

Unified analytical models of parallel and distributed computing

Michal Hanuliak

Dubnica Technical Institute, Dubnica Nad Vahom, Slovakia

Email address:

michal.hanuliak@gmail.com

To site this article:

Michal Hanuliak. Unified Analytical Models of Parallel and Distributed Computing. *American Journal of Networks and Communications*. Vol. 3, No. 1, 2014, pp. 1-12. doi: 10.11648/j.ajnc.20140301.11

Abstract: The optimal resource allocation satisfies the needed capacity of the used resources. To such analysis we can use both analytical and simulation methods. Principally analytical methods (AM) belong to the preferred method in comparison to the simulation method, because of their potential ability of more general analysis and also of ability to analyze massive parallel computers. This article goes further in developing AM based on queuing theory results in relation to our published paper in [9]. The extensions are in extending derived AM to whole range of parallel computers and also to sum up public acceptance of our published paper. The article therefore describes deriving of correction factor of standard AM based on M/M/m and M/M/1 queuing theory systems. In detail the paper describes derivation of a correction factor for standard AM to study more precise their performance. The paper contributions are in unified AM and in deriving correction factor in order to take into account real non-exponential nature of the inputs to the computing nodes and node’s communication channels. The derived analytical results were compared with performed simulation results in order to estimate the magnitude of improvement. Likewise the corrected AM were tested under various ranges of parameters, which influence the architecture of the parallel computers and its communication networks too. These results are very important in practical use.

Keywords: Parallel Computer, Communication System, Correction Factor, Analytical Model, Performance, Queuing System, Overhead Latencies, Modeling

1. Introduction

In [9] we have characterized developing periods of parallel computers and their basic classification from the point of programmer as potential developers of parallel algorithms. In relation to it and also to other performed classifications [1, 8] we have divided them as following

- synchronous parallel computers. To this group belong actually dominated parallel computers based on multiply cores, processors or mix of them too (symmetrical multiprocessors - SMP) and most of realized massive parallel computers (classic supercomputers) [31]. The practical example of such synchronous parallel computer is illustrated at Figure 1.
- asynchronous parallel computers. According the mentioned characteristics this group consist of actually dominant distributed parallel computers based on NOW (Network of workstation) module. To this group belong mainly computer networks based on network of workstation (NOW) module [16, 30]. The example of typical asynchronous parallel computer illustrates Figure 2.

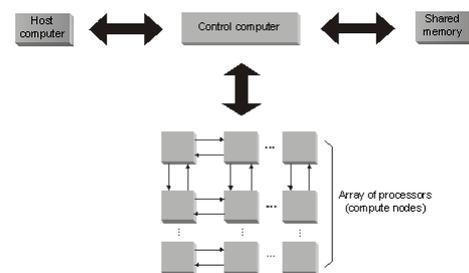


Figure 1. Example of synchronous parallel computer.

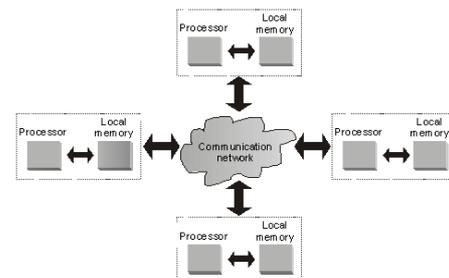


Figure 2. Example of asynchronous parallel computer.

Practical example of NOW module is representing at Figure 3. It represents also our outgoing architecture of laboratory parallel computer. On such modular parallel computer we have been able to study basic problems in parallel and distributed computing as load balancing, inter processor communication (IPC), modeling and optimization of parallel algorithms (effective PA) etc. [10, 11, 18]. The coupled computing nodes PC_1, PC_2, \dots, PC_i (workstations) could be single extreme powerful personal computers (PC) or SMP parallel computers. In this way parallel computing on networks of conventional PC workstations (single, multiprocessor, multicore) and Internet computing, suggest advantages of unifying parallel and distributed computing. Parallel computing and distributed computing have traditionally evolved as two separate research disciplines. Parallel computing has addressed problems of communication-intensive computation on highly - coupled processors [21, 22] while distributed computing has been concerned with coordination, availability, timeliness, etc., of more likely coupled computations [28, 34].

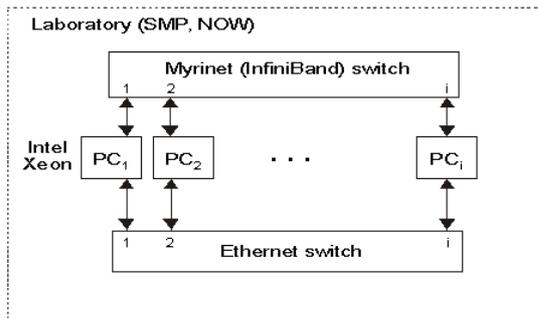


Figure 3. Practical example of NOW module.

Workstations could be connected using different network technologies such as off the shelf devices like Ethernet to specialized high speed communication networks (Infiniband, Quadrics, Myrinet) [35]. Such networks and the associated software and protocols introduce latency and throughput limitations thereby increasing the execution time of cluster - based computation [23, 26]. Researchers are engaged in designing algorithms and protocols to minimize the effect of these latencies [12, 13].

2. Abstract Models of Parallel Computer

For any realized parallel computer we can use one of the following two basic models according Figure 4 or Figure 5. The difference of both models technically consists in various types of memories (shared, distributed) because both of these memories are actually build from memory modules. On the other side applied using of various memory types of parallel computers is very different. But in abstract models existed differences are question of only defined technical parameters. Therefore recent trends point

to a convergence of research in parallel and distributed computing. Perhaps the most significant of these trends is architectural. Three architectural trends may be noted. First, increased communication bandwidth and reduced latency make geographical distribution of processing nodes less of a barrier to distributed computing.

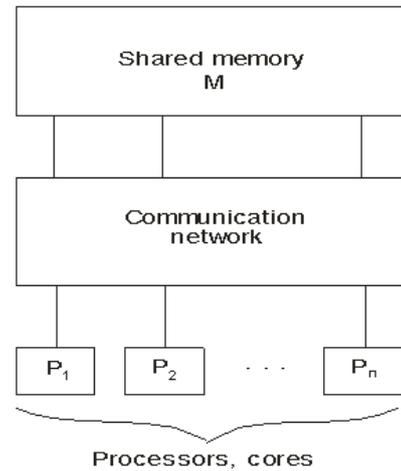


Figure 4. Abstract model of the synchronous parallel computer.

