DOI: 10.22184/1992-4178.2018.175.4.108.112

Управление предоставлением облачных вычислительных ресурсов в виртуальных дата-центрах

С. Назаров, д. т. н.[1], А. Барсуков, к. т. н.[2]

УДК 004.75 | ВАК 05.13.00

Рынок облачных технологий уверенно растет, отмечают эксперты, ежедневно появляются новые пользователи, особенно среди предприятий малого и среднего бизнеса. Ведущие коммерческие компании и государственные службы приходят к пониманию удобства применения облачных вычислений. Современные высокотехнологичные вычислительные мощности позволяют эффективно хранить, анализировать и обрабатывать данные.

Ведущие ИТ-компании и инженеры давно обсуждают плюсы и минусы использования облачных технологий. Эксперты прогнозируют проблемы и возможные потери конфиденциальных данных в «облаках» из-за широкого спроса и притока пользователей. К основным достоинствам облачных технологий относятся высокий уровень безопасности и конфиденциальности хранения данных, степень надежности и неограниченные вычислительные ресурсы. «Облако» удобно в использовании, поскольку требуется лишь установка простых приложений на любые пользовательские терминалы, и позволяет сэкономить на покупке лицензионных продуктов, ресурсов и программ. В результате применения облачных технологий уменьшаются расходы на приобретение дорогостоящих мощных компьютеров, серверов, сокращаются затраты на оплату труда ИТ-специалистов, обслуживающих локальный дата-центр.

Рассмотрим создание системы реального времени (СРВ), сочетающей в себе веб-серверы, вычислительная мощность которых формируется на основе арендованной инфраструктуры IaaS (Infrastructure-as-a-Service – инфраструктура как сервис). Арендатор создает собственный парк виртуальных серверов (виртуальный дата-центр) и разворачивает коммерческую систему для обслуживания пользователей, получая прибыль. В предложенной модели с точки зрения теории стратегических игр два партнера: СРВ (виртуальный дата-центр – арендуемая часть IaaS) и природа (сеть клиентов), как принято в теории игр именовать непредсказуемого партнера. Модель и система мониторинга СРВ позволяют управлять предоставлением облачной вычислительной мощности и сбалансировать расходы на аренду и прибыль, получаемую в результате обслуживания пользователей.

Остановимся подробнее на одном из вопросов организации функционирования широкого класса компьютерных систем, используемых в электронном бизнесе. Имеются в виду электронная коммерция, онлайновый маркетинг, оформление заявок, осуществление платежей, информационная поддержка доставки товаров, электронный документооборот, информационно-справочные системы, финансовые системы, системы взаимодействия с клиентами и т. п. [1–7]. Подобные системы используют интернет-технологии для передачи данных и предоставления веб-сервисов на основе специализированных веб-сайтов. Последние представляют собой системы реального времени на основе веб-серверов.

Современные облачные технологии позволяют получать требуемые для этого ресурсы в форме инфраструктуры IaaS, на базе которой предоставляются услуги аренды вычислительных ресурсов и систем хранения, таких как виртуальные серверы с заданной вычислительной мощностью и каналы связи нужной пропускной способности для доступа к хранилищам данных и внешним ресурсам. При этом клиент может использовать любые операционные системы и приложения [8–11].

Рассмотрим вопросы организации работы систем реального времени и требования к вычислительным ресурсам (количеству процессоров), обеспечивающим работу систем реального времени. Как правило, СРВ предназначены для своевременного и предсказуемого реагирования на запросы, поступающие в систему. Для таких систем характерен определенный набор запросов

[1] ЗАО «МНИТИ», главный специалист, действительный член Международной академии информатизации.

[2] ЗАО «МНИТИ», заместитель генерального директора, профессор Академии военных наук, действительный член Международной академии безопасности.

некоторого типа $Z=\{z_i, i=1, 2, \ldots, M\}$, для каждого из них в системе предусмотрена заранее разработанная программа P_i, хранящаяся в памяти системы. Основное требование к СРВ заключается в своевременности обработки запросов. Реакция на запрос z_i должна уложиться в заранее заданный интервал времени R_i.

Все системы реального времени принято подразделять на жесткие системы реального времени R_i, в которых недопустимо превышение заданного значения для реализации запроса z_i, и мягкие (гибкие системы реального времени). В последних допускается «опоздание» при обработке запроса, но повышается «стоимость» опоздания, которую обозначим c_i.

