Yanina SHESTAK [1], Larysa MURYTENKO [2]

Scientific supervisor: Olexander OKSIUK [3]

OPRACOWANIE METODYKI ROZWOJU SPRZĘTOWYCH TECHNIK RÓWNOWAŻENIA OBCIĄŻENIA ROZPROSZONYCH SIECI INFORMACYJNYCH

Streszczenie: Uwzględniono specyfikę wysokowydajnego i energooszczędnego przetwarzania dużych rozproszonych sieci informacyjnych. Zaproponowano matematyczny model przetwarzania rozproszonych sieci informacyjnych w oparciu o teorię wykresu zadań. W celu rozwiązania problemów związanych z harmonogramowaniem przeanalizowano podział systemu zadań oraz zasilanie. Dokonano wielu porównań wydajność opracowanego modelu matematycznego przetwarzania wielko-skalowych rozproszonych sieci informacyjnych z danymi statystycznymi.

Słowa kluczowe: sieci informacji rozproszonych, model matematyczny, sprzętowe równoważenie obciążenia, grupowanie, wykres zadań, algorytmy partycjonowania, algorytmy partycjonowania

METHODOLOGY OF DISTRIBUTED INFORMATION NETWORKS HARDWARE LOAD BALANCING TECHNIQUES DEVELOPMENT

Summary: Peculiarities of high performance and energy-efficient processing of large-scale distributed information networks organizing was considered. Mathematical model of distributed information network processing based on task graph theory was proposed. In order to solve scheduling problems there were analyzed tasks system partitioning and power supplying. Performance of the developed mathematical model of large-scale distributed information network processing was compared with statistical data.

Keywords: distributed information networks, mathematical model, hardware load balancing, clustering, task graph, partitioning algorithms, partitioning algorithms

---

[1] Taras Shevchenko National University of Kyiv, Faculty of Informaition Technology, Cyber Security and Information Protection: lucenko.y@ukr.net
[2] Taras Shevchenko National University of Kyiv, Faculty of Informaition Technology, Cyber Security and Information Protection: myrutenko.lara@gmail.com
[3] Dr.Sc., Professor, Head of Cybersecurity and Information Protection department of the Taras Shevchenko National University of Kyiv, oksiuk@ukr.net

1. Formulation of the problem

Development of the modern distributed information networks (DIN) requires particular analysis of peculiarities associated with nowadays rapid growth of technologies. Extension of workstation capabilities, adoption of multicore processors and CPU clock rate increase allow to organize scalable, high performance and energy-efficient system with multithreading architecture and infrastructure virtualization (Fig. 1).

