A comparison of linear programming and the genetic algorithm approaches to the problem of optimizing the server hardware resources for hosting virtual desktops

K A Makoviy¹, D K Proskurin¹, Yu V Khitskova¹² and Ya V Metelkin¹

¹Voronezh State Technical University, 20 let Oktyabrya str., 84, Voronezh, Russia, 394006
²Voronezh State University, Universitetskaya sq. 1, Voronezh, Russia, 394018

e-mail: makkatya@mail.ru, pdk@vgasu.vrn.ru, prosvetovau@list.ru, flow101@mail.ru

Abstract. The article presents a comparison of genetic algorithm and the linear programming method approaches for solving the optimization problem. The comparison is performed on the example of the task of server hardware resources optimization when placing Virtual Machines (VMs) during Virtual Desktop Infrastructure implementation.

1. Introduction

Cloud computing is the paradigm of providing services such as software, a platform, an infrastructure, via the Internet [1]. Desktop as a Service, DaaS, is one of the new technologies provided by cloud computing, and is based on desktop virtualization technology. Virtualization is the isolation of computing processes and computing resources from each other. This technology allows organizations to create transparent, scalable and flexible models of computing and applications, which increases the efficiency of using computing resources. The technology of virtualization has a great impact on the world of information technology and is becoming one of the most important topics of research and technology in the field of IT.

The use of virtualization technology corresponds to the growing popularity of Green Computing. This concept includes computational procedures, policies and functions related to the use of IT. The goal of Green Computing is to use computers and IT in a way that saves energy, money and protects the environment [2].

Virtual Desktop Infrastructure (VDI) is a technology for giving a user access to a remote desktop hosted in a cloud data center [3]. Users can access their desktop applications and work with them using any device (BYOD concept). Virtualization technology offers advantages such as ease of management, mobile computing, flexibility, data security and reduced total costs for business owners, for example, thin client devices designed to access a virtual desktop, use 1 to 5 watts [2]. In addition, these devices do not emit much heat and, as a rule, operate without a fan.

VDI users can access virtual machines using remote desktop protocols such as the Microsoft Remote Desktop (MS RDP) protocol, the Sun Application Link (ALP) application protocol, the Citrix High Definition Experience Protocol (HDX) protocol, the Internet protocol (PCoIP) or the protocol of blocked asynchronous transfer (BLAST) from VMware.

A special feature of virtualization technology are fairly high costs of the first stage [4], so the transition to this infrastructure should begin with a preliminary estimate of server resource requirements [5].

Currently, there are two fundamentally different approaches to the problem of optimizing the hosting of virtual servers on the hardware platforms in the literature – both static and dynamic. The static approach reduces the optimization problem to solving the problem of a bin packing problem or
the knapsack problem [6]. This approach is used when the number of virtual machines and their needs of hardware resources is known in advance.

The dynamic approach to the problem of optimization of server resources when implementing virtualization technology is based on the support of the most popular hypervisors, for example, VMWare ESX, Xen, Linux KVM, Hyper-V live migration technology which allows to move virtual machines "on the fly" without interrupting their work.

In these optimization models, the authors consider a set of identical hardware servers as inputs to the task of placing virtual machines with different resource requirements. In this paper, we propose a model for minimizing the cost of deploying a number of virtual machines, so there is a need to consider hardware servers with different characteristics, which will increase the utilization rate of the servers used. This problem belongs to the class of nondeterministic polynomial complexity (NP) [6], which is proved by reducing it to the bin packing. Usually different heuristics are used to solve this class of problems.

2. Formulation of the problem

In the process of implementation of the technology of virtual desktops it is necessary to purchase hardware servers that will host a custom virtual machine. Since the transition to this technology is economically justified in the case of virtualization of a large number of desktops [4], the task of minimizing costs at the stage of purchasing hardware has a great economic effect.

It is assumed that the equipment is selected based on the number of virtual workstations necessary for end users, taking into account the requirements of virtual machines for resources that can be determined in the following ways:

- recommendations of vendors, which includes requirements for resources depending on the applications that are running, studying the load on the IT infrastructure,
- use the utility VMware View Planner, designed to simulate various workload scenarios in a VMWare Horizon 7 environment,
- use the console utility esxtop, which allows to monitor all aspects of the virtualization server performance.

