VOL. 11, NO. 23, DECEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

DEVELOPMENT AND RESEARCH OF OPEN-LOOP MODELS THE SUBSYSTEM "PROCESSOR-MEMORY" OF MULTIPROCESSOR SYSTEMS ARCHITECTURES UMA, NUMA AND SUMA

A. I. Martyshkin
Penza State Technological University, Baydukov Proyezd / Gagarin Street, 1a/11, Penza, Penza region, Russia
E-Mail: dmitry.shkurkin@gmail.com

ABSTRACT

The articles explore a problem of the synthesis and analyze models of subsystem "processor-memory" multiprocessor systems. The prime object is to study the architecture of the subsystem "processor-memory" of modern high-performance computing systems, calculation and comparison of the performance, as well as a conclusion on the effect of conflict over access to shared resources on the overall performance of the whole system. The objects of study of this work are the subsystem "processor-memory" multiprocessor computer systems, existing varieties of architectural construction of this subsystem. During researches the comparative analysis of systems with architecture UMA, NUMA and SUMA was carried out. Their merits and demerits were revealed. In the inference the appropriate conclusions on operation are drawn. The considered models allow making an assessment of characteristics of the multiprocessor systems without creation of a real prototype. At the expense of it economic effect as the assessment of characteristics of the designed systems and a choice of the most optimal variants can be carried out without creation of real system is reached.

Keywords: multiprocessor system, subsystem "processor-memory", architecture, mathematical model, performance, queuing system.

1. INTRODUCTION

Increased productivity computing systems is directly related to the increasing of speed and storage capacity. Although the rapid growth of memory bandwidth, which is observed in recent years, the gap "CPU-Memory" is not reduced, but on the contrary – is increasing.

To research and the analysis of computing systems even more often apply elements of the queuing theory, namely queuing systems and networks (QS and NS). Any unit, any system module it is possible to present queuing systems in the form, and set of devices of the computing system is represented a network of mass service and quite easily gives in to research in case of the minimum financial expenses – it isn't required to build real system, there is enough its model [1-3].

2. GOAL SETTING

This article is research character. To solve this problem there were analyzed the literature describing research in this subject area to search for unsolved problems. Literary sources have been analyzed [4-11]. However, a number of issues related to the study of the subsystem "processor-memory" did not adequately reflect.

Open queuing networks fairly well studied and described in detail in the sources [1-3, 12-14]. We present part of expressions and formulas are used in the generated software package in this research, the non-necessity for the study of the developed models.

3. THE CALCULATION OF CHARACTERISTICS OF QUEUING NETWORK

If the network settings are set, you can define the following characteristics of each QS and the network as a whole: the average length of the queue of applications in the *i*-th QS - l_i and in the network - L; the average number of applications residing in the *i*-th QS - m_i and network - M; the average waiting time of service applications *i*-th QS - ω_i and network - W; the average residence time applications in the *i*-th QS – u_i and the network - U [1].

The average waiting time in the queue for the application of single-channel queuing equal to the quotient of the average queue length l_i on the intensity λ_i of the input at the i-th QS flow [1]

$$\omega_i = \frac{l_i}{\lambda_i} = \frac{\upsilon_i \cdot \rho_i}{1-\rho_i}.$$

For multi-channel QS [1]

$$\omega_i = \frac{l_i}{\lambda_i} = \frac{\upsilon_i \cdot \beta_i^{k_i}}{k_i! k_i \left(1 - \beta_i / k_i\right)^2} p_{0i}.$$

The average time of the application in the system is determined by its average delay in queue and the time of service in the i-th QS.

For single-channel queuing [1]

$$u_i = \omega_i + \upsilon_i = \frac{\upsilon_i}{1-\rho_i}.$$

For multi-channel QS [1]

$$u_i = \omega_i + \upsilon_i = \frac{\upsilon_i \cdot \beta_i^{k_i}}{k_i! k_i \left(1 - \beta_i / k_i\right)^2} p_{0i} + \upsilon_i.$$

Based on the determined characteristics of individual queuing network characteristics in general.

The average number of requests waiting for service in the network (that is, the average number of applications on the network lines)

$$L = \sum_{i=1}^{n} l_i.$$

The average number of applications residing on the network

$M = \sum_{i=1}^{n} m_i$.

Because each application may receive the service in the i-th QS in the mean α time, the waiting time service and stay in her system will increase in α time. The average waiting time applications in the network lines

$W = \sum_{i=1}^{n} \alpha_i \omega_i$,

and the residence time

$U = \sum_{i=1}^{n} \alpha_i u_i$.

Development of a model of the subsystem "processor-memory" UMA's architecture [4, 8, 9]. The structure of the UMA architecture (Unified Memory Access) and the count of transmissions is presented in Figure-1(a) and (b).

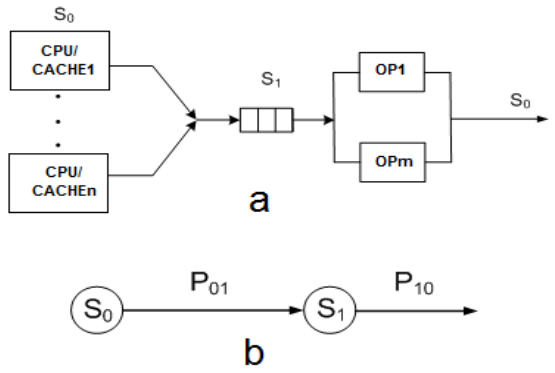

Figure-1. The structure (a) and the transmission graph (b) of UMA-system.