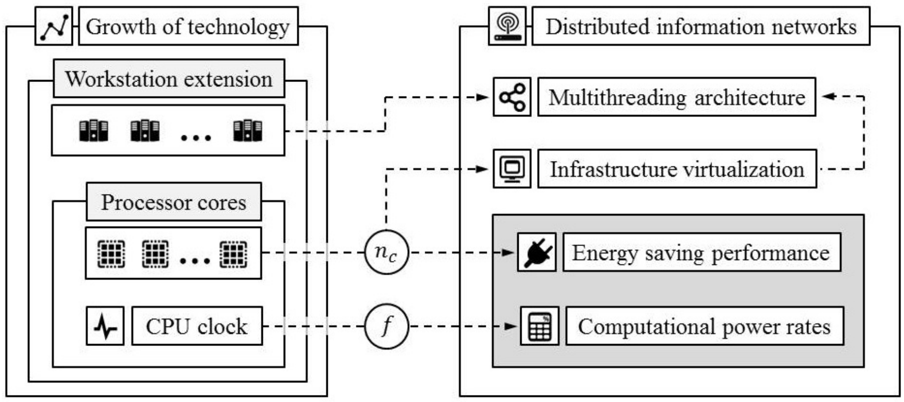

Figure 1. Recent trends in scalable distributed information networks development

Growth of multicore servers DIN performance is mostly based on the multithreading architecture principles. Thereby, mathematical model of the modern DIN should include simulation of parallel computing based on the set of n parallel tasks as a subset of the task graph G which consists of the set of nodes $K(i)$ and set of arcs $R(j)$ representing the precedence constraints. It should be noticed that task i must be executed before task j, which could be formulated as $i \prec$ j. Every task could be characterized by quantity of the CPU cores $n_c(i)$ which are used to perform task of the size $\mu_i \in \{\mu_1, \mu_r\}$ and execution requirements $\delta_i \in \{\delta_1, \delta_r\}$. The product $w_i = \delta_i \cdot \mu_i$ would be the work of task. Generalization of the DIN hardware load balancing model will be done based on the similar multicore CPU servers' network. Task graphs $G_i \in \{G_1, G_x\}$ simulate scheduling of the x parallel applications and multicore CPU work is represented by $P_i \in \{P_1, P_y\}$ function where $n_c(i) \in \{n_c(1), n_c(y)\}$ [1-5]. Optimization task requires to obtain minimal task graph size T^{min} for the r parallel tasks with size $\mu_i \in \{\mu_1, \mu_r\}$ and execution requirements $\delta_i \in \{\delta_1, \delta_r\}$. Obviously it could be performed through obtaining of the set of task work $\{w_1 \dots w_r\}$. It is also necessary to take into account total energy consumption. Thereby, finally developer has to calculate minimum of total energy function E_{Σ}^{min} for the set $\{p_1 \dots p_r\}$ up to the task graph size consumption:

$$\begin{cases} T^* \geq \sqrt[\alpha-1]{\frac{n_c(i)}{E} \cdot \left(\frac{\sum_{i=1}^{r} w_i}{n_я}\right)^{\alpha}} \\ E^* \geq n_я \cdot \frac{\left(\sum_{i=1}^{r} w_i / n_c(i)\right)^{\alpha}}{T^{\alpha-1}} \end{cases} \quad (1)$$

where α is parameter which characterize performance of the multicore CPU processing.
It should be also mentioned that task graph has to be directed acyclic graph and there are no k such that (i, k) and (k, j) exist. Thereby task graph could be distributed at $l \in \{1 \ldots y\}$ levels. Let us suppose that task i is on the level l^+ if number of the nodes on the longest path from the task i to the initial task equals l^+. Tasks from the same level can be scheduled as independent parallel tasks. List of the independent parallel tasks at the level l includes z groups: $(l, 1) \ldots (l, m) \ldots (l, z)$. Thereby, for the $m \in [1, z-1]$ group $m \in [1, z-1]$ could be obtained. Number of the obtained group CPU cores has to be calculated as:

$$\frac{n_c}{m+1} < n_c(i) \leq \frac{n_c}{m} \tag{2}$$

Proposed mathematical model simulates processing algorithms of dividing n_c cores into m clusters which consist of n_c/m CPU core computational power. Every single cluster processes one task of the group (l, m) where n tasks have to be processed simultaneously ($n \leq m$). Scheduling of the (l, m) group parallel tasks is based on the set of task size $\mu_i \in \{\mu_1, \mu_r\}$ and execution requirements $\delta_i \in \{\delta_1, \delta_r\}$ for the development of the task list for m server processors with obtaining of maximal requirements of every task to the processor. If multicore CPU with n_c cores divided into m clusters during processing tasks of the group (l, m), then group (l, m) could be also divided into subgroups $\{(l, m, 1) \ldots (l, m, m)\}$. Every subgroup of the $\{(l, m, 1) \ldots (l, m, m)\}$ set will be processed at one cluster.
Calculating of the energy and task schedule size objective functions minimums is performed after the clustering of the CPU cores. Thus, the process of the task graph is accomplished by determining the time of task execution i using the parameters from the set of maximum requirements δ_i. In this case, the scheduling algorithm includes following steps [1-5]:

1. Every single task must be distributed with respect to the cluster k∈[1,m];
2. The first unscheduled task of the list (m+1) has to be removed from the list and included to cluster k;
3. Go to (1) and repeat the algorithm processing until the set of list is completed.

At the Fig. 2 demonstrated set of algorithms based on the values of the task size $\mu_i \in \{\mu_1, \mu_r\}$, the maximum requirements $\delta_i \in \{\delta_1, \delta_r\}$, and the work of the task $w_i \in \{w_1 \ldots w_r\}$, which can be used for the task list scheduling.
At this investigation is proposed to represent task graph as hierarchical structure of lists, groups and subgroups. Parallel tasks includes y levels, tasks of the single level include z groups of $\{(l, 1) \ldots (l, z)\}$, tasks of the group m include z subgroups $\{(l, m, 1) \ldots (l, m, m)\}$. Thus, the optimal power distribution scheme can be determined by consumption of the graph size and power consumption within both groups and subgroups, allowing four levels to be constructed [1-5].
Optimizing the hardware resources allocation within a single subset of tasks implies execution of tasks within a subgroup that should carried out sequentially. The tasks list length of a subgroup has minimized as:

$$\left[T = \sqrt[\alpha-1]{\frac{\sum_i \mu_i \delta_i^{\alpha}}{E_i}} \right. \tag{3}$$