It is convenient to start the transition to a virtual infrastructure from a pilot project [7] to assess the feasibility, the required implementation time, the costs, the presence or absence of any side effects, and assess the effect size.

After assessing the requirements, you must select a set of servers for purchasing in such a way as to ensure the execution of a specified number of virtual machines and minimize the cost of purchased equipment.

The formulated problem is a non-linear integer programming problem. In the general case, such problems are solved by looking through all possible values, and using the existing methods of finding the optimal plan to reduce this search. In [5, 8], a heuristic approach is used, namely, decomposition is applied, the result of which are two subtasks of linear integer programming, solved by the branch and boundary method.

Another way to solve the problem is to use a genetic algorithm to find the optimal solution.

3. Mathematical model

In this work we solve the problem of providing a predetermined number of virtual machines to run on a set of servers. We consider the identical virtual machines with certain requirements for hardware resources. The servers are based on the models of server platforms, RAM of which, pre-installed by the manufacturer, can be expanded by purchasing additional memory modules. The amount of purchased memory is limited to the maximum amount of memory supported by this platform, as well as the number of free slots for memory modules located on the motherboard of this platform. To increase the amount of RAM you can use additional memory modules of different sizes and, accordingly, the cost.
We suppose that the performance of a virtual machine is acceptable if the amount of server memory is sufficient to run the virtual machine in RAM only, without using a paging file.

For the proposed model, we introduce the following variables:

- $S$ – set of models of server platform $s_i \in S$, $i = 1..m$, which can be used as hardware servers;
- $C$ – set of costs of models of server platforms, where $c_i \in C$, $i = 1..m$ – cost of the platform $s_i$;
- $N$ – number of server platforms which will be used in a final set $n_i \in N$, $i = 1..m$;
- $M$ – the set of maximum amounts of RAM that can be added to the server platform $s_i$, $m_i \in M$, $i = 1..m$;
- $R$ – the set of amountsof memory module, $\eta_j \in R$, $j = 1..k$, where $k$ – is a number of memory modules types;
- $C_v$ – the cost of $j$ RAM module, $c_{v_j} \in C_v$, $j = 1..k$;
- $P$ – the set of numbers server platform slots $p_i \in P$, $i = 1..m$.

To determine the optimal configuration in terms of cost, we need to find an objective function that reflects the full cost of the set of hardware servers. The total cost of the server consists of the value of based server platform model ($c_i$) and the cost of additional RAM modules ($\sum_{j=1}^{k} c_{v_j} a_{ji}$), where $a_{ji}$ - number of $j$ RAM modules on the server platform $s_i$.

Each of the platforms can occur in the final set more than once, and along with the repeated memory configurations, the optimal result set can have the same platforms with different options for filling the memory slots.

To determine the number of options for filling memory slots of server platforms, we present this task as a combinatorial task of determining the number of combinations with repetitions. The general formula for combinations with repetitions from $n$ to $k$ is the following formula:

$$\tilde{C}_n^k = \binom{n+k-1}{k} = \frac{(n+k-1)!}{k!(n-1)!}$$ (1)

where $k$ – the number of selectable items, $n$ – number of types of elements from which the choice is made.

For our task, memory slots on the server platform are considered as selectable elements, and the number of element types is the number of variants of the installed memory modules plus the absence of a memory module in the slot. According to (1) for the server platform $c_i$ the number of options for filling slots with memory modules $q_i$ will be:

$$q_i = \tilde{C}_{p_i}^{k+1} = \binom{k+p_i}{k}$$ (2)

The total number of options for filling slots with RAM modules of all servers is the sum of the number of options for each server $q_i$:

$$q = \sum_{i=1}^{m} q_i$$ (3)

Thus, the objective function will be the following:

$$F = \sum_{i=1}^{q} \sum_{j=1}^{m} (c_i + \sum_{j=1}^{k} c_{v_j} a_{ji}) n_i$$ (4)

To solve the optimization problem, it is necessary to take into account the problem constraints.

1. The total amount of memory added to the server platform should not exceed the maximum amount supported by the server platform:

$$\sum_{j=1}^{k} a_{ji} \leq m_i, \quad i = 1..n,$$ (5)

2. The total number of memory modules can not exceed the number of memory slots on the server:

$$\sum_{j=1}^{k} a_{ji} \leq p_i, \quad i = 1..n$$ (6)

3. The total amount of memory on all servers from the final set should provide enough memory to run predetermined number of virtual machines:

$$\sum_{i=1}^{m} (\sum_{j=1}^{k} r_{j} a_{ji} / V) \geq N_{v}$$ (7)

where $N_{v}$ – is a predetermined number of virtual machines, $V$ – the amount of memory required for a single virtual machine.

