TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE
AUTOMATIC CONTROL | AUTOMATYKA
4-AC/2013

SERGII TELENYK, OLEKSANDR ROLIK, MAKSYM BUKASOV,
DMYTRO HALUSHKO*

MODELS AND METHODS OF RESOURCE MANAGEMENT FOR VPS HOSTING

MODELE I METODY ZARZĄDZANIA ZASOBAMI DLA VPS HOSTING

A b s t r a c t

The paper summarizes models and methods of data center resource management for VPS hosting. The approach for the allocation of computing resources in the form of particles of a predetermined size (virtual nodes) was proposed. Different cases of this problem for both an excess and a lack of computing resources were considered. These problems belong to the classes of linear and nonlinear Boolean programming. To solve the mentioned problems, heuristic and guided genetic algorithms have been proposed. A comparison of their effectiveness was carried out.

Keywords: *virtualization, virtual private server, VPS, virtual node, resource management, guided genetic algorithm*

S t r e s z c z e n i e

Niniejszy artykuł podsumowuje modele i metody zarządzania zasobami centrum danych dla VPS hosting. Przedstawione w nim podejście dotyczy alokacji zasobów obliczeniowych w formie cząstek o określonym wymiarze (wirtualne węzły). Rozważono przy tym różne przypadki tego problemu, obejmujące zarówno nadmiar, jak i braki zasobów obliczeniowych. Problemy te należą do klas liniowego i nieliniowego programowania logicznego. Do ich rozwiązania wskazano odpowiednie heurystyki i nadzorowane algorytmy genetyczne. Artykuł podsumowano wnioskami na temat efektywności poszczególnych rozwiązań.

Słowa kluczowe: *wirtualizacja, prywatny serwer wirtualny, VPS, wirtualny węzeł, zarządzanie zasobami, nadzorowany algorytm genetyczny*

* Prof. D.Sc. Ph.D. Sergii Telenyk, e-mail: telenyk@acts.kpi.ua, Ph.D. Oleksander Rolik, Ph.D. Maksym Bukasov, M.Sc. Dmytro Halushko, Department of Automatic and Control in Technical Systems, National Technical University of Ukraine.

Definitions

VPS – virtual private server
VN – virtual node
S_i – server, $i = 1, ..., n$
r_i – resources of S_i [VN]
V_j – VPS, $j = 1, ..., m$
p_j – required resources for V_j [VN]
p_{0j} – committed resources for V_j [VN]
x_{ij} – equals to 1 if VPS V_j is deployed on server S_i, otherwise equals to 0

1. Introduction

Recently, through the development and spread of virtualization and cloud computing technologies, there has been a trend to consolidate computational resources, data storage and communicational equipment in data centers. The implementation of globalization ideas in the IT-area has caused the development of corporative, national and global IT-infrastructures as organized complexes of interconnected networks, information technologies and resources, end-users' equipment and their environment (organized complexes of information applications, user applications and information services) [1]. Interaction between components of an IT-infrastructure provides support for the collection of information, its storage and processing. A special management system supports the effective functioning of the IT-infrastructure. Forrester analytics classified the problems of IT-infrastructure management system into 15 groups [2].

One of the most significant problems is resource management in data centers. Like the other problems in the information and communication services area, the mentioned problem is focused on the user's needs and ensures an appropriate level of service [3–5].

There are several approaches to solving the resource management problem in data centers. Creating a tool that is able to associate changes in the IT-infrastructure state with a degradation of quality of services and to take the appropriate actions is complicated because of the large number of users, the complexity of the IT-infrastructure, the variety of the equipment types and other factors [6–8]. In general, the problem of resource management is relevant for different types of networks and technologies. Researchers and engineers have developed a lot of generalized and specific methods to solve this problem [9–11]. Among these methods, there are a few that take into consideration the peculiarities of the data center allocation models, particularly in the case of the resources limitation [12–16].

However, the development of IT empowers companies to create data centers that take into account the business and user requirements more thoroughly. On the other hand, it is necessary to develop new models and methods for resource allocation in order to ensure the effective use of the new features. Recently, there has been a trend to transfer web applications from virtual hosting to virtual private servers (VPS), so that they would provide higher cost-effectiveness compared to dedicated servers, and could also ensure the necessary guaranteed number of computational resources, which are not always achieved via virtual hosting.

Until recently, the VPS services were provided by a not so flexible scheme – the customer signed an agreement under which s/he was granted the VPS with a fixed amount of the resources. If the customer wanted to ensure the quality of the functioning of applications during peak loads, s/he was forced to request resources with reserve and also pay for them during the idle time while the load is reduced.

