

УДК 519.687

DOI: 10.24160/1993-6982-2018-1-98-105

## Методика перераспределения функционирующих виртуальных машин по серверам в дата-центре

А.А. Ларин, Л.И. Абросимов

Ежегодно возрастает необходимость в централизованной обработке и хранении данных, для этого вводятся в эксплуатацию дата-центры еще большей вместимости и вычислительной мощности. Для снижения расходов особое внимание уделяется оптимизации энергопотребления инфраструктуры такого центра. Рассмотрен способ повышения энергоэффективности всего дата-центра за счет перераспределения нагрузки с последующим отключением простаивающих серверов. Рассмотрен механизм миграций виртуальных машин для оптимизации потребления серверами.

Существующие методики для перераспределения виртуальных машин используют полный перебор, который невозможен в случае реальных размерностей, либо в них не учитывается критерий энергопотребления. Предложена методика, исключающая полный перебор и нацеленная на повышение эффективности по критерию суммарного энергопотребления серверов дата-центра за счет перераспределения функционирующих виртуальных машин по серверам в дата-центре. Она основана на эвристической процедуре выбора серверов назначения и состоит из статической части, в которой происходит сбор информации о ресурсах, и динамической части — в которой виртуальные машины распределяются по серверам в зависимости от начальных данных и установленных ограничивающих параметров.

Приведен модельный пример, демонстрирующий работоспособность методики. В нем раскрываются шаги методики, показан выбор конечных серверов назначения для миграций.

*Ключевые слова:* энергопотребление дата-центра, ресурсы серверов и виртуальных машин, методика перераспределения виртуальных машин, дата-центр.

*Для цитирования:* Ларин А.А., Абросимов Л.И. Методика перераспределения функционирующих виртуальных машин по серверам в дата-центре // Вестник МЭИ. 2018. № 1. С. 98—105. DOI: 10.24160/1993-6982-2018-1-98-105.

## A Methodology for Redistributing the Operating Virtual Machines among the Servers in a Data Center

А.А. Larin, L.I. Abrosimov

The need for centralized processing and storage of data is growing every year. Data centers of still greater capacity and computing power are put into operation for this purpose. To reduce their expenses the owners of data centers pay special attention to optimizing the energy consumed by the data center infrastructure. An approach for improving the data center's overall energy efficiency through redistributing its load with subsequently shutting down the idle servers is considered. A mechanism governing the migration of virtual machines for optimizing power consumption by the servers is outlined.

The existing methods used to redistribute virtual machines either perform a full search which is impossible in the case of real dimensions or disregard the power consumption criterion. A procedure is proposed that excludes the need to perform a full search and places focus on improving the efficiency according to the criterion of total energy consumption by the data center servers due to redistribution of the operating virtual machines among the data center servers. The proposed method is based on a heuristic procedure of selecting the destination servers and consists of a static part, which collects information about the available resources, and a dynamic part in which the virtual machines are distributed among the servers depending on the initial data and the preset limiting parameters.

A model example demonstrating the operability of the method is given which outlines the steps in applying the method and shows how the destination servers are selected for migrations.

*Key words:* data center energy consumption, resources of servers and virtual machines, virtual machines redistribution procedure, data center.

*For citation:* Larin A.A., Abrosimov L.I. A Methodology for Redistributing the Operating Virtual Machines among the Servers in a Data Center. MPEI Vestnik. 2018;1:98—105. (in Russian). DOI: 10.24160/1993-6982-2018-1-98-105.

## Введение

С каждым годом растет роль центров обработки данных в предоставлении необходимых вычислительных ресурсов для коммерческих и государственных организаций. В России за 2014 г. объем рынка облачных услуг возрос на 35 %, а к 2018 возрастет еще в 2,5 раза [1]. Крупнейшие коммерческие дата-центры (ДЦ) в среднем заполнены уже более чем на 75 % [2].

С увеличением темпов роста, владельцам центров обработки данных необходимо поддерживать на требуемом уровне, либо оптимизировать такие ключевые параметры, как потребление энергии, надежность, а также обеспечивать другие параметры, установленные соглашением об уровне услуг (Service Level Agreement, SLA).

Владельцы ДЦ ищут способы повышения энергоэффективности. Корпорация Google, оптимизировав потребление мощности охлаждающим оборудованием за счет анализа статистики потребления прошлых лет, смогла снизить энергопотребление на 40 % [3].

Помимо систем охлаждения, потребляют электроэнергию и другие подсистемы. На рис. 1 представлена диаграмма потребления электроэнергии различными подсистемами ДЦ.

Наименьшие затраты электроэнергии приходятся на освещение ДЦ, а также при потерях на источниках бесперебойного питания (ИБП). Большую часть электроэнергии потребляют система водоохлаждения (чиллеры) и до половины электроэнергии — серверное оборудование.

Заметим, что серверы, обслуживающие виртуальные машины (VM), часто бывают загружены частично или полностью не загружены. При этом даже простаивающий сервер, как показывают результаты исследования, значительно потребляет энергию [4].

