DE GRUYTER OPEN

Print ISSN: 1312-2622; Online ISSN: 2367-5357
DOI: 10.1515/itc-2015-0013

Load Optimization in a Grid Structure for Parallel Simulations of the Throughput of a Packet Switch Node

T. Tashev, V. Monov, R. Tasheva

Key Words: *Computer technologies; simulations; optimization; grid structures; crossbar switch.*

Abstract. *In the present paper we employ the grid structure of IICT-BAS for parallel computer simulations of the throughput of a crossbar switch node. In our simulations we use PIM-algorithm for non-conflict scheduling in a crossbar node with hotspot load traffic. The obtained simulation results enable us to propose a procedure for optimizing the load of a grid structure in order to minimize the overall time of performance*

1. Introduction

Modern digital information systems are built according to the principle of exchange of discrete portions of information called packets. Communication nodes in these systems are called router and switch. A crossbar switch node routes traffic from the input to output lines. The randomly incoming traffic must be controlled and scheduled to eliminate conflict at the crossbar. The goal of the traffic-scheduling for the crossbar switches is to maximize the throughput of packet through a switch and to minimize packet blocking probability and packet waiting time (Kang, et all., 2013). This is assured by the algorithm for calculation of non-conflict schedule which is running in the control unit of the node (Scheduler – *figure 1*) (Csaszar, et all, 2007).

The problem of calculating of a non-conflict schedule is NP-complete (Chen, et all, 1990). Increasing data volumes (Atanasova, 2010) and increasing the speed of transmission lines of communication require new, more efficient algorithms for the calculation of the conflict-free schedule. The efficiency of these algorithms can be verified with a formal or simulation tools.

The efficiency of the algorithms for switches in the first place can be evaluated by using bandwidth output channels (throughput) (Kolchakov, 2010, Kang, et all., 2013). The incoming traffic may be uniform or non-uniform. The study of an algorithm's throughput begins with modeling of the switch throughput under uniform load traffic (uniform i.i.d. Bernoulli traffic). The next step is to investigate the algorithm's throughput under non-uniform traffic (Chang, 2012).

In the previous paper (Tashev, et all, 2013) we proposed a numerical procedure for calculating the upper boundary of the throughput for crossbar switch node. If throughput of a crossbar node increases to a certain limit (monotonically), the procedure provides one unique solution. We performed simulations for a specific algorithm for non-conflict schedule, a model for incoming traffic and a load intensity. Our modeling of the throughput utilizes PIM-algorithm (Anderson, et all, 1993), Chao-model for hotspot load traffic (Chao-Lin, Yu, et all, 2007) and ρ=100% load intensity of each input (i.i.d. Bernoulli) (Tashev, 2011). In this case the throughput of the crossbar node increases monotonically to a certain limit. In case of a size n of the switch field: $n \in [3, 70]$ and the scale i of input buffers loading for Chao-model: $Chao_i$, $i \in [1,10]$ the results are shown in *figure 2* (Tashev, 2011).

To increase the accuracy of the numerical procedure

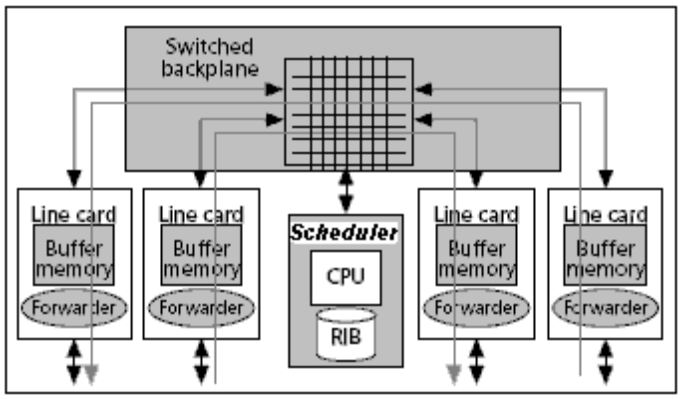

Figure 1. Third generation crossbar switch structure

the simulations should be performed for a large size of the switch field and in a large scale of input buffers. Such simulations and the necessary computations are commonly carried out by using grid-computer structures.

In this case we should solve the problem for optimal loading resources of the grid structure for parallel simulations.