Today, the VPS service providers usually allow customers to change their VPS options using the web management console. This enables customers to increase the VPS resources during periods of increased load to ensure the operational quality of their web applications, and to reduce the VPS resources during periods of the decreased load to save money. However, the major disadvantage of this approach is that it requires a customer to increase or decrease VPS resources in manual mode.

2. The problem

The IT-infrastructure resources (networks, servers, data storage, application etc.) require accounting and analysis of the compliance with user requirements to avoid customer outflow and financial losses. It is necessary to maintain the required level of information and communication services, including the peak load, because a lack of resources can cause a degradation of the service level. The solution of this problem by allocating additional resources is not always reasonable. It is necessary to create flexible solutions that are built on the load balancing and the resource allocation. This in turn requires appropriate mathematical models and methods to solve these problems.

Let's consider a situation with the VPS service provision when server resources are allocated by virtual nodes (VN), and are being accounted for in node-hours. While signing the contract for the provision of the services, a customer indicates a fixed number of VN that is guaranteed to be available at any time, and which will be paid for even during idle time. Also, let's assume a customer is able to specify the number of VNs to be additionally provisioned when necessary in case of the availability of appropriate resources in the cloud. A customer pays for those additional VNs only if they are used during peak loads. It is obvious that a customer is interested in obtaining additional resources while increasing his clients' request number to ensure the highest quality of service. A provider is also interested in providing additional resources to customers since they will have to pay more. Herewith, the provider guarantees to provide the resources to other customers, i.e. as additional resources may be used only those, which were not given to any other customer. It is necessary to develop cloud data centers, resource allocation and load models, and methods that meet the above features of the cloud IT-infrastructures. These models and methods are to be based on reasonable criteria for the providers and take into account resource, technological and other constraints.

The VPS providing service implies granting to a customer a virtual server in the form of a virtual machine (VM) that is actually running under hypervisor management on a single physical server [17]. In other words, a provider is unable to allocate to the VM the resources of different physical servers. Therefore, the main instrument of the resource allocation is the VM migration between servers to place the VMs most densely. To me, this phrase seems unclear, but ignore this comment if you think it would make sense to someone who understands the subject.

We suppose that there are several physical servers S_i, $i = 1, ..., n$, where VPS V_j, $j = 1, ..., m$ are running under hypervisor management.

It is clear that the following condition should be fulfilled for business models of the service provision mentioned above:

$$p_j \geq p_{oj}, \quad j = 1, ..., m \tag{1}$$

If ISP does not offer an opportunity to request additional non-guaranteed computing resources, i.e.

$$p_j = p_{oj}, \quad j = 1, ..., m \tag{2}$$

the problem is reduced to the problem that is described in [15] and can be solved by the proposed methods.

Let's impose the following constraints. Since each VPS can be located on only one server, the following condition should be fulfilled:

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, ..., m \tag{3}$$

As a provider guarantees that each VPS will receive the resources not less than p_{oj}, the condition is:

$$\sum_{j=1}^{m} x_{ij} p_{0j} \leq r_i, \quad i = 1, ..., n \tag{4}$$

In order to meet the user requirements and to maximize their own profit in the best way possible, a provider can solve one of the three problems according to the availability of the resources.

Problem 1. If the data center resources substantially exceed the users' requirements to the resources, a provider will attempt to distribute VPS among the servers in the densest way to release some servers that could be turned off to save power. However, a provider will try to satisfy all of the customers' needs in additional resources because payment for even a single VN exceeds savings from shutting down of that server (Fig. 1).

Instead of condition (4) let's impose the following constraints:

$$\sum_{j=1}^{m} x_{ij} p_j \leq r_i, \quad i = 1, ..., n \tag{5}$$

Let's denote by e_i the power of server S_i, when it is not running any VPS. The indication that any VPS does not run on server S_i will express as follows:

$$d_i = \prod_{j=1}^{m} \overline{x_{ij}}, \quad i = 1, ..., n \tag{6}$$

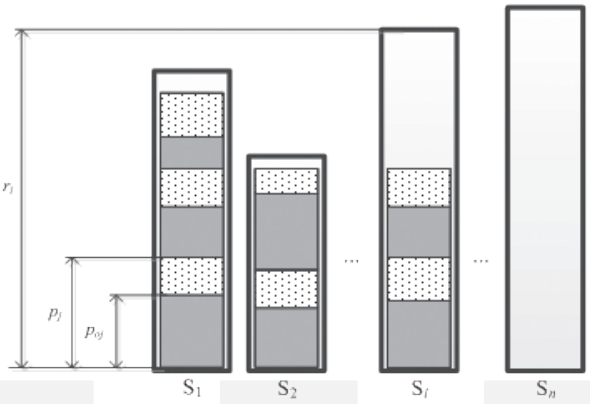

Fig. 1. Desired VPS distribution in the case of the resources excess

Then the problem of power consumption minimization can be formulated as follows:

$$\max \sum_{i=1}^{n} e_i \prod_{j=1}^{m} \overline{x_{ij}} \tag{7}$$

under the constraints (3) and (5). This problem is also similar to the problem that is described in [15] and can be solved by the proposed methods.

Problem 2. If the available resources do not significantly exceed the user requirements (problem 1 solution does not represent a solution that satisfies the requirements of (3) and (5)), a provider has to provide a guaranteed amount of VN for all the users, as well as meeting the maximum number of requests for additional resources in order to maximize own profit and customer satisfaction (Fig. 2). In order to meet this requirement, a provider has to place all of the VPS on the servers most tightly, but with less strict constraints: (4) instead of (5).

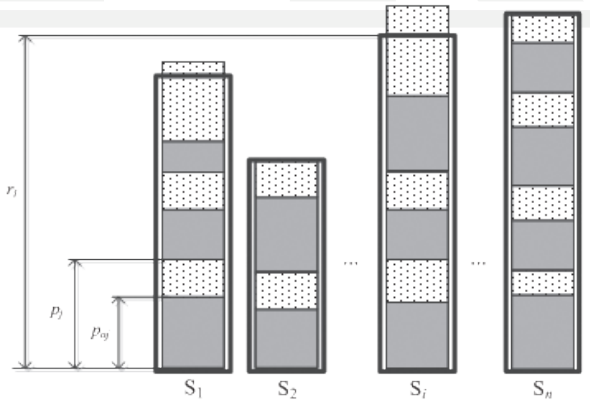

Fig. 2. Desired VPS distribution in the case of the resources lack

Then problem 2 can be formulated as follows:

$$\min \sum_{i=1}^{n} \left| r_i - \sum_{j=1}^{m} x_{ij} p_j \right| \tag{8}$$

under the constraints (3) and (4).

Problem 3. In the case of equipment failure, it becomes impossible to ensure all VPS with even the guaranteed amount of resources. The obvious solution is to support the most important services by providing the resources especially for VPS with related services at the expense of those which are less important. Since each VPS can be located on a single server, the following condition should be fulfilled:

$$\sum_{i=1}^{n} x_{ij} \le 1, \quad j = 1, ..., m \tag{9}$$

Let's denote by $w_j, j = 1,..., m$ the importance of services ensured by V_j.
Then the problem can be formulated as follows:

$$\max \sum_{j=1}^{m} \sum_{i=1}^{n} x_{ij} w_j \tag{10}$$

under the constraints (4) and (9).

3. Cloud IT-infrastructure resource allocation methods

The problems described above belong to a broad class of Boolean programming problems. Problems 1 and 3 are examples of the type of problem that is described in [15]. To solve problem 2 we use greedy and guided genetic algorithms, based on a new combination of ideas [18–21].

Greedy algorithm. Since we are interested in the most uniform distribution of VPS through the servers, let's formulate an idea of the algorithm as follows:

```
while (the list of unallocated VPS has at least one VPS)
{
    find VPS with the highest requirements to the resources;
    place that VPS on the least loaded server;
}
```

Using a greedy algorithm for the resource allocation between VPS is highly effective because the biggest VPS that requires a lot of the resources will be placed first. After that, all the free space on the servers will be filled by the smaller VPS.

Genetic algorithm (GA). Since each VPS can be placed on not more than a single server, for encoding genes let's move from $n*m$ matrix x_{ij} of the Boolean variables to the length m vector y_j of the discrete variables. Each element of that vector is the server's number $i = 1, ..., n$, which contains the appropriate VPS. For example:

$$x_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$y_j = [3 \quad 2 \quad 4 \quad 2 \quad 1 \quad 3 \quad 1 \quad 1]$$

This method of genes coding allows, firstly, to reduce the dimension of the problem, and secondly, to provide an automatic execution of the constraints (3). Therefore, the mutation operation will correspond to the VPS transferring from one server to another and the crossover operation, to the multiple VPS migrations between servers.