The figure shows: CPU/CACHE - n processing nodes; OPm – m memory modules; S0 - source applications, representing the n processing nodes; S1 - m memory modules in the form of a multi-channel device-servicing.

It is considered that applications create the simplest flows of requests, and holding times submit to the exponential law. This distribution will allow receiving results of certainly worse real values that, in turn, will allow making an assessment of the received results on top.

It is considered that memory has uniform address space, and controllers of memory contain in the composition buffer registers for data storage, written in memory or read from memory, having the volume sufficient in order that requests were not refused in service.

4. AN EXAMPLE AND SIMULATION RESULTS

To study this architecture we will take the Intel Pentium IV. The volume of cache L2 of the processor series Pentium 4 500 - 1 MB. Intel Pentium 4 500 series have a bus frequency of 200 MHz (the quantum duration T=5 ns) [15, 16], denoted by based technology Quad Pumped Bus as 800 MHz. This bus per cycle can transmit 4 ready to transfer 64-bit words. But if you are ready to transfer only one word and it will be transmitted for 1 clock cycle.

CPU clock frequency is taken to be Intel Pentium 4 500-2800 MHz. It is expected that the treatment will be made in the memory of every clock cycle. The data cache memory, according to statistics gets at least 99% of the requests. Therefore, based on the processor speed of 2, 8 GHz, we find that the frequency of queries in memory will be $\lambda = 2,8 \cdot 1\% = 0,028$ query/ns.

At the study of the CPU performance impact on the operation of the subsystem "processor-memory" discussed Intel Pentium 4 500 processor - 571 with a clock frequency 2,8-3,8 GHz [17-19].

The calculations use DDR memory with a frequency of 200 MHz. Because DDR bus standard labeled as 400 MHz. This means that can transmit two 64-bit ready word in one cycle length of 5 ns memory bus [15].

Memory Timings CL-RCD-RP according to the chip are 15 ns each (3-3-3).

The best case occurs when there is an appeal to open-line memory module. It is necessary to submit only CAS signal, and the memory module to produce data at an interval CL (15 ns). Statistically, this situation occurs in 55% of cases [20].

The worst case is when this is in a different row. At the same time signals are RAS (activation of the line) and after a time RCD-CAS (column activation). Thus, the memory module will require a time equal to CL + RCD (30 ns), to prepare the data. This embodiment is obtained with a probability of 40% [21].

Worst case, when the data is on an inactive page. In this case, the current page should be closed. The preparation of the data memory module requires a time equal to CL + RCD + RPT (45 ns). This variant accounts for 5% of the cases [22].

We define the mean time OP module $V_{OP} = 0,55 \cdot 15 + 0,4 \cdot 30 + 0,05 \cdot 45 = 22,5$ ns.

Calculate how much you want to bus time for the transmission controller 32-bit address. Bit bus – 64 bit. That is a 32-bit address will be given for 1 cycle $1 \cdot 5 = 5$ ns. For a 64-bit word from the memory to the processor through the controller requires one bus cycle "memory controller" and 1 bus cycle "controller-processor" $1 \cdot 5 + 1 \cdot 5 = 10$ ns.

By summing the values obtained, we obtain the average time to access memory: 22,5 + 5 + 10 = 37,5 ns. Thus, obtained during memory accesses of the processor.

Analysis of the impact of efficiency of the cache on the real capacity of the subsystem "processor-memory". Initial data: the number of service channels (OP units) in the NS K = 12; Load the number of sources (CPUs) CPU = 4; Queuing time one channel (the OP module) ν = 37, 5 ns [23, 24].

The intensity of the flow of requests in the simulation was changed as follows: the probability of a cache miss, %•(CPU clock frequency). Range probability of cache

misses - 0, 5-3%. The simulation results are shown in Figure-2.

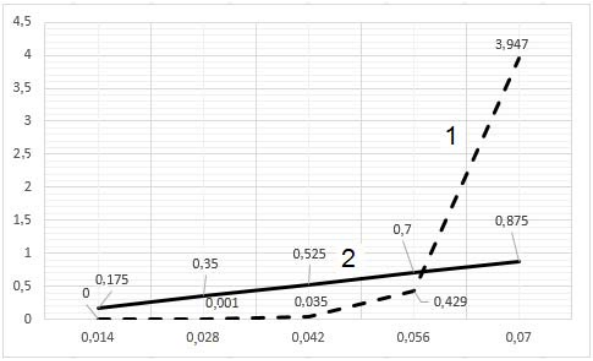

Figure-2. The dependence of load factor (1), and the length of the queue in front of the device (2) the probability of a cache miss.

When hit in the cache is 99 % of the requests the average queue length l is almost equal to 0, the response time consists of memory access time to the memory u = 37,507.

When the flow of requests 0,070 request/ns (size of cache-misses is 2, 5 %) queue length l=3,974 applications, the waiting time of requests in the queue (ω =14,098 ns) and a stay application to the QS (response time memory) u=51,598 ns), i.e. increased by 28 % [25].

When the cache-miss is 3 % the system is overloaded (ρ =1,05), as it reduces the throughput of the subsystem "processor-memory".

By increasing the effectiveness of the cache in 2 times (the number of cache hits is 99, 5 %) in queue are no requests, waiting time in the queue is 0.

Analysis of the impact of the number of processor nodes on the throughput of the subsystem "processor-memory". Initial data: the number of service channels (OP units) in the NS K = 8; number of load sources (CPUs) CPU = 2-8; service time applications with one channel (the OP module) ν = 37, 5 ns; the intensity of the flow of requests λ = 0,028 query/ns. The simulation results are shown in Figure-3.

