женных уязвимостей в случае возможности устранения уязвимости несколькими различными способами.

Библиографический список

1. **Kwiatkowska, M.,** Norman, G., Parker, D. (2010). Advances and Challenges of Probabilistic Model Checking. Proceedings of 48th Annual Allerton Conference on Communication, Control and Computing, Urbana–Champaign, 1691–1698.

2. **Sheyner, O.,** Wing, J. M., Jha, S., Lippmann, R., Haines, J. (2002). Automated Generation and Analysis of Attack Graphs. Proceedings of the 2002 IEEE Symposium on Security and Privacy, Berkley, 273–284.

3. **Сложность** проверки модели параллельных программных систем / С. С. Захарченко // Программные продукты и системы. – 2012. – № 98. – Вып. 2. – С. 39–42.

4. **Manna, Z.,** Pnueli, A. (1981). Verification of Concurrent Programs. Part I: The Temporal Framework. Stanford, 62 p.

5. **Krautsevich, L.,** Martinelli, F., Yautsiukhin, A. (2010). Formal approach to security metrics. What does «more secure» mean for you? Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, Copenhagen, 162–169.

6. **Idika, N.,** Bhargava, B. (2012). Extending Attack Graph-Based Security Metrics and Aggregating Their Application. IEEE Transactions on dependable and secure computing, 9 (1), 75–85.

7. **Dijkstra, E. W.** (2002). Cooperating Sequential Processes. The Origin of Concurrent Programming, New York, 65–138.

УДК 004.75

Н. А. Игнатов
Московский государственный университет путей сообщения

ПОСТРОЕНИЕ ЗАДАЧИ ОПТИМИЗАЦИИ ПРЕДОСТАВЛЕНИЯ ВИРТУАЛЬНЫХ РЕСУРСОВ В ЦЕНТРАХ ОБРАБОТКИ ДАННЫХ, ОСНОВАННЫХ НА ОБЛАЧНЫХ ТЕХНОЛОГИЯХ

Рассматривается проблема расчета оптимального объема необходимых для обеспечения работы приложений виртуальных ресурсов, запускаемых в центре обработки данных, построенном с применением облачных технологий. Под оптимальным объемом виртуальных ресурсов понимается сочетание таких характеристик, как мощность процессора, объем оперативной памяти и устройства хранения данных, пропускная способность сети. Они предоставляются конечным пользователям посредством запуска определенного количества экземпляров виртуальных машин за минимальную стоимость в условиях соблюдения требований к качеству обслуживания системы. В ходе исследования построена и решена с использованием симплекс-метода задача оптимизации процесса предоставления виртуальных ресурсов, относящаяся к классу задач линейного программирования.

методы оптимизации, виртуализация, центр обработки данных, виртуальная машина, качество обслуживания.

Введение

Сегодня в транспортной отрасли активно применяются информационные системы, обеспечивающие деятельность различных направлений: административно-организационного, научно-производственного, финансово-экономического, информационно-

аналитического, направления безопасности. Работа современных информационных систем заключается в обработке больших массивов данных, осуществлении вычислительных операций различной сложности, передаче данных и подразумевает использование центров обработки данных (далее – ЦОД) различного масштаба.

Построение ЦОД на базе облачных вычислений – это решение, позволяющее существенно сократить затраты на процесс автоматизации организации. При данном подходе отдельный экземпляр приложения компонуется m экземплярами виртуальных машин (далее – ВМ) $\{v_1, \ldots, v_m\}$, где значение m – любое фиксируемое или изменяемое во времени, основанное на текущей рабочей загрузке и запрашиваемой производительности. Вычисление объемов ресурсов, необходимых для приложения, состоит из набора независимых задач, которые могут быть смоделированы как запросы на обслуживание, отправляемые конечными пользователями на экземпляры виртуальных приложений.