Виртуализация представляет собой механизм создания множества виртуальных машин (VM) на одном физическом сервере. Современные средства виртуализации позволяют без остановки VM выполнять их перемещение (миграцию) на другие физические сервера.

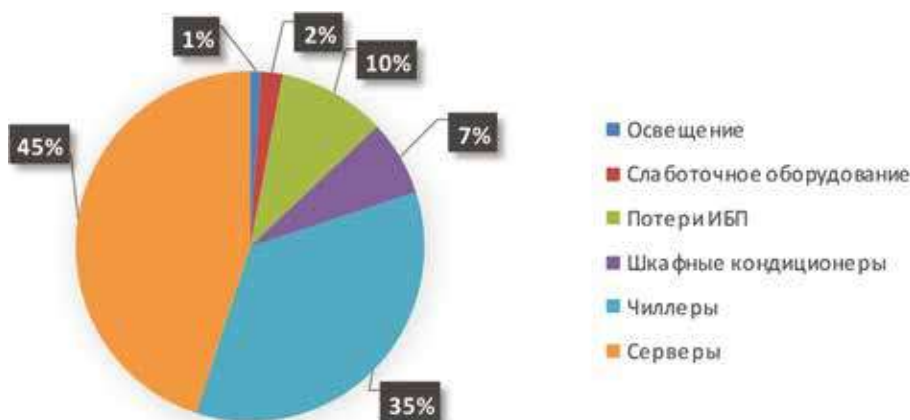


Рис. 1. Диаграмма потребления электроэнергии в дата-центре

Механизм миграций также влияет на энергопотребление оборудования.

Для оптимизации энергопотребления используется перераспределение функционирующих VM по серверам таким образом, чтобы требуемые VM ресурсы не превышали имеющиеся ресурсы серверов. Оптимизация происходит за счет перемещения VM с серверов с незначительной суммарной загрузкой ресурсов от VM на другие серверы, при этом все освободившиеся серверы отключаются, тем самым снижая общее энергопотребление ДЦ.

Оценка необходимого размера кластера под назначение определенного количества виртуальных машин проведена в [5]. Однако, такой подход подходит только для частных облаков, а не для динамически расширяемых облачных вычислений. Известно решение задачи по критерию оптимального размещения VM при миграциях методом решения задачи «упаковки объектов в контейнер», но данный подход относится к категории NP-трудных комбинаторных задач, решение которых сводится к полному перебору, что не применимо в случае больших размерностей, когда кластеры включают в себя несколько тысяч VM, к тому же этот метод не учитывает сокращение текущего энергопотребления серверов [6]. Многокритериальные способы оптимизации рассматриваются в [7], однако, в них энергопотребление не является основным критерием для оптимизации.

Описанная в настоящей работе методика позволяет выполнять миграции VM, учитывая энергопотребление серверов, время миграций, а также используемые вычислительные ресурсы серверов, с целью уменьшения энергопотребления вычислительных комплексов.

## Постановка задачи

Рассмотрена вычислительная сеть ДЦ в котором есть множество серверов  $S$ , их количество —  $N_s$ . Из технической документации для каждого  $i$ -го сервера известны максимум и минимум энергопотребления  $P_{\max_i}$  и  $P_{\min_i}$ , а также доступные ресурсы (количество ядер процессоров, оперативной памяти) каждого сервера  $R_{\max_i}^{\text{CPU}}$ ,  $R_{\max_i}^{\text{RAM}}$ .

Датчики мониторинга ДЦ снимают показатели энергопотребления серверов  $P_i$ . Известна ширина полосы пропускания  $W$  каналов связи между серверами. На контроллере управления ресурсами серверов показаны количество  $N_{vm}$  виртуальных машин  $VM$ , назначенных на сервера, принадлежность  $VM$  к определенному серверу (выражено матрицей  $X$ ), объем требуемых ресурсов по процессорам, оперативной памяти для каждой  $j$ -й  $VM$   $R_{vm_j}^{CPU}$ ,  $R_{vm_j}^{RAM}$  и текущая загрузка ресурсов для каждой  $j$ -й  $VM$   $Q_{vm_j}^{CPU}$ ,  $Q_{vm_j}^{RAM}$ .

Известна функциональная зависимость передаваемого объема данных каждым сервером при миграции  $VM$  от текущей загрузки ресурсов процессоров, памяти  $VM$   $V_i(Q_{vm_j}^{CPU}, Q_{vm_j}^{RAM})$ , которые могут быть определены методом замеров характеристик сервера при помощи программных агентов [8].

Для каждой  $VM_j$  известен список серверов  $\{s_i\}$ , доступных для выполнения миграций.

Системным администратором в зависимости от требуемых режимов задаются параметры коэффициентов:

- $\alpha_i$  — текущего энергопотребления, устанавливаемого по каждому серверу  $s_i$  и ограничивающего выбор серверов для выполнения миграции  $VM$  на сервер  $s_i$ ;
- $\beta_i$  — продолжительность времени миграции каждой  $VM$ , устанавливаемого по каждому серверу  $s_i$  и ограничивающего выбор серверов для выполнения миграции;
- $\gamma$  — определяющего конечный набор серверов, на которые выполняются миграции  $VM$ .