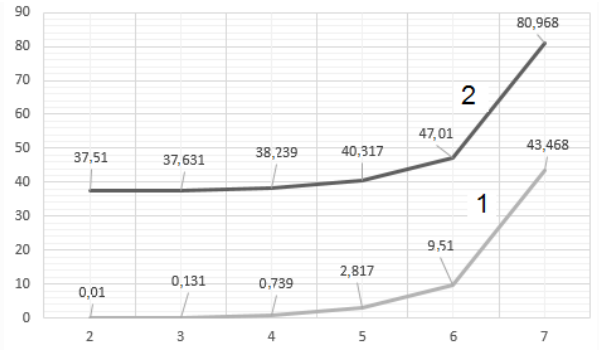

Figure-3. The dependence of the waiting time in the queue (1) and response time memory (2) of the number of processor nodes.

When CPU =2-5 in the studied system queue length l<1 (from 0,001 to 0,394 applications), the waiting time in queue is from 0,01 to 2,817 ns.

When CPU =6-7 number of entries in the queue reaches 8,52 applications, the waiting time in the queue increases to 43,468 ns, response time, memory is 80,968, which is 2,2 times larger than the value when CPU =2.

When CPUs =8 in the system experiencing the overload ρ=1,05.

Analysis of the impact of the number of memory real capacity act-the capacity of the subsystem "processor-memory". Initial data: the number of service channels (OP units) in the NS K = 4-10; sources of loads (CPUs) M= 4; the service time of requests with a single channel (the OP module) ν=37,5 ns; the intensity of the flow of requests λ=0,028 query/ns. Results are shown in Figure-4.

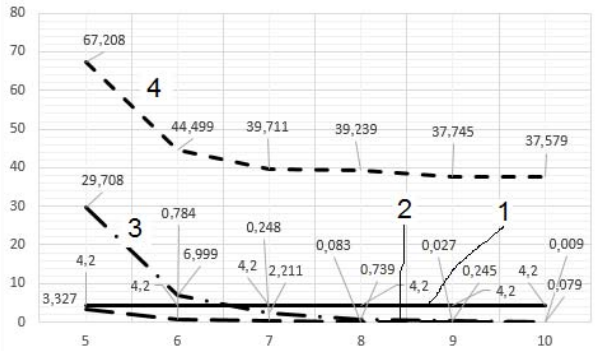

Figure-4. Dependence of average number of busy channels (1), queue length in front of the device (2), waiting time in the queue (3) and response time memory (4) by the number of memory modules.

As seen from the figure, the average number of occupied channels in the system for a given task flow intensity of 4, 2, i.e. no pre-exceeds 5. The average number of customers in the system at K> 5 and not more than 5. Thus, the optimal number of memory modules 6. It is confirmed by other characteristics, as for K>6 system characteristics change slightly and tend to 0.

When K = 10, the number of requests in the queue l close to 0, as well as the waiting time in the queue ω. The, response time, memory u consists of the memory access time.

Analysis of influence of time of storage access for real throughput of a subsystem "processor memory". As basic data for time of storage access mean value is taken, in real system it can be more than this value. Therefore, it is important to analyze model with bigger value of time of a memory access with a research objective of limit values.

Initial data: the number of service channels (OP units) in the NS K = 8; Load the number of sources (CPUs) CPU = 4; service time applications with one channel (the module OP) ν = 37,5-40 ns; the intensity of the flow of requests λ = 0,028 query/ns. Results of the study are shown in Figure 5.

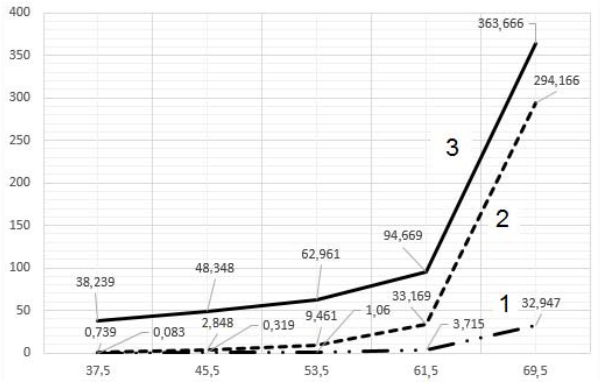

Figure-5. Dependence of the queue length before the device (1), the waiting time in the queue (2) and the response time of the memory (3) from the memory access time.

The figure shows that the critical state of the system is observed with considerable delay memory response (up to 77, 5 ns). This channel load factor ρ is 1,085 - overload occurs. With a slight increase in the time dos dumb to the memory (up to 45,5 ns) memory response time is increased by 10 ns compared with the value of the original system, the number of requests in the queue is less than 0,319 applications. These values are not critical, so little effect on system performance.

Development of a model of the subsystem "processor-memory" architecture NUMA. The structure of NUMA (Non Unified Memory Access) and the count of transmissions is presented in Figure 6, a and b.

