References

1. Kipjatkova I.S., Ronzhin A.L., Karpov A.A. *Automatic Processing of Spoken Russian Speech*. Saint Petersburg: Sankt-Peterburgskij institut informatiki i avtomatizatsii RAN, 2013, 314 p. (In Russ.)
2. Sapozhkov M.A. *Speech Signal in Cybernetics and Communication*. Moscow: Gosizdat. po voprosam svjazi i radio, 1963, 452 p. (In Russ.)
3. Sorokin V.N. *Theory of Speech Production*. Moscow: Radio i svjaz', 1985, 312 p. (In Russ.)
4. Flanagan D.L. *Analysis, Synthesis and Perception of Speech*. Moscow: Svjaz', 1968, 396 p. (In Russ.)
5. Tampel' I.B., Karpov A.A. *Automatic Speech Recognition*. Saint Petersburg: Universitet ITMO, 2016, 140 p. (In Russ.)
6. Vocoder Telephony. *Methods and Problems*. Ed. by A.A. Pirogova. Moscow: Svjaz', 1974, 536 p. (In Russ.)
7. Galunov V.I., Garbuk V.I. Acoustic theory of speech formation and the system of phonetic signs. Saint Petersburg: Saint Petersburg State University; Saint Petersburg Research Institute of Ear, Throat, Nose and Speech. URL: https://studylib.ru/doc/3831855/akusticheskaya-teoriya-recheobrazovaniya-i-sistema (accessed: 14.02.2021).
8. Tsviker E., Fel'dkeller R. *The Ear as a Receiver of Information*. Moscow: Svjaz', 1971, 255 p. (In Russ.)
9. Foulkes J.D. Computer identification of vowel types. *JASA*, 1961, vol. 33, no. 1, pp. 7–11 (given by [2]).
10. Fel'dkeller R., Tsviker E. *The Ear as a Receiver of Information*. Moscow: Svjaz', 1965, 104 p. (In Russ.)
11. Fant G. Acoustic theory of speech production: with calculations based on x-ray studies of Russian articulations. Mouton & Co, s'-Gravenhage, 1960 (given by [2]).

Received 29.03.2021

ТЕХНОЛОГИИ ТЕЛЕКОММУНИКАЦИЙ

УДК 004.942

МЕТОДЫ И МОДЕЛИ СЕРВИСА РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В КЛАСТЕРАХ С БАЛАНСИРОВКОЙ НАГРУЗКИ ЦЕНТРОВ ОБРАБОТКИ ДАННЫХ

Мочалов В.П., Линец Г.И., Братченко Н.Ю., Палканов И.С.
Северо-Кавказский федеральный университет, Ставрополь, РФ
E-mail: n.b.20062@yandex.ru

Современные центры обработки данных представляют собой комплексные решения по управлению предприятиями и корпорациями, по организации систем обработки и хранения данных, по эффективному распределению программных приложений между доступными ресурсами. Объектом исследования являются кластеры с балансировкой нагрузки центров обработки данных, содержащие определенное множество серверов приложений, файл серверов, систем хранения данных, систему ввода-вывода, связанных между собой системой коммутации и каналами связи. Целью работы является повышение эффективности функционирования виртуализированных центров обработки данных путем разработки методических подходов, рациональных методов и моделей решения задач распределения нагрузки по его аппаратно-программным средствам. Рассматриваются задачи построения взаимосвязанных математических методов и моделей, необходимых для построения специализированного программного обеспечения, способствующего решению задач создания рациональных планов распределения ресурсов центров обработки данных на заданном временном интервале. В основу предложенных методов и моделей положена система корректных отображений совокупности параметров запросов на известные характеристики физических ресурсов центров обработки данных. В качестве критерия оптимизации используются параметры максимизации производительности системы за определенный временной интервал.

Ключевые слова: *центры обработки данных, вычислительные кластеры, сервис балансировки нагрузки, эффективное распределения ресурсов, система отображений*

Введение

Информационная структура центра обработки данных (ЦОД) включает в себя: серверный комплекс, представленный в виде кластеров с балансировкой нагрузки (Load Balancing Cluster), в котором применяются многоплатформенные решения, обеспечивающие равномерное распределение нагрузки между серверами, средства памяти, включающие множество устройств хранения данных в виде дисковых массивов, систему хранения данных, средства копирования и восстановления, средства связи и коммутации, средства телекоммуникации, обеспечивающие взаимодействие между операторами и пользователями ЦОД.

Формальная модель кластера ЦОД приведена на рисунке 1. При использовании кластера с балансировкой нагрузки множество ресурсных запросов пользователей поступает на распределитель нагрузки, который по определенным правилам равномерно распределяет их между серверами кластера. При этом может производиться направление поступившего запроса на первый доступный ресурс кластера или выбор случайным образом ресурса из множества доступных и обеспечивающих реализацию данного запроса. В основу подобных решений могут быть положены также широко известные эвристические алгоритмы на основе жадных стратегий. Например, методы распределения нагрузки, заложенные в циклическую логику алгоритма Round Robin, формируют на любой запрос соответствующий IP-адрес, обеспечивая тем самым формирование запросов к серверам по круговому циклу и реализуя одинаковое количество запросов к каждому серверу.

И хотя время работы данного алгоритма не является слишком большим, он мало пригоден для практического применения. Данный алгоритм не учитывает доступные объемы ресурсов серверов, их состояние, производительность, текущую загрузку, количество поддерживаемых подключений. Более совершенные варианты этого алгоритма, разработанные специально для улучшения характеристик кластеров, Least Connections, Weighted Least Connections, Locality-Based Least Connections Scheduling, Destination Hash Scheduling, Source Hash Scheduling, Sticky Sessions, хотя и используют методы поиска серверов с наименьшей загрузкой и передачи им нового запроса, также не обеспечивают в полной мере реализацию оптимальной технологии балансировки нагрузки, достижения эффективности кластеризации, оптимальной загрузки серверов при минимальном времени реализации приложений.

