

Peter Hanuliak - Michal Hanuliak\*

## HYBRID ANALYTICAL PERFORMANCE MODELS OF PARALLEL COMPUTERS

*The paper is a follow up to the performance modelling of dominant parallel computers (NOW, GRID) with analytical hybrid models (mixed) based on combinations of a more precisely developed corrected standard analytical model (M/M/1 and M/M/m) and on an improved analytical model (M/D/1 and M/D/m). Firstly, the paper briefly describes the steps of development of parallel computer architectures and then it summarises the basic concepts of performance evaluation. In the case of using SMP parallel system as node computer, the suggested models use for all the activities of node combinations of M/M/m and M/D/1 or M/D/m and M/M/1 different (hybrid) queuing theory systems. The achieved results of both developed analytical models have been compared with the results of other alternative evaluation method, based on simulation, to verify the accuracy of developed analytical models. All developed and presented analytical models could be used for various real ranges of input parameters which influence the final performance of analysed connected computing nodes for the practice.*

**Keywords:** Parallel computer, computing node, network of workstation (NOW), Grid, analytical modelling, queuing theory, performance evaluation, queuing theory system, simulation.

List of used symbols and short cuts

- M/M/1, M/M/m, M/D/1, M/D/m - queuing theory systems
- Grid - high integration of computer based networks (NOW, LAN, WAN etc.)
- FIFO - queue for service on a first-come first-served basis
- HPC - high performance computing
- LAN - local area network
- NOW - network of workstations
- $p$  - number of computing nodes of parallel computer
- PC - personal computer
- SMP - symmetrical multiprocessor system
- WAN - wide area network
- ws - workstation as a computing node of NOW and Grid.

Queuing theory symbols

- $\lambda$  - arrival rate at entrance to a queue
- $m$  - number of servers in the queuing theory system
- $\rho$  - traffic intensity (dimensionless coefficient of utilisation)
- $\rho \equiv R = \lambda \cdot E(t_s)$  - traffic intensity ( $0 < \rho < 1$ ) of one service
- $\rho = \lambda \cdot E(t_s) / m$  - traffic intensity of  $m$  servicing servers
- $q$  - random variable for the number of customers in a system at steady state
- $w$  - random variable for the number of customers in a queue at steady state
- $\gamma$  - sum of individual extern intensities  $\gamma_i$  to the  $i$ -th NOW module in the Grid
- $N$  - number of used NOW modules
- $u_i$  - number of used communication channel in  $i$ -th computing node
- $\overline{U}$  - number of network computing nodes
- $x_i, x_{ij}$  - constant service time in  $i$ -th - computing node resp. in  $i$ -th computing node and  $j$ -th communication channel
- $E(t_s)$  - the mean service time of a server
- $E(q)$  - the mean number of customers in a system at steady state
- $E(w)$  - the mean number of customers in a queue at steady state
- $E(t_q)$  - the mean time spent in system (queue + servicing) at steady state
- $E(t_w)$  - the mean time spent in the queue at steady state
- $c_{ij}$  - correction factor

\* Peter Hanuliak, Michal Hanuliak

Dubnica Institute of Technology, Dubnica nad Vahom, Slovakia  
E-mail: phanuliak@gmail.com

## 1. Periods of development of parallel computers

Scientific supercomputers which dominated in the first period of parallel computers between 1975 and 1995 were specially designed for high performance computing (HPC). These parallel computers mostly used computing models based on data parallelism. Those systems were ahead of standard common computers in terms of their performance and price. Increased processor performance was reached by the massive use of various parallel principles in all forms of produced processors. The parallel principles were used not only in single PCs and workstations (Scalar or super scalar pipeline or symmetrical multiprocessor systems SMP) [1 and 2] but also in extremely powerful PCs in form of variously connected network of workstations (NOW, cluster). Gained experience with the implementation of parallel principles and intensive extensions of computer networks led to the use of connected computers for parallel solution. We can define this period as the second period of development. Development trends were actually heading towards building widely spread connected NOW networks with high computation and memory capacity (Grid). Conceptually Grid comes to the definition of meta computer [3], where meta computer could be understood as a big computer network consisting of a massive number of computing nodes, memories and other needed resources that jointly create an illusion of one single powerful supercomputer. These highly integrated forms of NOWs create various Grid systems or meta computers that could be defined as the third period of parallel computers.