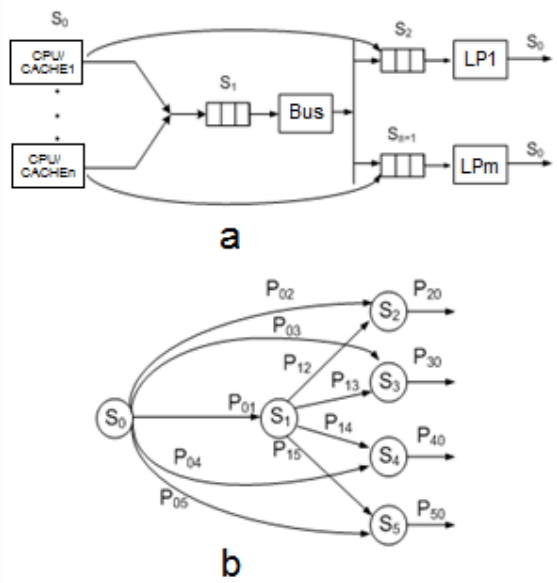

Figure-6. The structure (a) and the transmission graph (b) of the NUMA-system.

Here: CPU/CACHE - n processing nodes; Bus - common bus; LPm - m units of local shared memory; S_0 - source applications, representing a 4-weed assembly processes; S_1 - tire; S_2-S_5 - 4 local shared memory module.

Determination of initial data for NUMA architecture. This architecture is also seen on the example of a family of Intel Pentium IV 500. L2 cache processors series Pentium 4 500 - 1 MB.

Therefore, the frequency of requests in OP remains unchanged, namely λ=0,028 query/ns. When you study the effects of CPU performance on the subsystem "processor-memory" discusses the Intel Pentium 4 521 - 571 with a clock frequency of 2,8-3,8 GHz.

In the calculations, we use the DDR memory modules with a frequency of 200 MHz. On the module's OP fair the calculations made above. V_{OP} = 22, 5 (ns)

Since in the model this architecture the bus "controller-processor" is a separate element, then the average time of memory access will consist of the average time-on operation OP and 1 quantum bus "memory controller" to transfer words from memory. Received: 22, 5+5 =27,5 ns. If the processor access its local memory module, the average time of memory access will amount to 27, 5 ns. When handling the CPU module to the local memory of another processor node would require an additional 2 cycles of the bus "controller-processor": the transmission controller 32-bit address and to transfer 64-bit words from the memory via the controller to the processor. 5•2=10 ns. Thus, the bus delay will be 10 ns.

We also believe that 90 % of requests of processor nodes flow to its local memory module, and 10 % of appeals to the local memory module of another processor node.

Analysis of the impact of efficiency of the cache on the real capacity of the subsystem "processor-memory". Initial data: the number of OP units in the NS K = 4; Load the number of sources (CPUs) CPU = 4; service time applications with one channel (the module OP) ν = 27,5 ns; latency bus - 10 ns.

The intensity of the flow of requests in the simulation was changed as follows: the probability of a cache miss,%•(CPU clock frequency). Range probability of cache misses - 0, 5-3%. Results of the study are shown in Figure-7.

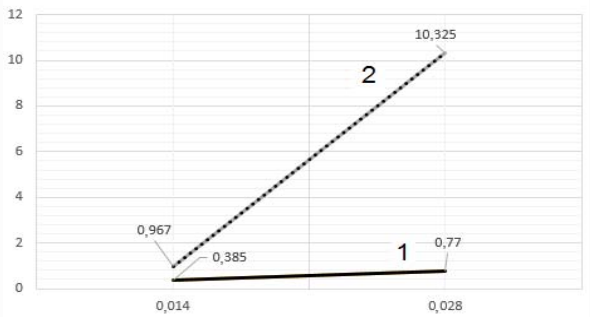

Figure-7. Dependence of load factor (1), and the length of the queue in front of the device (2) the probability of a cache miss.

When hit in the cache 99 % of the requests (the flow of applications to the memory of the processor request 0,028 query /ns): the average number of customers in the system m = 13,517 applications; the average queue length l = 10,325

applications; latency applications in the queue ω= 92,191 ns; the residence time of application to the QS (a memory response time) = 120,691 ns.

When the flow of applications query 0,042 query/ns (the value of a cache miss is 1, 5%) observed in the system overload - ρ = 1,155.

By increasing the efficiency of the cache memory 2 times (the number of cache hits is 99, 5%) Response time memory is 45,774 ns, which is compared with the original one is better about 38%. At the same queue length l <1.

Analysis of the impact of the number of processor units and memory modules on the real capacity of the subsystem "processor-memory". Initial data: the number of OP units in the NS K = 4-7; number of load sources (CPU) CPU = 4-7; service time applications with one channel (the module OP) ν = 27, 5 ns; latency bus – 10 ns; the intensity of the flow of requests λ = 0,028 query/ns. Results of the study are shown in Figure-8.

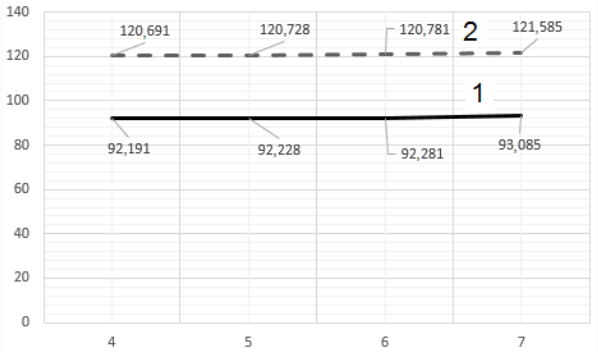

Figure-8. The dependence of the waiting time in the queue (1) and the memory response time (2) of the number of processor units and memory modules.

Despite identical time of storage access, the queue length, number of requests in system, wait time in queues and time of the response of memory increase in case of increase in number of processing nodes and modules of memory. It occurs in connection with increase in probability of the appeal of several processing nodes to the remote module of memory.

In general, an increase in performance slightly (the memory response time increases by about 1 ns). Such as 90% of storage accesses make processing node to "the" local module.