В основе широко используемых систем распределения нагрузки современных гипервизоров (VMware Infrastructure, VMware ESXi Server, Microsoft Hyper-V) лежат как методы прямого распределения приложений, так и настройки, использующие статистику прогнозирования использования вычислительных ресурсов [1–3; 7]. Например, при применении планировщика ресурсов на базе алгоритмов платформы OpenStack не учитывается в полной мере динамически изменяющаяся, непредсказуемая интенсивность нагрузки, динамика происходящих процессов, не обеспечивается в полной мере решение задачи балансировки нагрузки, распределения ресурсов физических серверов между виртуальными машинами (ВМ) и сетевыми приложениями в реальном времени, а значит, и гарантированное обеспечение качества обслуживания [4–6; 9].

Повышение эффективности функционирования аппаратно-программной платформы ЦОД вызывает необходимость разработки рациональных методов и моделей реализации сервиса распределения ресурсов в кластерах с балансировкой нагрузки, оптимизационных алгоритмов распределения запросов и программных приложений на физических серверах в реальном времени.

Для исключения проблемы перегрузки ресурсов виртуального ЦОД, ввиду ограниченности его производительности, размеров оперативной памяти вычислительных серверов, пропускной способности системы коммутации и каналов передачи данных, обеспечения соответствия между типом запросов и техническими параметрами запрашиваемых ресурсов, на отношения между параметрами системы запросов и соответствующих им характеристик ресурсов ЦОД необходимо наложить ряд ограничений. В качестве критерия оптимальности могут быть использованы параметры максимизации производительности системы, достигаемые за фиксированный период времени.

Математическая постановка задачи распределения ресурсов в кластерах с балансировкой нагрузки ЦОД

Каждый запрос на реализацию прикладных функций может выполняться на одном из серверов кластера при соблюдении корректных отно-