4. In order to get a solution that makes sense, we will add a constrains for numbers of servers and RAM modules to be integer:
\[ n_{il}, a_{ji} \geq 0, \quad n_{il}, a_{ji} \quad \text{integers} \quad (8) \]

The problem of minimizing the objective function (4) with allowance for constraints (5-8) is an optimization problem, which is proposed to solve using the genetic algorithm.

4. The approach to solving the problem using linear programming methods

At the first stage of our numerical solution, we fill the memory slots of the hardware servers in the optimal way with memory slots in order to achieve a certain degree of memory capacity from the maximum possible. As a step, we took a step of 25%, that is, we are considering options with memory capacity at 25%, 50%, 75% and 100%. At the same time, for each hardware platform, we minimize the cost of purchasing the required amount of memory by selecting the necessary memory slots. Such a problem can be formulated as follows:

\[ C_{vi} = \min \sum_{j=1}^{k} c_{vj} a_{ji} \quad (9) \]

subject to:

\[
\left\{ \begin{array}{l}
\sum_{j=1}^{k} a_{ji} q_j = m_i q_i \\
\sum_{j=1}^{k} a_{ji} \leq p_i \\
i = 1..m
\end{array} \right. \quad (10)
\]

where \( q \) – the degree of filling, which can be 0.25, 0.5, 0.75 and 1, which corresponds to the filling of the server's RAM on, respectively, 25%, 50%, 75% and 100%.

We will consider a set of received solutions as a set of hardware platforms, that is, on the basis of one platform we get four servers in a variety of hardware platforms, from which the choice is made. The corresponding problem can be formulated as follows:

\[ \min \sum_{i=1}^{m} \sum_{j=1}^{4} (c_i + c_{vi} q_j) a_{ij} \quad (11) \]

where \( c_{vi} \) – is a result of (9) subject to (7).

This problem is a problem of integer linear programming. A numerical solution was obtained using the Matlab package.

5. Genetic algorithm

The process of solving the optimization problem by the genetic algorithm includes four preparatory stages:

1) development of the coding structure of the solution, allowing to evaluate the solution obtained, called the presentation scheme;
2) creation of the initial population;
3) choosing an operator to generate descendants;
4) definition of a rule for selecting decisions for participation in further breeding.

For chromosome encoding, a list of server bundles is used, the elements of which are instances of the Serv class. The chromosome is a set of genes, each of which contains the number of servers (bundles), the serial number of the gene locus corresponds to the serial number of the equipment in the list of server bundles. The following properties are defined for the Serv class:

- the server platform name;
- the sequence number of the server platform;
- the value of the server platform model;
- the number of RAM slots;
- the number of RAM 4Gb modules;
- the number of RAM 8Gb modules;
- the number of RAM 16Gb modules;
- the number of RAM 32Gb modules;
- the total cost;
- the maximum amount of memory supported by this server platform model.

The first population, consisting of a set of chromosomes, is randomly generated. After receiving the first population of chromosomes, they are sorted in ascending value of total cost, and then crossed
in pairs (an even to odd). The position of the split point is chosen randomly. When two chromosomes are crossed, two new chromosomes are obtained. These resulting chromosomes are subjected to a random mutation, or directed mutation with a predetermined probability, and, after that, become members of a new generation of the population. In addition, a chromosome from the previous generation with the best result of a total cost (the lowest total cost) adds to the new generation, in order to not to lose the best result achieved in the previous stage.

By random mutation, we mean adding a unit to a locus with a randomly selected position. Under the directed mutation we mean the following procedure. Among the non-empty chromosome loci, a locus with a minimum cost of a gigabyte of memory and a locus with a maximum cost of a gigabyte of memory are defined. The number of servers at the locus with the maximum cost of gigabytes is reduced by one, and at the locus with the minimum cost of gigabytes is increased by one until the total memory of the chromosome reaches the required value. The process of genetic selection is limited to a given number of generations. The necessary number of generations was determined experimentally by several calculations for the same task parameters, increasing the number of servers was accompanied by the need to increase the size of the population and the number of generations.

