Applying the P-Medians in the Design of Modern Systems-on-Chip

Elena Suvorova, Nadezhda Matveeva, Lev Kurbanov
Saint-Petersburg State University of Aerospace Instrumentation
Saint-Petersburg, Russia
suvorova@aanet.ru, {nadezhda.matveeva, lev.kurbanov}@guap.ru

***Abstract*—In this paper we consider using p-medians searching algorithms in the design of modern systems-on-chip. This mathematical apparatus can be used for decision of some tasks that faced before developer. We consider the types of systems-on-chip, for which the p-median problem is useful. We describe different methods of calculating the P-medians. Also we examine which criteria can be used for searching P-medians. In this paper detailed describe the solving of the p-median problem for homogeneous systems-on-chip.**

I. INTRODUCTION

Zillions of all of the entities of the world from mobile devices to human beings are integrated into an interacting community providing control and actuation through internet. Nowadays, we are witnessing formation of a new technological marvel: Internet of Things(IoT), [1]. Modern smart systems [2] and devices of IoT[3] based on compact high performance computer systems which are realized as System-on-chip (SoC). They complexity scales, capturing the system's functionality has become increasingly difficult. Network-on-chip replaces bus in communication inside SoC. Count of IP-blocks inside SoC constantly grows.NoC technology is a relatively new approach to signaling that enables not only more efficient interconnects but also more efficient design and verification processes for modern SoCs [4].IoT technology and Smart Spaces technologyrequire new characteristics from hardware, [5,6]. The search for P-medians in the graph can be used for solve different tasks of designof computing systems. The task of multiple medians (P-medians) is used for solve minisum location problems.

II. USE CASES

It is very important to arrangeIP-blocks on chip with required parameters such as distance, throughput and etc.A distance affects a transmission delay.A throughput should satisfy requirements.The search for P-medians is useful for following tasks. First, if system contains several controllers of access to external memory, then information about their optimal position on chip is important for developer.Second, if system should communicate with different external interfaces such as PCIExpress, USB, HDMI and others, then information about their optimal position on chip is important for developer. Third, if specialized IP-blocks are included in SoC, then may be very interested to know how to locate CPU and these blocks inside chip with minimum distance or other characteristics.

For solving these problems, system can be presented as graph.Optimality criterion is to minimize the amount (or other functions) distances (cost) of the nodes of the graph to subset of vertices of the same graph - ϑp, called the median, or a subset of the p-median graph ($|\vartheta\ p| = P$). The graph may include one or more p-medians. The median of graph is the vertex of the graph in which the sum of the shortest distance from it to other nodes of the graph is the minimum possible.

For search P-medians in the graph is searched P points, in which the sum of the shortest distances between each of them to the rest of the nodes of the graph is the minimum possible. For other tasks also evaluated throughput and other parameters.

III. METHODS OF CALCULATING THE P-MEDIANS

Themethods of linear programming or directed tree search can be used to find the median graph. When using linear programming methods for a number of graphs can be obtained fractional coefficients. We described several methods of calculating the p-mediansbelow.

A. An approximate algorithm

A heuristic method based on vertex substitution is described by Teitz and Bart [7]. The method proceeds by choosing any p vertices at random to form the initial set S which is assumed to be an approximation to the p-median set $\overline{X_p}$. The method then tests if any vertex $x_j \epsilon\ X — S$ could replace a vertex $x_i\ \epsilon S$ as a median vertex and so produce a new set $S' = (S \cup \{x_j\})$ — $\{x_j\}$whose transmission $\sigma(S')$ is less than $\sigma(S)$. Ifthe substitution of vertex x_i by x_j is performed thus obtaining a set S' which is a better approximation to the p-median set $\overline{X_p}$. The same tests are performed on the new set S' and so on, until a set S is obtained for which no substitution of a vertex in S by another vertex in $X — \overline{S}$produces a set with transmission less than $\sigma(\overline{S})$,[8]. This final set $\overline{S}$ is taken as required approximation toX_p.The above method does not always give the best answer, [9].

B. Method "traversing a path"

An alternative approach via linear programming is given by Marsten [10], who shows that the solution of the p-median problem is an extreme point of a certain polyhedron H, and that all other p-medians for $1 \leq p \leq n$ are also extreme points of H. By using lagrange multipliers and parametric linear programming, Marsten gives a method of traversing a path among a few of the extreme points of H. This path successively generates the p-medians of the graph G in descending order of p, although certain p-medians (for some values of p) may be missed and never generated, or,

ISBN 978-952-68397-0-7 (paperback), ISBN 978-952-68397-1-4 (PDF)

conversely, extreme points of H may be generated which do not correspond to p-medians of G. Thus, although this method is both theoretically and computationally attractive, it may fail to produce the p-median of a graph for the specific value of p that may be required. In work[10] is described the case of a complete 33-vertex graph all of whose p-medians have been successfully generated for p = 33, 32, ..., 10, but whose 9-median and 8-median could not be obtained by this method.