Численные значения для всех указанных безразмерных параметров лежат в интервале  $(0, 1)$  и необходимы для контроля процесса перераспределения  $VM$  системным администратором ДЦ.

Требуется уменьшить энергопотребление в ДЦ с учетом того, что время миграций  $VM$  не должно превышать  $t_{vm_j}$  (время получаемое из условий SLA на каждую  $VM$ ).

Рассматриваемая задача может быть сформулирована следующим образом. Заданы множества серверов  $s_i \in S$  ( $i = \overline{1, N_s}$ ), каждый из которых характеризуется параметрами энергопотребления  $P_{max_i}$ ,  $P_{min_i}$ ,  $P_i$ , доступными ресурсами процессоров, памяти  $R_{max_i}^{CPU}$ ,  $R_{max_i}^{RAM}$ , и виртуальных машин  $VM_j \in VM$  ( $j = \overline{1, N_{vm}}$ ) каждая из которых характеризуется требуемыми значениями по ресурсам процессоров, памяти  $R_{vm_j}^{CPU}$ ,  $R_{vm_j}^{RAM}$  и текущими их значениями  $Q_{vm_j}^{CPU}$ ,  $Q_{vm_j}^{RAM}$ , а также ограничениями по времени миграции  $t_{vm_j}$  каждой  $VM$ .

При ограничениях, обеспечивающих условие достаточности ресурсов серверов для обслуживания виртуальных машин, требуется изменить исходное распределение виртуальных машин по серверам назначения, чтобы сократить суммарное энергопотребление серверов

ДЦ за счет изменения количества работающих серверов, обслуживающих виртуальные машины.

## Методика

В основу методики решения поставленной задачи перераспределения функционирующих виртуальных машин по серверам, учитывая, что задача относится к комбинаторным NP-трудным задачам, положена эвристическая процедура упорядоченного перебора вариантов в области допустимых решений.

Методика включает себя статический этап, в котором происходит сбор данных от системы мониторинга при нагрузочном тестировании серверов для определения зависимости потребляемой мощности от нагрузок, и динамический этап, в котором на основе собираемых данных принимается решение о перераспределении  $VM$ .

### Статический этап

По каждому  $i$ -ому серверу собирается статистика об энергопотреблении  $P_i$  при всех возможных загрузках ресурсов сервера  $Q_{vm_j}^{CPU}$ ,  $Q_{vm_j}^{RAM}$ . В результате выполнения этого этапа устанавливается зависимость энергопотребления сервера от загрузки ресурсов памяти  $P_i(Q_{vm_j}^{CPU}, Q_{vm_j}^{RAM})$ .

### Динамический этап

Системой мониторинга собираются данные о состоянии множества  $VM$ .

На этом шаге становятся известными данные о резервировании для каждой  $VM$  ресурсов сервера  $R_{vm_j}^{CPU}$ ,  $R_{vm_j}^{RAM}$ , данные о фактической загрузке ресурсов  $Q_{vm_j}^{CPU}$ ,  $Q_{vm_j}^{RAM}$ , параметры SLA для каждой  $VM$   $t_{vm_j}$ .

Расположение каждой  $j$ -й  $VM$  на  $i$ -м сервере выражается матрицей  $X$ , где каждый элемент  $x_{ij}$  — бинарная переменная

$$x_{ij} = \begin{cases} 1, & \text{при } VM_j \in S_i; \\ 0, & \text{при } VM_j \notin S_i, \end{cases}$$

при этом

$$\sum_{i=1}^{N_s} \sum_{j=1}^{N_{vm}} x_{ij} = N_{vm}. \quad (1)$$

Для каждого  $i$ -го сервера множества  $S$  вычисляется потенциальная мощность потребления (мощность при максимуме использования зарезервированных ресурсов  $VM$ ):

$$P_{\Pi_i} = P_{min_i} + \sum_{j=1}^{N_{vm}} P_i(R_{vm_j}^{CPU}, R_{vm_j}^{RAM})x_{ij}. \quad (2)$$

Отключаются сервера для которых выполняется равенство  $P_{\Pi_i} = P_{min_i}$ , при этом  $N_{ds}$  равен числу отключенных серверов (если таких нет, то  $N_{ds} = 0$ ).

Для остальных серверов проверяется, что

$$P_{\Pi_i} \leq P_{min_i} + \Delta\alpha_i, \quad (3)$$

где  $\Delta\alpha_i = (P_{max_i} - P_{min_i})\alpha_i$ .

При выполнении условия (3), номер сервера добавляется в множество **S1**.

Для каждого сервера множества **S1** проверяется условие на соответствие заданному времени миграции каждой принадлежащей ему *VM*

$$\frac{V_i(Q_{vm_j}^{CPU}, Q_{vm_j}^{RAM})}{W} < t_{m_j} \beta_j \text{ для всех } VM_j \in S_i. \quad (4)$$

Если условие выполнено, то номер сервера добавляется в множество **S2**.