Analysis of influence of time of storage access for real throughput of a subsystem "processor memory". As basic data for time of storage access mean value is taken, in real system it can be more than this value. Therefore, it is important to analyze model with bigger value of time of a memory access with a research objective of limit values.

Initial data: the number of OP modules in the NS K = 4; load the number of sources (CPUs) CPU = 4; service time applications with one channel (the module OP) ν = 27, 5-37,5 ns; latency bus – 10 ns; the intensity of the flow of requests λ = 0,028 query/ns. Results of the study are shown in Figure-9.

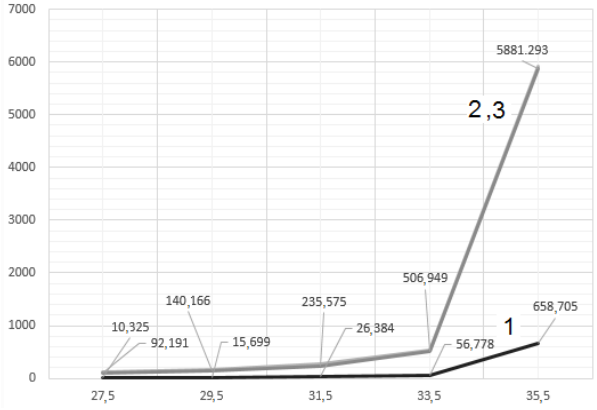

Figure-9. The dependence of the queue length before the device (1), the waiting time in the queue (2) and the response time of the memory (3) from the memory access time.

With a significant increase in the memory access time (up to 29, 5-31, 5 ns) response time, memory increases in 1,4-2,2 times, respectively. With a further increase in the value of access to the memory at 35,5 ns delay before the response of memory reaches 5917,793 ns, which is 49 times larger than the original value. System overload occurs at 37, 5 ns memory access.

Development of the "processor-memory" model subsystem SUMA's architecture. The proposed architecture AMD SUMA combines the advantages of UMA and NUMA allowing you to create a flexible and highly scalable computing systems (Figure 10, a). Here, each CPU has its own memory controller and high bandwidth interprocessor bus Hyper Transport (HT) allows you to get rid of the huge delays when accessing memory of another CPU. AMD has finally named their scheme is not NUMA, but SUMA - Slightly Uniform Memory Architecture, which is "almost uniform" memory architecture.

Basis of SUMA - serial bus Hyper Transport. The server versions of AMD processors can be integrated with up to three independent records HT, operating at frequencies up to 1 GHz (2 GHz DDR, taking into account the data transfer mode) and a width of 16 bits (4 GB/s) in each direction. Part HT-links used for point-to-point connections between the CPU, the part is activated to connect peripheral devices. Each CPU integrated controller "local" OP. Memory access "foreign" place the CPU by HT bus, and done this "redirection" of requests is absolutely transparent for the actual computing core CPU - it provides a built-in switch (Crossbar), running at full CPU frequency. It provides the "automatic" route passing through the CPU messages from peripherals and other CPUs, including the service of "foreign" to the OP requests.

Historically HT was developed by AMD as the processor bus of new generation especially for architecture

of AMD64. This bus is urged to provide throughput not smaller, than at OP, and the minimum time delays on data transfer and messages.

Data is transferred by the scheme DDR - there is an additional line for the clock signal, data is synchronized at the beginning and end of each clock pulse (that is, data is transmitted per clock cycle twice). The base clock frequency HT bus - 200 MHz (i.e., the frequency of data transmission - 400 MHz). All subsequent clock speeds are defined as multiples of a given - 400 MHz, 600 MHz, 800 MHz (HT 1.0-1.1), 1000MHz (last revision HT 1.x and HT 2.0), 1200 and 1400 MHz (HT 2.0).

Mathematical model of architecture SUMA and the transmission graph shown in Figure-10 (b), (c).

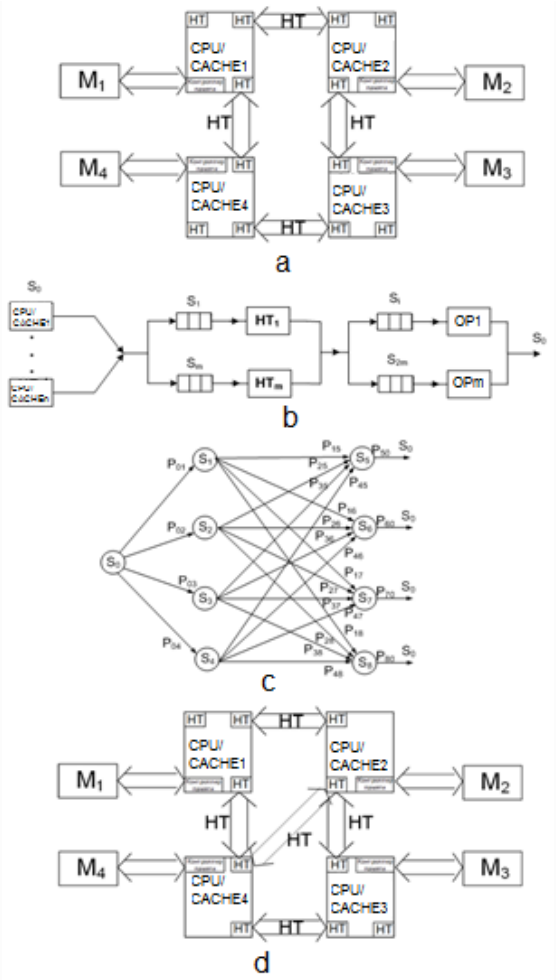

Figure-10. A four-computer system with an architecture based on SUMA (a), model (b) Transmission Count (c), a variant of a four computer system architecture SUMA (d).