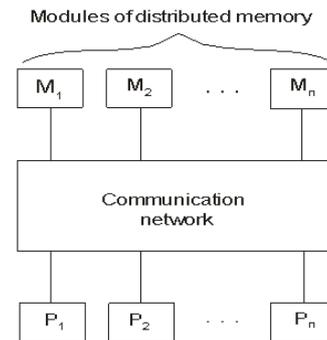


Figure 5. Abstract model of asynchronous parallel computers.

Second, the development of architecture transparent programming language, such as Java, provides a virtual computational environment in which nodes appear to be homogenous. Finally server machines in client/server (manager/worker) computing are increasingly adopting multiprocessor architecture, often multiple processors with a shared memory in a single workstations and symmetric multiprocessors (SMP) [36]. While such architectures are less scalable than computer networks (NOW, clusters), some concurrent programs with high communication traffic may execute on them more efficiently [19, 25].

3. Analytical Performance Evaluation of Parallel Computer in Queuing Theory

To the behavior analysis of parallel computer including their communication networks there have been developed various analytical models based on queuing theory results. Queuing theory is very good if you have to analyze a single

independent computing node of sequential or parallel computers [6, 15]. But analysis of used dominant parallel computers (NOW, Grid - network of NOW modules) generally lead to multiple computing node case. The first problem, in comparison to a single node case, is existence of traffic dependency in any real networks. What comes out of one computing node feeds the second one and what comes out from second node could not be Poisson [3]. Typically we could get into trouble even in single SMP parallel computer based on multicore or multiprocessor platform. Suppose there is communication traffic entering the system M/M/1 (Poisson arrivals and exponential service times) as simplest applied queuing theory system. One must ask the question is what comes out of the same nature as input traffic? It turns out that there is a fundamental theorem which says: only in the case when you have exponential service times (M/M/1, M/M/m) will you get something coming out which is also Poisson in nature [14].

There is a very important relationship for praxis which says: if you have Poisson arrivals and exponential service time what comes out has also Poisson characteristics. There is also a more general result which says: not only will the output be Poisson, if the input is Poisson, but we can consider also multiple service with defections, renegeing, balking and in some cases finite queue storage. But it has been proved that the outgoing process will be exactly Poisson only on the assumption of unlimited length of queue in size.

If all the nodal traffic has the property that it is Poisson, then even in a complicated network we can do under some conditions network analysis on a node-by-node basis. In fact, however, that is not yet true, because in communication networks of parallel computers the time a communication message spends in one node is related to the time it spends in another node, because the service one is looking for is network communication. That is one very nasty problem, but there have been developed some solutions.

The second serious problem is blocking as consequence of always real limited technical resources. If one node is blocked, the node feeding could not enter more data into that node. Consider a communication network in which you are given the location of computing nodes and the required communication traffic between pairs of computing nodes. Then according mentioned theorem says that if you have Poisson traffic into an exponential server you get Poisson traffic out; but a message maintains its length as it passes through the network, so the service times are dependent as it goes along its path. Thus, one thing we want to do is to get rid of that dependence. We can do this by making an independence assumption; we just assume that the dependence does not exist. We manage this by allowing the communication message to change its length as it passes through the communication network. Every time it hits a new computing node, we are going to randomly choose the message length so that we come up with an exponential distribution again. With that assumption, we can then solve

the queuing problem of communication in parallel computers. Let us assume infinite storage at all points in the network of coupled computing nodes and refer to the problem M/M/1, where the question mark refers to the modified input process. We then run simulations, with and without the independence assumption for a variety of networks. The reason why it is good to do it is that a high degree of mixing takes place in a typical communication network; there are many ways into a node and many ways out of the node. The dependence between service times and between adjacent messages on a line need not be highly correlated even if there were originally. With that assumption, it is possible to use appropriate results of the queuing theory for modeling parallel computers including their communication systems. We can use the results from essentially single node queuing theory, since now we have independent arrival and services processes. Neither the queuing model nor the simulation hit the blocking effect; so blocking remains as nasty problem.

The assumption of independence permits us to break also the massive parallel computer into independent computing nodes, and allowed all node analysis to take place. The reason we had to make that assumption was because the communication message maintains the same length as they pass through the network. If we accept the independence assumption, it turns out that the queuing theory contains a number of results for cases where the service at a node is an independent random variable in an arbitrary network of queues. A basic theorem is due to Jackson [29, 32]. Jackson's result essentially gives us the probability distribution for various numbers of messages at each of the nodes in such a network. These nodes are essentially Markovian queues, and Markovian queuing theory is relatively simple in the nearest neighbor cases. There are more general cases, which one has to solve which involve solving a set of linear simultaneous equations for the probabilities of finding queue lengths at various points. Typically, in the complicated network, one looks for something called the product form solution. This term comes from Jackson's theorem, which expresses the joint probability of the numbers of customers at each queue being a particular combination is the product of their individual probabilities of having that number [27].

Product form networks have the property that they can be regarded as independently operating queues, where steady state can be expressed as both a set of global balance equations on each queue. Local flow balance says that the mean number of customers entering any queue from all others must equal the number leaving it to go to all others, including customers which leave and rejoin the same queue immediately.

3.1. Application of Queuing Theory

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 awaiting lines (queues). 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 port. Awaiting line is just that: a place where jobs queue for service. To make a queuing model work, jobs or customers or communication message (blocks of data, packets) or anything else that requires the sort of processing provided by the server, are inserted into the network. A basic simple example could be the single server abstract model as single queuing theory system. In such model, 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 jobs entering and leaving the system, is called an open queuing system model [3, 33].