In this paper we will research in what conditions for parallel simulations the values of the time load of processors of grid structure will be equal. In particular, for scale i=const, how to divide the interval [*n1*, *n2*] modeling the size of switch field *n* by subintervals for this aim.

2. Conditions for the Simulation

Our simulation of the throughput utilizes a PIM-algorithm specified by the apparatus of Generalized nets (Atanassov, 1997). The utilized GN-model for the PIM-algorithm is specified in (Tashev, Monov, 2012). The transition from a GN-model to executive program is performed as in (Tashev, Vorobiov, 2007) using the program package VFort (Vabushkevich, 2009). The source code has been executed by means of Win XP SP2 and IBM PC compatible computer with Intel Pentium IV 3 GHz and 2 GB RAM.

We utilize a family of patterns of a traffic matrix ***T*** for a non-uniform traffic simulation based on the hotspot (Chao) model (Tashev, 2011). This model is given by: $\lambda_{ii} = 0{,}5\rho$ for $i = j$ and $\lambda_{ij} = 0{,}5\rho/(n\text{-}1)$ otherwise, $i, j \in 1, \ldots, n$, where ρ is the load intensity (i.i.d. Bermoulli) of each input (Chao-Lin, Yu, et all, 2007).

The times of simulations for Chao_i, $i \in [1, 10]$ and $n \in [3, 70]$ are shown in *figure 3* (the average values of time by 10 000 simulations for each n). In the figures below, Chao_i is denoted as C-i for $i = 1, 2, \ldots$ The time of simulation depends linearly by scale *i* of Chao-model and depends approximately on the third power of the size *n* of commutation field.

Let the source code has been tested on Vfort and then compiled by means of the grid-structure BG01-IPP of the Institute IICT – Bulgarian Academy of Sciences (http://www.grid.bas.bg).

The resulting executive code is performed on the grid-structure. The results for throughput and time executions of PIM-algorithm with Chao_1 are shown in *figure 4* (10^4 simulations for each *n*, sequentially).

The results for throughput are identical with IBM PC. The results for time execution are similar – approximately on the third power. We can proceed to large-scale simulations using a grid structure. A main restriction is the time for execution. How to divide the interval [*n1*, *n2*] by subintervals for parallel execution each, for our aim?

3. Calculating of Subintervals

Let's assume that the variable *n* is continuous and designate it with *x*. Then the time execution is of the type $y=a.x^3$, where *a* is a constant depending on the hardware used. In this case the area S under the curve $y=a.x^3$ is equal to the general time of simulation. Therefore, if we divide this area into two equal parts, we will receive the required subintervals (along *x*) for the case of two parallel executed tasks. We may divide the area into three subintervals, etc. This leads to the following calculations:

$$S(x) = \int_A^B ax^3 dx \rightarrow S(x) = \frac{a}{4}x^4 \perp_A^B$$

For our case A=0, B>A. We will designate the time for simulation from dimension A=0=n1 to dimension B (x_1=B=n2 is given) as

(1) $S_1(x_1) = (1/4).a.x_1^4$

The time for simulation equal to the half of the latter will be:

(2) $S_{1/2}(x_{1/2}) = (1/4).a.x_{1/2}^4$

If $S_{1/2}$ is calculated as half of S_1

(3) $S_{1/2}(x_{1/2}) = (1/2).(1/4).a.x_1^4$

We can write

(4) $(1/4).a.x_{1/2}^4 = (1/2).(1/4).a.x_1^4$

And then

(5) $x_{1/2}^4 = (1/2).x_1^4$,

therefore

(6) $x_{1/2}=(1/2)^{1/4}.x_1$

Analogically when we divide into three equal parts

(7) $x_{1/3}=(1/3)^{1/4}.x_1$, $x_{2/3}=(2/3)^{1/4}.x_1$

Analogically four equal parts, etc.

(8) $x_{1/4}=(1/4)^{1/4}.x_1$
$x_{2/4}=(2/4)^{1/4}.x_1$
$x_{3/4}=(3/4)^{1/4}.x_1$

Of course, the boundary of last interval is

(9) $x_{4/4}=(4/4)^{1/4}.x_1 = x_1$

The results $x_{i/k}$ up to 8 equal parts are given in *the table.*

Numbers of boundary of subintervals

Tasks *k*	**2**	**4**	**8**
i **Subintervals 1/8**			0,594604
2/8		0,707107	0,707107
3/8			0,782542
4/8	0,840896	0,840896	0,840896
5/8			0,889139
6/8		0,930605	0,930605
7/8			0,967168
8/8	1	1	1

Using the formula $x_{i/k} = (i/k)^{1/4}.x_1$ we can calculate the boundaries for each *i*-subinterval in a given number *k* of subintervals.