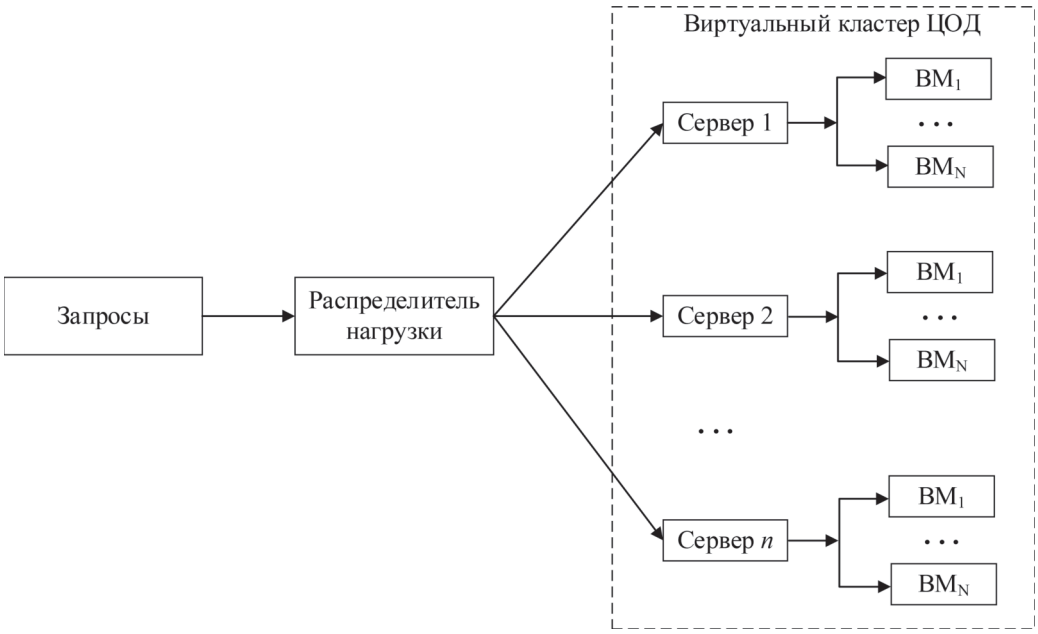

Рисунок 1. Формальная модель кластера ЦОД

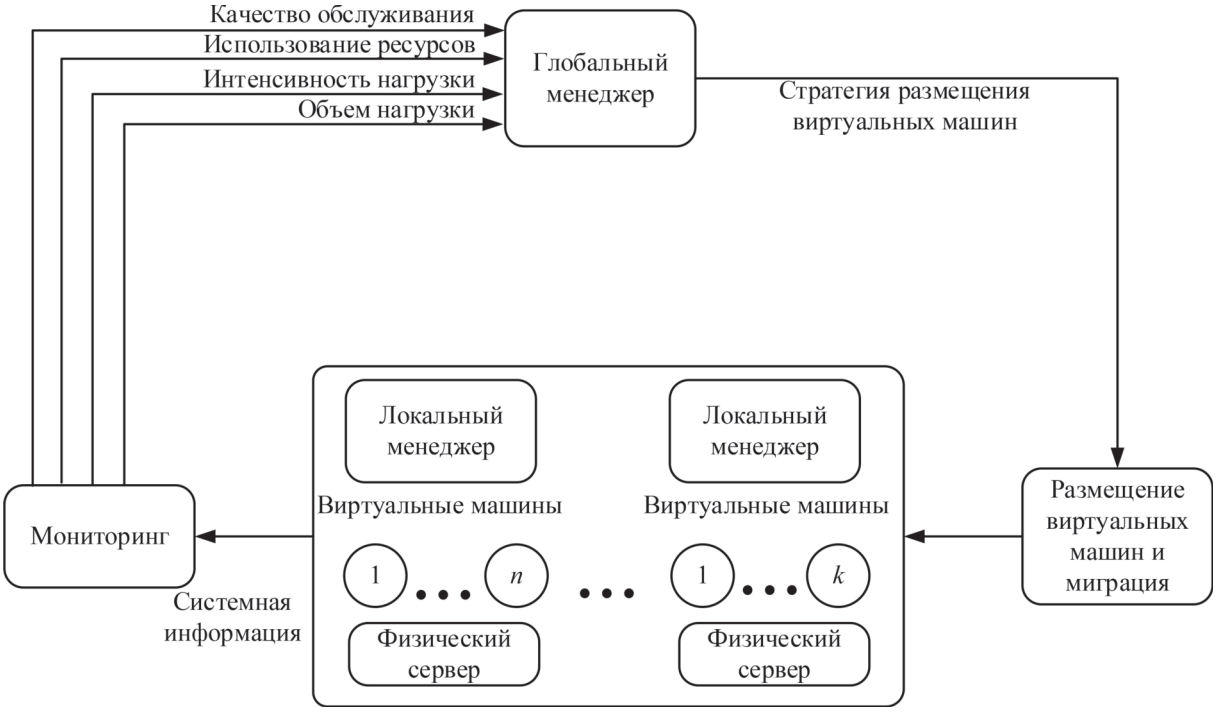

Рисунок 2. Структура системы управления аппаратно-программной платформы ЦОД

шений между параметрами запроса, реализуемого программными приложениями разного уровня сложности и соответствующими характеристиками физических ресурсов ЦОД. Структура системы управления аппаратно-программной платформы ЦОД приведена на рисунке 2.

По требованию пользователей ЦОД система управления производит формирование ВМ с определенными ресурсными параметрами. Локальный менеджер осуществляет контроль загрузки физических серверов и размещение запросов на ресурсы ВМ. Система мониторинга ЦОД проводит передачу полученной информации глобальному менеджеру, который осуществляет миграцию ВМ и управляет перераспределением физических серверов. При этом последовательно реализуются следующие задачи:

– прием запросов от абонентов на реализацию услуг;

– распределение запросов по серверам кластера;

– выбор сервера в кластере, способного реализовать запросы пользователей;

– направление запросов к выбранному серверу кластера;

– распределение реализуемых программных приложений по серверам кластера;

– формирование совокупности виртуальных машин, реализующих приложения;

– прием результатов решения задач;

– отправление полученных результатов пользователям.

Состав и структуру серверного комплекса ЦОД [5; 8] можно представить в виде кортежа

$$H = \left(P \cup M \cup K, L\right), \quad (1)$$

где P – множество серверов кластера; M – множество устройств хранения данных; K, L – множество ресурсов телекоммуникационного оборудования.

Запросы на реализацию приложений можно формализовать следующим образом:

$$G = \left(W \cup S, E\right), \quad (2)$$

где W – множество атрибутов запросов; S, E – характеристики запрашиваемых ресурсов кластера.

Отношение между характеристиками доступных физических ресурсов и параметрами запросов можно формализовать в виде следующего отображения:

$$G \to H = \left\{W \to P, S \to M, E \to \left\{K, L\right\}\right\}. \quad (3)$$

Данное отображение будет корректным при выполнении системы ограничений на отношения между запросами и ресурсами.

1. Модель ограничений при распределении запросов по серверам кластера:

– каждое приложение i, $i = \overline{1,n}$ выполняется только на одном сервере j, $j = \overline{1,m}$:

$$\sum_{j=1}^{m} x_{ij} = 1, \quad x_{ij} \in \left\{0,1\right\}, \quad (4)$$

где m – число серверов; $i = \overline{1,n}$ – число приложений; причем время реализации приложений, распределенных на j сервер, не должно превышать заданного значения $t_{\text{зад.}}$:

$$\sum_{i=1}^{m} x_{ij}\tau_{ij} \le t_{\text{зад.}}, \quad \tau_{ij} = Q_i / S_j, \quad (5)$$

где Q_i – сложность реализации приложений; S_j – характеристика производительности сервера.

Приведенный далее метод распределения запросов по серверам кластера обеспечивает вполне приемлемые результаты при краткосрочном планировании:

– упорядочиваем все реализуемые приложения по возрастанию длительности их выполнения;

– определяем такой порядковый номер p приложения, при котором

$$t_1 = \sum_{i=1}^{p} \tau_i, \quad (6)$$

где τ_i – время реализации i-го приложения, имеющего наименьшее отклонение от $t_{\text{зад.}}$:

$$t_1 - t_{\text{зад.}} \to \min; \quad (7)$$

– переходим к порядковому номеру приложения $\left(p+1\right)$ и повторяем эту процедуру применительно к оставшимся приложениям. Последовательно получаем t_2, t_3 и т. д.;

– выделенные группы соответствующих запросов распределяем на соответствующих серверах кластера.