To implement the genetic algorithm, an application was developed in the Java language. The application provides the ability to import source data about server platforms and their characteristics from an Excel file, as well as specifying the size of required RAM, the size of the population, the number of generations, and % random and directed mutations.

### Table 1. Server platforms hardware specifications.

| Preset RAM, Gb | Nr. of preset RAM modules | Maximum RAM volume, Gb | Nr. Of RAM slots | Cost, USD | CPU          |
|----------------|--------------------------|------------------------|------------------|-----------|--------------|
| ML150 Gen9 NHP | 4                        | 512                    | 16               | 1580      | E5-2603v3-1.60 |
| ML150 Gen9 Hot Plug | 8 | 1 | 512 | 16 | 1700 | E5-2609v3-1.90 |
| ML150 Gen9 NHP | 8 | 1 | 512 | 16 | 1960 | E5-2609v3-1.90 |
| ML350p Gen8 | 8 | 2 | 384 | 24 | 3300 | E5-2620-2.00 |
| ML350p Gen8 | 8 | 2 | 384 | 24 | 4300 | E5-2630-2.30 |
| ML350p Gen8 | 32 | 4 | 384 | 24 | 4440 | E5-2620-2.00 |
| ML350e Gen8 Hot plug | 8 | 2 | 192 | 12 | 1874 | E5-2420-1.90 |
| ML350p Gen8 E5-2620 Hot Plug | 16 | 2 | 384 | 24 | 3556 | E5-2620-2.00 |
| ML350p Gen8 E5-2620 | 8 | 2 | 384 | 24 | 3169 | E5-2620-2.00 |
| ML350e Gen8 Hot plug | 2 | 1 | 96 | 12 | 1624 | E5-2407-2.20 |
| ML350p Gen8 HPM | 16 | 2 | 384 | 24 | 7100 | E5-2640v2-2.00 |
| ML350e Gen8v2 Hot Plug | 8 | 1 | 192 | 12 | 1660 | E5-2407v2-2.40 |
| ML350 Gen9 | 8 | 1 | 384 | 24 | 2510 | E5-2609v3-1.90 |
| ML350 Gen9 | 16 | 1 | 384 | 24 | 3590 | E5-2620v3-2.40 |
| ML350 HPM Gen9 | 32 | 2 | 768 | 24 | 6518 | E5-2630v3-2.40 |

### 6. Input data

Usually organizations use the same equipment of a certain vendor, which simplifies maintenance and maintenance, so it's reasonable to choose the hardware platforms offered by one manufacturer. As part of this work, we are considering server platforms HP Proliant ML for small and medium business. Hardware characteristics and cost of solutions are taken from the site of one of the suppliers of HP Proliant servers [9] (Table 1).

The hardware specifications of the server platforms that are used for selection are shown in Table 1. The amount and the number of the preset RAM module, cost of which is included to a total server platform cost is given in a Table 1. RAM of each server platform can be extended by RAM modules of...
2GB, 4GB, 8GB, 16GB, 32GB and cost, respectively, 26, 136, 215, 315, 840 USD. The requirement of 4 GB per one virtual machine is used for the calculation.

7. Results and Discussion
The results obtained with the help of a genetic algorithm and integer linear programming are given below. To compare the results, three different sets of input data were used:
1) you need to host 1500 virtual machines, the choice is made from the first 10 servers of Table 1;
2) you need to host 1000 virtual machines, the choice is made from all 15 servers in Table 1;
3) you need to host 700 virtual machines, the choice is made from the first 5 servers of Table 1.

Table 2. 1500 VM, a choice of 10 server platforms, linear programming.

| CPU              | Nr. of CPU mod. 2GB | Nr. of RAM mod. 4GB | Nr. of RAM mod. 8GB | Nr. of RAM mod. 16GB | Nr. of RAM mod. 32GB | Cost   |
|------------------|----------------------|----------------------|----------------------|----------------------|----------------------|--------|
| ML150 Gen9 NHP   | 1                    | 6                    | 0                    | 0                    | 7                    | 3941   |
|                  | 25%                  |                      |                      |                      |                      |        |
| E5-2603v3-1.60   |                      |                      |                      |                      |                      |        |

Table 3. 1500 VM, a choice of 10 server platforms, genetic algorithm.

| CPU              | Nr. of CPU mod. 2GB | Nr. of RAM mod. 4GB | Nr. of RAM mod. 8GB | Nr. of RAM mod. 16GB | Nr. of RAM mod. 32GB | Cost   |
|------------------|----------------------|----------------------|----------------------|----------------------|----------------------|--------|
| ML150 Gen9 NHP   | 17                   | 0                    | 0                    | 10                   | 5                    | 8930   |
|                  | E5-2603v3-1.60       |                      |                      |                      |                      |        |

Tables 2,3 show the results of the solution to the optimization task of placing 1500 virtual machines requiring 4 GB of RAM each, provided that the first 10 servers are selected from the server platforms listed in Table 1. The first column shows the serial number of the server platform list from Table 1.