Here: CPU/CACHE - n processing nodes; OP - memory modules; HT - Hyper Transport channels; S_0 - processor nodes; S_1-S_4 - Hyper Transport bus; S_5-S_8 - memory modules.

Bus HT specifically optimized for this mode of operation and provides extremely low latency treatment in "another" memory, and high (up to 4 GB/s) bandwidth when accessing memory adjacent modules. The bus is full duplex, i.e. can simultaneously transmit data at this rate in the "both sides" (8 GB/s total). The memory model is obtained an inhomogeneous, but the differences in "their" speed and "foreign" sites OP obtained relatively small.

Such an organization of the memory subsystem has one important fact with respect to each of the specified CPU OP can be not only "their" (belonging to the memory controller of the CPU) and "foreign" (owned by a neighboring controller). In the case that access to the OP with TSP1, located in the address space TSP3 controller, electrical signals need to pass in two serially connected bus HT (CPU1> CPU2> CPU3 or CPU1> CPU4> CPU3). If made not 4 I/O channel, but 2 – the two connection HT freed can be joined with each other. The resulting structure is shown in Figure-10(d).

There is the possibility of building eight-processor systems of this architecture. Thus at the extreme CPU 4 on HT given the same bus for input/output, and in the central – all three are involved in an interprocessor communications. However, it should be noted that, presumably, in this system the delay would increase strongly.

The initial data for the architecture of SUMA. Example of x86-64 architecture will take CPU AMD Athlon64 class a frequency of 2800 MHz (Athlon 64 FX 57). The frequency of HT is for the model 1000 MHz (Tcycle = 1 ns).

When studying the effect of CPU performance on the subsystem "processor-memory" reviewed CPU AMD Athlon64 class with a clock frequency of 2,2-2,8 GHz.

Regarding the OP module fair the calculations made above. Vop = 22,5 ns. Transfer 64-bit words from the module memory at 200 MHz the memory bus takes 1 clock cycle. 1•5= 5 ns.

Thus, the average address to local memory will occur at the time of 22.5 + 5 = 27, 5 ns.

To transfer 32-bit address data from the CPU to the controller via high speed internal bus believe that it takes one clock cycle. 1•1 = 1 ns.

This same word will be passed to CPU for 1 clock cycle internal bus 1•1 = 1 ns.

Obtain, taking into account the mathematical model is constructed, all of which appeal to the bus will be 2 ns.

Analysis of the impact of efficiency of the cache on the real capacity of the subsystem "processor-memory". Initial data: the number of modules in the OP QS = 4; the number of CPU load source = 4; service time applications with one channel (the module OP) ν = 27, 5 ns; HT bus latency = 2 ns.

The intensity of the flow of requests in the simulation was changed as follows: the probability of a cache miss, %•(CPU clock frequency). Range probability of cache misses - 0,5-3%. The results of the study are shown in Figure-11.

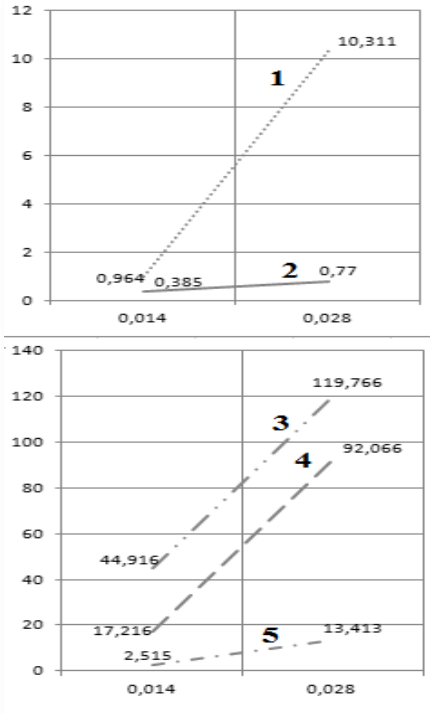

Figure-11. The dependence of the queue length before the device (1), a load factor (2), the response time of memory (3), the waiting time in the queue the application (4) and the average number of orders in the system (5) of the probability of a cache miss.

When hit in the cache 99 % of the queries (the flow of applications from CPU to memory 0,028 query/ns): the average number of tasks in the system, m = 13,413 tasks; the average queue length l = 10,311 tasks; task waiting time = 92,066 ns; the residence time of the task in the NS (memory response time) = 119,766 ns.

When the flow of tasks query 0,042 / ns (the value of a cache miss is 1, 5%), the system is overloaded (load factor is 1,155).

By increasing the efficiency of the cache memory 2 times (the number of cache hits is 99, 5%) Response time memory is 44,916 ns, which is compared with the original one preferably about 37, 5%.

Analysis of the CPU performance impact on the real capacity of the subsystem "processor-memory". Initial data: the number of modules OP in the QS = 4; the number of CPU load source = 4; service time applications with one channel (the module OP) n = 27,5 ns; HT bus latency = 2 ns; Task flow rate l = 0,028-0,038 query/ns.
The simulation results are shown in Figure-12.