3.2. Background of Analytical Models

We will now turn our attention to some suitable network of queuing theory systems, the notation used to represent them, the performance quantities of interest, and the methods for calculating them. We have already introduced many notations for the quantities of interest for random variables and stochastic processes. For the analysis of parallel computers including their communication network communication networks we have been developed in [9] following analytical models

- standard analytical model on the basis of Jackson's theorem. In this case we suppose decomposition to the independent computing nodes on the basis of the M/M/m systems with infinite buffer. This together with the independence assumption reduces a very difficult problem to an open network of independent M/M/m queuing theory queues. This analytical model produced the worst results (upper limit)

- improved analytical model on the basis of the modern node's multiprocessor system [4], as computing node of parallel computer. This analytical model extends used analytical models considering the delays through the activities of communication processor and through awaiting these services with the more realistic M/D/m queuing theory systems. Satisfying conditions this improved analytical model will be produced the best results (lower limit)

- results of any other developed analytical model will be between results of these two analytical models (standard, improved).

Queuing theory systems are classified according to various characteristics, which are often summarized using Kendall's notation [7, 9]. The basic parameters of queuing theory systems are as following

- λ - arrival rate at entrance to a queue
- m - number of identical servers in the queuing system
- ρ - traffic intensity (dimensionless coefficient of utilization)
- 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

- $E(ts)$ - the expected (mean) service time of a server
- $E(q)$ - the expected (mean) number of customers in a system at steady state
- $E(w)$ - the expected (mean) number of customers in a queue at steady state
- $E(tq)$ - the expected (mean) time spent in system (queue + servicing) at steady state
- $E(tw)$ - the expected (mean) time spent in the queue at steady state.

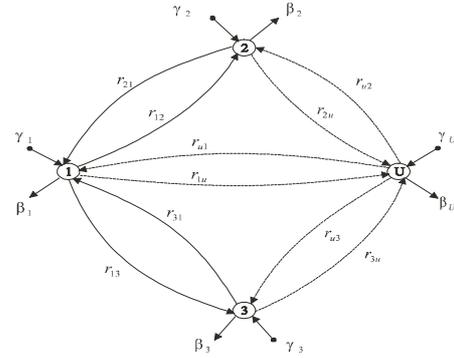


Figure 6. Parallel computer model including its communication network.

Communication demands (parallel processes, IPC data) arrive at random at a source node and follow a specific route in the communication networks towards their destination node. Data lengths of communicated IPC data units (for example in words) are considered to be random variables following distributions according Jackson theorem. Those data units are then sent independently through the communication network nodes towards the destination node. At each node a queue of incoming data units is served according to a first-come first-served (FCFS) discipline.

At Figure 6 we illustrate generalization of any parallel computer including their communication network as following

- computing nodes U_i ($i=1, 2, 3, \dots, U$) of any parallel computer are modeled as graph nodes
- network communication channels are modeled as graph edges r_{ij} ($i \neq j$) representing communication intensities (relation probabilities).

The other used parameter of such abstract model are defined as following

- $\gamma_1, \gamma_2, \dots, \gamma_U$ represent the total intensity of input data stream to individual network computing nodes (the summary input stream from other connected computing nodes to the given i -th computing node. It is given as Poisson input stream with intensity λ_i demands in time unit
- r_{ij} are given as the relation probabilities from node i to the neighboring connected nodes j
- $\beta_1, \beta_2, \dots, \beta_U$ correspond to the total extern output stream of data units from used nodes (the total output stream to the connected computing nodes of the given node).

The created abstract model according Figure 6 belongs in queuing theory to the class of open queuing theory systems

(open queuing networks). Formally we can adjust abstract model adding virtual two nodes (node 0 and node U+1 according Figure 7 where

- virtual node 0 represent the sum of individual total extern input intensities $\gamma = \sum_{i=1}^U \gamma_i$ to computing nodes U_i
- virtual node U+1 represent the sum of individual total intern output intensities $\beta = \sum_{i=1}^U \beta_i$ from computing nodes u_i .

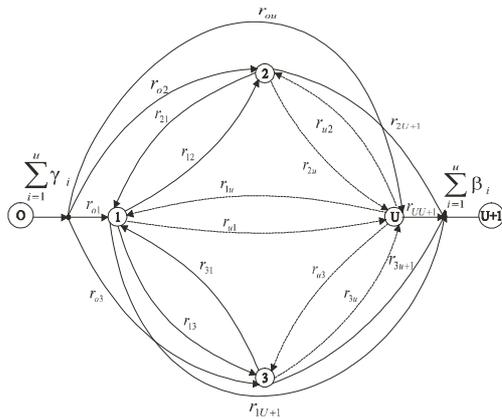


Figure 7. Adjusted abstract model.

3.3. Standard Analytical Model

In [9] we have presented open multi nodes module NOW consisting of U computation nodes including their communication network. Model of i-th computing node illustrates Figure 8. For the whole delay in NOW we have derived following final result based on M/M/m and M/M/1 queuing theory systems as

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

where $\frac{\lambda_i \cdot E(t_q)_i}{\gamma}$ and $\frac{\lambda_{ij} \cdot E(t_q)_{ij}}{\gamma}$ defined individual

contribution of computation queue delay (M/M/m) and communication channel delay (M/M/1) of every computing node to the whole delay. The meanings of used and not explained parameters are as following

- λ_i - the whole number of incoming demands to the i-th node, that is the sum both of external and internal inputs to the i-th node
- λ_{ij} - the whole input flow to the j-th communication channel at i-th node
- $E(t_q)_i$ - the average servicing time in the program queue (the waiting in a queue and servicing time) in the i-th node
- $E(t_q)_{ij}$ - the average servicing time of the j-th queue of the communication channel (the queue waiting time and servicing time) at i-th node.

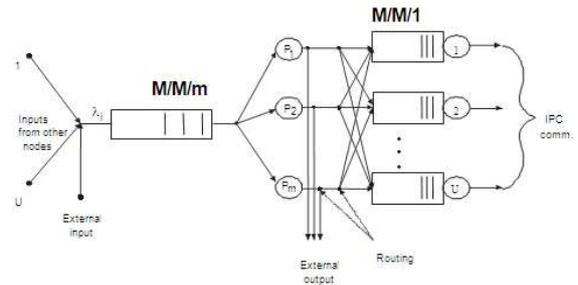


Figure 8. Standard analytical model of i-th computing node.