Table 2 additionally shows the percentage of filling the memory slots with RAM modules, corresponding to the value of $q$ in (9) and (10).

The main parameters of the calculation using the genetic algorithm were the following: the size of the population was 3000, 150 generations, the percentage of random and directed mutations was 15, the results are given in the Table 3.

Table 2 shows the results of solving the optimization task of hosting 1000 virtual machines requiring 4 GB of RAM each, provided that the first 15 servers are selected from the server platforms.

| CPU              | Nr. of CPU mod. 2GB | Nr. of RAM mod. 4GB | Nr. of RAM mod. 8GB | Nr. of RAM mod. 16GB | Nr. of RAM mod. 32GB | Cost   |
|------------------|----------------------|----------------------|----------------------|----------------------|----------------------|--------|
| ML150 Gen9 NHP   | 1                    | 0                    | 0                    | 13                   | 1                    | 6895   |
|                  | E5-2609v3-1.90       |                      |                      |                      |                      |        |

| CPU              | Nr. of CPU mod. 2GB | Nr. of RAM mod. 4GB | Nr. of RAM mod. 8GB | Nr. of RAM mod. 16GB | Nr. of RAM mod. 32GB | Cost   |
|------------------|----------------------|----------------------|----------------------|----------------------|----------------------|--------|
| ML150 Gen9 NHP   | 1                    | 0                    | 2                    | 1                    | 10                   | 6897   |
|                  | E5-2603v3-1.60       |                      |                      |                      |                      |        |

Tables 4,5 show the results of solving the optimization task of hosting 1000 virtual machines requiring 4 GB of RAM each, provided that the first 15 servers are selected from the server platforms.
listed in Table 1. The main parameters of the calculation using the genetic algorithm were the following: population size - 4000, 300 generations, percent random mutations - 10% and directed mutations - 15%, the results are given in Table 5.

### Table 4. 1000VM, a choice of 15 server platforms, linear programming.

| Filling | CPU                | Nr. | Nr. of RAM mod. 2GB | Nr. of RAM mod. 4GB | Nr. of RAM mod. 8GB | Nr. of RAM mod. 16GB | Nr. of RAM mod. 32GB | Cost  |
|---------|--------------------|-----|---------------------|---------------------|---------------------|-----------------------|-----------------------|-------|
| 2       | ML150 Gen9 HotPlug | 50% | 1                   | 0                   | 0                   | 1                     | 13                    | 6850  |
|         | ML350 Gen9         | 75% | 1                   | 4                   | 0                   | 0                     | 17                    | 7969  |
|         | ML350 Gen9         | 100%| 9                   | 0                   | 0                   | 1                     | 21                    | 10180 |
|         | Total cost         |     |                     |                     |                     |                       |                       | 106439 USD |
|         | Total RAM          |     |                     |                     |                     |                       |                       | 3870 GB |

### Table 5. 1000VM, a choice of 15 server platforms, genetic algorithm.

| CPU                | Nr. | Nr. of RAM mod. 2GB | Nr. of RAM mod. 4GB | Nr. of RAM mod. 8GB | Nr. of RAM mod. 16GB | Nr. of RAM mod. 32GB | Cost  |
|--------------------|-----|---------------------|---------------------|---------------------|-----------------------|-----------------------|-------|
| 1 ML150 Gen9 NHP   | 1   | 0                   | 0                   | 0                   | 6                     | 6                     | 8510  |
| 1 ML150 Gen9 NHP   | 5   | 0                   | 0                   | 0                   | 6                     | 8                     | 10190 |
| 1 ML150 Gen9 NHP   | 7   | 0                   | 0                   | 0                   | 10                    | 2                     | 6410  |
| 13 ML350 Gen9      | 1   | 0                   | 1                   | 3                   | 17                    | 1                     | 9486  |
| Total cost         |     |                     |                     |                     |                       |                       | 113816 USD |
| Total RAM          |     |                     |                     |                     |                       |                       | 4008 GB |