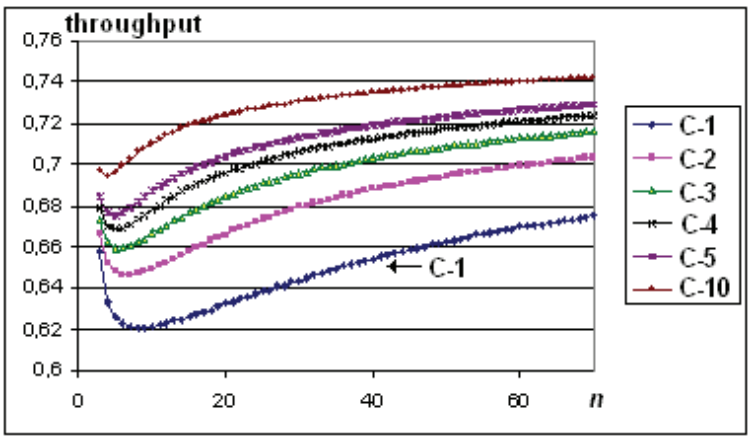

Figure 2. Throughput of PIM-algorithm with Chao (C-1, 2, 3, 4, 5, 10) traffic (Tashev, 2011)

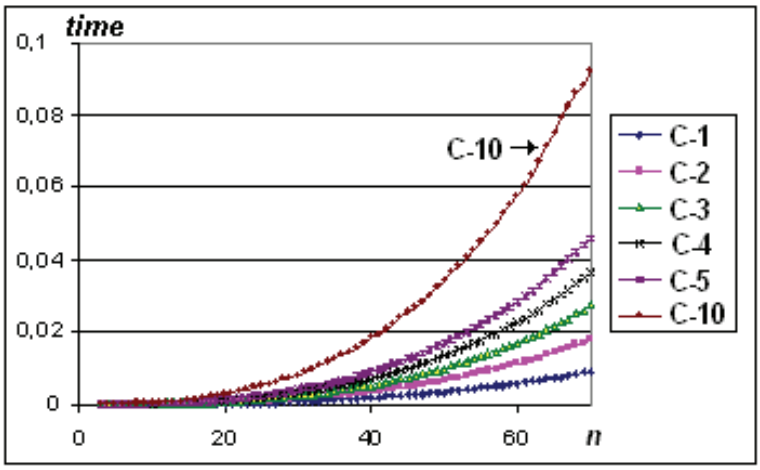

Figure 3. Simulation time of PIM-algorithm with Chao (C-1, 2, 3, 4, 5, 10) traffic (Tashev, 2011)

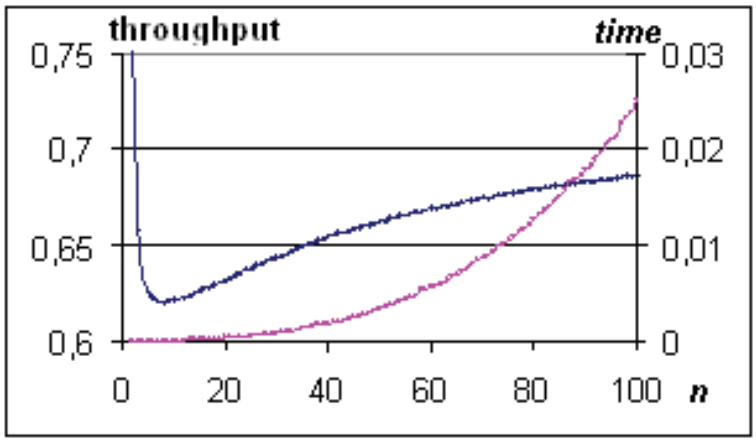

Figure 4. Throughput and time of PIM-algorithm with $Chao_1$ traffic by grid structure