C. Direct tree search method

The algorithm for finding the p-medians based on directtree search has the following structure. Based on any of the forms describing the graph is constructed matrix M dimension VxV. J-th column of M contains all the vertices of the graph, arranged in order of increasing distance from the x_j. The first vertex j-th column is x_j.

The search begins with the sequential scan of all nodes of the graph (for example, from the vertex of number 1 to the vertex with the highest number).x_j initially is attached to $m1_j$, then to the $m2_j$, and so on until all possibilities are tried. For each of the possible attachment x_j defines all the possible options of attachment x_{j+1} and etc., until not be executed all vertices attachment.

Each of the possible attachment of vertices forms a partial solution - a subset of vertices that can be p-median. For this subset is calculated cost.Once obtained all the possible partial solutions, are selected those, which the cost is minimal. These solutions are the p-medians of graph.

With this method, you can find all the possible p-median. However, the number of potential solutions is the number of combinations in p nodes of n nodes, where n - the number of nodes of the graph G. It follows from (1).If number of nodes in the graph and p close in meaning to the n / 2 will increase, then the number of combinations will increase rapidly and time to find the true solution will also increase. This method is best used for small graphs.

$$C_n^p = \frac{n!}{(n-p)! \cdot p!} \quad (1)$$

IV. Classification the System-on-Chip for p-median search

To search p-median System-on-Chip are classified by type of nodes.

A. Homogeneous systems

The system consists of homogeneous cores of the same area (Fig. 1).For example, this idea is used in the homoGENEous Processor arraY (GENEPY) platform for 4G applications, [11]. Also The Epiphany architecture defines a multicore, scalable, shared-memory, parallel computing fabric, [12]. It can be used in modern high performance computer systems for IoT devices.

B. Heterogeneous systems

The system consists of heterogeneous coreswith various sizes (Fig. 2), [13].A typical heterogeneous multiprocessor System-on-Chip (MPSoC)in mobile applications consists of multiple CPU cores of varying capabilities, GPU cores, DSP cores, and crypto accelerators and such cores differ widely in their physical size and their bandwidth requirements, [14].For example,Exynos is a series of ARM-based System-on-Chips (SoCs) developed and manufactured by Samsung Electronics, [15]. And Tegra is a SoC series developed by Nvidia for mobile devices, [16]. They areheterogeneous SoC.

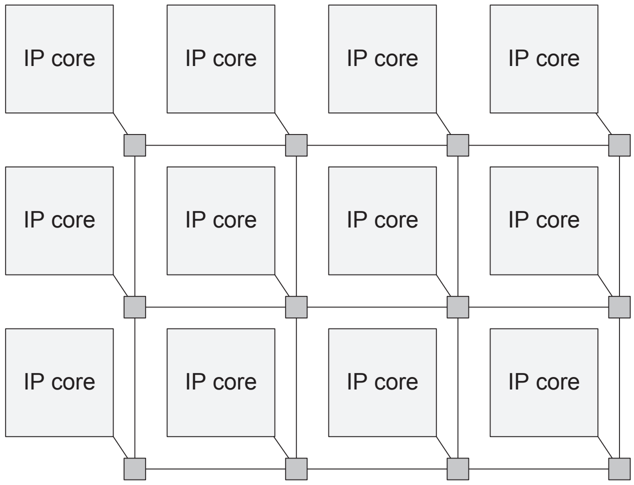

Fig. 1. Example of homogeneous system

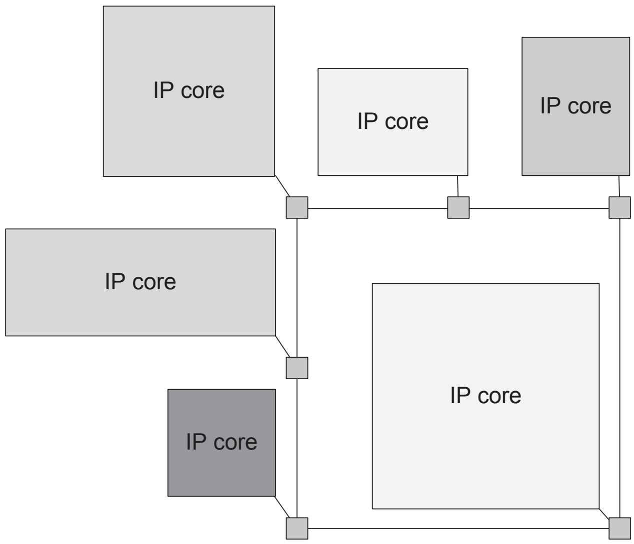

Fig. 2. Example of heterogeneous system

C. Hybrid systems

The system comprises parts of the system, which consist of a homogeneous cores of the same area, also the system includes parts, which are composed of dissimilar cores of different size (Fig. 3).