## 2. Typical architectures of modern parallel computers

### 2.1. Symmetrical multiprocessor system

Symmetrical multiprocessor system (SMP) is a multiple system using the same processors or cores which are implemented on motherboard in order to increase the whole performance of such system [4]. Typical common characteristics are as follows:

- each processor or core (Computing node) of the multiprocessor system can access the main memory (Shared memory),
- I/O channels or I/O devices are allocated to individual computing nodes according to their demands,
- integrated operation system coordinates cooperation of whole multiprocessor resources (Hardware, software etc.).

### 2.2. Network of workstations

Actually there has been an increasing interest in the use of networks of workstations (NOW) connected together by high speed networks for solving large computation intensive problems.

This trend is mainly driven by the cost effectiveness of such systems as compared to massive multiprocessor systems with tightly coupled processors and memories (Supercomputers). Parallel computing on a network of workstations connected via high speed networks has given rise to a range of hardware and network related issues on any given platform [5 and 6]. With the availability of cheap personal computers, workstations and networking devices, the recent trend is to connect a number of such workstations to solve computation intensive tasks in a parallel way on such clusters. Network of workstations [7] has become a widely accepted form of high performance computing (HPC). Every workstation in a NOW is treated similarly as a processing element in a multiprocessor system. However, workstations are far more powerful and flexible than processing elements in conventional multiprocessors (Supercomputers).

Typical example of networks of workstations also for solving complex problems is shown in Fig. 1. The individual workstations are mainly extremely powerful personal workstations based on multiprocessor or multicore platform SMP (Symmetrical multiprocessor systems).

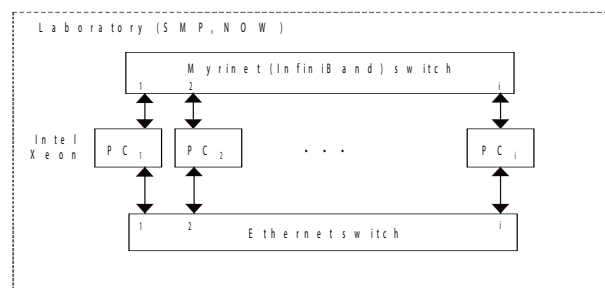


Fig. 1 Practical example of NOW module

We have studied such serious problems in parallel computing (Parallel and distributed computing) as load balancing, inter processor communication IPC, modelling and optimisation of parallel algorithms. etc. [8 and 9]. The coupled computing nodes  $PC_1, PC_2, \dots, PC_i$  (Workstations) could be single most powerful personal computers or SMP parallel computers. In this way, parallel computing on networks of conventional PC workstations (Single, multiprocessor, multicore) and Internet computing indicates advantages of unifying parallel and distributed computing [10].

### 2.3. Grid systems

In general, Grids represent a new way of managing and organising computer networks and mainly their deeper resource sharing [2]. Grid systems are expected to operate on a wider range of other resources as processors (CPU), as memory modules, data modules, network computing nodes, software files etc. All these resources typically exist within nodes that are geographically distributed and span multiple administrative domains. It is

obvious that existing HPC parallel computers (Supercomputers etc.) could be members of such Grid systems, too. Illustration of Grid modules based on integrated NOW networks are in Fig. 2.

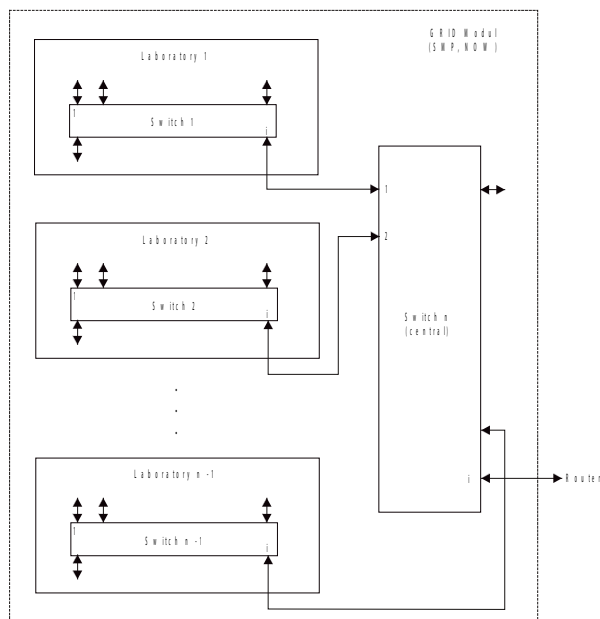


Fig. 2 Architecture of Grid module

Conceptually, Grids arise from a structure of virtual parallel computer based on computer networks (NOW). In general, Grids represent a new way of large integration and managing of resources as a large network of NOW modules. This expression is closely connected with the term massive computational Grid with the basic characteristics as follows:

- wide area network of integrated free computing resources - it is a massive number of inter connected networks, which are connected through high speed connected networks while the whole massive system is controlled by the network operation system, which makes an illusion of a powerful computer system (Virtual supercomputer) [11 and 12],
- grants a function of meta computing that means computing environment, which enables individual applications and functionality of all system resources,
- Grid system combines distributed parallel computation with remote computing from user workstations.

### 3. Abstract models of computing nodes

#### 3.1. Abstract model of SMP computing node with shared memory

Basic abstract model of SMP parallel computer with shared memory is in Fig. 3.

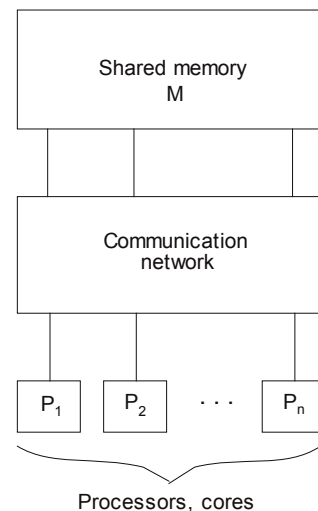


Fig. 3 Abstract model of SMP

#### 3.2. Abstract model of SMP with distributed memory

Abstract model of parallel computer with distributed memory is in Fig. 4.

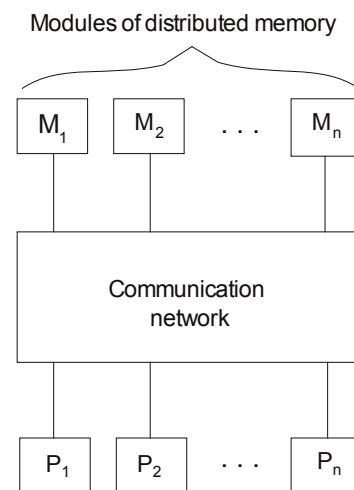


Fig. 4 Abstract model of NOW

### 4. The role of performance

Quantitative evaluation and modelling of hardware and software components of parallel systems are critical for the delivery of high performance [13]. Performance studies apply to initial design phases as well as to the procurement, tuning and capacity planning analysis. As performance cannot be expressed by quantities independent of the system workload, the quantitative characterisation of resource demands of application and of their behaviour is an important part of any performance evaluation

study [14 and 15]. Among the goals of parallel system performance analysis belong: to assess the performance of a system or a system component or an application, to investigate the match between requirements and system architecture characteristics, to identify the features that have a significant impact on the application execution time, to predict the performance of a particular application on a given parallel system, to evaluate different structures of parallel applications [16 and 17].

## 5. Application of queuing theory systems

The basic premise behind the use of queuing models for computer systems analysis is that the components of a computer system can be represented by a network of servers (Resources) and waiting lines (Queues) [18 and 19]. A server is defined as an entity that can affect, or even stop, the flow of jobs through the system. In a computer system, a server may be the CPU, I/O channel, memory, or a communication channel. Awaiting line is just that place where jobs (Customers) queue for service. To make a queuing model work, jobs or customers of data message blocks (Packets), which require any sort of processing, are inserted into the network. In such a system, jobs arrive at some rate, queue for service on a first-come first-served basis, receive service, and exit the system. This kind of model, with customers entering and leaving the system, is called an open queuing system model [14].

## 6. Results

### 6.1. Corrected standard analytical model

The described standard analytical model [1] supposes that the inter arrival time to the node's communication channels has the exponential distribution. This assumption is not true mainly in the important cases of high communication utilisation. The node servicing time of parallel processes (Computation complexity) could vary from nearly deterministic (in the case of balanced parallel processes) to exponential (in the case of unbalanced

ones). From this, in the case of node's high processors utilisation, the outputs from individual processor of node's multiprocessor may vary from the deterministic interval time distribution to exponential one. These facts violate the assumption of the random exponential distribution and could lead to erroneous value of the whole node's delay computation. Worst of all, this error could be even greater if the node of utilisation were higher. From the above-mentioned we derived the correction factor which accounts for the measure of violation for the exponential distribution assumption. The inter arrival input time distribution to each node's communication channel depends on  $\rho_i$ , where  $\rho_i$  is the overall processor utilisation at the node  $i$ . But because only the part  $\lambda_{ij}$  from the total input rate  $\lambda_i$  for node  $i$  goes to the node's communication channel  $j$ , it is necessary to weigh the measure of influence of the whole node's processors utilisation through the value  $\lambda_{ij}/\lambda_i$  for channel  $j$  as  $\rho_i \cdot (\lambda_{ij}/\lambda_i)$ .