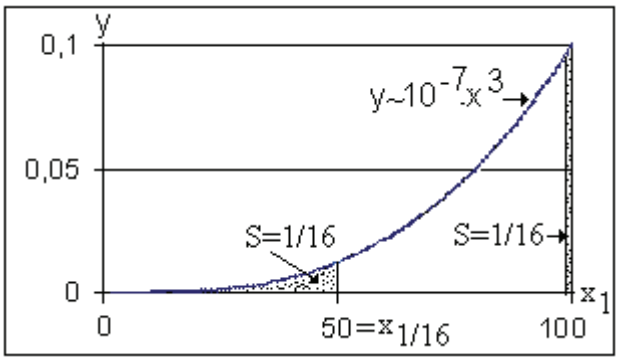

Figure 5. Analog approximation of half-time execution

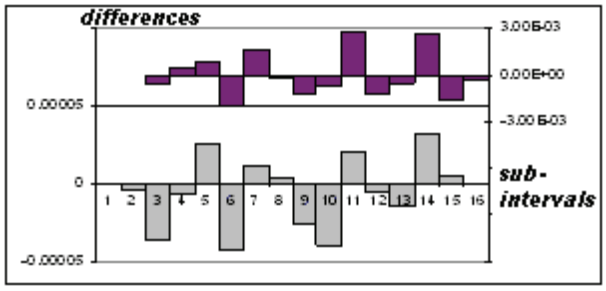

Figure 6. Differences between different time for S (top – analog four digits) and between different *x*

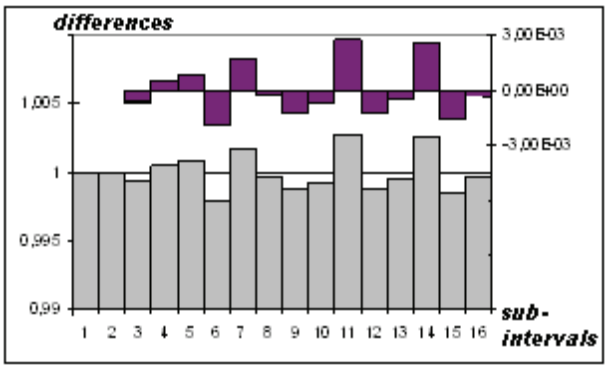

Figure 7. Differences – normalized by average for $S_{1/16}$ (down)

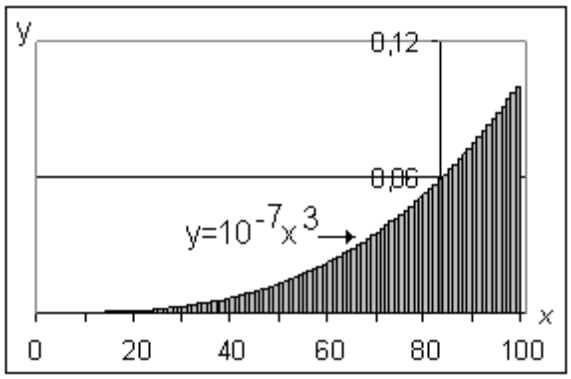

Figure 8. Discrete approximation of half-time execution

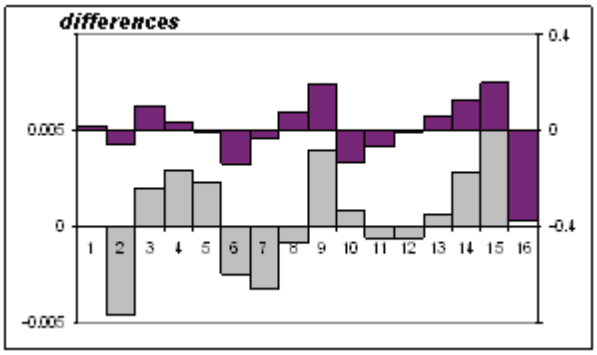

Figure 9. Differences between the times for $S_{1/16}$ (top) and x (down – discrete two digits)

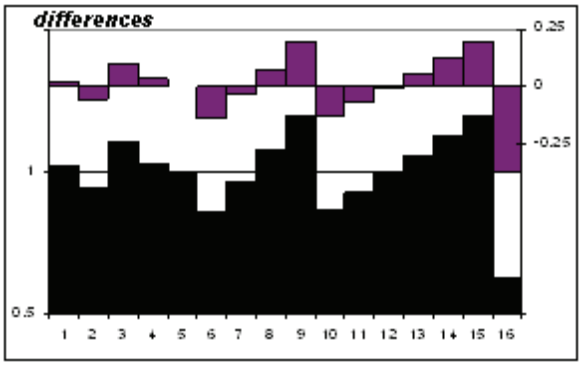

Figure 10. Differences between different S (down – normalized)