2. Распределение программных приложений $F = \left\{f_1, f_2, \ldots, f_n\right\}$ по $k \in K$ ВМ осуществляется с учетом ограниченных объемов ресурсов ЦОД. При этом необходимо определить число задействованных ВМ и состав распределенных на них программных приложений с учетом следующих ограничений:

– приложение может быть реализовано только на одной из ВМ

$$\sum_{n} a_{jn} \le 1, \quad (8)$$

где

$$a_{jn} = \begin{cases} 1, \text{ если приложение} \\ \text{выполняется на } n \text{ ВМ;} \\ 0, \text{ в противном случае,} \end{cases}$$
$$j = \overline{1,n}, \quad n = \overline{1,k};$$

– назначенные на ВМ приложения запрашивают суммарные объемы ресурсов не больше имеющихся в наличии:

$$\sum_{n} a_{jn} C_j \le C_n, \quad \sum_{n} a_{jn} m_j \le M_n, \quad (9)$$

где C_n, M_n – производительность и память j ВМ;

– реализуется условие неделимости приложений для всех временных интервалов t_n их выполнения:

$$\sum_{t=0}^{t_n} \sum_{n} a_{jn} = t_j \sum_{n} a_{jn}. \quad (10)$$

Данная задача относится к классу NP-полных. Для её решения можно использовать приведенный ниже эвристический алгоритм [10; 12]:

– упорядочиваем программные приложения $j \in r$ в порядке убывания их запросов на объёмы ресурсов ВМ;

– ранжируем ВМ $k \in K$ в порядке возрастания их производительности. Для каждого очередного приложения $j\left(k\right)$ на каждом временном интервале t_n подбираем ВМ, способную реализовать соответствующее приложение;

– если ВМ не удается подобрать, то приложение не будет выполнено. В противном случае на

ВМ фиксируются требуемые объемы ресурсов. Данный алгоритм имеет полиномиальную сложность и обеспечивает близкое к оптимальным распределение [11; 15].

3. Модель ограничений при распределении виртуальных машин по серверам кластера. ЦОД содержит определенное множество серверов приложений S_i, $i=\overline{(1,L)}$ и файл-серверов H_j, $j=\overline{(1,M)}$, необходимо распределить по ним множество BM_k, $k=\overline{(1,N)}$ при соблюдении следующих ограничений:

– каждая ВМ может быть размещена только на одном сервере

$$\sum_{i=1}^{L}\sum_{j=1}^{M}\sum_{k=1}^{N} x_{ijk}=1, \qquad (11)$$

где

$$x_{ijk}=\begin{cases}1, \text{если } BM_k \text{ располагается}\\ \text{на } S_i \text{ или } H_j;\\ 0, \text{в противном случае;}\end{cases}$$

– требования ВМ, предъявляемые к ресурсу оперативной памяти RAM физических серверов, должны удовлетворять ограничению

$$\sum_{i=1}^{L}\sum_{j=1}^{M}\sum_{k=1}^{N} RAM_k x_{ijk} < RAM_{ij}. \qquad (12)$$

Используемые ресурсы всех ВМ не должны быть больше ресурсов серверов:

– ограничение на производительность серверов

$$\sum_{i=1}^{L}\sum_{j=1}^{M}\sum_{k=1}^{N} CPU_k x_{ijk} < CPU_j; \qquad (13)$$

– ограничение на объем памяти дисков

$$\sum_{i=1}^{L}\sum_{j=1}^{M}\sum_{k=1}^{N} S_k x_{ijk} < S_{ij}; \qquad (14)$$

– ограничение на производительность системы ввода-вывода данных

$$\sum_{i=1}^{L}\sum_{j=1}^{M}\sum_{k=1}^{N} IOPS_k x_{ijk} < IOPS_{ij}. \qquad (15)$$

Физические серверы и системы хранения данных соединены между собой системой коммутации. Поэтому при распределении ресурсов физических серверов необходимо учитывать основные параметры системы коммутации и каналов связи между ВМ и системой хранения данных.

Суммарная пропускная способность данной системы должна удовлетворять следующим условиям:

$$\sum_{i=1}^{n}\sum_{l=1}^{L} r_i l < \Pi, \quad \sum_{i=1}^{n}\sum_{l=1}^{L} r_i l < K, \qquad (16)$$

где $r_i\, l$ – требуемая пропускная способность i-й ВМ при использовании l виртуальных каналов; П и K – суммарная пропускная способность физических каналов и системы коммутации.

В качестве критериев оптимальности при распределении ресурсов ЦОД будем использовать показатели:

– наиболее полной загрузки оборудования ЦОД

$$K_1=\sum_{i=1}^{L}\sum_{j=1}^{M}\sum_{k=1}^{N} CPU_k x_{ijk} RAM_k x_{ijk}; \qquad (17)$$

– максимальной производительности системы ввода-вывода

$$K_2=\sum_{i=1}^{L}\sum_{j=1}^{M}\sum_{k=1}^{N} IOPS_k x_{ijk}; \qquad (18)$$

– максимальной производительности системы коммутации и передачи данных

$$K_3=\sum_{i=1}^{n}\sum_{l=1}^{L} r_i l. \qquad (19)$$

Задача оптимального распределения ресурсов ЦОД с учетом рассмотренных ограничений является NP-трудной, поэтому для ее решения необходимо воспользоваться не полиномиальными алгоритмами офлайн- или онлайн-размещения [13; 17].

Предлагаемый в данной работе алгоритм производит последовательную обработку запросов на реализацию услуг. Для этого осуществляется их первоначальное ранжирование по убыванию запрашиваемых ресурсов. При этом внешний планировщик ЦОД выделяет каждой ВМ максимальное значение ресурсов CPU, RAM, disk, IOPS и т. д. Определяется физический ресурс с минимальной остаточной суммой его параметров, и идет размещение на нем ВМ в соответствии со следующей схемой.