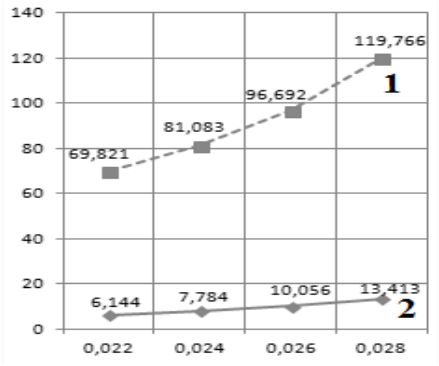

Figure-12. Dependence of the memory response time (1) and the average number of tasks in the system (2) of the CPU performance.

It is obvious that with the growth of CPU time, memory response increases (from Figure 7 – from 69,821 to 119,766 ns). The value of this characteristic is increased by 1, 7 times. The average number of tasks in the system increases from 6,144 to 13,413, i.e. more than 2 times. The waiting time in the queue is also increased approximately 2-fold.

Analysis of the influence of the time of memory access to the real capacity of the subsystem "processor-memory". In this architecture, there is the likelihood that the block of memory IP, located in 2 connected in series tires. With this in mind, the memory access time is increased by 2 ns. Let's consider this case.

Initial data: the number of service channels (OP units) in the QS = 4; the number of CPU load source = 4; time maintenance tasks one channel (the module OP) ν = 27, 5-29, 5 ns; HT bus latency = 2 ns, task flow rate λ = 0, 028 query/ns.

The resulting graph is shown in Figure-13.

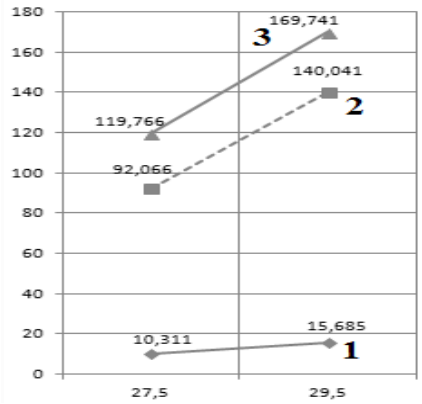

Figure-13. Dependence applications queue length (1), the waiting time in the queue (2) and the response time of the memory (3) from the memory access time.

Based on the results, we find that when referring to a local unit PU remote queue length l memory is increased by 1, 5 times, while waiting in the queue increases by 47,275 ns and a memory response time is

increased by 30% compared with the reference to the adjacent memory module CPU.

Comparison of simulation results for SUMA and NUMA architectures. Features of the system based on SUMA architecture with a four organizations slightly less from such a system, NUMA architecture, due to a similar organization. At the same time congestion and delay at a high-speed rail a lot less. Tire load results are shown in Figure-14.

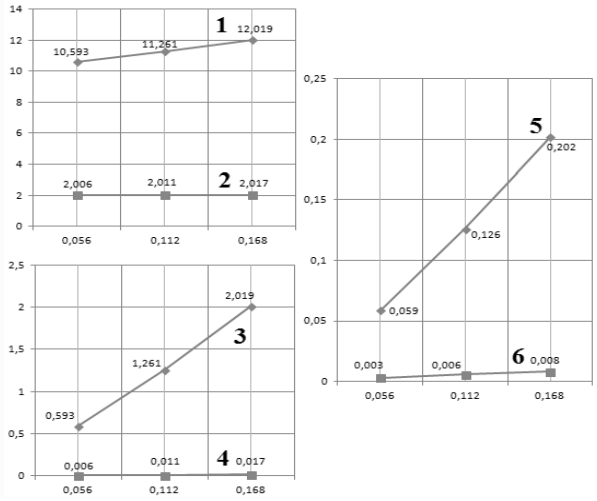

Figure-14. Dependence of the memory response (1, 2), the waiting time in the queue (3, 4) and the average number of customers in the system (5, 6) on the probability of a cache miss in the four processes systems architectures, NUMA and SUMA accordingly.

5. CONCLUSIONS

By results of the conducted researches it is possible to make the following outputs.

When functioning in the multitask mode the flow of requests continuously increases in MVS of architecture of UMA that explains bigger number of the serviced requests. At the same time latency of memory of this system is lower, than in case of the single-task mode. This results from the fact that CPU, without expecting the response of memory, make new request. At the same time viability of system is higher as even in case of a high flow of requests the system isn't overloaded in difference from the first where the subsystem of memory doesn't cope with high intensity of requests.

The architecture of NUMA differs in the fact that in case of sharp increase of a flow of requests in memory modules on the common bus there is a sharp increase in time of the response of memory as the multiprocessor's bus is permanently occupied, it leads to growth of wait time. Though in case of a constant flow of requests stability of system operation is watched.

The developed models can be used in case of design of the multiprocessor computing systems. These developments give an opportunity to make an assessment of characteristics of the multiprocessor systems and their subsystems without creation of a real prototype. At the expense of it economic effect as the assessment of characteristics of the designed systems and a choice of the most optimal variants can be carried out without creation of real system is reached.

In article mathematical models of a subsystem "processor memory" which with ease can be used in case of design of new multiprocessor computing systems and enhancement of already available are offered. The considered options give the chance to make an assessment of characteristics of the multiprocessor systems and their subsystems without creation of a real prototype. At the expense of it economic effect as the assessment of characteristics of the designed systems and a choice of the most optimal variants can be carried out without creation of real system is reached.

Results of the conducted researches allow to give the grounds for a row of outputs. MVS of architecture of NUMA differs in the fact that in case of sharp increase of a flow of requests in memory modules on the common bus there is a sharp increase in time of the response of memory as the multiprocessor's bus is permanently occupied, it leads to growth of wait time. Though in case of a constant flow of requests stability of system operation is watched.