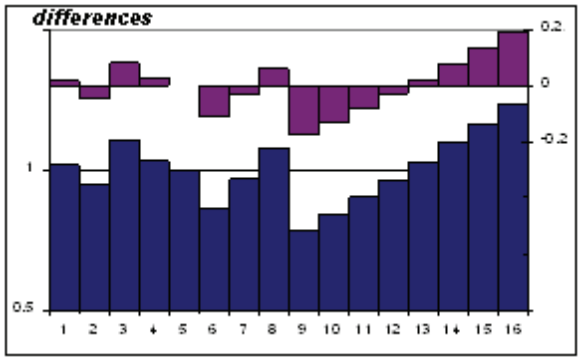

Figure 11. Differences between diff. S for x shift left by 1

4. Accuracy of the Load Optimization with Subintervals

Let's assume that the variable x is analog (*figure 5*). We calculate the differences between different x and different $S_{1/16}$ (x) ($y=x^3$) time executions – for 16 subintervals. We build a line for differences as shown in *figure 6* (different x for 16 tasks – down).

We have two lines: $\Delta_1 = x$ (accuracy of four significant digits – *the table*) – x (exact value) (down), $\Delta_2 = y$ (accuracy of four significant digits for x – *the table*) – y (exact value of y) (up). Differences between different x are smaller than 0.005%. Time execution, normalized for $S_{1/16}$ is shown in *figure 7*. Over a continuous x the time differences are smaller than 0.3% relative to the theoretical time of $S_{1/16}$. The results for subintervals (x is analog) is very good.

Let's assume that the variable x is discrete (*figure 8*). For $n2$=100, a=1.10^{-7}, 2 subintervals, the error for continuous x is 2% (good). We choose the boundary of subintervals to be the nearest integer.

For $n2$=100, a=1.10^{-7}, 16 subintervals, the differences between different S for discrete x reaches 40% (bad – *figure 9*). The differences between different discrete x are smaller than 0.5%.

We choose the boundary of subintervals to be the nearest integer. In *figure 10* are shown the times for S normalized by average for $S_{1/16}$.

Can we reduce the difference of 40%? We may choose for boundaries of subintervals the nearest integer x "plus" shift to the left by 1 (minus 1). The results of this choice are shown on *figure 11* ($n2$=100, a=1.10^{-7}, 16 subintervals). The difference between different S reaches 25%. That's the

best we can get, as the following shifts do not improve the situation.

We expect that this is possible when you select the interval type [1/2.*n2*, *n2*], *n2e* ≥ 200. This requires new research. Now we have to check the real situation: the execution time on the grid structure.

5. Grid Simulation Using the Subintervals

The results of the computational experiments for a given patterns for hotspot model of incoming traffic and model of PIM-algorithm, by means of the grid-structure BG01-IPP (http://www.grid.bas.bg).

For $Chao_{10}$, *n2*=100 ($y \approx 3.10^{-7}x^{2,9}$), four subintervals, the differences between different S for discrete *x* reaches 3.5% (*figure 12*).

For $Chao_5$, *n2*=200 ($y \approx 4.10^{-8}x^{3,2}$), 4 subintervals, the differences between different S for discrete *x* reaches 43% (*figure 13*).

For $Chao_4$, *n2*=200 ($y \approx 3.10^{-8}x^{3,1}$), the differences between different S for discrete *x* reaches 43% (*figure 14*).

We observed large as well as small differences in the execution time of simulations, when using patterns $Chao_3$ and $Chao_8$ (four tasks). Above we have shown the extreme cases.

Why is that so? Such differences are expected in 16 tasks. Why in the last two cases we have such differences? Before the simulation for $Chao_{10}$ (*n2*=100), the grid structure has three tasks (and two tasks expect of the resource) as shown in *figure 15*. This is a low load on the grid. We run four tasks and load becomes seven tasks (four ours and three others, as shown in *figure 16*). The tasks are in various blades (wn24, wn05, wn10 for other tasks and wn01 for our tasks).