1. Производится ранжирование ВМ по показателю [4] $V_i=\sum_{j=1}^{K} C_j V_j^i$, где V_j^i – требование BM_i к ресурсу j, $j=\overline{1,K}$; C_j – коэффициент значимости ресурса.

2. В соответствии с исходной схемой выбирается ВМ с наибольшими требованиями к ресурсам и размещается на физическом элементе с минимальным ресурсом. Наиболее целесообразно использовать при этом итерационный алгоритм, основанный на подходах динамического программирования.

Например, итерационный жадный алгоритм работает следующим образом [14; 16]. В качестве входных процессов a_i множества $S=\{a_1, a_2, \ldots, a_n\}$ выступают конечные моменты f_i их реализации. Последовательно выбранные процессы объединяются процедурой Greedy

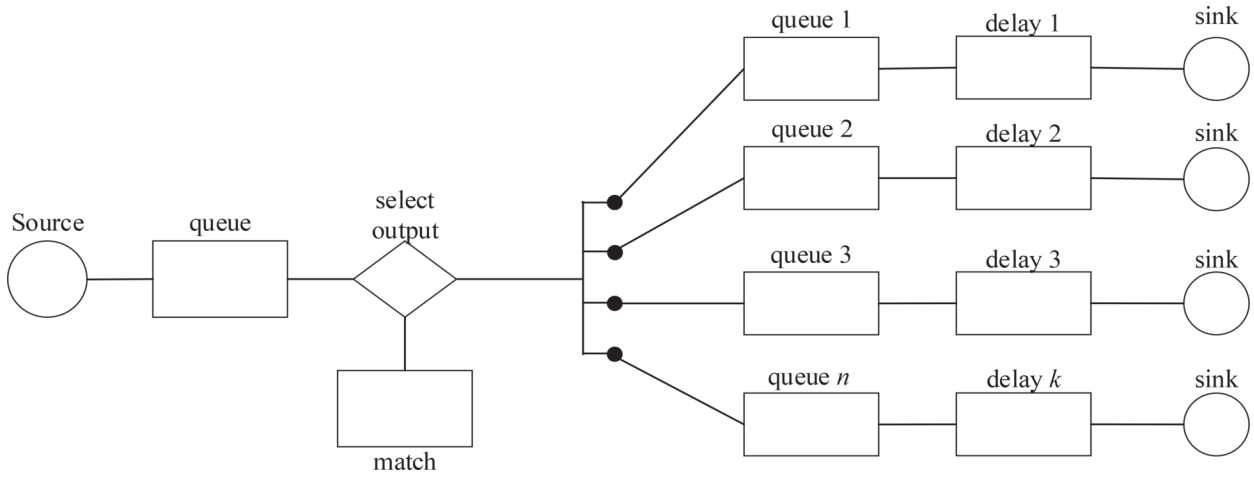

Рисунок 3. Модель кластера серверов

Activity Selector (s, f) в множество A, в котором f_i является максимальным временем окончания всех процессов $f_i = \max\{f_k : a_k \in A\}$. Greedy Activity Selector (s, f):

$$n \leftarrow length[s]; \quad A \leftarrow \{a_1\};$$
$$i \leftarrow 1; \quad for\ m \leftarrow 2\ to\ n;$$
$$do\ if\ S_m \geq f_i; \quad then\ A \leftarrow A \cup \{a_m\};$$
$$i \leftarrow m; \quad return\ A.$$

1. Внешний планировщик ЦОД осуществляет оценку производительности размещенной на сервере ВМ.

2. Если производительность ВМ ниже установленного уровня, внешний планировщик перемещает её на физический элемент с более высоким уровнем ресурсов и переходит на шаг 3.

3. Если производительность ВМ выше установленного уровня, внешний планировщик перемещает её на физический элемент с более низким уровнем ресурсов и переходит на шаг 3.

4. В случае невозможности распределения ВМ осуществляется переход к процедуре ограниченного перебора. Глубина перебора, определяющая максимальное число серверов, для которых назначается распределение, должна обеспечивать требуемый баланс между качеством сервиса и временем распределения ВМ, то есть находить приемлемое решение за допустимое время. Качество сервиса можно оценить вероятностью блокировки P_δ запросов на обслуживание, иначе – вероятностью отсутствия допустимых серверов в единицу времени. Для этого можно использовать формулу Эрланга

$$P_\delta = \frac{a^n / n!}{\sum_{k=0}^{n} a^k / k!}, \qquad (20)$$

где P_δ – вероятность блокировки; n – число серверов; a – интенсивность запросов.

Значение данной вероятности определяется рекуррентным способом [18]:

$$P_\delta = B(n, a) = \frac{B(n-1, a)}{B(n-a) + n/a}, \qquad (21)$$

где $n = 1, 2, ...,\ B(0, a) = 1$.

В случае достаточно больших значений n и a можно уменьшить время вычисления P_δ путем задания разницы $P_\delta(i) - P_\delta(i-1) < \delta$ и остановки выполнения процедуры перестановки при её достижении [10]. Значение $P_\delta(i) - P_\delta(i-1) < \delta$ определяет здесь точность вычислений.

5. Если ресурс не найден, то происходит размещение следующей ВМ, требование к уровню ресурсов предыдущей ВМ понижаются и она становится в общую упорядоченную очередь.

6. Если ресурс найден, то ВМ удаляется из очереди.

7. Выбор наилучшего по времени размещения при соблюдении всех ограничений.

Модельный эксперимент

Формальная модель кластера серверов с модулем распределения нагрузки, представленная на рисунке 3, разработана на основе характеристик платформы Huawei 2488, процессора Intel(R) Xeon(R) Gold 6154 в среде кроссплатформенного программного обеспечения AnyLogic 7 [19; 20]. При реализации модели, а также разработке системы распределения запросов использованы элементы библиотеки Enterprise Library.