When comparing characteristics of the bus we will draw a conclusion that the bus Hyper Transport (SUMA architecture) really possesses higher throughput, and its loading is rather small that is explained by the special organization of an interprocessor exchange. That is the multiprocessor SUMA system possesses bigger real throughput from the point of view of an exchange with memory, unlike architecture with the common multiprocessor's bus.

High scalability of system allows increasing advantage of the bus interface of SUMA systems repeatedly. It is necessary to carry to advantages of this architecture also that the controller of memory is integrated into a CPU crystal at the expense of what the address to him happens without involvement of the front-side bus, and the controller works at CPU frequency.

ACKNOWLEDGEMENT

The work has been done with the financial support of RFBR (Grant No. 16-07-00012 A).

REFERENCES

[1] Aliyev T.I. 2009. The Basics of Discrete System Modeling. SPb.: SPbGU ITMO. p. 363.

[2] Matalytskiy M.A., Tichonenko O.M. and Koluzayeva Ye.V. 2011. The Queuing Systems and Networks: the Analysis and Applications: The Monograph. Grodno: GrGU, 1, pp. 816.

[3] Lozhkovskiy A.G. 2012. The Queuing Theory in Telecommunications: Textbook. Odessa: ONAS named after A.S. Popov, pp. 112.

[4] Tanenbaum A. and Bos H. Modern operating systems. The 4th Edition. SPb.: Piter. p. 1120.

[5] Martyshkin, A.I. & Yasarevskaya, O.N. 2015. Mathematical modeling of Task Managers for Multiprocessor systems on the basis of open-loop queuing networks. ARPN Journal of Engineering and Applied Sciences. 10(16): 6744-6749.

[6] Abramov V.M. 2006. Stochastic Analysis and Applications. 24(6): 1205- 1221.

[7] Martyshkin A.I. 2015. Research of algorithms planning processes in Real Time Systems. In collection: Modern methods and means of the processing of spatial-temporal signals collection of articles XIII All-Russian scientific and technical conference. Edited I.I. Salnikov, pp. 118-124.

[8] Martyshkin, A.I. 2011. Research of memory subsystem with buffered transactions on models of queuing. 21st Century: The Resumes of the Past and the Challenges of the Present Plus: Scientific and Methodological Journal. Penza: PGTA. № 3 (03). pp. 124-131.

[9] Martyshkin A.I. 2015. Development and research of open-loop models of the subsystem "processor-memory" of Multiprocessor systems architectures UMA and NUMA. Herald of Ryazan State University of Radio Engineering. (54-1): 121-126.

[10] Martyshkin, A.I. 2015. Mathematical modeling of the hardware memory buffer for Multiprocessor systems. In collection: Optical-electronic instruments and devices in systems of pattern recognition, image processing, and character information, the collection of materials of XII International scientific-technical conference. pp. 247-249.

[11] Martyshkin, A.I. 2016. Analytical model for the analysis of architectures of Microprocessor systems with memory NUMA and COMA. In collection: Modern scientific research: theoretical and practical aspects. Collection of articles of International scientific-practical conference. Responsible editor: Sukiasyan Asatur Albertovich. pp. 58-61.

[12] Kempa, Wojciech M. 2010. Stochastic Models. 26(3): 335-356.

[13] Masuyama Hiroyuki and Takine, 2003. Tetsuya. Stochastic Models. 19(3): 349-381.

[14] Nadarajah Saralees. 2008. Stochastic Analysis and Applications. 26(3): 526-536.

[15] Mikhalev V. 2012. Performance Test Results QNX Neutrino. Modern Automation Technology: Scientific and Technical Journal. (2): 82-88.

[16] Certificate of state registration, e-number 2013611118. Program complex for measuring the performance of the functions of operating systems.

[17] Martyshkin A.I. 2015. Implementation of the hardware memory buffer for Multiprocessor system. In collection: New information technologies and systems for the collection of scientific articles of the XII International scientific-technical conference. pp. 96-99.

[18] Martyshkin A.I. 2015. Hardware development of memory buffer devices for Multiprocessor systems. Fundamental research. (12-3): 485-489.

[19] Martyshkin A.I. 2016. To the question of assessing service time of the applications when performing exchange operations in Multiprocessor systems-on-chip with shared memory. In collection: Priorities of the world science: scientific experiment and discussion Materials of X international scientific conference. pp. 81-87.

[20] Martyshkin, A.I. 2016. To the problem of the construction of memory subsystem of Multiprocessor systems, in collection: Information technologies in the economic and technical problems Collection of scientific works of International scientific-practical conference, pp. 256-258.

[21] Martyshkin, A.I. 2016. Investigation of performance high performance Multiprocessor systems on open queuing networks. Innovations in science. № 54, pp. 179-184.

[22] Martyshkin, A.I. 2016. To the question of hardware support for process synchronization in Multiprocessor operating real-time systems. New science: From idea to result. № 2-3 (66), pp. 129-131.

[23] Martyshkin, A.I. 2016. Modeling of Multiprocessor systems memory subsystem with a buffer device with multiple queues, based on open queuing networks. Innovations in science. № 55-2, pp. 96-102.

[24] Martyshkin, A.I. 2016. To the problem of research memory subsystem of Multiprocessor computer systems. In collection: Science in modern society: patterns and

trends collection of articles of International scientific-practical conference: in 2 parts, pp. 55-58.

[25] Certificate of state registration, e-number 2013611117. The program package for the calculation of probability-time characteristics of stochastic queuing networks.