При постановке задачи предоставления виртуальных ресурсов функцию от набора параметров принимают в качестве меры «нежелательных» свойств или стоимости, которую нужно минимизировать. При этом минимизация стоимости должна быть выполнена с условием соблюдения требований к качеству обслуживания (далее – QoS) информационной системы. Основными параметрами QoS системы являются: максимальное допустимое значение времени отклика на запрос конечного пользователя и количество запросов на предоставление виртуальных ресурсов, получивших отказ в обслуживании. Выполнение данных параметров важно, так как они имеют определяющее воздействие на мнение пользователя о приложении. Если время реагирования будет очень большим или запросы будут отклоняться, то приложение не будет использоваться, что приведет к снижению его рентабельности.

При этом необходимо рассмотреть построение задачи оптимизации предоставления виртуальных ресурсов. Задача оптимизации сформулирована как задача оптимизации целевой функции при условии выполнения ряда заданных ограничений.

1 Построение задачи оптимизации

Будем рассматривать задачу оптимизации в конечномерном пространстве. Это означает, что мера качества, или целевая функция, которая должна быть минимизирована, является функцией конечного числа переменных.

Рассмотрим задачу составления наиболее экономного, то есть наиболее дешевого, плана предоставления ресурсов, удовлетворяющего определенным установленным требованиям QoS. Подобная задача возникает в связи с необходимостью организовать предоставление виртуальных ресурсов набору пользователей, объединенных в группы по определенным характеристикам.

Предполагается, что известен перечень разновидностей ВМ из n штук, которые будем обозначать буквами $F_1, F_2, \ldots, F_n$. Кроме того, рассматриваются также такие характеристики ВМ, как мощность процессора, объем оперативной памяти, объем устройства хранения данных и др. Обозначим эти характеристики буквами $N_1, N_2, \ldots, N_m$. Предположим, что для любого типа ВМ F_i, где $i = 1, \ldots, n$, известны его технические характеристики, т. е. количественное содержание в одном экземпляре данного типа ВМ указанных компонент. В таком случае можно составить таблицу, содержащую характеристики типов ВМ:

$$\begin{array}{c} \\ N_1 \\ \vdots \\ N_m \end{array} \begin{array}{c} \begin{array}{ccc} F_1 & \ldots & F_n \end{array} \\ \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix} \end{array}. \qquad (1)$$

Элементы этой таблицы образуют матрицу размера $m \times n$, которую назовем матрицей виртуальных ресурсов.

Допустим, что мы составили распределение мощностей $x^T = (x_1, x_2, \ldots, x_n)$ на некоторый промежуток времени, для которого известно точное количество пользователей

(далее – сессия). Другими словами, мы планируем предоставить каждой группе пользователей на сессию:

x_1 экземпляров ВМ типа F_1,

x_2 экземпляров ВМ типа F_2,

……

x_n экземпляров ВМ типа F_n.

Нетрудно вычислить, какие объемы оперативной памяти и устройства хранения данных, а также значение производительности процессора получит каждая группа пользователей на сессию. Именно характеристика N_1 присутствует в этом распределении мощностей в общем количестве:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n,$$

поскольку в x_1 экземплярах ВМ типа F_1, согласно матрице виртуальных ресурсов, содержится $a_{11}x_1$ единиц характеристик N_1. К этому количеству добавляется объем $a_{12}x_2$ характеристики N_1 из x_2 экземпляров ВМ типа F_2 и т. д. Аналогично можно определить и количество всех остальных характеристик N_i в составленном распределении мощностей $x^T = (x_1, x_2, \ldots, x_n)$.

Допустим далее, что имеются требования, касающиеся предоставления определенного количества каждой из характеристики N_i, где $i = 1, \ldots, m$, каждой группе пользователей в планируемый срок. Например, суммарный объем выделяемой оперативной памяти не должен быть меньше заданного. Выразим эти требования вектором $b^T = (b_1, b_2, \ldots, b_m)$, i компонента которого указывает минимально необходимое содержание характеристики N_i в распределении мощностей. Это означает, что координаты x_i вектора x^T должны удовлетворять следующей системе ограничений:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\geq b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\geq b_2, \\ &\ldots\ldots \end{aligned} \qquad (2)$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \geq b_m.$$

Из смысла задачи очевидно, что все переменные $x_1, x_2, \ldots, x_n$ неотрицательны. На основании этого к ограничениям (2) добавляются неравенства:

$$x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0. \qquad (3)$$

Понятно, что любое распределение мощностей $x^T = (x_1, x_2, \ldots, x_n)$ должно удовлетворять условиям (2) и (3). Но таких распределений мощностей может быть бесконечно много, поэтому для выбора одного из них вводят критерий стоимости – цену набора.

