, V.39, 1996, № 7, pp. 54-61.

[7] V.A. Romanchuk, v.N. Ruchkin, "Software Development for Neuroprocessor Systems Analyses" in RSREU Bulletin, V. 32, 2010, № 2, pp.61–67.