D. System with fixed position certain IP-blocks

Systems in which the location of certain cores is fixed and can't be changed (Fig. 4). For such systems it is typical that on the physical layer the length of the communication lines, which can be used for interaction with this core, is limited.

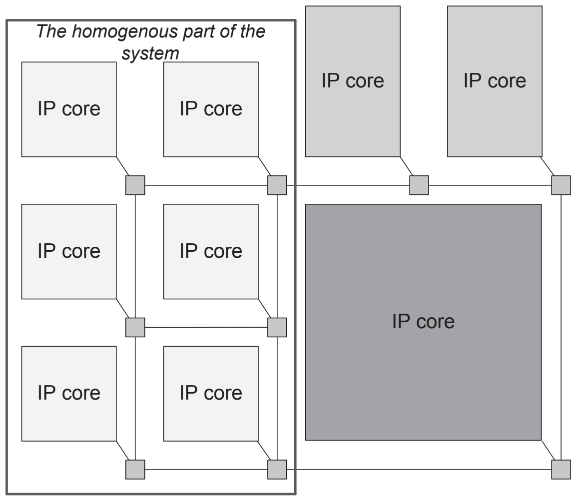

Fig. 3. Example of hybrid system

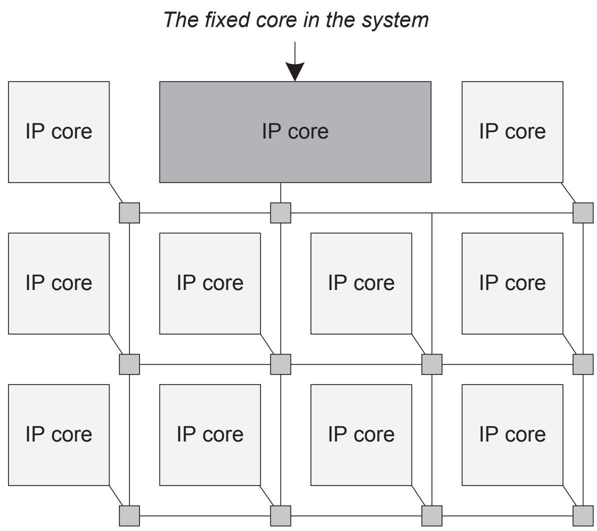

Fig. 4. Example of the system with the fixed core

For homogeneous systems IP-cores can be easily represented as a node in the graph, since all the vertices of the same type (and have the same functional and area). For such systems, the place where will be located P-median nodes is determined only by the restrictions that are defined by the user. For systems in which not all nodes can occupy any position, must take into account features of their possible locations within the chip and take this into account in solving P-median problem.

V. Search P-median in the Homogeneous Systems-on-Chip

To find the p-medians, there are the following criteria:

1) Distance.

2) Throughput.

3) Load of P-medians.

4) Support for multiple criteria.

A. Distance

In the graph are searched P nodes, which have minimum distance to all other nodes in the system. In evaluating the distance from one node to all others can be used a variety of approaches.

1) Evaluation of distances by the shortest route.

2) Evaluation of distances according to the routing rules.

The distance corresponds to the number of links that connect the nodes to each other. If necessary, the distance used as the number of nodes.Consider the following example. Fig. 5 shows the system which consists of 5 nodes. The distance from median node to the not median node will not exceed an established value d. As an example, we define p = 2, d = 1. Searching the p-medians will be implemented by a direct tree search method.

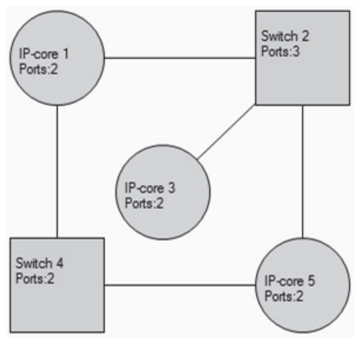

Fig. 5. Example of the system for searching p-median

For solving this problem is necessary to make the distance matrix D for a given system (Fig. 6) in which the $d_{i,j}$ - distance from the i-th node to the j-th.In this example, the evaluation of the distance is calculated according to shortest distance. Therefore it is necessary to make a matrix of shortest distances MD (Fig. 7), in which $md_{i,j}$ - shortest distance from the i-th node to the j-th node. To create this matrix, you can use the Wave algorithm.