Пусть стоимость ВМ типа $F_1, F_2, \ldots, F_n$ равна соответственно $c_1, c_2, \ldots, c_n$. Следовательно, стоимость всего распределения мощностей $x^T = (x_1, x_2, \ldots, x_n)$ может быть записана в виде ценовой функции:

$$c_1x_1 + c_2x_2 + \ldots + c_nx_n. \qquad (4)$$

Окончательная формулировка задачи о распределении виртуальных ресурсов: среди всех векторов $x^T = (x_1, x_2, \ldots, x_n)$, удовлетворяющих ограничениям (2) и (3), необходимо выбрать такой вектор, для которого выражение (4) принимает минимальное значение. На этом процесс формализации задачи о составлении распределения мощностей закончен. Задачу поиска наиболее дешевого распределения виртуальных ресурсов можно кратко выразить в следующей матрично-векторной форме:

$$\begin{aligned} c^T x &\to min, \\ Ax &\geq b, \\ x &\geq 0, \end{aligned} \qquad (5)$$

где A – матрица размерности $m \times n$; b, c, x – векторы-столбцы соответствующей размерности.

Мы выразили реальную задачу в строгой математической форме, что в дальнейшем позволит применить весь богатый арсенал математических методов для ее решения.

Рассмотренная задача относится к классу задач линейного программирования.

2 Вычисление экономного плана предоставления ресурсов

Используя описанный выше алгоритм, вычислим комбинацию ВМ с наименьшей стоимостью, удовлетворяющую системным требованиям приложения Δ, выполняемого в ЦОД, основанном на облачных технологиях. В рассматриваемой системе используется пять типов виртуальных машин (табл.), используемых для обеспечения работы конечного множества экземпляров приложения Δ, требующих одинаковых объемов ресурсов.

Здесь vCPU (virtual CPU) – виртуальный процессор, производительность которого измеряется в ECU (EC2 Compute Unit). Согласно информации [6], один ECU соответствует 1–1,2 ГГц процессора AMD Opteron или Intel Xeon 2007 года производства.

Так, про экземпляр ВМ типа m1.large можно сказать, что он содержит 7,5 Гб оперативной памяти, два процессора по два ECU каждый и устройство хранения данных в 820 Гб, при этом его базовая стоимость составляет 0,260 долларов в час. Данные типы ВМ были выбраны из набора экземпляров ВМ, разработанных компанией Amazon EC2 [7], являющейся одним из лидеров на рынке виртуализации. Обозначим типы ВМ m1.small, m1.medium, m1.large, t1.micro, c1.medium как F_1, F_2, F_3, F_4, F_5 соответственно, а технические характеристики ВМ через N_1, N_2, N_3, N_4, N_5, где

N_1 – количество виртуальных процессоров: {1, 1, 2, 2, 1},

N_2 – мера производительности процессора: {1, 2, 4, 5, 1},

N_3 – объем оперативной памяти в Гб: {1,7, 3,75, 7,5, 1,7, 0,615},

N_4 – устройство хранения данных в Гб: {160, 410, 820, 350, 0},

N_5 – мера производительности сети, где очень низкая производительность – 1, низкая – 2, средняя – 3: {2,3,3,3,1}.

На основании (1) и введенных обозначений построим матрицу виртуальных ресурсов ЦОД:

	F_1	F_2	F_3	F_4	F_5
N_1	1	1	2	2	1
N_2	1	2	4	5	1
N_3	1,7	3,75	7,5	1,7	0,615
N_4	160	410	820	350	0
N_5	2	3	3	3	1

Составим распределение мощностей $x^T = (x_1, x_2, \ldots, x_5)$ на сессию. Другими словами, мы планируем каждому пользователю на сессию:

x_1 экземпляров ВМ типа F_1,

x_2 экземпляров ВМ типа F_2,

……

x_5 экземпляров ВМ типа F_5.