Внешние запросы (события), на которые СРВ должна реагировать, можно разделить на периодические (возникающие через регулярные интервалы времени) и непериодические (непредсказуемые). Если в систему поступает M потоков периодических запросов и запрос z_i поступает с периодом T_i, а на его обработку затрачивается t_i времени системы, то все потоки могут быть своевременно обработаны в однопроцессорной системе только при выполнении условия:

$$\sum_{i=1}^{M}\frac{t_i}{T_i}\le 1. \qquad (1)$$

Системы реального времени, удовлетворяющие этому условию, считают поддающимися планированию или планируемыми.

Классический пример статического алгоритма планирования реального времени для прерываемых периодических процессов – алгоритм DMS (Date Monotonic Scheduling) [12]. Основанный на статических приоритетах алгоритм подходит для планирования независимых периодических процессов с заданным директивным временем выполнения. Как было показано Лю (Liu) и Лейлэнд (Layland) в [13], использование статических приоритетов целесообразно только при не слишком высокой загруженности центрального процессора. Алгоритм DMS гарантированно работает в любой системе периодических процессов при условии:

$$U=\sum_{i=1}^{M}\frac{t_i}{T_i}\le M\left(2^{\frac{1}{M}}-1\right). \qquad (2)$$

Например, U ≤ 0,8284 для двух процессов. Когда количество процессов M стремится к бесконечности, это выражение имеет вид:

$$\lim_{M\to\infty} M\left(\sqrt[M]{2}-1\right)=\ln 2\approx 0{,}69\ldots$$

Гиперболическая граница (2) – более жесткое условие планирования, чем то, которое было представлено Лю и Лейлэнд, то есть имеем:

$$\prod_{i=1}^{M}(U_i+1)\le 2, \qquad (3)$$

где U_i – использование ЦП для каждой задачи. Приблизительная оценка заключается в том, что DMS может удовлетворить все предельные сроки, если загрузка процессора составляет менее 69,32%. Другие 30,7% ЦП могут быть выделены для задач с меньшим приоритетом (не для реального времени). Известно, что случайно созданная система периодических задач будет соответствовать всем предельным срокам, когда использование ЦП составляет 85% или меньше, однако это зависит от знания точной статистики задачи (периодов, крайних сроков), которая не может быть гарантирована для всех наборов задач.

Таким образом, алгоритм DMS надежен при относительно невысокой загрузке процессора. К тому же статическое планирование, используемое алгоритмом, не всегда возможно. Другой недостаток алгоритма DMS – неэффективность для планирования непериодических процессов и непостоянных временных интервалов использования центрального процессора. А именно эти условия характерны для систем мягкого реального времени.

Наиболее подходящий алгоритм планирования для таких систем – EDF (Earliest Deadline – First) – процесс с ближайшим сроком завершения первым. Это динамический алгоритм планирования, не требующий периодичности процессов и постоянства временных интервалов использования центрального процессора. Каждый раз при поступлении в систему процесс объявляет о своем присутствии и о сроке выполнения задания. Планировщик хранит список процессов, отсортированный по срокам выполнения. Алгоритм запускает процесс с ближайшим по времени сроком. В случае перехода нового процесса в состояние готовности система сравнивает его срок выполнения со сроком текущего процесса. Если у нового процесса график более жесткий, он прерывает работу текущего процесса. Алгоритм EDF работает с любым набором процессов, для которого возможно планирование. При этом коэффициент загрузки процессора может достигать 100%.

ПОСТАНОВКА ЗАДАЧИ

Алгоритм EDF можно считать достаточно неплохим для планирования работы систем мягкого реального времени. В целом с учетом непредвиденного характера нагрузки систем рассматриваемого класса следует ввести некоторую метрику (показатель) функционирования системы. Наиболее целесообразной метрикой считается «штраф» за опоздание в обслуживании запросов, поступающих в систему. Размер штрафа C_i, получаемого системой за опоздание с обработкой процесса z_i, можно сформировать следующим образом:

- $C_i^- = K_1(D_i, t_i)$, если имеется дефицит производительности процессоров СРВ, в результате которого превышено время обработки запроса t_i по сравнению с заданным директивным значением D_i (в данном случае $t_i > D_i$);
- $C_i^+ = K_2(D_i, t_i)$, если имеется избыточная производительность процессора СРВ, в результате чего запрос z_i обрабатывается быстрее директивного значения (в данном случае $t_i < D_i$);
- $C_i = 0$, если $t_i = D_i$.