Mathematical abstract model of  $i$ -th node for NOW system is illustrated in Fig. 5.

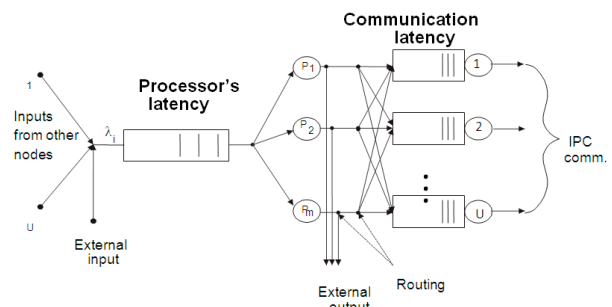


Fig. 5 Abstract model of  $i$ -th node of NOW

To clarify the influence of the node's processor utilisation on the average delay of communication channel we tested the 5-noded experimental parallel computer. The processing time was varied to develop various workloads of node's processors. Performed tests showed a remarkable fact - by decreasing the node's processor workload the assumption of the exponential inter arrival message time distribution to the communication

Achieved results for correction factor

Table 1

| Processor utilisation<br>at node 1 | Average channel<br>delay at node<br>1 - simulation<br>[msec] | Standard analytical model       |                          | Corrected analytical model      |                          |
|------------------------------------|--|---------------------------------|--------------------------|---------------------------------|--------------------------|
|                                    |  | Average channel<br>delay [msec] | Relative<br>error<br>[%] | Average channel<br>delay [msec] | Relative<br>error<br>[%] |
| 0.6                                | 21.97  | 22.27                           | 1.4                      | 22.03                           | 0.3                      |
| 0.7                                | 21.72  | 22.27                           | 2.5                      | 21.92                           | 0.9                      |
| 0.8                                | 21.43  | 22.27                           | 3.9                      | 21.70                           | 1.3                      |
| 0.9                                | 21.05  | 22.27                           | 5.8                      | 21.45                           | 1.9                      |
| 0.95                               | 20.91  | 22.20                           | 6.5                      | 21.31                           | 1.9                      |

channel is more effective. The achieved results are summarised in Table 1 for one of the communication channels at node 1.

Graphical illustration of achieved results is in Fig. 6.

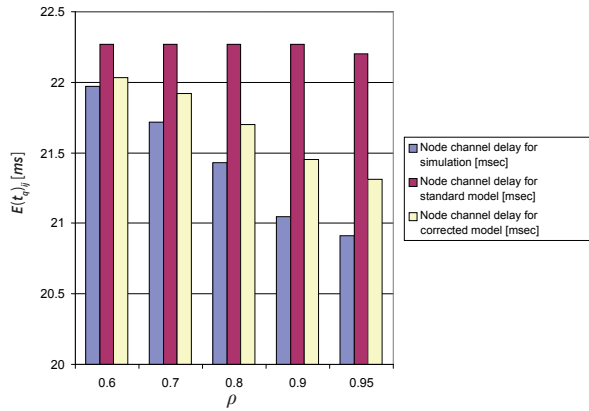


Fig. 6 The influence of the exponential time distribution and its correction

Extensive testing have proved that if we increase utilisation of communication channel and it develops saturation of the communication channel queue then the average queue waiting time is less sensitive to the nature of inter arrival time distributions. This is due to the fact that the messages (Communicating IPC data) wait longer in the queue, which significantly influences the increase of the average waiting time and the error influence of the non-exponential inter arrival time distribution is decreased. To incorporate this knowledge into the correlation factor we investigated the influence of weighting  $p_i(\lambda_{ij}/\lambda_i)$  on the value  $(1 - \rho_{ij})^x$  for various values  $x$ . The performed experiments showed the best results for the value  $x = 1$ . Derived approximation of the average queue waiting time of the communication channel  $j$  at the node  $i$ , which eliminates violence of the exponential inter arrival time distribution, is then given as

$$\frac{\rho_i \cdot (1 - \rho_{ij}) \cdot \lambda_{ij}}{\lambda_i}$$

The final correction factor of the communication channel  $j$  at the node  $i$ , which we named as  $c_{ij}$  is as follows

$$c_{ij} = 1 - \frac{\rho_i \cdot (1 - \rho_{ij}) \cdot \lambda_{ij}}{\lambda_i}$$