The intern input flow to i-th node is defined as the input from all other connected computing nodes. We can express it in two following ways

- through solving a system of linear equations in matrix form as $\bar{\lambda} = \gamma + \bar{\lambda} \cdot R$
- using of two data structures in form of tables and that is the routing table (RT) and destination probability tables (DPT).

The used model were build on assumptions of modeling incoming demands to program queue as Poisson input stream and of the exponential inter arrival time between communication inputs to the communication channels. The idea of the previous models were the presumption of decomposition to the individual nondependent computing nodes together with the independence presumption of the demand length, that is the demand length is derived on the basis of the probability density function $p_i = \mu e^{-\mu t}$ for $t > 0$ and $f(t) = 0$ for $t \leq 0$ independent always at its input to the node. On this basis it was possible to model every used communication channel as the queuing theory system M/M/1 and derive the average value of delay individually for every communication channel. The whole delay was then simply the sum of the individual delays of the every used communication channel.

To improve the mentioned problems we suggested improved analytical model, which extends the used standard analytical model to more precise analytical model (improved analytical model) supposing that

- we consider to model computation activities in every node of NOW network as M/D/m system (assumption of inputted balanced parallel processes to every node)
- we consider an individual communication channels in i-th node as M/D/1 systems. In this way we can take into account also the influence of real non exponential nature of the inter arrival time of inputs to the communication channels (assumption of nearly equal IPC communication complexity).

Both published analytical models (standard, improved) are not fulfilled for every input load, for all parallel computer architectures and for the real character of computing node service time distributions. These changes may cause at some real cases imprecise results. Another survived problem of the used standard analytical model is assumption of the exponential inter arrival time between message inputs to the communication channels in case of

unbalanced communication complexity of parallel processes. To remove mentioned changes we derived a correction factor to standard analytical model.

4. Corrected Standard Analytical Model

The standard analytical model 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 utilization. The node servicing time of parallel processes (computation complexity) could vary from nearly deterministic (in case of balanced parallel processes) to exponential (in case of unbalanced ones). From this in case of node's high processors utilization 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 whole node's delay calculation. Worst of all this error could be greater the higher is the node utilization. From these causes we have derived the correction factor which accounts the measure of violation for the exponential distribution assumption.

The inter arrival input time distribution to each node's communication channel depends on ρ_i , where ρ_i is the overall processor utilization at the node i . But because only the part λ_{ij} from the total input rate λ_i for node i go to the node's communication channel j , it is necessary to weight the influence measure of the whole node's processors utilization through the value λ_{ij} / λ_i for channel j as $\rho_i \cdot (\lambda_{ij} / \lambda_i)$.

Table 1. Achieved results for correction factor.

Processor utilization at node 1	Average channel delay at node 1 - simulation [msec]	Standard analytical model		Correct analytical model	
		Average channel delay [msec]	Relative error [%]	Average channel delay [msec]	Relative error [%]
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	21,91	22,20	6,5	21,31	1,9

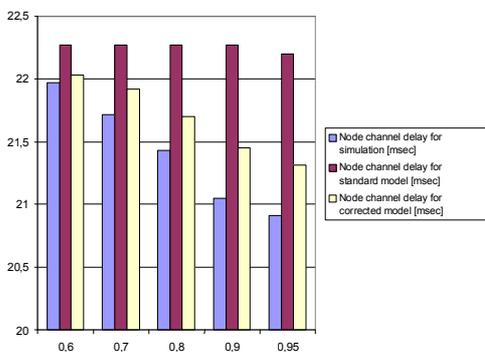


Figure 9. The influence of the exponential time distribution and its correction.

To clarify the node's processor utilization influence to the average delay of communication channel we have tested the 7-noded experimental parallel computer. The processing time was varied to develop the various workloads of node's processors. The achieved results are summarized at Table 1 for one of communication channels at the node 1. Graphical illustration of achieved results is at Figure 9.

Extensive testing have proved, that if we increase utilization of communication channel and that develops saturation of communication channel queue then 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 what significantly influenced 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 for the correlation factor we investigated the influence of the weighting ρ_i (λ_{ij} / λ_i) through the value $(1 - \rho_i)^x$ 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 finally correction factor of the communication channel j at the node i , which we have named as c_{ij} is as following

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

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

$$W_{ij}'(LQ) = c_{ij} \cdot W_{ij}(LQ)$$

The standard analytical model we can simply correct in such a way that instead of $W_{ij}(LQ)$ we will consider its corrected value $W_{ij}'(LQ)$. In this way derived improved standard analytical model we have defined as corrected standard analytical model. From the performed tests it is also remarkable that decreasing of the node's processors workload the assumption of the exponential inter arrival message time distribution to the communication channel is more effective.

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. Comparison of the relative errors is illustrated in the Figure 10.

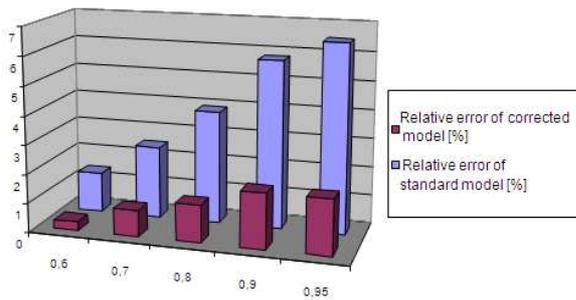


Figure 10. Comparison of relative errors.

At Table 2 there are results of the channel utilization influence to the average waiting time for the communication channel of 7-noded communication network. For this case the channel utilization was influenced through communication speed changes.

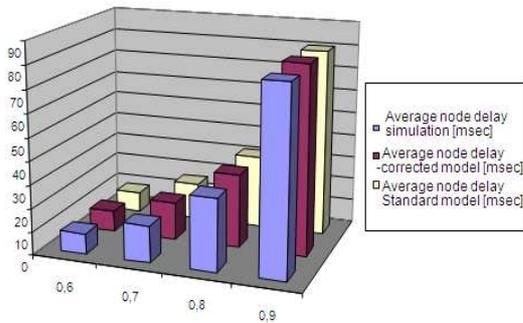


Figure 11. The channel utilization influence to the total node delay.