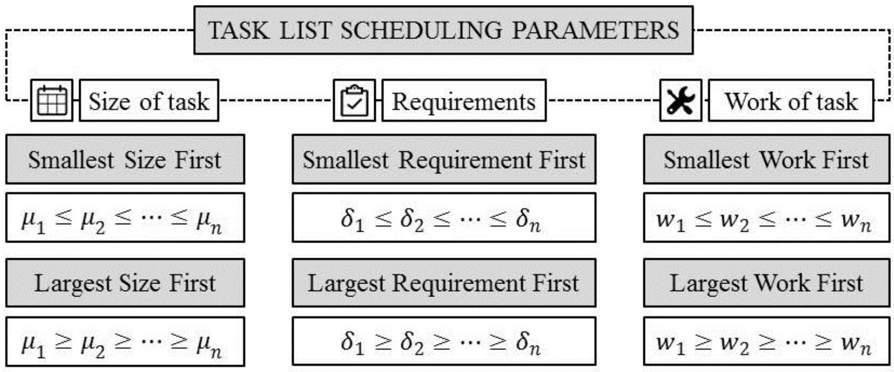

Figure 2. Algorithm of task list scheduling by key parameters

Optimizing the hardware resources allocation at the level of interaction between task subgroups implies obtaining of tasks list length and energy consumption for cluster *k*:

$$\begin{cases} E_k = \frac{E \cdot \sum_i \mu_{k|i} \delta_{k|i}^{\alpha}}{\sum_j (\sum_i \mu_{j|i} \delta_{j|i}^{\alpha})} \\ T_k = \sqrt[\alpha-1]{\frac{\sum_i \mu_{k|i} \delta_{k|i}^{\alpha}}{E}} \end{cases} \tag{4}$$

Optimizing the hardware resources allocation at the level of interaction between task groups should be done for all z groups of the list. Due to partition rules every j group includes i subgroups $(i = j)$. Minimization of T will be done up to the optimal distribution of E_j:

$$\begin{cases} E_j = \left(\frac{\sqrt[\alpha]{\sum_j \mu_{j|i} \delta_{j|i}^{\alpha}}}{\sum_i \sqrt[\alpha]{\sum_j \mu_{j|i} \delta_{j|i}^{\alpha}}} \right) \cdot E \\ T = \sqrt[\alpha-1]{\frac{\sum_i \sum_j \mu_{j|i} \delta_{j|i}^{\alpha}}{E}} \end{cases} \quad => \ T = \sum_l \frac{\sum_i \sqrt[\alpha]{\mu_{l|j|i} \delta_{l|j|i}^{\alpha}}}{\sqrt[\alpha-1]{E_l}} \tag{5}$$

Finally, optimizing the hardware resources allocation at the level of interaction between task lists also should be done through estimation of optimal distribution of energy:

$$\begin{cases} E_l = \left(\frac{\sum_j \sqrt[\alpha]{\sum_i \mu_{l|j|i} \delta_{l|j|i}^{\alpha}}}{\sum_l \sum_j \sqrt[\alpha]{\sum_i \mu_{l|j|i} \delta_{l|j|i}^{\alpha}}} \right) \cdot E \\ T = \sqrt[\alpha-1]{\frac{\sum_l \sum_j \sum_i \mu_{l|j|i} \delta_{l|j|i}^{\alpha}}{E_l}} \end{cases} \tag{6}$$

Thereby normalized schedule length (NSL) and normalized energy consumption (NES) values at the frame of the current investigation and proposed mathematical model could be obtained as [6-7]:

$$\begin{cases} NSL = n_c \left(\dfrac{\sum_l \sqrt[\alpha]{\sum_j \sum_i \mu_{l|j|i} \delta_{l|j|i}^{\alpha}}}{W} \right)^{\frac{\alpha}{\alpha-1}} \\ NEC = n_c \cdot \dfrac{\left(\sum_l \sqrt[\alpha]{\sum_j \sum_i \mu_{l|j|i} \delta_{l|j|i}^{\alpha}} \right)^{\alpha}}{W} \end{cases} \tag{7}$$

For estimation of the proposed model performance there were performed simulation of the processing of task graphs for the tree-structured computations, partitioning algorithms, linear algebra task graphs and “diamond dags” methods. Obtained results were compared with statistic data [4] that was obtained for uniform, binomial and geometric distribution (Fig. 3).
The results of the comparison show sufficiently high prediction accuracy of the developed mathematical model.