Полученное множество **S2** сортируется по возрастанию значения  $P_{\min_i}$ , а для серверов с одинаковым значением  $P_{\min_i}$  — по критерию минимального значения отношения  $P_{\Pi_i} / P_{\min_i}$ .

Округленное до наименьшего целого произведение  $N_s \gamma$  определяет максимум количества серверов отсортированного множества **S2**, из которых формируется множество **S3**. Остальные номера серверов добавляются в множество **M**:  $M = S - S3$ .

Миграция выполняется, начиная с первого элемента множества **S3**, на сервера множества **M** при условии достаточности ресурса памяти на новом сервере, а также при условии, что энергопотребление всех переносимых *VM* на новом сервере меньше потенциального энергопотребления на исходном сервере.

Для этого по каждому *m* серверу множества **M** вычисляется остаток ресурса памяти сервера

$$R_{\text{ост},m}^{\text{CPU}} = R_{\text{max},m}^{\text{CPU}} - \sum_{j=1}^{Nvm} R_{vm_j}^{\text{CPU}} x_{mj}; \quad (5)$$

$$R_{\text{ост},m}^{\text{RAM}} = R_{\text{max},m}^{\text{RAM}} - \sum_{j=1}^{Nvm} R_{vm_j}^{\text{RAM}} x_{mj}. \quad (6)$$

Для миграции всех *VM* с сервера *k* множества **S3** на сервер *m* множества **M**, должны выполняться условия

$$\begin{cases} R_{\text{ост},m}^{\text{CPU}} > \sum_{j=1}^{Nvm} R_{vm_j}^{\text{CPU}} x_{kj}; \\ R_{\text{ост},m}^{\text{RAM}} > \sum_{j=1}^{Nvm} R_{vm_j}^{\text{RAM}} x_{kj}; \\ \sum_{j=1}^{Nvm} P_m(R_{vm_j}^{\text{CPU}}, R_{vm_j}^{\text{RAM}}) x_{mj} < \\ < P_{\min_k} + \sum_{j=1}^{Nvm} P_k(R_{vm_j}^{\text{CPU}}, R_{vm_j}^{\text{RAM}}) x_{kj}. \end{cases} \quad (7)$$

Если для какого-либо *k*-го сервера условия не выполняются, то он исключается из множества **S3**.

Если для *k*-го сервера условия выполняются, то среди серверов множества **M**, подходящих под эти условия, *VM* переносится на сервер *m*, для которого минимально значение  $\sum_{j=1}^{Nvm} P_m(R_{vm_j}^{\text{CPU}}, R_{vm_j}^{\text{RAM}}) x_{mj}$ , после этого сервер *k* отключается; число отключенных серверов  $N_{ds}$  увеличивается на 1.

Остаток ресурса на сервере *m* становится равным

$$R_{\text{ост},m}^{\text{CPUnew}} = R_{\text{ост},m}^{\text{CPU}} - \sum_{j=1}^{Nvm} R_{vm_j}^{\text{CPU}} x_{kj}; \quad (8)$$

$$R_{\text{ост},m}^{\text{RAMnew}} = R_{\text{ост},m}^{\text{RAM}} - \sum_{j=1}^{Nvm} R_{vm_j}^{\text{RAM}} x_{kj}. \quad (9)$$

Сервер *k* исключается из **S3**.

По аналогии проверяются оставшиеся сервера множества **S3**. Результат перераспределений содержится в матрице  $X^{\text{final}}$ . Мощность ДЦ после выполнения миграций равна

$$P_{\text{ДЦ}} = \sum_{k=1}^{Nds} P_{\min_k} + \sum_{k=1}^{Nds} \sum_{j=1}^{Nvm} P_k(Q_{vm_j}^{\text{CPU}}, Q_{vm_j}^{\text{RAM}}) x_{kj} + \sum_{m=1}^{Nds} \sum_{j=1}^{Nvm} P_m(Q_{vm_j}^{\text{CPU}}, Q_{vm_j}^{\text{RAM}}) x_{mj}^{\text{final}}, \quad (10)$$

где  $P_{\text{ДЦ}}$  — потребляемая мощность ДЦ до выполнения шагов методики.

По предложенной методике разработан алгоритм, представленный на рис. 2.

### Пример

Рассмотрена вычислительная сеть ДЦ с кластером из 5 серверов. Множество  $S = \{1, 2, 3, 4, 5\}$ . Пропускная способность канала между серверами одинакова  $W = 10$  Гбит/с. Функция передаваемого объема данных  $V_i(Q^{\text{CPU}}, Q^{\text{RAM}})$  при миграциях *VM* равна объему *VM*,  $\alpha$ ,  $\beta$ ,  $\gamma$  имеют одинаковые значения для всех серверов и равны 0,6; 0,9; 0,35, соответственно. Исходные данные заданы в виде табл. 1.

На статическом этапе методики для каждого из серверов установлена функция зависимости потребляемой мощности от загрузки  $P_i(Q^{\text{CPU}}, Q^{\text{RAM}})$ , она имеет линейную зависимость в диапазоне от  $P_{\min_i}$  до  $P_{\max_i}$ , возрастающая прямо пропорционально  $Q^{\text{CPU}}$ .