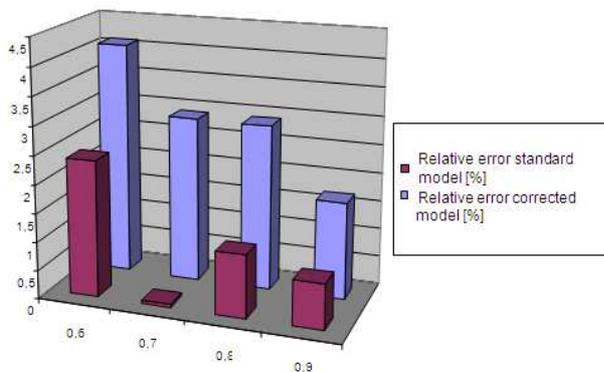


Figure 12. Influence of channel utilization to the accuracy of analytical model.

Table 2. The results of the channel influence.

Processor utilization at node 1	Average channel delay at node 1 - simulation [msec]	Standard analytical model		Correct analytical model	
		Average channel delay [msec]	Relative error [%]	Average channel delay [msec]	Relative error [%]
0,6	8,89	9,25	4,1	8,68	2,4
0,7	15,92	16,38	2,9	15,91	0,06
0,8	31,04	31,94	2,9	31,39	1,1
0,9	79,76	81,08	1,7	80,38	0,8

The achieved results in Table 2 are illustrated at Figure 11. The relative errors are incorporated in the Table 2. The influence of communication channel utilization to the result accuracy of the analytical models is at the Figure 12. From these achieved results follow that decreasing of the node’s communication channel utilization the difference between simulated results and the standard analytical model increases.

5. The Achieved Results

The developed corrected model was intensive tested and compared with simulation results. We varied various range of parameters, which influence architecture of parallel computer – number of computing nodes, topology of communication network, communication network load, communication speed of node’s communication channels etc. in order to verify stability and accuracy of the corrected model under various conditions. The following tables and graphs present essential parts of achieved results of the done experiments. In each tested case we compared percentages errors between corrected model and simulation results. In Table 3 are results for 5-noded communication network, in which all nodes are fully interconnected (every node with all other nodes). This rather small communication network has its causes through limitations of the simulation method than analytical one.

Table 3. The comparison of used methods

Processor number	Processor utilization p	The whole delay for simulation [msec]	Correct analytical model		Average message queue delay [msec]
			The whole delay [msec]	Relative error [%]	
4	0,24	25,44	24,34	4,52	0,01
3	0,31	25,45	24,39	4,35	0,05
2	0,47	25,94	24,83	4,47	0,30
1	0,94	51,91	54,79	5,55	15,60

For each node of tested communication network the number of node’s processors and communication channels were the same. The processors utilization was varied through various numbers of processors in each node. Because the average queue waiting time is directed proportional to node’s processor utilization, decreasing the processing utilization decrease also the delays in node’s queue. This in turn influences the whole node’s delay of communication channel. It has been remarkable that high node’s processor utilization influence considerable node’s communication delay through channel queue waiting time. The comparison of the whole delay for simulation and corrected analytical model is at the Figure 13. The comparison of their relative errors is at the Figure 14.

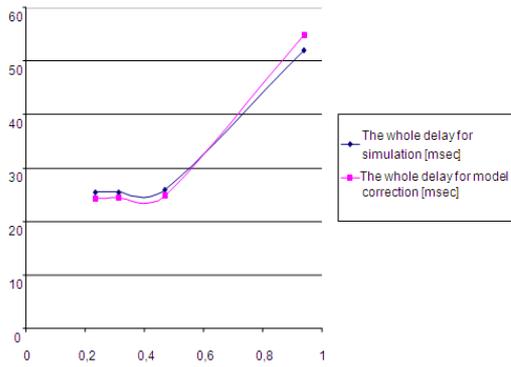


Figure 13. The accuracy of corrected model.

The relative error of the corrected model illustrates the Figure 14.

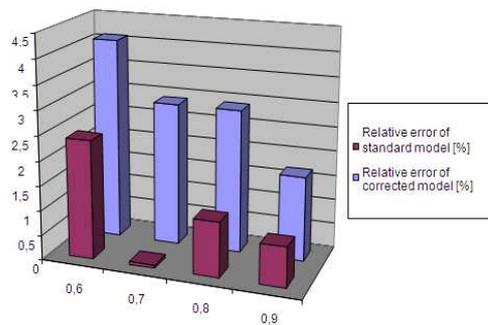


Figure 14. Comparison of relative errors.

The Tables 4 and 5 represent the results for the total message delay in alternatively analyzed 5-noded test model. Again all computing nodes have the same processors and the same communication channels to have the possibility to achieve uniform results for processor and channel utilization in individual nodes. For this tested net they were used always on processor in each node and the utilization of this processor was changed through the variation of message servicing time. Table 4 presents the results for the case of communication channel speeds of the used communication channels to 50%.

Table 4. Results for 50% communication channel utilization

Processor utilization	The whole delay for simulation [msec]	Correct analytical model		Average message queue delay [msec]
		The whole delay [msec]	Relative error [%]	
0,6	21,57	21,10	2,22	0,9
0,7	23,76	23,39	1,60	1,8
0,8	28,22	28,18	0,14	3,8
0,9	39,30	40,25	2,40	9,0
0,95	58,22	61,83	6,20	19,83

The comparison of the total message delays through simulation and corrected model for the 50% communication channel utilization is at the Figure 15.

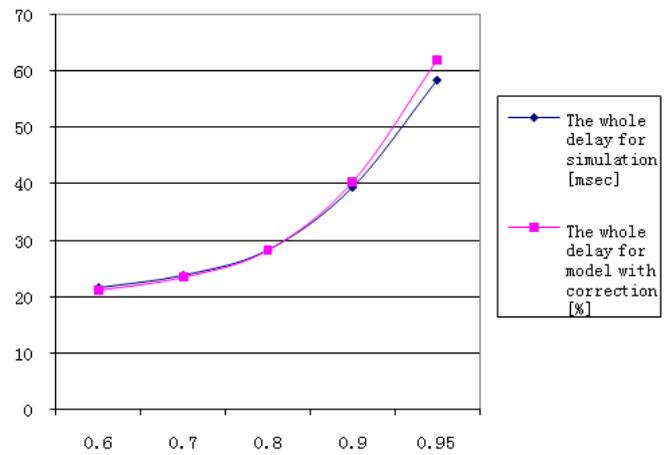


Figure 15. The comparison for 50% communication channel utilization.