During the simulation for $Chao_5$ and $Chao_4$, the grid structure has performed more than 20 tasks (our tasks are four). Loading of resources has increased.

From this we can conclude that the constant *a* has different values depending on the grid load. We must keep this in mind. The building of *the table* is just the beginning of the calculation optimization. For this purpose, we have to optimize our simulations using a practical procedure.

6. The Procedure for Calculating the Subintervals

We give an informal description of the procedure for determining the subintervals for simulations the throughput of the switch.

Step 1. We have a program for simulations that is already tested. We determine the interval [1,n2] for one hour sim. with one task. We run serial simulation.

Step 2. We check the coefficients in $y=a.x^3$. We run 5-hour serial simulation (~interval [2, 1.5 *n2*]) by two tasks.

Step 3. We check the coefficients in $y=a.x^3$. We run

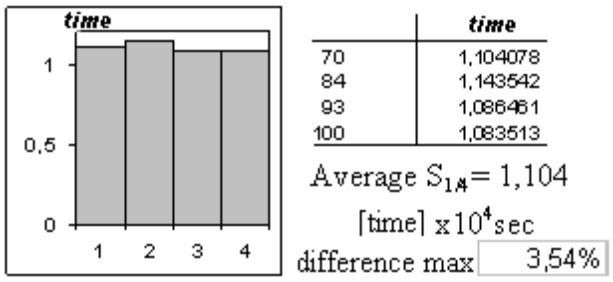

Figure 12. Time S for $Chao_{10}$, n2=100 with four tasks

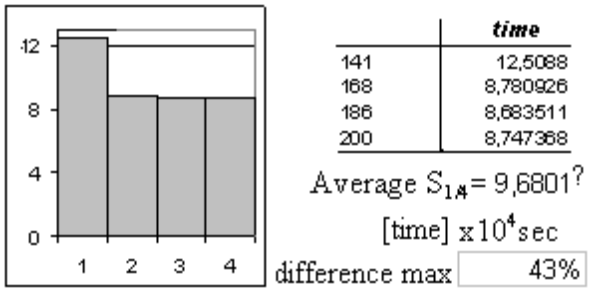

Figure 13. Time S for $Chao_5$, n2=200 with four tasks

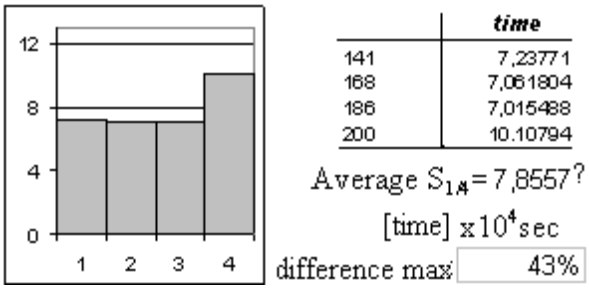

Figure 14. Time S for $Chao_4$, n2=200 with four tasks

16-hour serial simulation (~interval [2, 2 *n2*]) by four tasks.

Step 4. We determine the differences of time for the tasks.

Step 5. We choose the interval simulation according to our authorized resources (CPU and time), starting by five tasks, $^1/_2$ autorized time by using expansion the table.

Step 6. We run large-scale simulations with *k* subintervals.

Step 7. We obtain the results. If the results satisfy us, we stop. If the results do not satisfy us, we repeat from **Step 4**.

Approbation of the procedure is done with the patterns for hotspot model of load traffic and model of PIM-algorithm. For $Chao_{10}$, *n2*=200 ($y \approx 5.10^{-7}x^{2,8}$), four subintervals, the differences between different S for discrete *x* reaches 2.1% (*figure 17*).