	1	2	3	4	5
1	0	1	-	1	-
2	1	0	1	-	1
3	-	1	0	-	-
4	1	-	-	0	1
5	-	1	-	1	0

Fig. 6. The distance matrix for given system

After determined the shortest distance between the nodes of the graph, it is necessary to build the matrix M = [mhj] (Fig. 8). J-th column of M contains all the nodes of the graph, arranged in order of increasing distance from the xj. The first vertex j-th column is xj. For clarity in the indices of elements of the matrix M, we write the distance between the xj and mj.

	1	2	3	4	5
1	0	1	2	1	2
2	1	0	1	2	1
3	2	1	0	3	2
4	1	2	3	0	1
5	2	1	2	1	0

Fig. 7. The shortest distance matrix for given system

1	2	3	4	5
1_0	2_0	3_0	4_0	5_0
2_1	1_1	2_1	1_1	2_1
4_1	3_1	1_2	5_1	4_1
3_2	5_1	5_2	2_2	1_2
5_2	4_2	4_3	3_3	3_2

Fig. 8. MatrixM

In the problem statement there is a distance limit(d = 1) between the median and the not median nodes. In the matrix M elements, whose distance is greater than d, are removed from consideration. On the Fig. 9 nodes, which are markedindark gray, will not be considered in the later decision.

1	2	3	4	5
1_0	2_0	3_0	4_0	5_0
2_1	1_1	2_1	1_1	2_1
4_1	3_1	1_2	5_1	4_1
3_2	5_1	5_2	2_2	1_2
5_2	4_2	4_3	3_3	3_2

Fig. 9. MatrixM, with the application of adistance limit

The remaining elements of the matrix M will be considered in the search for p-median. xj element is a potential median node, mj,i - the potential not median node. Since p = 2, then the median set Mm must contain 2 nodes, therefore, the non median set Mn must contain N-p elements, where N - number of nodes in the graph. In other words, the union of the sets Mm and Mn must cover the total set of nodes in the graph G(2).

$$Mm \cup Mn = G \quad (2)$$

Fig. 10 shows three solutions, which are derived by directed tree search for the given parameters p = 2, d = 1. Nodes, which are encircled by a bold line, are median nodes. Dashed links, which originate from the median nodes, show the relationship between the median and non median nodes.

Fig. 11 shows the algorithm for finding p-medians with a distance limit. This algorithm allows to find only that solutions, which have minimal sum of distance between median and not median nodes.

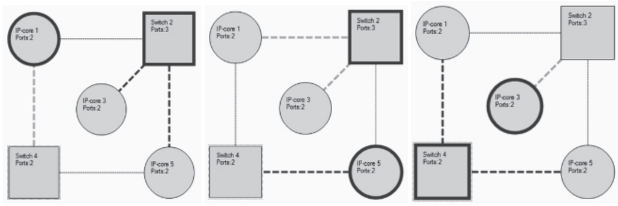

Fig. 10. Solutions of the p-median problem for given system with a distance limit, d=1