Relative errors of corrected model for 50% communication channel utilization illustrate Figure 16.

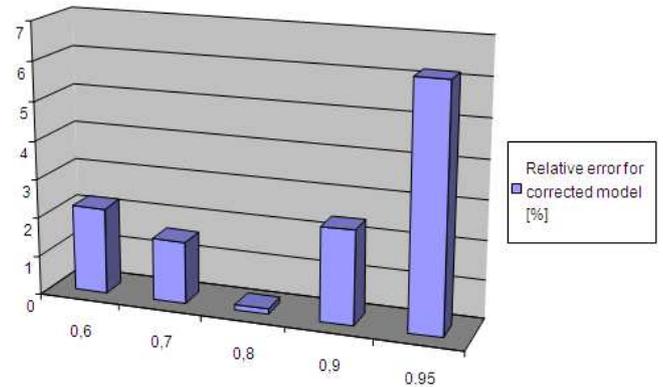


Figure 16. The relative errors for 50% channel utilization.

The portion of communication message queue delay to total communication message delay illustrates for 50% channel utilization Figure 17.

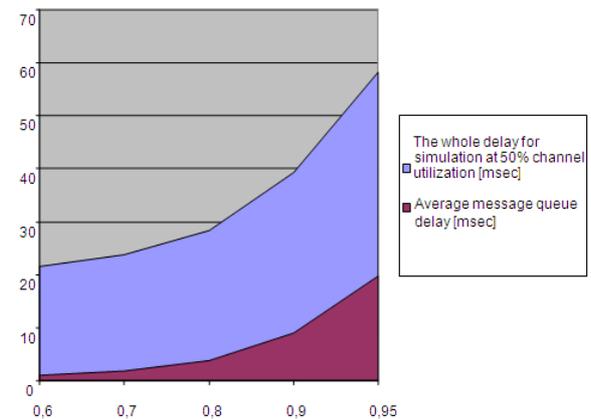


Figure 17. The portion of message queue delay to total message delay.

The Table 5 represents the results for the case of communication speed reduction to achieve the increased channel utilization to 80%.

Table 5. The results for 80% channel utilization

Processor utilization	The whole delay for simulation [msec]	Correct analytical model		Average message queue delay [msec]
		The whole delay [msec]	Relative error [%]	
0,6	72,33	68,87	5,02	0,9
0,7	74,46	71,03	4,82	1,8
0,8	78,76	75,62	4,20	3,8
0,9	90,09	87,48	3,00	9,0
0,95	104,35	108,93	4,39	19,83

The comparison of the total delays for simulation and developed analytical model at 80% utilization is at the Figure 18.

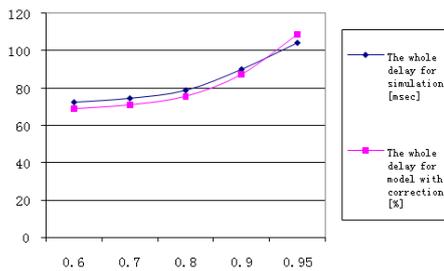


Figure 18. The comparison at 80% channel utilization.

The relative error of the corrected analytical model at 80% channel utilization illustrates Figure 18.

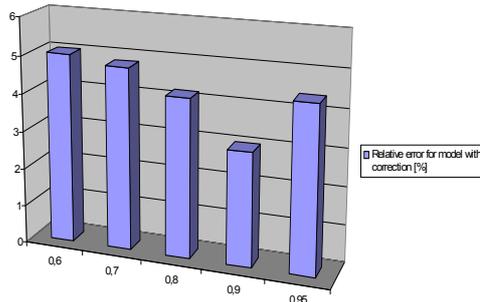


Figure 19. Relative errors at 80% channel utilization.

The influence of channel utilization to the total node influence of the utilization to whole communication channel delay illustrates Figure 20.

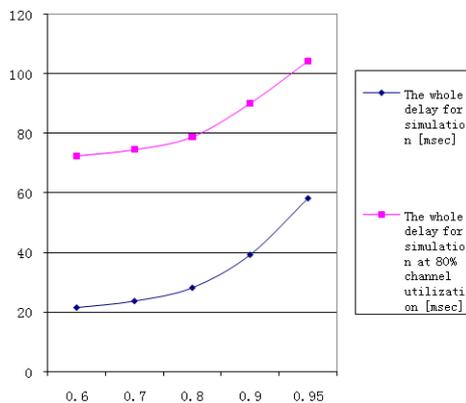


Figure 20. Influence of the utilization to the total delay.

In both considered cases the decreasing of processor utilization causes the decreasing of average node’s channel communication delay. Therefore the communicating data are waiting in the node’s channel queue shorter time and the total communication channel delay is lower. In contrary the decreasing of communication channel speed increase the channel utilization and then the communicating data have to wait longer in the communication channel queues and increase the total message node delay. These facts are clear also from the achieved results because all achieved results in Table 5 are greater than in Table 4.

The corrected analytical model allows modeling of delay influences through computation complexity of performed parallel processes influences of node’s communication activities and also to correct violation influence of exponential inter arrival time distribution. The tested results has proved, that the corrected analytical model provides very precision results in the whole range of input workload of processor utilization, communication channels and network topologies with relative error, which does not exceed 6% and in most cases were in the range up to 5%.

To prove the effectiveness of corrected analytical model it is not satisfactory to document its accuracy but also his improvement in relation to standard analytical model. It is necessary to emphasize, that the effect of considering the delay influence caused through the processors communication activities to the total message delay intensify mainly under two following conditions

- if the average number of passed nodes at the message transport is high, the message queue delay has inconsiderable effect to the total message delay
- if the processor utilization is higher the average message queue waiting time is increased what influence the whole delay.

Table 6 represents the results and relative error for the average value of the total message delay in the 7 – nodes network through corrected analytical model and standard model.