The fitness function is a very important concept in the genetic algorithms. It is a measure of an individual's fitness in the population. In our case, the fitness function has to take the large values for the most uniform VPS distribution on the servers when the entire major and most of the additional user resource requirements are satisfied. As the fitness function, we use the number of VNs which are allocated to the VPS. If all the basic and additional requirements for the resources were satisfied, i.e. the constraints (4) and (5), the fitness function will be equal to the number of VNs that were listed in the users' requests and were allocated for VPS.

If all the basic and only part of the additional users requirements were satisfied, i.e. all the constraints (4) are satisfied and (5) are not satisfied, the fitness function will be equal to the number of VNs that were actually allocated for VPS.

If the constraint (4) is not satisfied, the fitness function is penalized, as a result, it takes a small value and the corresponding individual becomes less attractive, reducing the likelihood of its participation in progeny formation.

This fitness function reflects the real economic situation, when a provider increases its income from the provision of a larger number of the VNs on a customers' order and responses in the case of failure to provide the guaranteed amount of resources.

To solve the problem, we use the guided genetic algorithm. The basic idea of this algorithm is to provide a balance between the 'research' and the 'use' by introducing a system of rules, based on which an operator (crossover (C) or mutation (M) for obtaining population of the next epoch is chosen [16]. These rules have to work in such a way that approach to the best solution is controlled If the approaching steps are quite large, it is necessary to speed up the search, and if they slow down, it is necessary to expand the useful schemas. At the same time, these rules should create a barrier for convergence to non-optimal solutions. Therefore, we dedicate every single epoch of the search process to the 'use' of the existing material (fixing optimal solutions) or to the 'research' of the new areas of the space solutions. The selection will happen in each case by keeping in the new epoch only the best individuals from a set of the previous epoch population, and by obtaining 'children' on its basis.

When the number of 'children' is too small, the genetic material collection will be filled in with an additional infusion of randomly generated individuals, this ensures the diversity of the population's genetic material.

In order to form a guided GA system of rules, it is necessary to choose the process parameters for evaluation of convergence and necessity of the specific algorithm operator (C or M) for each particular epoch. Let's introduce the following parameters.

Population growth rate. During the generation of the new epoch population by using one of the operators (C or M), the obtained number of 'children' could be very small in comparison to the population size. This fact indicates a degeneracy in the population due to the insufficient variety of genetic material and requires an appropriate action (an additional 'infusion' of new genetic material)

$$k = \frac{l}{N}$$

where:

l – the number of obtained 'children',
N – population size.

Population prospects. During the GA execution, it is necessary to have a parameter that would characterize the coming degree of the current population to the optimum.

As is known, the search algorithm stops when the maximum value of fitness function matches (or is close enough) to the mean value of fitness function of the entire population.

Let's introduce the 'population prospects' index as a maximum value of the fitness function to its average value ratio for the current epoch population. We assume that the optimal criterion is a maximization of the fitness function value.

$$\rho = \frac{f_{\max}(y)}{f_{\text{avg}}(y)}$$

where:

$f(y)$ – objective function of search (fitness function),
$f_{avg}(y)$ – average value of the current population objective function.

As we can see from the formulated parameter $\rho \geq 1$. The prospects value equals 1 in the case when the fitness of all the individuals within the population is equal.

Convergence speed. It is necessary to have a parameter that would describe the variation trends of GA convergence during transition from epoch to epoch.

Let's introduce the convergence speed concept as the difference between the prospect values of the previous $(i-1)$ and the current (i) epochs' population:

$$\Delta_{i-1,i} = \rho_{i-1} - \rho_i$$

Let's state a set of rules for the selection of the operation in order to obtain the next epoch population.

If the population growth rate is smaller than the threshold, then it is necessary to make an 'infusion' of the new genetic material (operator G).

The formalization of the rule is as follows:

$$\textbf{IF } (k \le k_0) \textbf{ AND } (N \le N_{\max}) \textbf{ THEN G}$$

where:

k_0 – limit for the value of the population growth rate,
$N_{\max}$ – constraint on the population size.

If the convergence speed becomes negative during the transition from $(i - 1)$ to (i) epoch, the crossover (C) operator will be used to form the next epoch population.

Assuming that during the transition from epoch to epoch the population fitness does not decrease (only the best individuals are being selected), convergence speed could be negative only if current epoch prospects become greater than in the previous one. This becomes possible when the new maximum value of the optimum (the best solution to the problem) is found. In order to save the better *best?* solution it is necessary to move from the 'research' to the 'use' strategy and, therefore, to the crossover.