We got uniformity in load – 2.1% max differences is a better value than previous experiments. The procedure is applicable to simulations of throughput of other algorithms and different patterns of load traffic (with the complexity of calculation of $O(n^2)$ to $O(n^5)$).

```
                                        Req'd  Req'd   Elap
Jobname           SessID NDS   TSK Memory Time  S Time
----------------- ------ ----- --- ------ ----- - -----
transform16         7906     16 256    --  24:00 R 05:32
n24+wn24+wn24+wn24+wn24+wn24+wn24+wn24

transform16          --      16 256    --  24:00 Q   --

transform8           --       8 128    --  24:00 Q   --

TiB.sh              18668     1   8     --  240:0 R 04:00
n05
TiC.sh              13240     1   8     --  240:0 R 03:58
n10

[tashotr@wn02 ~]$ qsub -q cm ./Chao10-02-70-10k---rh2.job
```

Figure 15. Grid load before our simulations: three tasks

```
                                        Req'd  Req'd   Elap
   Jobname           SessID NDS   TSK Memory Time  S Time
-- ----------------- ------ ----- --- ------ ----- - -----
   transform16         7906     16 256    --  24:00 R 06:04
l+wn24+wn24+wn24+wn24+wn24+wn24+wn24+wn24

   transform16          --      16 256    --  24:00 Q   --

   transform8           --       8 128    --  24:00 Q   --

:h TiB.sh              18668     1   8     --  240:0 R 04:32
5+wn05
:h TiC.sh              13240     1   8     --  240:0 R 04:31
)+wn10
   Chao10-02-70-10k    7495     1   1     --  72:00 R 00:30

   Chao10-71-84-10k    7520     1   1     --  72:00 R 00:30

   Chao10-85-93-10k    7909     1   1     --  72:00 R 00:29

   Chao10-94-100-10    8916     1   1     --  72:00 R 00:29

                                                  our task

 Run   Hld   Wat   Trn   Ext Status
 ---   ---   ---   ---   --- ----------
   7     0     0     0     0 Active
```

Figure 16. Grid load during our simulations ($Chao_{10}$)

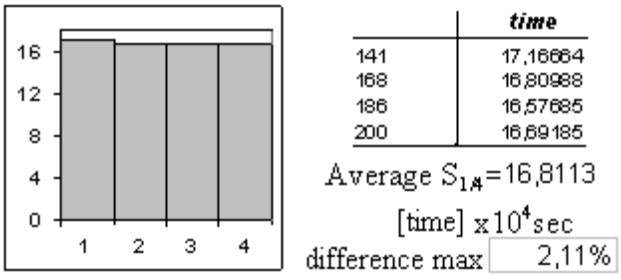

Figure 17. Time S for $Chao_{10}$, n2=200 with four tasks

7. Conclusions

In the present paper the possibility is investigated for "a priory" distribution of the resources of a grid-structure at parallel simulation of a task with calculation complexity $O(n^3)$.

An equation is derived for dividing the interval [1,x] of the argument *n* into finite number of subintervals which determine the processor tasks of the grid-structure. The aim is for the separated tasks to finish simultaneously – i.e. the loading to be uniform. Theoretically the suggested formula offers less than 2% time lag difference in the task execution.

During the fulfilled simulations on the grid-structure BG01-IPP (Institute of information and communication technologies – Bulgarian Academy of Sciences) are observed bigger differences in the time for parallel execution of four task. These computer simulations investigated the throughput of a crossbar switch node with PIM-algorithm and hotspot load traffic.

A procedure is suggested for decreasing of the differences. When testing procedure we obtained 2.1% time lag difference in the task execution. This procedure is useful for the initial period of a series of simulations.

Acknowledgements

The research work reported in the paper is supported by the project AComIn "Advanced Computing for Innovation", grant 316087, funded by the FP7 Capacity Programme (Research Potential of Convergence Regions).