To vary the processor utilization we modified the extern input flow in the same manner for each used node. For both analytical models (standard, corrected) are in Table 6 represented the whole delays (end-to-end) and relative errors in relation to simulated results. Comparison of whole delays illustrates Figure 21.

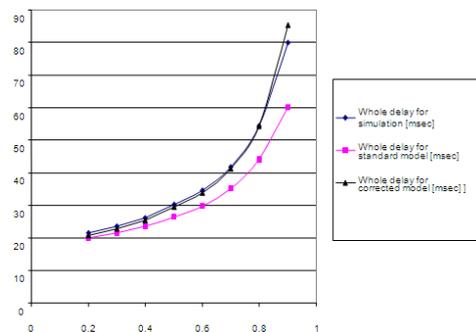


Figure 21. Comparison of the used models.

Table 6. Comparison of considered models.

The whole Processor delay for utilization simulation [msec]	Standard analytical model		Corrected analytical model		
	The whole delay [msec]	Relative error [%]	End to end delay [msec]	Relative error [%]	
0,2	21,45	20,06	6,48	20,83	2,89
0,3	23,53	21,58	8,29	22,85	2,89
0,4	26,24	23,49	10,48	25,51	2,78
0,5	30,16	26,51	12,10	29,44	2,39
0,6	34,69	29,79	14,12	33,92	2,22
0,7	41,67	35,19	15,55	41,38	0,70
0,8	54,25	44,08	18,75	54,43	0,33
0,9	80,01	60,38	24,53	85,47	6,82

The relative errors comparison of both analytical models to simulation results according Figure 22 shows clearly the improvement of corrected analytical model. If the processor utilization varied from 20 to 90% the relative error of standard model changes from 6 to 24%. This is due the influences of communication message queue delays, the nature of inter arrival input to communication channels in the case of high processor utilization. In contrary the corrected analytical model in all cases has the relative number not greater than 7%. The achieved results in Table 6 indicate also other important critical fact. The derived corrected model produces more precise results in the whole range of node's processor utilization including the range of their higher utilization (in range 0,5 – 0,9) which are the most interesting to practical use.

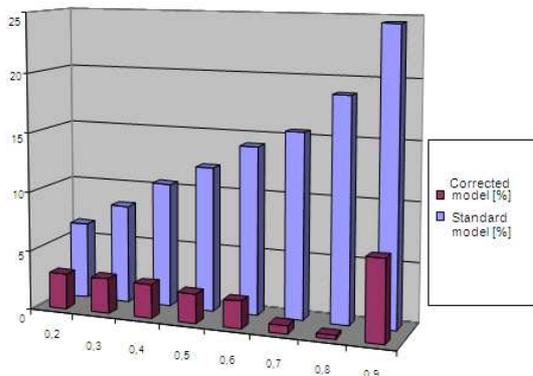


Figure 22. The comparison of relative errors.

We also point out, that accuracy contribution of corrected analytical model was achieved without the increasing the computation time in comparison to standard analytical model. It is also remarkable to emphasize increasing influence of the simulation complexity for the analysis of real massive parallel computers including their communication networks. The simulation models require three orders of magnitude more computation time for testing such complex parallel systems.

6. Conclusions

Performance evaluation of existed computers (sequential, parallel) used to be a very hard problem from their birthday.

This involves the investigation of the control and data flows within and between components of computers including their communication networks. The aim is to understand the behavior of the systems, which are sensitive from a performance point of view. It was, and still remains, not easy to apply typical analytical method (queuing theory) to performance evaluation of used computers because of their high number of not predictable parameters and for sequential computers the existence of only one control stream. Using of actual parallel computers (SMP-multiprocessor, multicore, NOV, Grid) open more possibilities to apply a queuing theory results to analyze more precise their performance. This imply existence of many inputs streams (control, data), which are inputs to modeled queuing theory systems and which are generated at various used resources by chance (theoretic assumption for good approximation of Poisson distribution). Therefore we could model computing nodes of parallel computers as M/D/m or M/M/m and their communication channels as M/D/1 or M/M/1 queuing theory systems in any existed parallel computer.

In relation to it this paper describes the deriving and testing of a correction factor of used standard analytical models in order to extend behavior analysis of parallel computers with another more precise analytical model. The developed corrected standard model was extensively tested and the results were compared with both standard analytical model and simulation results too. The results clearly show that the correction factor contributes to better results with a negligible increase in processing time. Its advantage, in comparison to possible used simulation method, is its ability to analyze also large existed communication networks of massive parallel computers (MPC). In this way we hope that more precise corrected analytical model would help in effective resource projecting of suggested parallel computers and parallel algorithms (effective PA) too. In this way parallel computing on networks of conventional personal workstations (single, multiprocessor) and Internet computing, awaits to unify advantages of parallel and distributed computing at their performance modeling too. In summary according input technical parameters of parallel computer we can apply to performance modeling of developed analytical models in actually typical following cases

- NOW based on workstations (single, multiprocessors or multicores, mix of them)
- Grid (network of NOW networks)
- mixed parallel computers based on SMP, NOW and Grid
- metacomputers (massive Grid etc.).

From a point of user application of any analytical method is to be preferred in comparison with other possible methods (simulation), because of universal and transparent character using of achieved results in form of used parameters. We can apply developed more precise analytical models to performance modeling of any parallel computer or parallel algorithm too. To practical applied

using we would like to advise following

- running of unbalanced parallel processes where λ is a parameter for incoming parallel processes with their exponential service time distribution as $E(ts) = 1/\mu$ (corrected standard model)

- in case of potential considering incoming units of parallel processes (data block, packet etc.) at using model based on M/M/m and M/M1 queuing theory systems it would be necessary to recalculate at entrance incoming parallel processes to wanted data units. The way how to recalculate them to such units at first node entrance we would like to refer in next paper

- running of parallel processes (λ parameter for incoming parallel processes with their deterministic service time $E(ts) = 1/\mu = \text{constant}$). The deterministic servicing times are a very good approximation of balanced parallel processes (M/D/m) with nearly equal amount of communication data blocks for every parallel process (M/D/1)

- in case of using analytical model using M/D/m and M/D/1 we can consider λ parameter also for incoming units of parallel processes (data block, packet etc.) with their average service time for considered unit t_i , where $E(ts) = 1/\mu = t_i = \text{constant}$.