The formalization of this rule is as follows:

$$\textbf{IF } (\Delta_{i-1,i} < 0) \textbf{ THEN C}$$

If GA convergence speed and value of the population prospects of the current epoch are not smaller than the threshold (nature of the convergence process is uncertain, there is no convergence to the 'local' optimum), then crossover should be used to obtain the next epoch population.

The formalization of the rule:

$$\textbf{IF } ((\rho_i \ge \rho_0) \textbf{ AND } (\Delta_{i-1,i} \ge \Delta_0)) \textbf{ THEN C}$$

where:

ρ_0 – limit value of the population prospects,
Δ_0 – limit value of the convergence speed.

If the convergence speed or the population prospects value of the current epoch are smaller than the corresponding limit values, then mutation should be used to obtain the next epoch population. In this case, the search for the optimal solution converges to a certain 'local' optimum and it is necessary to move from the 'use' to the 'research' strategy, and, therefore, to the mutation.

The formalization of the rule is as follows:

$$\textbf{IF } ((\rho_i < \rho_0) \textbf{ OR } (\Delta_{i-1,i} < \Delta_0)) \textbf{ THEN M}$$

As we can see from the proposed rules for the guided GA process management, it is necessary to set the limit values for the population growth rate, the convergence speed and

the prospects. By adjusting these factors, it is possible to regulate the GA convergence search speed and the nature of the basic processes. These processes take place at the stage of rules modification, which allows for the organizing of a kind of feedback in terms of the optimality of the obtained results.

4. Experimental results

The effectiveness of the proposed algorithms was estimated as follows. A cluster of 10 servers were divided into 16 VNs. Each of the proposed algorithms solved the problem of resource allocation for the cases of small and medium-sized VPS with respect to the server size. Two series of experiments were performed, which differed by the spread of the guaranteed number of VNs. In each experiment, the number of the additional VNs, which had been ordered by the users, ranged from 0 to half of the guaranteed amount of the VNs. The requirements for the virtual machines were chosen according to the following table:

Table 1

Input data for the experiment

Series of experiments 1 (small variation in the user requirements)				Series of experiments 2 (average variation in the user requirements)			
p_{0min}	p_{0max}	p_{min}	p_{max}	p_{0min}	p_{0max}	p_{min}	p_{max}
1	2	1	3	1	2	1	3
1	3	1	5	1	3	1	5
2	4	2	6	1	4	1	6
2	5	2	7	1	5	1	7
3	6	3	9	1	6	1	9

For each case, 20 samples were randomly generated. The average results of the solutions are presented in the Fig. 3. The guaranteed and desired average number of VNs was laid off as the x-axis respectively. The number of VNs that was successfully assigned to the VPS was laid off as the y-axis. The results of the heuristic algorithm are labeled as «E», the genetic one – as «GA», the series of the experiments are labeled with numbers 1 and 2.

As can be seen, if VPS has low resource requirements relative to the size of the servers, both algorithms give good results close to the possible maximum (16 VN*10 servers = = 160 VN on cluster). With an increase in the users' requirements (which has to be guaranteed by a provider) it becomes more difficult to place densely the larger VPS to servers and efficiency of the both algorithms decreases because the servers have unused VN. Therefore, the efficiency of GA is consistently higher than the heuristic algorithm.