### Table 6. 700 VM, a choice of 5 server platforms, linear programming.

| Filling | CPU                | Nr. | Nr. of RAM mod. 2GB | Nr. of RAM mod. 4GB | Nr. of RAM mod. 8GB | Nr. of RAM mod. 16GB | Nr. of RAM mod. 32GB | Cost  |
|---------|--------------------|-----|---------------------|---------------------|---------------------|-----------------------|-----------------------|-------|
| 2       | ML150 Gen9 HotPlug | 50% | 1                   | 0                   | 0                   | 1                     | 13                    | 6850  |
| Total cost |                 |     |                     |                     |                     |                       |                       | 75350 USD |
| Total RAM |                |     |                     |                     |                     |                       |                       | 2816 GB |

Tables 6,7 show the results of solving the optimization task of placing 700 virtual machines requiring 4 GB of RAM each, provided that the first 5 servers are selected from the server platforms.
listed in Table 1. The main parameters of the calculation using the genetic algorithm were as follows: population size - 4000, 300 generations, random mutations percent - 10% and directed mutations - 12%, the results are given in Table 7.

| CPU                     | Nr. of RAM mod. 2GB | Nr. of RAM mod. 4GB | Nr. of RAM mod. 8GB | Nr. of RAM mod. 16GB | Nr. of RAM mod. 32GB | Cost  |
|-------------------------|----------------------|---------------------|---------------------|----------------------|----------------------|-------|
| ML150 Gen9 NHP          | E5-2603v3-1.60       | 8                   | 0                   | 0                    | 11                   | 4     | 8405  |
| ML150 Gen9 NHP          | E5-2603v3-1.60       | 1                   | 0                   | 0                    | 12                   | 0     | 5360  |
| ML150 Gen9 Hot Plug     | E5-2609v3-1.90       | 1                   | 0                   | 1                    | 0                    | 8     | 4356  |
| **Total cost**          |                      |                     |                     |                      |                      | 76956 USD |
| **Total RAM**           |                      |                     |                     |                      |                      | 2800GB |

The obtained results show the principle applicability of both proposed methods to the solution of the problem of optimizing server hardware resources in the process of deploying the Virtual Desktop Infrastructure. For all tested combinations, the approach based on linear programming methods shows better results, although the order of the values of the objective function obtained by both methods coincides. One of the problems of the genetic algorithm on functions that have in addition to the global minimum a large number of local minima is the tendency to roll into the local minimum and "get stuck" there. To solve this problem, it is necessary to use additional algorithmic solutions.

8. References

[1] Ali M and Miraz MH 2013 Cloud Computing Applications Proceedings of the International Conference on Cloud Computing and eGovernance 1

[2] Agrawal S, Biswas R and Nath A 2014 Virtual desktop infrastructure in higher education institution: Energy efficiency as an application of green computing Communication Systems and Network Technologies (CSNT), Fourth International Conference 601-605

[3] Kochut A 2009 Power and performance modeling of virtualized desktop systems MASCOTS'09 IEEE International Symposium 1-10

[4] Makoviy K A and Khitskova Y V 2015 Economic basis of VDI deployment in institution of higher education IT-infrastructure Modern economy: problem and solutions 2 75-81

[5] Proskurin D K and Makoviy K A 2017 The task of selecting server resources for Virtual Desktop Infrastructure deployment Bulletin of Voronezh State Technical University 13(4) 26-32

[6] Speitkamp B and Bichler M 2010 A mathematical programming approach for server consolidation problems in virtualized data centers IEEE Transactions on services computing 3(4) 266-278

[7] Makoviy K A and Shipilov N V 2016 The pilot project of virtualization of workplaces in the computer class of Voronezh GASU Scientific Bulletin of Voronezh State Architecture and Construction University. Series: Student and Science 10 113-117

[8] Makoviy K, Proskurin D, Khitskova Y and Metelkin Y 2017 Server hardware resources optimization for virtual desktop infrastructure implementation CEUR Workshop Proceedings 1904 178-183

[9] Servers and Accessories of Hewlett-Packard (Access mode: http://www.proliant.ru/files/File/HP_proliant_price_09_15.xls)