Перейдем к динамическому этапу методики.

Системой мониторинга получены данные о состоянии *VM* (табл. 2).

Таблица 1

### Исходные данные кластера серверов

Параметры	Номер сервера <i>i</i>				
	1	2	3	4	5
$P_{\max_i}$ , Вт	600	700	600	600	800
$P_{\min_i}$ , Вт	200	300	200	250	400
$R_{\max_i}^{\text{RAM}}$ , Гб	64	80	64	64	80
$R_{\max_i}^{\text{CPU}}$ , количество ядер	32	48	32	24	48

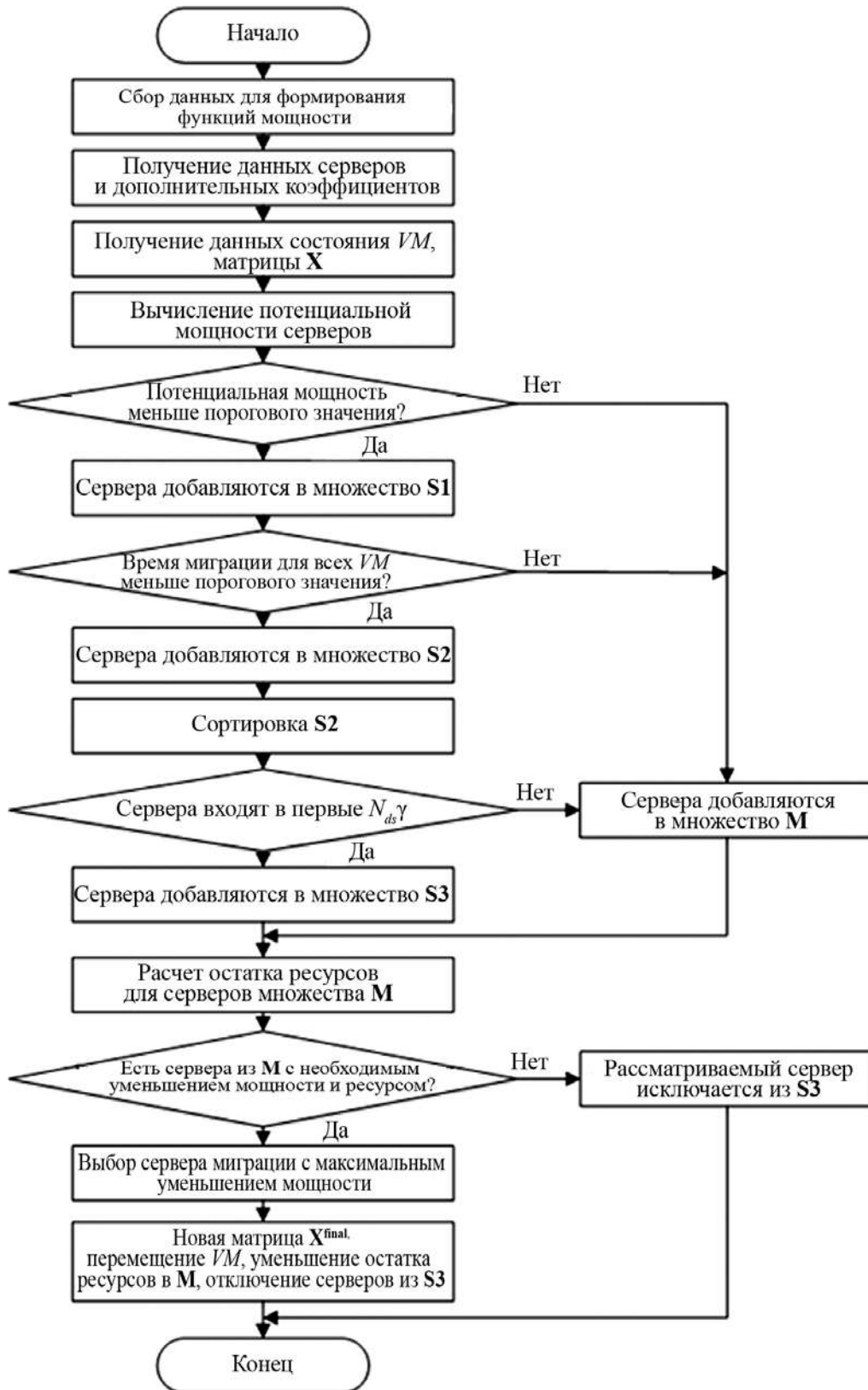


Рис. 2. Алгоритм методики перераспределения ресурсов

Таблица 2

## Данные о VM серверов

Номера VM <sub>j</sub>	1	2	3	4	5	6	7	8
$R_{vm_j}^{RAM}$ , Гб	30	8	8	2	4	2	4	8
$R_{vm_j}^{CPU}$ , количество ядер	16	4	4	12	4	4	12	16
$Q_{vm_j}^{RAM}$ , Гб	10	3	4	1	2	2	2	3
$Q_{vm_j}^{CPU}$ , количество ядер	6	2	4	6	2	2	6	10
Принадлежность $i$ серверу	2	1	4	3	1	5	4	5
$t_{vm_j}$ , с	3	3	2	2	2	2	4	3