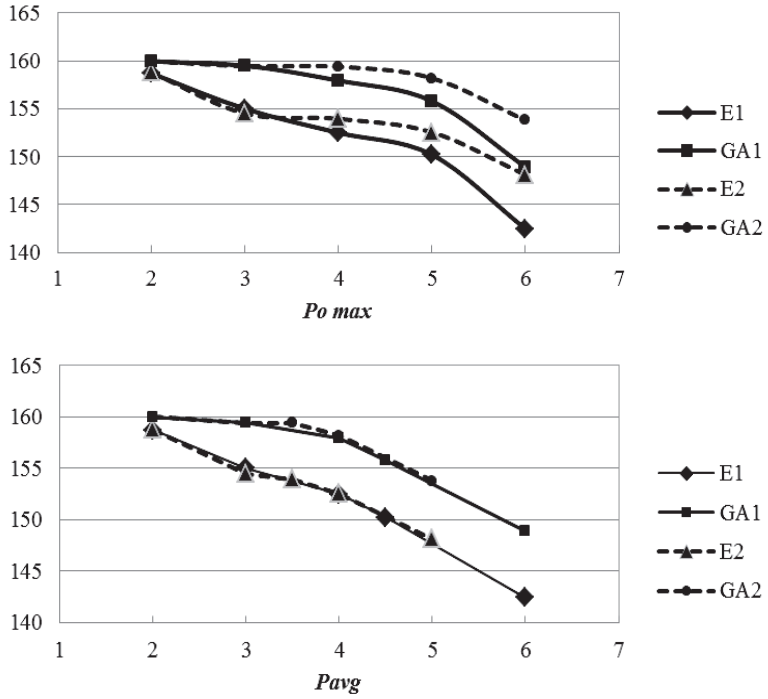

Fig. 3. Comparison of the effectiveness of the heuristic and genetic algorithms

5. Conclusions

Models and methods for solving the resource allocation problem in data centers that provide VPS services in cases where a customer is able to order services in pre-defined units (virtual nodes) were proposed.

Formulated problems were reduced to problems of Boolean programming. Heuristic and guided genetic algorithms [16] were used.

The results of the experiments confirmed the efficiency of the proposed approach as well as the appropriate level of time and costs for service providers.

References

[1] Bon J., Pieper M., Veen A., *Foundations of IT Service Management, based on ITIL*, Van Haren Publishing, 2005.

[2] Garbani J.-P., Mende T., *Market Overview: The IT Management Software Market In 2008*, Forrester Research, March 2008.

[3] *SLA Management Handbook. Volume 1. Executive Overview. GB 917.1. Version 2.1.*, TeleManagement Forum.Morrittown, NJ, January 2005.

[7] Cai Z., Chen Y., Kumar V., Milojicic D., Shwan K., *Automated Availability Management Driven by Business Policies*, 10th IFIP/IEEE Symposium on Integrated Management (IM), 2007, 264-273.

[8] Buco M.J., Chang R.N., Luan L.Z., Ward C., Wolf J.L., Yu P.S., *Utility computing SLA management based upon business objectives*, IBM Systems Journal, Vol. 43, No. 1, 2004, 159-178.

[9] Chen H., Huang L., Kumar S., Kuo C.C., *Radio resource management for multimedia QoS support in wireless network*, Kluwer Academic Publishers, Boston 2004.

[10] Herminghaus V., Scriba A., *Storage Management in Data Centers*, Springer 2009.

[11] *Information Storage and Management: Storing, Managing, and Protecting Digital Information*, EMC Education Services, John Wiley & Sons, 2010.

[12] Wolf J., Yu P., *On balancing load in a clustered web farm*, ACM Transactions on Internet Technology, Vol. 1, #2, 2001, 231-261.

[13] Ardagna D., Trubian M., Zhang L., *SLA based profit optimization in multi-tier web application systems*, Proc. of Int'l Conference On Service Oriented Computing, New York 2004, 173-182.

[14] Kimbrel T., Steinder M., Sviridenko M., Tantawi A., *Dynamic application placement under service and memory constraints*, Proc. of Int'l Workshop on Efficient and Experimental Algorithms, Santorini Island, May 2005, 1-12.

[15] Теленик С.Ф., Ролик О.І., Букасов М.М., Лабунський А.Ю., *Моделі управління віртуальними машинами при серверній віртуалізації*, Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка, Київ 2009, № 51, 147-152.

[16] Теленик С.Ф., Ролік О.І., Букасов М.М., Андросов С.А., *Генетичні алгоритми вирішення задач управління ресурсами і навантаженням центрів оброблення даних*, 'Автоматика. Автоматизація. Електротехнічні комплекси та системи', 2010, № 1(25), 106-120.

[17] Marshall D., Reynolds W.A., McCrory D., *Advanced Server Virtualization: VMware and Microsoft Platforms in the Virtual Data Center*, Auerbach Publications, 2006.

[18] Turban E., Aronson J.E., *Decision support systems*, Prentice Hall, New Jersey 2001.

[19] Nong Y., *The Handbook of Data Mining*, Arizona State University Publishers, New Jersey 2003.

[20] Leung K.S., Duan Q.H., Xu Z.B., Wong C.K., *A new model of simulated evolutionary computation – convergence analysis and specifications*, IEEE Transactions on evolutionary computation,Vol. 5, #1, 2001, 3-16.

[21] Chambers D.L., *Practical handbook of genetic algorithms*, 2nd ed., Applications Vol. 1, Chapman & Hall, 2001.