ТАБЛИЦА. Типы виртуальных машин

№	Стоимость в час, $	Название	vCPU	ECU	ОЗУ, Гб	Устройство хранения, Гб	Производительность сети (оценка)
1.	0,065	m1.small	1	1	1,7	160	Низкая (2)
2.	0,130	m1.medium	1	2	3,75	410	Средняя (3)
3.	0,260	m1.large	2	4	7,5	820	Средняя (3)
4.	0,165	t1.micro	2	5	1,7	350	Средняя (3)
5.	0,020	c1.medium	1	1	0,615	0	Очень низка (1)

Пусть нам известно, что для работы одного экземпляра приложения Δ требуются следующие минимальные значения технических характеристик: 0,35 единиц виртуальных процессоров производительностью в один ECU, 0,256 Гб оперативной памяти, устройство хранения данных объемом 2Гб и низкая производительность сети. Тогда вектор ограничений имеет вид $b^T = (0.35,1,0.256\ ,2\ ,2)$ и мы получаем:

$$\begin{gathered} 1x_1 + 1x_2 + 2x_3 + 2x_4 + 1x_5 \geq 0,35, \\ 1x_1 + 2x_2 + 4x_3 + 5x_4 + 1x_5 \geq 1, \\ 1,7x_1 + 3,75x_2 + 7,5x_3 + \\ +1,7x_4 + 0,615x_5 \geq 0,256, \\ 160x_1 + 410x_2 + 820x_3 + 350x_4 + 0x_5 \geq 2, \\ 2x_1 + 3x_2 + 3x_3 + 3x_4 + 1x_5 \geq 2. \end{gathered} \quad (6)$$

Из смысла задачи очевидно, что все переменные $x_1, x_2, \ldots, x_n$ неотрицательны. На основании этого к ограничениям (5) добавляются неравенства:

$$x_1 \geq 0, x_2 \geq 0, \ldots, x_5 \geq 0. \quad (7)$$

Из табл. можно вывести вектор стоимости ВМ типа $F_1, F_2, \ldots, F_5$, который записывается следующим образом:

$$c^T = (0,065,\ 0,130,\ 0,260,\ 0,165,\ 0,020).$$

Следовательно, стоимость всего распределения мощностей $x^T = (x_1, x_2, \ldots, x_5)$ может быть записана в виде ценовой функции:

$$\begin{gathered} 0,065x_1 + 0,130x_2 + 0,260x_3 + \\ +0,165x_4 + 0,020x_5 \to \min. \end{gathered}$$

Задачу поиска наиболее дешевого распределения виртуальных ресурсов можно выразить в следующей матрично-векторной форме:

$$\begin{gathered} 0,065x_1 + 0,130x_2 + 0,260x_3 + \\ + 0,165x_4 + 0,020x_5 \to \min, \end{gathered}$$

$$1x_1 + 1x_2 + 2x_3 + 2x_4 + 1x_5 \geq 0,35,$$

$$1x_1 + 2x_2 + 4x_3 + 5x_4 + 1x_5 \geq 1,$$

$$1,7x_1 + 3,75x_2 + 7,5x_3 + 1,7x_4 + 0,615x_5 \geq 0,256,$$

$$160x_1 + 410x_2 + 820x_3 + 350x_4 + 0x_5 \geq 2,$$

$$2x_1 + 3x_2 + 3x_3 + 3x_4 + 1x_5 \geq 2,$$

$$x_1 \geq 0, x_2 \geq 0, \ldots, x_5 \geq 0.$$

Построенная задача относится к классу задач линейного программирования**,** а поиск ее решения осуществлялся с помощью симплекс-метода, который был разработан американским ученым Дж. Данцингом.

При решении данной задачи для получения оптимального плана в основном симплекс-методе было проведено 3 итерации. Время вычисления оптимального плана с применением программного пакета MathCad на персональном компьютере с системными параметрами: процессор – IntelCore 2 Duo, 2,00 ГГц, оперативная память – 2 Гб – составило 6,5 секунд.