На начальном этапе матрица  $X$  имеет вид

$i$	$j$							
	1	2	3	4	5	6	7	8
1	0	1	0	0	1	0	0	0
2	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0
4	0	0	1	0	0	0	1	0
5	0	0	0	0	0	1	0	1

Потенциальная мощность серверов, учитывая, что потребляемая мощность от загрузки имеет линейную зависимость (по изменению загрузки CPU), равна

$$P_{П1} = 200 + \frac{(600-200) \cdot 4}{32} + \frac{(600-200) \cdot 4}{32} = 300;$$

$$P_{П2} = 300 + \frac{(700-300) \cdot 16}{48} = 433,33;$$

$$P_{П3} = 200 + \frac{(600-200) \cdot 12}{32} = 350;$$

$$P_{П4} = 250 + \frac{(600-250) \cdot 4}{24} + \frac{(600-250) \cdot 12}{24} = 483,33;$$

$$P_{П5} = 400 + \frac{(800-400) \cdot 4}{48} + \frac{(800-400) \cdot 16}{48} = 566,66.$$

По каждому серверу проверяется условие :

$$300 < 200 + (600 - 200) \cdot 0,6 = 440 \Rightarrow i = 1 \in \mathbf{S1};$$

$$433,33 < 300 + (700 - 300) \cdot 0,6 = 540 \Rightarrow i = 2 \in \mathbf{S1};$$

$$350 < 200 + (600 - 200) \cdot 0,6 = 440 \Rightarrow i = 3 \in \mathbf{S1};$$

$$483,33 > 250 + (600 - 250) \cdot 0,6 = 460 \Rightarrow i = 4 \notin \mathbf{S1};$$

$$566,66 < 400 + (800 - 400) \cdot 0,6 = 640 \Rightarrow i = 5 \in \mathbf{S1}.$$

Для сервера 4 не выполняется условие, следовательно, множество  $\mathbf{S1}$  имеет вид  $\{1, 2, 3, 5\}$ .

На основе значений загрузки  $Q_{vm_j}^{RAM}$  каждой VM проверяется для каждого сервера множества  $\mathbf{S1}$  соответствие условию времени миграции:

для  $i = 1$

$$\frac{3 \cdot 8}{10} = 2,4 < 3 \cdot 0,9 = 2,7;$$

$$\frac{2 \cdot 8}{10} = 1,6 < 2 \cdot 0,9 = 1,8 \Rightarrow i = 1 \in \mathbf{S2};$$

для  $i = 2$ :

$$\frac{10 \cdot 8}{10} = 8 > 3 \cdot 0,9 = 2,7 \Rightarrow i = 2 \notin \mathbf{S2};$$

для  $i = 3$

$$\frac{1 \cdot 8}{10} = 0,8 < 2 \cdot 0,9 = 1,8 \Rightarrow i = 3 \in \mathbf{S2};$$

для  $i = 5$

$$\frac{2 \cdot 8}{10} = 1,6 < 2 \cdot 0,9 = 1,8;$$

$$\frac{3 \cdot 8}{10} = 2,4 < 3 \cdot 0,9 = 2,7 \Rightarrow i = 5 \in \mathbf{S2}.$$

Для сервера 2 не выполняется условие, следовательно, новое множество  $\mathbf{S2}$  имеет вид  $\{1, 3, 5\}$ . В табл. 3 представлены необходимые данные для сортировки множества  $\mathbf{S2}$ .

Сначала сервера сортируются по убыванию  $P_{\min_i}$ . Затем, для серверов с одинаковым  $P_{\min_i}$ , происходит сортировка по возрастанию  $P_{П_i}/P_{\min_i}$ . Таким образом, отсортированное множество  $\mathbf{S2}$  выглядит как  $\mathbf{S2}^{\text{sort}} = \{5, 1, 3\}$ . С учетом того, что  $N_s \gamma = 5 \cdot 0,35 = 1,75$  и округлив до меньшего целого, получим 1. Следовательно, только первый элемент отсортированного множества  $\mathbf{S2}^{\text{sort}}$  входит в новое множество  $\mathbf{S3} = \{5\}$ , а множество  $\mathbf{M}$  имеет вид  $\{1, 2, 3, 4\}$ .

Рассчитаем  $R_{\text{ост}}$  для множества  $\mathbf{M}$ :

$$R_{\text{ост1}}^{\text{CPU}} = 32 - (4 + 4) = 24; \quad R_{\text{ост1}}^{\text{RAM}} = 64 - (8 + 4) = 52;$$

$$R_{\text{ост2}}^{\text{CPU}} = 48 - (16) = 32; \quad R_{\text{ост2}}^{\text{RAM}} = 80 - (30) = 50;$$

$$R_{\text{ост3}}^{\text{CPU}} = 32 - (12) = 20; \quad R_{\text{ост3}}^{\text{RAM}} = 64 - (2) = 62;$$

$$R_{\text{ост4}}^{\text{CPU}} = 24 - (4 + 12) = 8; \quad R_{\text{ост4}}^{\text{RAM}} = 64 - (8 + 4) = 52.$$

Для множества  $\mathbf{S3}$  (начиная с первого элемента) проверяются условия предоставления ресурсов и минимизации мощности относительно всех серверов множества  $\mathbf{M}$ .