References

1. Kang, K., K. Park, L. Sha, Q. Wang. Design of a Crossbar VOQ Real-Time Switch with Clock-Driven Scheduling for a Guaranteed Delay Bound. – *Real-Time Systems,* 49, January 2013, Issue 1, 117-135.
2. Csaszar, A., G. Enyedi, G. Retvari, M. Hidell, P. Sjodin. Converging the Evolution of Router Architecture and IP Networks. – *IEEE Network,* 4, 2007, 8-14.
3. Chen, T., J. Mavor, Ph. Denyer, D. Renshaw. Traffic Routing Algorithm for Serial Superchip System Customisation. – *IEE Pro. – E.part,* 137, Jan 1990, No. 1, 65-73.
4. Atanasova, T. E-Home – Data Aggregating for Increasing Energy Efficiency. Cooperative Science Workshop. Modeling and Control of Information Processes. Proceedings, High School of Telecommunication and Posts, Sofia, Bulgaria, 69-75 (in Bulgarian).
5. Kolchakov, K. An Approach for Performance Improvement of Class of Algorithms for Synthesis of Non-conflict Schedule in the Switch Nodes. Proc. of the 11th Int. Conf. CompSysTech'10, 17-18 June 2010, Sofia, Bulgaria. ACM Press, ICPS, 471, 2010, 235-239.
6. Chang, H., G. Qu, S. Zheng. Performance of CTC(N) Switch under Various Traffic Models. Springer, Lecture Notes in Electrical Engineering. http://www.springerlink.com/content/1876-1100/, 126, 2012, 785-793.
7. Tashev, T., N. Bakanova, R. Tasheva. Upper Bound Research of the Throughput of a Communication Node with Hotspot Load Traffic. – *International Journal Information Technologies & Knowledge*, 7, 2013, No. 2, 182-189 (in Russian).
8. Anderson, T., S. Owicki, J. Saxe and C. Thacker. High Speed Switch Scheduling for Local Area Networks. – *ACM Trans. Comput. Syst.,* 11, Nov. 1993, No. 4, 319-352.
9. Chao-Lin, Yu, C.-S. Chang, D.-S. Lee. CR Switch: A Load-Balanced Switch with Contention and Reservation. – *IEEE/ACM Transactions on Networking,* 17, October 2007, No. 5, 1659–1671.
10. Tashev, T. Modelling Throughput Crossbar Switch Node with Nonuniform Load Traffic. Proc. of the International Conference DCCN-2011, 26-28 October 2011, Moscow, Russia. Moscow, R&D Company "Information and Networking Technologies", 96-102 (in Russian).
11. Atanassov, K. Generalized Nets and System Theory. Acad. Press "Prof. Marin Drinov", Sofia, 1997, Bulgaria.
12. Tashev, T., V. Monov. Modeling of the Hotspot Load Traffic for Crossbar Switch Node by Means of Generalized Nets. Proc. of Intelligent Systems (IS), 6th IEEE Int. Conference, 6-8 Sept. 2012, Sofia, Bulgaria, 187-191.
13. Tashev, T., V. Vorobiov. Generalized Net Model for Non-Conflict Switch in Communication Node. Proc. of Workshop DCCN'2007, 10-12 September 2007, Moscow, 158-163.
14. Vabushkevich, P. VFort. http://www. nomoz.org/site/629615/vfort.html (last checked Apil 14, 2015), 2009.

Manuscriot received on 08.05.2015

Tasho Tashev *graduates the Leningrad Institute of Cinema Engineering (now St. Petersburg Institute of Cinema and TV, Russia) in 1984 as a Radioelectronics Engineer. He received M. Sc. degree in Applied Mathematics from Center of Applied Mathematics, Technical University – Sofia, Bulgaria. The current position is assistant professor in Institute of Information and Communication Technologies – Bulgarian Academy of Sciences. His field of interest includes mathematical modeling, distributed information systems design, methods and tools for network models.*

Contacts:
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences, Acad. G. Bonchev St., bl. 2,
Sofia, Bulgaria
e-mail: ttashev@iit.bas.bg

Assoc. Prof. Dr ***Vladimir Monov*** *received his M.Sc. degree in Electrical Engineering with qualification in Control Engineering and Automation from Technical University, Sofia and Ph.D. degree in Technical Sciences, Bulgarian Academy of Sciences. Since 2010, Dr V. Monov has been the Head of Modeling and Optimization Department at the Institute of Information and Communication Technologies, Bulgarian Academy of Sciences. His professional interests and research activity are in the areas of systems and control theory, information and communication systems, business management systems, applied mathematics and operations research.*

Contacts:
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences, Acad. G. Bonchev St., bl. 2,
Sofia, Bulgaria
e-mail: vmonov@iit.bas.bg

Radostina Tasheva *graduates the Sofia University, Bulgaria in 1984 as a Physicist. She received Ph.D. degree in Astronomy from Department of Astronomy, Faculty of*
Physics, Sofia University in 1991. From 1997 she is assistant professor in Department of Applied Physics, Technical University of Sofia, Bulgaria. Her field of interest includes processes of star formation, galaxies with active nuclei, computer modeling.

Contacts:
Technical University of Sofia
Department of Applied Physics
8 Kliment Ohridski Blvd., bl. 10,
1000 Sofia, Bulgaria
e-mail: rpt@tu-sofia.bg