With the derived correction factor  $c_{ij}$  we can define the corrected average queue waiting time as

$$E(t_{q_{ij}})'(LQ) = c_{ij} \cdot E(t_{q_{ij}})(LQ)$$

where  $E(t_{q_{ij}})(LQ)$  is defined in [13] as

$$\frac{\lambda_{ij} \cdot E(t_{q_{ij}})}{\gamma}$$

Based on the derived correction factor, the whole latency in modelled NOW is as

$$E(t_q)_{now} = \frac{1}{\gamma} \left[ \sum_{i=1}^U \left( \lambda_i \cdot E(t_{q_i}) + \sum_{j=1}^{u_i} c_{ij} [\lambda_{ij} \cdot E(t_{q_{ij}})] \right) \right] \quad (1)$$

where  $\frac{\lambda_i \cdot E(t_{q_i})}{\gamma}$  and  $c_{ij} \frac{\lambda_{ij} \cdot E(t_{q_{ij}})}{\gamma}$  define individual contribution of computation queue delay and communication channel delay of every node to the whole delay. We named this model based on M/M/m and corrected M/M/1 results as a corrected standard analytical model. The average delay values of the node's communication channel achieved through simulation are compared with the results of the standard analytical model (Exponential inter arrival time distribution) and with the results of the corrected standard model as illustrated in Fig. 7.

## 6.2. Other real analytical models

### 6.2.1. Analytical model with M/M/m and M/D/1 systems

This model is a mixture of analysed models. We get the first part of the final total average time  $E(t_{q_{now}})$  from relation (1) of the previous chapter. The second part is the sum of all communication channels considered as M/D/1 system [17]. Then, for  $E(t_{q_{now}})$  we can finally get:

$$E(t_q)_{now} = \frac{1}{\gamma} \left[ \sum_{i=1}^U \left( \lambda_i \cdot E(t_{q_i}) + \sum_{j=1}^{u_i} (E(t_w)_{ij}(LQ) + \bar{x}_{ij}) \right) \right] \quad (2)$$

### 6.2.2. Model with M/D/m and M/M/1 systems

In this alternative analytical model the first part of the final total average time  $E(t_{q_{now}})$  is the derived expression in [17] for M/D/m system. We get the second part from relation (1) of the previous chapter as the sum over all communication channels

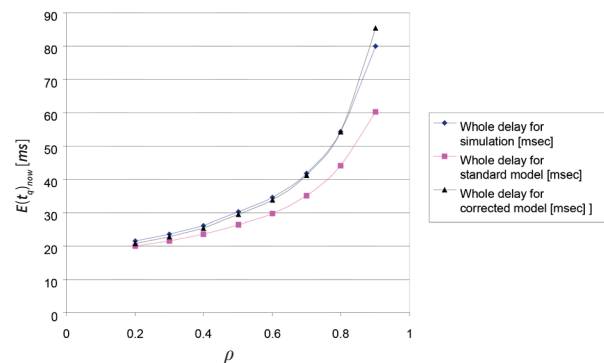


Fig. 7 Comparison of the used models

considering them as corrected M/M/1 system. Then, for  $E(t_{q_{now}})$  we finally get for this model:

(3)

$$E(t_{q_{now}}) = \frac{1}{\gamma} \left[ \sum_{i=1}^U \left( (E(t_{w_{ij}})(PQ) + \bar{x}_{i_j}) + \sum_{j=1}^{u_i} c_{ij} [\lambda_{ij} \cdot E(t_{q_{ij}})] \right) \right]$$

Comparison of the relative errors (in relation to simulation results) is illustrated in Fig. 8.