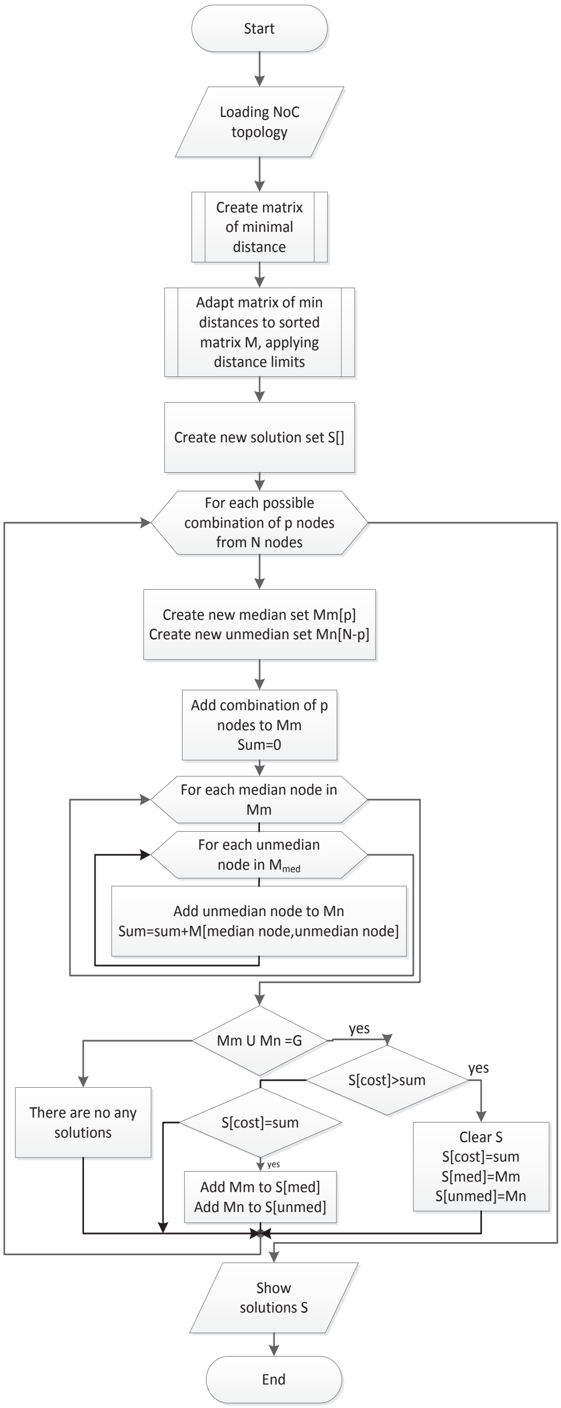

Fig. 11. Algorithm for finding p-medians with a distance limit

B. Throughput

In the graph are searched P nodes, from which data can be transmitted to all other nodes in the system with user-specified throughput. In evaluating the throughput from one node to all others can be used a variety of approaches.

1) Evaluation of throughput by the shortest route.

2) Evaluation of throughput in accordance with the rules of the routing (XY-routing, routing table).

3) Evaluation of throughput by several routes of the data transmission between a pair of nodes.

In this paper we consider the case where throughput is evaluated by the shortest route.

Throughput is equal to the minimum value of all the throughputs of the links, which are included within the shortest route. If there are several shortest routes, you need to find all the minimum values of throughputs, and choose the maximum value from it(3).

$$Thr_{i,j} = \max\{Thr_{\min}\} \quad (3)$$

Fig. 12 shows an example of such a situation. To transfer data from node 1 to node 2, there are 3 shortest routes: 1-3-2 with $Thr_{min} = 3$, 1-4-2 with $Thr_{min} = 2$, 1-5-2 with $Thr_{min} = 1$. Throughput between node 1 and node 2 in this case is equal to 3.

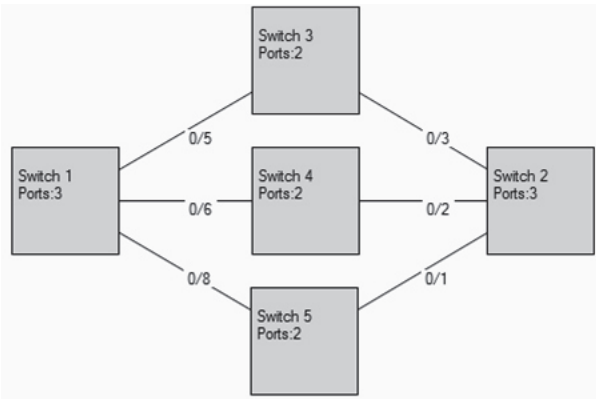

Fig. 12. Determination of the throughput in case of the several shortest routes

To solve the p-median problem with set throughput limit, generated the matrix of minimum throughputs between the nodes of the system. The structure of the system is shown in Fig. 13.

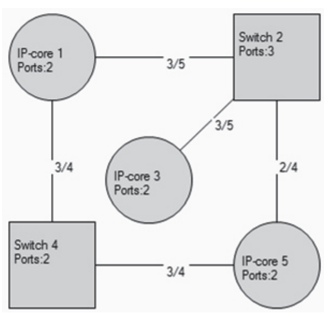

Fig. 13. Example of the system for searching p-median with a throughput limit

On the links you can see the layout of the form x/y, where x - the load of link, y - throughput of link. The loading is taken into account, when calculating throughput, so that the true value of the throughput is equal to the difference between the link's throughputand its loading. Matrix of the minimum throughputs is shown in Fig. 14.

	1	2	3	4	5
1	0	2	2	1	2
2	2	0	2	1	2
3	2	2	0	1	2
4	1	1	1	0	1
5	2	2	2	1	0

Fig. 14. Matrix of the minimum throughputs

Further, it is necessary to sort matrix of throughputs in order of increasing throughputs from a given node to the all others. Sorted matrix is shown in Fig. 15.

1	2	3	4	5
1_0	2_0	3_0	4_0	5_0
4_1	4_1	4_1	1_1	4_1
2_2	1_2	1_2	2_1	1_2
3_2	3_2	2_2	3_1	2_2
5_2	5_2	5_2	5_1	3_2

Fig. 15. Sorted matrixof the minimum throughputs

The column of matrix is a sorted list of vertices in ascending order of throughput from a given node to all other indicating the value of the minimum throughput to the corresponding vertex of the system. The column number corresponds to the given vertex.