Using developed more precise analytical models (in [9] and in this paper too) we are able to apply them so to SMP parallel computers (parallel computing) as to dominant distributed computers (NOW, Grid, metacomputer). In developed unified parallel computer models it is possible to study load balancing [2], inter-process communication (IPC) mechanisms, transport protocols, performance prediction etc. We would also like to analyze

- blocking problem (consequence of exhausted used resources)

- extending waiting times caused by limited technical resources (blocking consequence)

- the role of various routing algorithms

- to prove, or to indicate experimentally, the role of the supposed independence assumption, if you are looking for higher moments of delay [20]

- to verify the suggested model also for node limited buffer capacity and for other servicing algorithms than assumed FIFO (First In First Out)

- necessary unified decomposition strategies for parallel and distributed computing

- intensive testing [17], measurement and observation in order to obtain estimates of such system variables [5, 24].

Acknowledgements

This work was done within the project “Modeling, optimization and prediction of parallel computers and algorithms” at University of Zilina, Slovakia. The author gratefully acknowledges help of project supervisor Prof. Ing. Ivan Hanuliak, PhD.

References

- [1] Abderazek A. B., Multicore systems on chip - Practical Software/Hardware design, Imperial college press, 200 pp., 2010
- [2] Arora S., Barak B., Computational complexity - A modern Approach, Cambridge University Press, 573 pp., 2009
- [3] Dattatreya G. R., Performance analysis of queuing and computer network, University of Texas, Dallas, USA, 472 pp., 2008
- [4] Dubois M., Annavaram M., Stenstrom P., Parallel Computer Organization and Design, 560 pages, 2012
- [5] Dubhash D.P., Panconesi A., Concentration of measure for the analysis of randomized algorithms, Cambridge University Press, UK, 2009
- [6] Gelenbe E., Analysis and synthesis of computer systems, Imperial College Press, 324 pages, 2010
- [7] Giambene G., Queuing theory and telecommunications, 585 pp., Springer, 2005
- [8] Hager G., Wellein G., Introduction to High Performance Computing for Scientists and Engineers, 356 pages, July 2010
- [9] Hanuliak M., Hanuliak P., Performance modeling of parallel computers NOW and Grid Vol. 2/5, Am. J. of Networks and Comm., Science GP, USA, 112-124 pp., 2013
- [10] Hanuliak J., Hanuliak I., To performance evaluation of distributed parallel algorithms, Kybernetes, Volume 34, No. 9/10, UK, 1633-1650 pp., 2005
- [11] Hanuliak P., Hanuliak I., Performance evaluation of iterative parallel algorithms, Kybernetes, Volume 39, No.1, UK, 107-126 pp., 2010
- [12] Hanuliak P., Analytical method of performance prediction in parallel algorithms, The Open Cybernetics and Systemic Journal, Vol. 6, Bentham Open, UK, 38-47 pp., 2012
- [13] Hanuliak P., Complex performance evaluation of parallel Laplace equation, AD ALTA – Vol. 2, issue 2, Magnanimitas, Czech republic, 104-107 pp., 2012
- [14] Harchol-Balter Mor, Performance modeling and design of computer systems, Cambridge University Press, UK, 576 pp., 2013
- [15] Hillston J., A Compositional Approach to Performance Modeling, University of Edinburgh, Cambridge University Press, UK, 172 pages, 2005
- [16] Hwang K. and coll., Distributed and Parallel Computing, Morgan Kaufmann, 472 pages, 2011
- [17] John L. K., Eeckhout L., Performance evaluation and benchmarking, CRC Press, 2005
- [18] Kshemkalyani A. D., Singhal M., Distributed Computing, University of Illinois, Cambridge University Press, UK, 756 pages, 2011
- [19] Kirk D. B., Hwu W. W., Programming massively parallel processors, Morgan Kaufmann, 280 pages, 2010

- [20] Kostin A., Ilushechkina L., Modeling and simulation of distributed systems, Imperial College Press, 440 pages, 2010
- [21] Kumar A., Manjunath D., Kuri J., Communication Networking, Morgan Kaufmann, 750 pp., 2004
- [22] Kushilevitz E., Nissan N., Communication Complexity, Cambridge University Press, UK, 208 pages, 2006
- [23] Kumar A., Manjunath D., Kuri J., Communication Networking, Morgan Kaufmann, 750 pp., 2004
- [24] Kwiatkowska M., Norman G., and Parker D., PRISM 4.0: Verification of Probabilistic Real-time Systems, In Proc. of 23rd CAV'11, Vol. 6806, Springer, 585-591 pp., 2011
- [25] Le Boudec Jean-Yves, Performance evaluation of computer and communication systems, CRC Press, 300 pages, 2011
- [26] McCabe J., D., Network analysis, architecture, and design (3rd edition), Elsevier/ Morgan Kaufmann, 496 pages, 2010
- [27] Miller S., Probability and Random Processes, 2nd edition, Academic Press, Elsevier Science, 552 pages, 2012
- [28] Misra Ch. S., Woungang I., Selected topics in communication network and distributed systems, Imperial college press, 808 pages, 2010
- [29] Natarajan Gautam, Analysis of Queues: Methods and Applications, CRC Press, 802 pages, 2012
- [30] Peterson L. L., Davie B. C., Computer networks – a system approach, Morgan Kaufmann, 920 pages, 2011
- [31] Resch M. M., Supercomputers in Grids, Int. J. of Grid and HPC, No.1, 1 - 9 pp., 2009
- [32] Riano I., McGinity T.M., Quantifying the role of complexity in a system's performance, Evolving Systems, Springer Verlag, 189 – 198 pp., 2011
- [33] Ross S. M., Introduction to Probability Models, 10th edition, Academic Press, Elsevier Science, 800 pages, 2010
- [34] Wang L., Jie Wei., Chen J., Grid Computing: Infrastructure, Service, and Application, CRC Press, 2009
- [35] www pages
- [36] www.top500.org
- [37] www.intel.com