На данном рисунке представлены следующие блоки: Source – блок генерации; Queue – блок формирования очереди; Delay – блок задержки; Match – блок согласования; Sink – блок удаления транзактов.

Уровень запрашиваемых ресурсов ВМ равномерно распределен в интервале 20–80 % уровня ресурсов сервера. Соотношение процессорного

времени и оперативной памяти распределено равномерно. На пропускную способность системы коммутации и каналов связи наложены ограничения до 1 Гб/с. Длительность миграции ВМ задавалась нормальным законом распределения.

Исследование характеристик системы осуществлено в три этапа: путем последовательного использования предлагаемых методов распределения запросов, запросов и ВМ, ВМ и приложений. Количественная оценка качества распределения ресурсов ЦОД проведена путем сравнения времени обработки запросов получаемых при использовании предложенных алгоритмов и алгоритма Round Robin.

Предлагаемые методы распределения нагрузки обеспечивают допустимые показатели времени обработки запросов при числе серверов кластера не более 15. С увеличением числа серверов время распределения растет по экспоненциальному закону. Из результатов моделирования, представленных в таблице, следует, что наибольшую производительность обеспечивают методы, последовательно реализующие алгоритмы рационального распределения запросов, ВМ и приложений. При этом время обработки запросов, по сравнению с применением алгоритма Round Robin, уменьшается в среднем на 15 %.

Таблица. Время обработки запросов кластером ЦОД

Число приложений	Число серверов	T, мс			
		T_0	T_1	T_2	T_3
50	5	4576	4112	3742	2976
100	7	5844	5231	4806	3217
150	10	6733	5820	4173	3580
200	12	6915	6113	5633	4512
300	15	7062	6549	5321	4893

Примечание. Здесь T_0 – время обработки запросов при реализации алгоритма Round Robin; T_1 – время обработки запросов при реализации метода распределения запросов; T_2 – время обработки запросов при совместной реализации метода распределения запросов и ВМ; T_3 – время обработки запросов при совместной реализации метода распределения запросов, ВМ и приложений.

Заключение

В работе рассмотрена задача построения эффективного механизма реализации запросов пользователей на реализацию услуг ЦОД за счет использования рациональных способов распределения его ресурсов. Представленные в данной статье методы и модели сервиса распределения ресурсов в кластерах с балансировкой нагрузки используют систему корректных отображений совокупности параметров запросов на известные характеристики физических ресурсов ЦОД, итерационный жадный алгоритм.

Сокращение времени вычислений достигается при этом путем введения ограничений на допустимую глубину перебора. Рассмотрены методы эффективного распределения запросов по серверам кластера, модель распределения программных приложений по множеству виртуальных машин кластера, метод распределения виртуальных машин по серверам кластера и соответствующая ему модель. Разработаны имитационные модели алгоритмов функционирования системы управления виртуализацией кластеров ЦОД (см. Свидетельства о регистрации программы для ЭВМ № 2019664647 от 11.11.2019 и № 2020617795 от 17.07.2020), проведено их экспериментальное исследование. Использование предлагаемых методов позволит обоснованно выполнять распределение программных приложений по ВМ корпоративного ЦОД, выбирать состав виртуальных машин и решать задачи рационального их размещения по физическим серверам ЦОД.

Литература

1. Mochalov V.P., Linets G.I., Bratchenko N.Yu., Govorova S.V. An analytical model of a corporate software-controlled network switch // Scalable Computing. 2020. № 21 (2). P. 337–346.
2. Боев В.Д. Компьютерное моделирование: пособие для практических занятий, курсового и дипломного проектирования в AnyLogic7. СПб.: ВАС, 2014. 432 с.
3. Taihoon K., Soksoo K. Analysis of Security Session Reusing in Distribution Server System // Computational Science and Its Applications. ICCSA 2006. Springer, 2006. P. 1045.
4. Хританков А. Модели и алгоритмы распределения нагрузки. Алгоритмы на основе сетей СМО // Информационные технологии и вычислительные сети. 2009. № 3. С. 33–48.
5. Иванисенко И., Кириченко Л., Радивилова Т. Методы балансировки с учетом мультифрактальных свойств нагрузки // Information Contentand Processing. 2015. Vol. 2, № 4. P. 345–368.
6. Панченко Т.В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю.Ю. Тарасевича. Астрахань: АГУ, 2007. 87 с.
7. Holland J.H. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. Cambridge: MIT Press, 1992. P. 211.