Very often for the systems is necessary to find a set of p-median nodes which must be satisfied that the throughput from p-median node to any of the other nodes will be not less, than a user-defined value. For such problems, in sorted matrix of minimum throughputs easily eliminated those nodes, which throughput is less, than a user-defined value, and searching of P-median is implemented in an adapted matrix.

Consider a situation that presented the matrix in Fig. 15 is given by the restriction that the throughput between the nodes of the system should not be less, than 2. This means that in an adapted matrix nodes, which throughput is less than 2, will be removed. Adapted matrix is shown in Fig. 16, where the dark gray marked nodes will be removed from consideration in the later solving.

1	2	3	4	5
1_0	2_0	3_0	4_0	5_0
4_1	4_1	4_1	1_1	4_1
2_2	1_2	1_2	2_1	1_2
3_2	3_2	2_2	3_1	2_2
5_2	5_2	5_2	5_1	3_2

Fig. 16. Adapted matrix of the minimum throughputs

Fig. 17 shows four solutions, which are derived by directed tree search method for the given parameters p = 2, thr = 2.

Nodes, which are encircled by a bold line, are median nodes. It should be noted that the node 4 in this case is the median node, which has not any attached of other nodes.In the fact that after was given throughput limit (thr = 2), we "cut off" 4-th node from the rest of the nodes, because from 4-th node comes out 2 links, which the real throughput is equal 1.

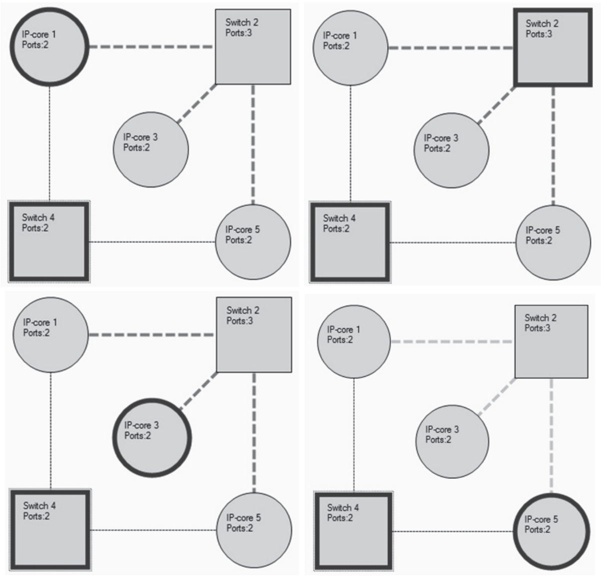

Fig. 17. Solutions of the p-median problem for given system with a throughput limit, thr=2

Search P-medians by throughput allows to find such nodes in the system which have minimum possible throughput to other nodes of the system. Also you can find nodes for which will be found the greatest value of throughput for the given throughput limitations below.

It helps the designer to identify the location of specialized blocks for which it is important to have a maximum throughput in relation to the other nodes of the system. To such blocks can be referred processors, controllers of access to external devices, etc.

Algorithm for finding the p-medians with throughput limit is the same as the algorithm for distance limit, which was described on Fig.11. But a matrix of minimum throughput is used here instead of the distance matrix. Also, in contrast to the algorithm with distance limit, the selection is done by making the maximum sum of throughput.

C. Load of p-medians

For systems in which the location of the P-medians is already set, the actual problem is the evaluation of the load, which is distributed to selected nodes. Consider the following example illustrated in Fig. 18. As the p-medians were selected nodes 1 and 4. It is known that the 2-nd node transmits data to 1-st node with an intensity of 100 Mbit/s, the 3-rd node transmits data to 1-st node with an intensity of 200 Mbit/s, the 5-th node transmits data to 4-th node with intensity of 500 Mbit/s.

The total intensity of the transmitted data through the nodes that are selected as the p-medians is the sum of all intensities: 100 + 200 + 500 = 800 Mbit/s. It is known that the number of p-medians is equal 2. The average load on each median node is 800/2 = 400 Mb/s. We can evaluate how much the actual load on each of the median nodes differs from the average load. It turns out that the load on node 1 is less than the average on 100 Mbit/s, which is (100/400) * 100% = 25%, and the load on 2-nd node more than the average on 100 Mbit/s, which is also 25%.