Сервер  $k$  — сервер из множества  $S_3$ , для которого ведется поиск сервера миграции,  $m$  — сервер миграции из множества  $M$ :

для  $m=1, k=5$

$$\begin{cases} 24 > 4+16 = 20; \\ 52 > 2+8 = 10; \\ \frac{(600-200) \cdot 4}{32} + \frac{(600-200) \cdot 16}{32} = \\ = 250 < 400 + 33,33 + 133,33 = 566,66; \end{cases}$$

для  $m=2, k=5$

$$\begin{cases} 32 > 4+16 = 20; \\ 50 > 2+8 = 10; \\ \frac{(700-300) \cdot 4}{48} + \frac{(700-300) \cdot 16}{48} = \\ = 166,67 < 400 + 33,33 + 133,33 = 566,66; \end{cases}$$

для  $m=3, k=5$

$$\begin{cases} 20 = 4+16 = 20; \\ 62 > 2+8 = 10; \\ \frac{(600-200) \cdot 4}{32} + \frac{(600-200) \cdot 16}{32} = \\ = 250 < 400 + 33,33 + 133,33 = 566,66; \end{cases}$$

для  $m=4, k=5$

$$\begin{cases} 8 < 4+16 = 20; \\ 52 > 2+8 = 10; \\ \frac{(600-250) \cdot 4}{32} + \frac{(600-250) \cdot 16}{32} = \\ = 291,67 < 400 + 33,33 + 133,33 = 566,66. \end{cases}$$

На сервере 4 не выполняется условие предоставления ресурса. На сервере 3 ресурс заполняется до предела, в то время как в исходном неравенстве требуется использовать строго большее количество свободного ресурса, поэтому перемещение  $VM$  с сервера 5 возможно на сервера 1 или 2. Перемещение происходит на любой сервер с наименьшим значением суммы

$$\sum_{j=1}^{Nvm} P_m(R_{vm_j}^{CPU}, R_{vm_j}^{RAM}) x_{mj}.$$

Для сервера 1:

$$\frac{(600-200) \cdot 4}{32} + \frac{(600-200) \cdot 16}{32} = 250;$$

для сервера 2:

$$\frac{(700-300) \cdot 4}{48} + \frac{(700-300) \cdot 16}{48} = 166,67.$$

Следовательно,  $VM$  с сервера 5 перемещаются на сервер 2. Остаток ресурсов на сервере 2 равен

$$R_{ост2}^{CPUnew} = 32 - (4+16) = 12;$$

$$R_{ост2}^{RAMnew} = 50 - (2+8) = 40.$$

Сервер 5 отключается. Следующий элемент множества  $S_3$  отсутствует. Выполнение методики заканчивается. Число отключенных серверов  $N_{ds} = 1$ . Потребляемая мощность ДЦ уменьшена на

$$400 + \frac{(800-400) \cdot 2}{48} + \frac{(800-400) \cdot 10}{48} - \frac{(700-300) \cdot 2}{48} - \frac{(700-300) \cdot 10}{48} = 400.$$

Результирующая матрица  $X^{final}$  после перестановок выглядит как:

$i$	$j$							
	1	2	3	4	5	6	7	8
1	0	1	0	0	1	0	0	0
2	1	0	0	0	0	1	0	1
3	0	0	0	1	0	0	0	0
4	0	0	1	0	0	0	1	0

Таким образом, получено, что восемь  $VM$  могут быть обслужены четырьмя серверами. Рассмотренный пример иллюстрирует этапы методики, которая может быть применена для решения задач реальной размерности.

## Выводы

Методика на основе предложенной эвристической процедуры повышает энергоэффективность ДЦ, при этом исключает полный перебор вариантов при решении задач реальной размерности.

На основе программной реализации методики ведутся работы по оценке эффективности при масштабировании решаемых задач перераспределения функционирующих виртуальных машин по серверам в дата-центре.

## Литература

1. **Облачный** провайдинг: экономика, стратегии, бизнес-модели [Электрон. ресурс] URL: <http://www.iks-consulting.ru/raitings-220.html> (дата обращения 03.01.2017)
2. **Обзор:** облачные сервисы 2014 [Электрон. ресурс] URL: [http://www.cnews.ru/reviews/cloud\\_2014/review\\_table/655b2c19b5e2ed63f1792efd5a3e786121fd69d2/](http://www.cnews.ru/reviews/cloud_2014/review_table/655b2c19b5e2ed63f1792efd5a3e786121fd69d2/) (дата обращения 23.12.2016)
3. **Google's DeepMind Trains AI to Cut its Energy Bills by 40%** [Электрон. ресурс] URL: <http://www.wired.co.uk/article/google-deepmind-data-centres-efficiency> (дата обращения 25.11.2017)
4. **Pedram M., Hwang I.** Power and Performance Modeling in a Virtualized Server System [Электрон. ре-

сурс] URL: <http://www.mpedram.com/Papers/Virtual-system-modeling-greencom10.pdf> (дата обращения 20.02.2017)

5. **Аверьянихин А.Е., Котельницкий А.В., Муравьев К.А.** Методика расчета оптимального числа узлов кластера виртуализации частного облака виртуальных рабочих столов по критерию эффективности // Международный научно-исследовательский журнал. 2016. № 5 (47). С. 6—13.