Коэффициент K_1 можно трактовать как удельные финансовые издержки, связанные с потерями системой клиентов, которые отказались от ее услуг из-за неудовлетворительного обслуживания. Коэффициент K_2 можно трактовать как удельные финансовые издержки, обусловленные эксплуатацией СРВ избыточной производительности.

Будем считать, что СРВ на платформе IaaS строится как многопроцессорная (и возможно, многоядерная) система с возможностью динамического выделения некоторого числа процессоров (от 1 до n) для обслуживания входящих запросов. Выбор количества работающих в конкретный момент времени процессоров зависит от интенсивности поступления запросов в систему, которая в общем случае слабо предсказуемая либо непредсказуемая. В условиях полной неопределенности можно попробовать решить задачу на основе теории стратегических игр.

РЕШЕНИЕ ЗАДАЧИ

С точки зрения теории стратегических игр в данной игре два партнера: СРВ (арендуемая часть IaaS) и природа (сеть клиентов), как принято в теории игр именовать полностью непредсказуемого партнера [14]. Стратегии СРВ обозначим $R_1, R_2, \ldots, R_n$, а стратегии природы – $P_1, P_2, \ldots, P_k$. Предположительно потребность в производительности в некоторые периоды функционирования (например, рабочий день, вечер, праздничные дни и т. п.) составляет $P_1, P_2, \ldots, P_k$, а СРВ может состоять из $1, 2, \ldots, n$ процессоров, обеспечивающих соответственно значения реальной производительности $R_1, R_2, \ldots, R_n$.

Схематично матрицу выигрышей можно записать в следующем виде:

	P_1	P_2	…	P_k
R_1	C_{11}	C_{12}	…	C_{1k}
R_2	C_{21}	C_{22}	…	C_{2k}
…	…	…	…	…
R_n	C_{n1}	C_{n2}	…	C_{nk}.

(4)

Здесь C_{ij} – штраф, получаемый системой при имеющейся производительности системы реального времени R_i и требуемой производительности P_j. Предположим, что с использованием тех или иных методов матрица (4) получена. В соответствии с теоремой стратегических игр для нашего случая, когда значения из множеств $P = \{P_1, P_2, \ldots, P_k\}$ и $R = \{R_1, R_2, \ldots, R_n\}$ могут принимать конечное число, оптимальное решение заключается в поиске смешанных стратегий.

Из теории стратегических игр следует, что при использовании смешанных стратегий есть, по крайней мере, одно оптимальное решение с ценой игры V, которое находится между верхним и нижним значениями [14–16]. Следует заметить, что всегда $V > 0$.

Допустим, оптимальная смешанная стратегия СРВ складывается из стратегий $R_1, R_2, \ldots, R_n$ с вероятностями, равными $p_1, p_2, \ldots, p_n$ ($p_1 + p_2 + \ldots + p_n = 1$), а оптимальная стратегия клиентской сети – из стратегий $P_1, P_2, \ldots, P_k$, которые применяются с вероятностями, равными $q_1, q_2, \ldots, q_n$ ($q_1 + q_2 + \ldots + q_n = 1$). Если СРВ применяет оптимальную стратегию, а клиентская сеть чистую стратегию P_j ($j = 1, 2, \ldots, k$), то средний штраф, получаемый системой, составит: $C_j = p_1 \cdot c_{1j} + p_2 \cdot c_{2j} + \ldots + p_n \cdot c_{nj}$ ($j = 1, 2, \ldots, k$).