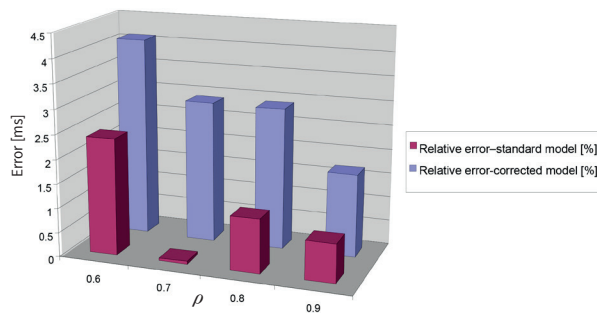


Fig. 8 Influence of channel utilisation on the accuracy of analytical models

### 6.3. Analytical model of real Grid systems

We defined the Grid system as a network of NOW network modules. Let N be the number of individual NOW networks or similar clusters. Then final total average time  $E(t_{q_{grid}})$

$$E(t_{q_{grid}}) = \frac{1}{\alpha} \left[ \sum_{i=1}^N E(t_{q_{i_{now}}}) \right] \quad (4)$$

where

- $\alpha = \sum_{i=1}^N \gamma_i$  represents the sum of individual total external intensities to the i-th NOW module in the Grid
- $E(t_{q_{i_{now}}})$  corresponds to individual average times in i-th NOW module ( $i=1, 2, \dots, N$ ).

## 7. Conclusion and perspectives

Performance evaluation of computers has generally been a very difficult problem since the birth of computers. It was very hard to apply any analytical methods to performance evaluation of sequential computers because of high number of non-predictable parameters. Secondly, endless user demands to increase computer performance were fulfilled more quickly via continuing technology improvements or computer architecture changes. For a long time incorporation of various forms of parallel principles has created more stable conditions to apply performance evaluation methods mainly for dominant parallel

computers (SMP, NOW and Grid) to open more possibilities to apply mainly analytical methods (Queuing theory) in order to analyse performance of parallel computers. This implies to one known queuing theory result - many inputs to queuing theory system, which create shared stream and which are generated at various independent resources by chance, could be a very good approximation of Poisson distribution, as a basic assumption, to solve such coupled parallel systems in an analytical way. Therefore, we are able to model parallel computing nodes (Multiprocessors, multicores, workstations etc.) of any dominant or perspective parallel computer (SMP, NOW, Grid, meta computer) as M/D/m or M/M/m queuing theory systems and computing node's communication channels as M/D/1 or M/M/1 queuing theory systems respectively.

We have compared the relative errors of the developed models (standard, corrected) in relation to the developed simulation model. The relative errors of standard analytical model are from 7% to 25%. This is due to the influences of computing node queue delays, the nature of inter arrival inputs to the communication channels mainly in the case of high processor utilisation. The developed corrected analytical model provides more precise results in the whole range of input workload of processor utilisation, communication channels and network topologies with relative error, which does not exceed 14% and even in the most important cases of the high processor utilisation ( $\rho > 0.6$ ) the relative errors according to Fig. 8 are in the range up to 9%.

Then, such a very flexible modelling tool based on preferred analytical solutions shows a real path to a very efficient and practical performance analysis tool including massive NOW networks or other types of massive computer networks (Grid, meta computer). To summarise it - all developed more precise analytical models could be applied to the performance modelling of dominant parallel computers in typical cases as follows:

- a single computing node based on SMP parallel computer,
- NOW based on workstations (single, SMP),
- Grid (Network of NOW network modules),
- mixed parallel computers (SMP, NOW, Grid),
- meta computers (Massive Grid).

For our further research work, in relation to dominant trends in parallel computers (SMP, NOW, Grid) based of powerful workstations, we will be looking for preferred analytical models [20] in which load balancing, inter-process communication (IPC) in both parallel and distributed computing [21], effective transport protocols, influence of various parallel computer architectures, performance prediction etc. could be studied. We would also like:

- to analyse the role of adaptive routing in considered analytical models,
- to prove or to indicate experimentally the role of the independence assumption if we are looking for higher moments of overhead latencies (IPC communication, synchronisation, parallelisation, parallel computer architecture etc.)



- to verify analytical models also for node's limited resources capacities – buffers, communication channels etc., and for other existing queue servicing algorithms as standard assumed FIFO (First in First out).