8. Michalewicz Z. Genetic algorithms + Data Structures=Evolution Programs // Springer-Verlag Berlin Heidelberg. 1992. P. 387.
9. Цой Ю.Р., Спицын В.Г. Генетический алгоритм // Представление знаний в информационных системах: учебное пособие. Томск: Изд-во ТПУ, 2006. 146 с.
10. Mitchell M. An Introduction to Genetic Algorithms Cambridge: MIT Press, 1999. 158 p.
11. Periaux J., Sefrioui M. Evolutionary computationalmethods for complex design in aerodynamics // AIAA-98-0222. Reno, 1998. P. 15.
12. Periaux J., Chen H.Q., Mantel B., Sefrioui M., Sui H.T. Combining game theory and genetic algorithms withapplication to DDM-nozzle optimization problems // Finite ElemAnal Des. 2001. Vol. 37 (5). P. 417–429.
13. Жирков А. Суперкомпьютеры: развитие, тенденции, применение. Обзор HPC-решений Eurotech // СТА. 2014. № 2. С. 16–20.
14. Taihoon K., Soksoo K. Analysis of security session reusing in distribution server system // Computational Science and Its Applications. ICCSA 2006. Springer, 2006. P. 1045.
15. Beloglazov A., Buyya R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers, Concurrency and Computation // Practice and Experience (CCPE). 2012. Vol. 24 (13). P. 1397–1420.
16. Mochalov V.P., Bratchenko N.Yu., Yakovlev S.V. Analytical model of object request broker based on Corba standard // Journal of Physics: Conference Series. 2018. Vol. 1015 (2). DOI: 10.1088/1742-6596/1015/2/022012.
17. Mochalov V.P., Bratchenko N.Yu., Yakovlev S.V. Analytical model of integration system for program components of distributed object applications // 2018 International Russian Automation Conference (RusAutoCon). 2018. № 8501806. DOI: 10.1109/RUSAUTOCON.2018.8501806.
18. Mochalov V., Bratchenko N., Linets G., Yakovlev S. Distributed management systems for infocommunication networks: A model based on tm forum frameworx // Computers. 2019. Vol. 8 (2). P. 45. DOI: 10.3390/computers8020045.
19. Mochalov V.P., Bratchenko N.Yu., Yakovlev S.V. Process-Oriented Management System for Infocommunication Networks and Services Based on TM Forum Frameworx // 2019 Proceedings International Russian Automation Conference (RusAutoCon). 2019. № 8867619. DOI: 10.1109/RUSAUTOCON.2019.8867619.
20. Vdovin P.M., Kostenko V.A. Algorithm for Resource Allocation in Data Centers with Independent Schedulers for Different Types of Resources // Computer and Systems Sciences International. 2014. Vol. 53, № 6. P. 854–866. DOI: 10.1134/S1064230714050141.

Получено 29.01.2021

Мочалов Валерий Петрович, д.т.н., профессор кафедры инфокоммуникаций (ИК) Северо-Кавказского федерального университета (СКФУ). 355028, Российская Федерация, г. Ставрополь, пр-т Кулакова, 2, корпус 9. Тел. +7 865 295-69-97. E-mail: mochalov.valery2015@yandex.ru

Линец Геннадий Иванович, д.т.н., заведующий кафедрой ИК СКФУ. 355028, Российская Федерация, г. Ставрополь, пр-т Кулакова, 2, корпус 9. Тел. +7 865 295-69-97. E-mail: kbytw@mail.ru

Братченко Наталья Юрьевна, к.ф.-м.н., доцент кафедры ИК СКФУ. 355028, Российская Федерация, г. Ставрополь, пр-т Кулакова, 2, корпус 9. Тел. +7 865 295-69-97. E-mail: n.b.20062@yandex.ru

Палканов Илья Сергеевич, аспирант кафедры ИК СКФУ. 355028, Российская Федерация, г. Ставрополь, пр-т Кулакова, 2, корпус 9. Тел. +7 865 295-69-97. E-mail: ilya0693@yandex.ru

METHODS AND MODELS OF THE RESOURCE ALLOCATION SERVICE IN CLUSTERS WITH LOAD BALANCING OF DATA CENTERS

Mochalov V.P., Linets G.I., Bratchenko N.Yu., Palkanov I.S.
North Caucasus Federal University, Stavropol, Russian Federation
E-mail: n.b.20062@yandex.com

Modern data processing centers are complex solutions for the management of enterprises and corporations, for the organization of data processing and storage systems, for the efficient distribution

of software applications between available resources. The object of the research is load-balancing clusters of data processing centers containing a certain set of application servers, file servers, data storage systems, an input-output system interconnected by a switching system and communication channels. The aim of the work is to increase the efficiency of the functioning of virtualized data centers by developing methodological approaches, rational methods and models for solving the problems of load distribution by its hardware and software. The problems of constructing interrelated mathematical methods and models necessary for building specialized software that provide a solution to the problems of creating rational plans for the allocation of data center resources at a given time interval are considered. The proposed methods and models are based on the system of correct mappings of the set of query parameters for the known characteristics of the physical resources of the data center. As an optimization criterion, the parameters of maximizing system performance for a certain time interval are used.

Keywords: *data processing centers, computing clusters, load balancing service, efficient resource allocation, display system*

DOI: 10.18469/ikt.2021.19.2.04

Mochalov Valeriy Petrovich, North-Caucasus Federal University, 2, build. 9, Kulakov Avenu, Stavropol, 355028, Russian Federation; Professor of Infocommunications Department, Doctor of Technical Science. Tel. +7 865 295-69-97. E-mail: mochalov.valery2015@yandex.ru

Linets Gennadiy Ivanovich, North-Caucasus Federal University, 2, build. 9, Kulakov Avenu, Stavropol, 355028, Russian Federation; Head of Infocommunication Department, Doctor of Technical Science. Tel. +7 865 295-69-97. E-mail: kbytw@mail.ru

Bratchenko Natalya Yurevna, North-Caucasus Federal University, 2, build. 9, Kulakov Avenu, Stavropol, 355028, Russian Federation; Associated Professor of Infocommunications Department, PhD in Physics and Mathematics. Tel. +7 865 295-69-97. E-mail: n.b.20062@yandex.ru

Polkanov Ilya Sergeevich, North-Caucasus Federal University, 2, build. 9, Kulakov Avenu, Stavropol, 355028, Russian Federation; PhD student of Infocommunications Department. Tel. +7 865 295-69-97. E-mail: ilya0693@yandex.ru