Partitioning algorithms	**Distribution**					
	Uniform		Binomial		Geometric	
	NSL	NEC	NSL	NEC	NSL	NEC
Tree-structure computations	4,1%	7,1%	16,8%	4%	3,7%	0,6%
Partitioning algorithms	3,4%	8,6%	9%	2,3%	6,1%	4%
Linear algebra task graphs	4,1%	9,5%	3%	1,8%	6,8%	12,7%
“Diamond dags” methods	5%	19%	**3,8%**	**8,6%**	6,8%	15,6%

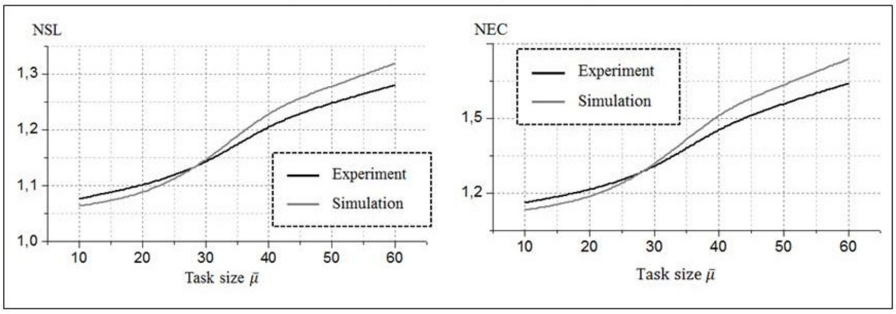

Figure 3. Estimation of the task graph simulation relative error

2. Conclusion

Analysis of high distributed information networks organizing methodology has shown that extension of workstation capabilities allowed to organize scalable, high performance and energy efficient system with multithreading architecture and infrastructure virtualization. Growth of multicore network servers’ performance is mostly based on the concurrency of operations which is related to the multithreading architecture principles.
It was supposed that mathematical model of distributed information network processing should be based on task graph theory. Task graph model included simulation of parallel computing based on the set of parallel tasks as a subset of the task graph which consists of the set of tasks and set of arcs representing the precedence

constraints. Proposed mathematical model simulates processing algorithms of dividing CPU cores into clusters which consist of CPU core computational power. At this investigation is proposed to represent task graph as hierarchical structure of lists, groups and subgroups. Thus, the optimal power distribution scheme can be determined by consumption of the graph size and power consumption within both groups and subgroups, allowing four levels to be constructed. Distributed information network hardware load balancing optimization problem was formulated as a mathematical task of normalized schedule length and normalized energy consumption objective functions minimums search.

Performance of the developed mathematical model of large-scale distributed information network processing was compared with statistical data obtained for the tree-structured computations, partitioning algorithms, linear algebra task graphs, “diamond dags” methods and uniform, binomial and geometric distribution. Estimation of the task graph simulation relative error show sufficiently high prediction accuracy of the developed mathematical model.

REFFERENCES

1. LI K.: Performance analysis of power-aware task scheduling algorithms on multiprocessor computers with dynamic voltage and speed, IEEE Transactions on Parallel and Distributed Systems, 2008, vol. 19, no. 11, 1484–1497.
2. LI K.: Energy efficient scheduling of parallel tasks on multiprocessor computers, Journal of Supercomputing, 2012, vol. 60, no. 2, 223–247.
3. LI K.: Power allocation and task scheduling on multiprocessor computers with energy and time constraints, Energy-Efficient Distributed Computing Systems, A. Y. Zomaya and Y. C. Lee, eds., Chapter 1, pp. 1-37, John Wiley & Sons, 2012.
4. KHAN S. U. Handbook on Data Centers. New York, NY: Springer. 2015.
5. LI K.: Algorithms and analysis of energy-efficient scheduling of parallel tasks, Handbook of Energy-Aware and Green Computing, Vol. 1 (Chapter 15), I. Ahmad and S. Ranka, eds., pp. 331-360, CRC Press/Taylor & Francis Group, 2012.
6. TAKABI H., JOSHI J. B. D., AHN G.-J.: Security and privacy challenges in cloud computing environments, IEEE Security & Privacy, 2011, vol. 8, no. 6, 24–31.
7. ARDAGNA D., PANICUCCI B., TRUBIAN M., ZHANG L.: Energy-aware autonomic resource allocation in multitier virtualized environments, IEEE Transactions on Services Computing, 5(1), 2-19.