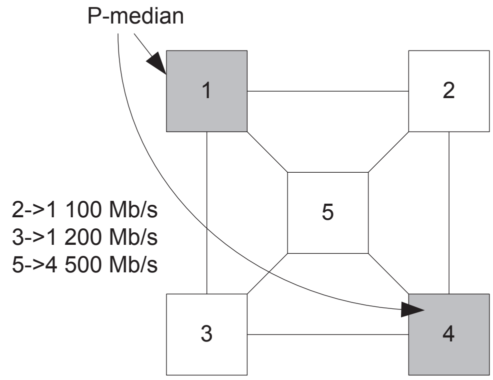

Fig. 18. Example of system for load evaluation of p-medians

With this information, the designer can evaluate the percentage ratio of the average load of median nodes and load of each selected node as the p-median. In addition, if the designer knows beforehand border of overload, he may set them as constraints and the software when searching for p-medians can evaluate each of the found solutions for compliance with the valid values and create a solution that will meet the stated requirements, if they are attainable for the system with the specified parameters. If you can't find a solution, the user will have the information that the restrictions, which he presented to the system is not feasible in the current configuration.

D. Using multiple criteria in solving P-median problem

For some computer systems it is very important to choose the location of specific nodes in the system with several requirements. To solve such problems, it is important to take into account the distance between nodes, throughput, and load of P-medians. For such problems, it is possible to use the incremental search of p-medians.

For multicriteria p-medians problem, you can search solutions in the following ways:

1) At the first step you get the set of solutions with the distance limit. After that you find solutions satisfying the constraints on throughput among solutions with distance limit.

2) At the first step you get the set of solutions with the throughput limit. After that you find solutions satisfying the constraints on distance among solutions with throughput limit.

3) Constraints on distance and troughput are taking into account during the search of the solution.

In this paper we described first method of solving multicriteria p-median problem. If you want to analyze the distance and throughput, you get all the possible solutions using only the principle of the search p-median in accordance to the criterion of the distance, then you get a solution that meets the criteria of throughput. Incremental search of p-medians is shown in Fig. 19.

As an example of solving p-median problem with several restrictions, we consider the system shown in Fig. 13. For example, we define a limit distance limit d = 1, throughput limit thr = 2, and the load limit for p-medians ld = 15%. First, we should find solutions that satisfying distance limit, then from these solutions are selected solutions that satisfying throughput limit. Further, in the obtained solutions we should to verify that the load of p-medians is within the limits set by the user. The solution of this example is shown in Fig.20.

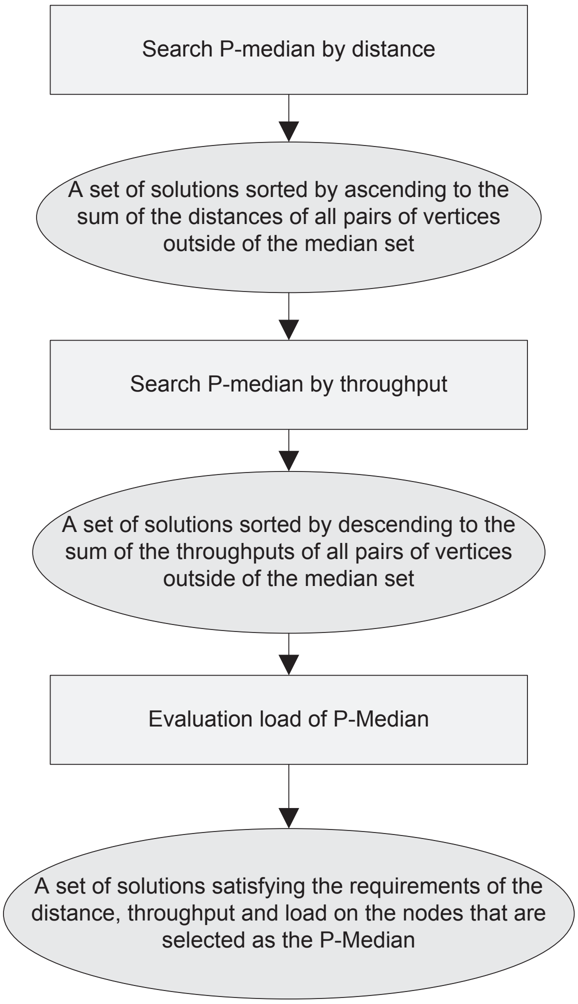

Fig. 19. Step-by-step search of p-medians with several requirements

It is known that the load of the 2-nd node is equal 4, and the load of the 4-th node is equal 3. The average value of the load at the p-medians is equal (4 + 3) / 2 = 3.5. Loads of both nodes are different from the average load of 0.5. We can evaluate how much the actual load on each of the median nodes differs from the average load: (0.5 / 3.5) * 100% = 14.3%. This means that the 2-nd node is overloaded by 14.3%, and the 4-th node underloaded by 14.3%. Since the load limit was set to 15%, the solution satisfies the requirements.