References

1. Mochalov V.P., Linets G.I., Bratchenko N.Yu., Govorova S.V. An analytical model of a corporate software-controlled network switch. *Scalable Computing*, 2020, no. 21 (2), pp. 337–346.
2. Boev V.D. *Computer Simulation: A Manual for Practical Studies, Coursework and Diploma Design in Anylogic7*. Saint Petersburg: VAS, 2014, 432 p. (In Russ.)
3. Taihoon K., Soksoo K. Analysis of Security Session Reusing in Distribution Server System. *Computational Science and Its Applications*. ICCSA 2006. Springer, 2006, p. 1045.
4. Hritankov A. Load balancing models and algorithms. Algorithms based on QS networks. *Informatsionnye tehnologii i vychislitel'nye seti*, 2009, no. 3, pp. 33–48. (In Russ.)
5. Ivanisenko I., Kirichenko L., Radivilova T. Balancing Methods Taking into Account Multifractal Load Properties. *Information Contentand Processing*, 2015, vol. 2, no. 4, pp. 345–368. (In Russ.)
6. Panchenko T.V. *Genetic Algorithms: Study Guide*. Ed. by Yu.Yu. Tarasevich. Astrahan': AGU, 2007, 87 p. (In Russ.)
7. Holland J.H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Cambridge: MIT Press, 1992, p. 211.
8. Michalewicz Z. *Genetic algorithms + Data Structures=Evolution Programs*. Springer-Verlag Berlin Heidelberg, 1992, p. 387.
9. Tsoj Ju.R., Spitsyn V.G. *Genetic Algorithm. Representation of Knowledge in Information Systems: A Tutorial*. Tomsk: Izd-vo TPU, 2006, 146 p. (In Russ.)

10. Mitchell M. *An Introduction to Genetic Algorithms*. Cambridge: MIT Press, 1999, 158 p.
11. Periaux J., Sefrioui M. Evolutionary computationalmethods for complex design in aerodynamics. AIAA-98-0222. Reno, 1998, p. 15.
12. Periaux J. et al. Combining game theory and genetic algorithms withapplication to DDM-nozzle optimization problems. *Finite ElemAnal Des*, 2001, vol. 37 (5), pp. 417–429.
13. Zhirkov A. Supercomputers: development, trends, applications. Eurotech HPC solutions overview. *CTA*, 2014, no. 2, pp. 16–20. (In Russ.)
14. Taihoon K., Soksoo K. Analysis of security session reusing in distribution server system. *Computational Science and Its Applications*, ICCSA 2006, Springer, 2006, p. 1045.
15. Beloglazov A., Buyya R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation. Practice and Experience* (CCPE), 2012, vol. 24 (13), pp. 1397–1420.
16. Mochalov V.P., Bratchenko N.Yu., Yakovlev S.V. Analytical model of object request broker based on Corba standard. *Journal of Physics: Conference Series*, 2018, vol. 1015 (2). DOI: 10.1088/1742-6596/1015/2/022012.
17. Mochalov V.P., Bratchenko N.Yu., Yakovlev S.V. Analytical model of integration system for program components of distributed object applications. *2018 International Russian Automation Conference (RusAutoCon)*, 2018, no. 8501806. DOI: 10.1109/RUSAUTOCON.2018.8501806.
18. Mochalov V., Bratchenko N., Linets G., Yakovlev S. Distributed management systems for infocommunication networks: A model based on tm forum frameworx. *Computers*, 2019, vol. 8 (2), pp. 45. DOI: 10.3390/computers8020045.
19. Mochalov V.P., Bratchenko N.Yu., Yakovlev S.V. Process-Oriented Management System for Infocommunication Networks and Services Based on TM Forum Frameworx. *2019 Proceedings International Russian Automation Conference (RusAutoCon)*, 2019, no. 8867619. DOI: 10.1109/RUSAUTOCON.2019.8867619.
20. Vdovin P.M., Kostenko V.A. Algorithm for Resource Allocation in Data Centers with Independent Schedulers for Different Types of Resources. *Computer and Systems Sciences International*, 2014, vol. 53, no. 6, pp. 854–866. DOI: 10.1134/S1064230714050141.

Received 29.01.2021

УДК 681.7.068

ИССЛЕДОВАНИЕ ПОТЕНЦИАЛЬНЫХ ВОЗМОЖНОСТЕЙ ОЦЕНИВАНИЯ КОЭФФИЦИЕНТА ПЕРЕДАЧИ ОСНОВНОЙ МОДЫ НА ОСНОВЕ АНАЛИЗА ПЕРЕКРЫТИЯ РАДИАЛЬНОГО РАСПРЕДЕЛЕНИЯ ПОЛЕЙ В ДИСКРЕТНОМ ПРЕДСТАВЛЕНИИ

Пашин С.С.
Поволжский государственный университет телекоммуникаций и информатики, Самара, РФ
E-mail: pashinstanislav@outlook.com

Представлено исследование потенциальных возможностей оценивания коэффициента передачи основной моды на основе анализа перекрытия радиального распределения полей в дискретном представлении, рассмотрен процесс прохождения основной (фундаментальной) моды LP01, характеризуемой наиболее простым распределением поля, через соединение пары одномодовых волоконных световодов, выполненного с некоторым радиальным рассогласованием при разбросе значений радиуса пятна моды взаимодействующих мод. Осуществлен ввод дополнительной поправки при выборе верхней границы области дискретного представления поля моды, которая обеспечивает устранение нежелательного выброса погрешности оценки в области верхней границы диапазона исследуемых значений осевого смещения. Приведены результаты экспериментальной верификации предложенного подхода, которые подтверждают возможность проведения оценки коэффициента передачи основной моды LP01 при прохождении соединения одномодовых оптических волокон.