To model single computing nodes we can also use other more complicated single queuing theory systems as the analysed ones (M/M/1, M/M/m, M/D/1, and M/D/m). The reasons for choosing these single queuing theory systems are following:

- to finish performance analysis of network of coupled queuing theory systems we need results of chosen single queuing theory systems M/M/m and M/D/m [22],
- we need their results to compute approximation relation for M/D/m,
- M/M/1 and M/M/m models could be used to compare their results with other models M/D/1 and M/D/m respectively,
- the results of analysed models M/M/m and M/D/m are necessary to finish performance analysis of decomposed communication network of computing nodes.

### Acknowledgement

This work was carried out within the project “Complete modelling, optimisation and prediction of parallel computers and algorithms” at the University of Zilina, Slovakia. The authors gratefully acknowledge the help of the project supervisor Prof. Ing. Ivan Hanuliak, PhD.

### References

- [1] HAGER, G., WELLEIN, G.: *Introduction to High Performance Computing for Scientists and Engineers*, 356 pp, 2010, CRC Press.
- [2] PATTERSON, D. A., HENNESSY, J. L.: *Computer Organization and Design*, 4<sup>th</sup> ed., Morgan Kaufmann, 914 pp., 2011.
- [3] WANG, L., JIE WEI, CHEN, J.: *Grid Computing: Infrastructure, Service, and Application*, CRC Press, 2009.
- [4] DUBOIS, M., ANNAVARAM, M., STENSTROM, P.: *Parallel Computer Organization and Design*, 560 p., 2012, Cambridge university press.
- [5] CENEK, J.: Evaluation of a Heuristic Algorithm for Transit Line Planning, *Communications - Scientific Letters of the University of Zilina*, No. 3, pp. 44-48, 2013.
- [6] HANULIAK, J., HANULIAK, I.: To Performance Evaluation of Distributed Parallel Algorithms, *Kybernetes*, vol. 34, No. 9/10, 1633-1650, 2005.
- [7] HANULIAK, P., HANULIAK, M.: Mod. of Single Comp. Nodes of Par. Comp., *AJNC (Am. J. of Networks and Comm.)*, Science PG, vol. 3, Special No. 1, 57-69, 2014.
- [8] RIANO, L., MCGINITY, T. M.: *Quantifying the Role of Complexity in A System's Performance*, Evolving Systems, Springer Verlag, 189-198, 2011.
- [9] GABRISOVA, L., JANACEK, J.: Design of Capacitated Emergency Service System, *Communications - Scientific Letters of the University of Zilina*, No. 2, pp. 42-48, 2015.
- [10] PETERSON L., DAVIE B. C.: *Computer Networks - A System Approach*, Morgan Kaufmann, 920 p., 2011.
- [11] HARCHOL, B. M.: *Performance Modelling and Design of Computer Systems*, Cambridge University Press, 576 p., 2013.
- [12] HANULIAK, M., HANULIAK, P.: Performance Modelling of Par. Comp. NOW and Grid, *AJNC - Am. J. of Networks and Communication*, Science PG, 112-124, 2013.
- [13] ARORA S., BARAK B.: *Computational Complexity - A Modern Approach*, Cambridge University Press, 573 p., 2009.
- [14] DATTATREYA, G. R.: *Performance Analysis of Queuing and Computer Network*, 472 pp., 2008, University of Texas: Dallas.
- [15] LE BOUDEC J. Y.: *Performance Evaluation of Computer and Communication Systems*, CRC Press, 300 p., 2011.
- [16] MEERSCHAERT, M.: *Mathematical Modelling*, 384 p., 2013, Elsevier.
- [17] NATARAJAN, G.: *Analysis of Queues - Methods and Applications*, 802 p., CRC Press, 2012.
- [18] HANULIAK P., HANULIAK M.: Modell. of Single Comp. Nodes of Par. Comp., *AJNC (Am. J. of Networks and Comm.)*, Science PG, vol. 3, Spec. No. 1, 57-69, 2014, USA.
- [19] KOSTIN, A., ILUSHECHKINA. L.: *Modelling and Simulation of Distributed Systems*, Imperial College Press, 440 p., 2010.
- [20] BRONSON R., COSTA G. B., SACCOMAN J. T.: *Linear Algebra - Algorithms, Applications, and Techniques*, 536 p., Elsevier, 2014.
- [21] HANULIAK, J.: Modelling of Comm. Complexity in Parallel Computing, *AJNC (Am. J. of Networks and Comm.)*, Science PG, vol. 3, Spec. No. 1, 29-42, 2014.
- [22] GAUTAM, N.: *Analysis of Queues: Methods and Applications*, CRC Press, 802 p., 2012.