6. **Соловьев В.П., Удовиченко А.О.** Метод планирования размещения группы виртуальных машин с перераспределением ресурсов // Программные продукты и системы. 2012. № 1. С. 134—137.

7. **Ворожцов А.С., Тутова Н.В., Тутов А.В.** Методика оптимального распределения виртуальных серверов в центрах обработки данных // T-Comm – Телекоммуникации и транспорт. 2015. № 7. С. 5—10.

8. **Абросимов Л. И., Крамаренко М. Д., Гончаренко О. С.** Лабораторные работы по исследованию вероятностно-временных характеристик прокси-сервера [Электрон. ресурс] URL:<http://network-journal.mpei.ac.ru/cgi-bin/main.pl?l=ru&n=27&pa=15&ar=1> (дата обращения 28.03.2017)

---

## References

1. **Oblachnyy** Provayding: Ekonomika, Strategii, Biznes-modeli [Elektron. Resurs] URL: <http://www.iksconsulting.ru/raitings-220.html> (Data Obrashcheniya 03.01.2017) (in Russian).

2. **Obzor:** Oblachnye Servisy 2014 [Elektron. Resurs] URL: [http://www.cnews.ru/reviews/cloud\\_2014/review\\_table/655b2c19b5e2ed63f1792efd5a3e786121fd69d2/](http://www.cnews.ru/reviews/cloud_2014/review_table/655b2c19b5e2ed63f1792efd5a3e786121fd69d2/) (Data Obrashcheniya 23.12.2016) (in Russian).

3. Google's DeepMind trains AI to cut its energy bills by 40% [Elektron. resurs] URL: <http://www.wired.co.uk/article/google-deepmind-data-centres-efficiency> (Data Obrashcheniya 25.11.2017)

4. **Pedram M., Hwang I.** Power and Performance Modeling in a Virtualized Server System [Elektron. Resurs] URL: <http://www.mpedram.com/Papers/Virtual-system-modeling-greencom10.pdf> (Data Obrashcheniya 20.02.2017)

5. **Aver'yanihin A.E., Kotel'nitskiy A.V., Murav'ev K.A.** Metodika Rascheta Optimal'nogo Chisla Uzlov Klastera Virtualizatsii Chastnogo Oblaka Virtual'nyh Rabochih Stolov po Kriteriyu Effektivnosti. Mezhdunarodnyy Nauchno-issledovatel'skiy Zhurnal. 2016;5 (47):6—13. (in Russian).

6. **Solov'ev V.P., Udovichenko A.O.** Metod Planirovaniya Razmeshcheniya Gruppy Virtual'nyh Mashin s Pereraspredeleniem Resursov. Programmnye Produkty i Sistemy. 2012;1:134—137. (in Russian).

7. **Vorozhtsov A.S., Tutova N.V., Tutov A.V.** Metodika Optimal'nogo Raspredeleniya Virtual'nyh Serverov v Tsentrah Obrabotki Danyih. T-Comm – Telekommunikatsii i Transport. 2015;7:5—10. (in Russian).

8. **Abrosimov L. I., Kramarenko M. D., Goncharenko O. S.** Laboratornye Raboty po Issledovaniyu veroyatnostno-vremennyh Harakteristik Proksi-servera [Elektron. Resurs] URL:<http://network-journal.mpei.ac.ru/cgi-bin/main.pl?l=ru&n=27&pa=15&ar=1> (Data Obrashcheniya 28.03.2017) (in Russian).

---

## Сведения об авторах

**Ларин Александр Анатольевич** — аспирант кафедры вычислительных машин, систем и сетей НИУ «МЭИ», e-mail: [kamitor@yandex.ru](mailto:kamitor@yandex.ru)

**Абросимов Леонид Иванович** — доктор технических наук, профессор кафедры вычислительных машин, систем и сетей НИУ «МЭИ», e-mail: [AbrosimovLI@mpei.ru](mailto:AbrosimovLI@mpei.ru)

---

## Information about authors

**Larin Aleksandr A.** — PhD-student of Computing Machines, Systems and Networks Dept., NRU MPEI, e-mail: [kamitor@yandex.ru](mailto:kamitor@yandex.ru)

**Abrosimov Leonid I.** — Dr.Sci. (Techn.), Professor of Computing Machines, Systems and Networks Dept., NRU MPEI, e-mail: [AbrosimovLI@mpei.ru](mailto:AbrosimovLI@mpei.ru)

*Статья поступила в редакцию 17.02.2017*