Оптимальный план предоставления виртуальных ресурсов можно записать так:

$$\begin{gathered} x_1 = 0,0125,\ x_2 = 0,\ x_3 = 0,\ x_4 = 0,\ x_5 = 1,98, \\ F(x) = 0,065 \cdot 0,0125 + \\ + 0,020 \cdot 1,98 = 0,0403. \end{gathered} \quad (8)$$

Полученное решение может говорить о том, что для обеспечения наиболее дешевого предоставления ресурсов, необходимых для работы одного экземпляра приложения Δ, потребуется 0,0125 экземпляров ВМ типа 1 и 1,98 экземпляров ВМ типа 5, а минимальная стоимость такого предоставления составит 0,0403 долларов в час.

Зная данные параметры, нетрудно вычислить, сколько потребуется экземпляров ВМ типа 1 и 5 для предоставления ресурсов на запросы любого конечного числа пользователей на сессию с конечным временем.

Наряду с полученным результатом известно, что основные существующие средства виртуализации (например, Windows Azure [8]) используют принцип предоставления

каждому экземпляру приложения отдельной виртуальной машины и только после анализа работы выполняемых экземпляров приложений платформа выделяет дополнительные или сокращает неиспользуемые экземпляры ВМ. Данный подход приводит к повышенной стоимости использования облачных сервисов.

С учетом вышесказанного прикладной эффект полученного решения заключается в снижении стоимости использования облачных сервисов за счет минимизации количества используемых ВМ.

Так, при «классическом» подходе для обеспечения работы 1000 экземпляров виртуального приложения Δ с заданными характеристиками потребуется 1000 экземпляров ВМ типа m1.small стоимостью 0,065 долларов в час (табл.), при этом стоимость услуги составит 65 долларов в час. Наряду с этим при использовании предложенного алгоритма, с учетом результата (8), стоимость услуги составит 40,3 доллара в час, что на 38 % сократит стоимость работы 1000 экземпляров приложения Δ.

Заключение

В данной статье рассмотрен алгоритм построения, а так же решения задачи оптимизации процесса предоставления ресурсов в облачных системах. Задача сформулирована как необходимость оптимизации целевой функции при условии выполнения ряда заданных ограничений.

Рассмотрена схема расчета оптимального плана предоставления ресурсов, заключающегося в поиске наиболее дешевого сценария предоставления виртуальных ресурсов, с учетом требований по качеству обслуживания, предъявляемых к системе.

В результате проведения научно-исследовательской работы была построена и решена задача оптимизации процесса предоставления виртуальных ресурсов. Так, зная минимальные системные требования одного экземпляра приложения, а также соотношения технических и ценовых характеристик виртуальных машин, можно вычислить оптимальный план предоставления виртуальных ресурсов для любого конечного множества пользователей на сессию с конечным временем.

Наряду с проведенной работой интерес для дальнейшего исследования представляет внедрение методов оптимизации в инструменты балансировки загрузки систем, основанных на облачных технологиях.

Библиографический список

1. **Методы** оптимизации. Кн. 1 / Ф. П. Васильев. – Москва : МЦНМО, 2011. – 620 с.

2. **Методы** оптимизации. Кн. 2 / Ф. П. Васильев. – Москва : МЦНМО, 2011. – 434 с.

3. **Теоретические** основы проектирования компьютерных сетей / В. М. Вишневский. – Москва : Техносфера, 2003. – 512 с.

4. **Ahmed, T.,** Singh, Y. (2012). Analytic Study of Load Balancing Techniques Using Tool Cloud Analyst. *International Journal of Engineering Research and Applications*, 1027–1030.

5. **Calheiros, R. N.,** Ranjan, R., Buyya, R. (2011). Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments. *Proceedings of the 2011 International Conference on Parallel Processing*, Taipei, 295–304.

6. **Хостинг** Amazon EC2 [Электронный ресурс]. – Режим доступа: http://www.kushnerov.com.

7. **Amazon** EC2 instances [Электронный ресурс]. – Режим доступа: http://aws.amazon.com.

8. **Windows** Azure Platform [Электронный ресурс]. – Режим доступа: http://download.microsoft.com/documents/rus/cloud/Windows_Azure_A4_2012.pdf.