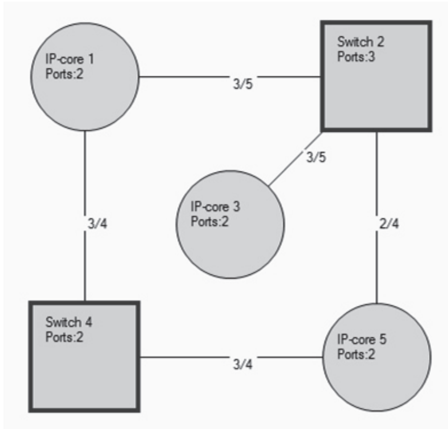

Fig. 20. Solution of the p-median problem for given system with following requirements: d=1, thr =2, ld =15%

VI. Conclusion

Solving the p-median problem helps to find nodes, with a minimum distance to other nodes and a maximum throughput for data transmission. Also we can set another criteria and requirements in solving p-median problem, which are important in the system-on-chip design.

We can get information about placement controllers for the external memory using p-medians algorithm. Access to external memory often is the bottleneck. It is usefull to allocate controlles with minimum distances to all system-on-chip nodes.

The most reliable way to solve p-median problem is using the "directed tree search" method, but we must remember, that this algorithm performs an exhaustive search of all possible solutions. And for large graphs this method is time consuming.In this paper we considered p-median problem with some criteria such as distance, throughput and load of the medians nodes. We describe how to use several parameters for finding p-median. The use of these criteria in solving the p-median problem allows optimally distribute the nodes in the system. Optimal allocation of nodes helps to improve system performance and reliability.

Acknowledgment

The research leading to these results has received financial support from the Ministry of Education and Science of the Russian Federation according to the base part of the state funding assignment in 2015, project № 1810.

References

[1] N. Bari, G. Mani, S. Berkovich, "Internet of Things as a Methodological Concept", *Conference on Computing for Geospatial Research and Application (COM.Geo)*, 2013, pp. 48 – 55.
[2] M. Glesner, F. Philipp, "Embedded systems design for smart system integration", *IEEE Computer Society Annual Symposium*, Aug. 2013, pp. 32 – 33.
[3] C. Subhas, Internet of Things Challenges and Opportunities, *Springer International Publishing*, 2014.
[4] Konstantinos Tatas, Kostas Siozios, Dimitrios Soudris, Axel Jantsch, Designing 2D and 3D Network-on-Chip Architectures, Springer, 2014.
[5] S. Balandin, H. Waris, "Key properties in the development of smart spaces," *in Proc. 5th Int'l Conf. Universal Access in Human-Computer Interaction (UAHCI '09).* Part II: Intelligent and

Ubiquitous Interaction Environments, *LNCS 5615*, C. Stephanidis, Ed. Springer-Verlag, 2009, pp. 3–12.

[6] Jayavardhana Gubbia, Rajkumar Buyyab, Slaven Marusic, Marimuthu Palaniswami. “Internet of Things (IoT): A vision, architectural elements, and future directions”, *Future Generation Computer Systems.* Volume 29, Issue 7, September 2013, pp. 1645 1660.

[7] A. Yaspan, “On finding a maximal assignment”, *Ops. Res*., 14, 1966, pp. 646-651.

[8] N. Christofides, Graph theory. An algorithmic approach, Academic press inc., 1978.

[9] F. Glover, “Maximum matching in a convex bipartite graph”, *Nav. Res. Log. Quart*., 14, 1967, pp. 313-316.

[10] H. W. Kuhn, “The Hungarian method for the assignment problem”, *50 Years of Integer Programming 1958-2008*, 2009, pp. 29-47.

[11] Camille Jalier, Didier Lattard and Ahmed Amine Jerraya, Gilles Sassatelli, Pascal Benoit and Lionel Torres, "Heterogeneous vs Homogeneous MPSoC Approaches for a Mobile LTE Modem", *Design, Automation & Test in Europe Conference & Exhibition (DATE),* 2010, pp. 184 – 189.

[12] Epiphany Architecture Reference, Web: http://www.adapteva.com/docs/epiphany_arch_ref.pdf

[13] Y. Sheynin, E. Suvorova, “Placement of different type nodes in a Network-on-chip graph", Web: http://www.design-reuse.com/articles/18925/network-on-chip-graph.html.

[14] B.Phanibhushana, S. Kundu, "Network-on-Chip Design for Heterogeneous Multiprocessor System-on-Chip", *Symposium on VLSI (ISVLSI)*, 2014, pp. 486 491.

[15] Exynos System-on-Chip family, Web: http://www.samsung.com/semiconductor/products/exynos-solution/application-processor/

[16] Tegra System-on-Chip, Web: http://www.nvidia.com/object/tegra-4-processor.html