Особенность оптимальной стратегии СРВ состоит в том, чтобы при произвольном поведении противника (клиентской сети) она обеспечивала штраф не больший, чем цена игр V. Отсюда имеем систему ограничений:

$$\left.\begin{aligned} &p_1 \cdot c_{11} + p_2 \cdot c_{21} + \ldots + p_n \cdot c_{n1} \le V, \\ &p_1 \cdot c_{12} + p_2 \cdot c_{22} + \ldots + p_n \cdot c_{n2} \le V, \\ &\ldots \\ &p_1 \cdot c_{12} + p_2 \cdot c_{22} + \ldots + p_n \cdot c_{n2} \le V. \end{aligned}\right\} \quad (5)$$

Систему (5) можно преобразовать, разделив по частям на V:

$$\left.\begin{aligned} &c_{11} \cdot x_1 + c_{21} \cdot x_2 + \ldots + c_{n1} \cdot x_n \le 1, \\ &c_{12} \cdot x_1 + c_{22} \cdot x_2 + \ldots + c_{n2} \cdot x_n \le 1, \\ &\ldots \\ &c_{1k} \cdot x_1 + c_{2k} \cdot x_2 + \ldots + c_{nk} \cdot x_n \le 1. \end{aligned}\right\} \quad (6)$$

Здесь, $x_1 = P_1 / V$, $x_2 = P_2 / V$, …, $x_n = P_n / V$. Из условия $p_1 + p_2 + \ldots + p_n = 1$ следует, что

$$x_2 + \ldots + x_n = 1 / V. \quad (7)$$

Значения величин $p_1, p_2, \ldots p_n$ должны быть подобраны таким образом, чтобы гарантированное значение штрафа СРВ было по возможности минимальным, то есть чтобы достигалось

$$V = \min \text{ или } 1 / V = \max.$$

Таким образом, задача сводится к нахождению таких значений $x_1, x_1, \ldots, x_n$, чтобы

$$x_1 + x_2 + \ldots + x_n = \max. \quad (8)$$

Кроме того, должны выполняться дополнительные граничные условия, а именно, $p_i \geq 0$ ($i = 1, 2, \ldots, n$), следовательно, имеем:

$$x_i = P_i / V \geq 0 \text{ для } i = 1, 2, \ldots, n. \quad (9)$$

Из этого следует, что нахождение оптимальной смешанной стратегии СРВ сводится к решению классической задачи линейного программирования с целевой функцией (8), ограничениями (6) и (9).

В результате решения этой задачи по определенным значениям $x_1 + x_2 + \ldots + x_n$ из уравнения (7) можно определить значение V, а затем из соотношений (9) значения $p_1, p_2, \ldots, p_n$, которые определяют оптимальную смешанную стратегию СРВ.

ПРИМЕР

Задана матрица игры (табл. 1), в которой стратегии СРВ (арендуемая часть IaaS) обозначены $R_1, R_2, \ldots, R_5$. Каждая стратегия предполагает включение в работу одного, двух, …, пяти процессоров. Возможные стратегии клиентов, обозначенные $P_1, P_2, \ldots, P_4$, предусматривают нужную производительность, обеспечиваемую загрузку 0,5; 1,25; 2,5 и 4,5 процессоров (здесь 0,5 следует понимать как загрузку одного процессора на 50%, соответственно это относится к другим данным). Пусть за дефицит производительности СРВ получает штраф, равный 5 условным единицам ($K_1 = 5$), если не достает производительности, обеспечиваемой одним процессором, то есть за избыточную производительность система получает штраф в 4 единицы за каждый лишний выделенный процессор ($K_2 = 4$). Например, для совокупности стратегий $\{R_2, P_3\}$ клиенту необходимо 2,5 процессора, система предоставляет 2 процессора, имеет место нехватка производительности. Штраф составляет $5 \cdot (2{,}5 - 2) = 2{,}5$ штрафной единицы.

Для совокупности стратегий $\{R_4, P_2\}$ клиенту нужно 1,25 процессора, система предоставляет 4 процессора. В этом случае имеет место избыточная производительность, штраф за которую составляет $4 \cdot (4 - 1{,}25) = 11$ штрафных единиц.

Как видно из решения (табл. 2), цена игры в данном примере равна 7,5 штрафной единицы. Решение определяет использование стратегий R_2, R_3 и R_4 с вероятностями, соответственно равными 0,36451; 0,27097 и 0,36451. Это позволяет считать целесообразным выбор числа процессоров СРВ из соотношения

$$N = n(R_2) \cdot p_2 + n(R_3) \cdot p_3 + n(R_4) \cdot p_4 = \\ = 2 \cdot 0{,}36451 + 3 \cdot 0{,}27097 + 4 \cdot 0{,}36451 \approx 3.$$

* * *

Использование представленной модели предполагает создание системы реального времени в форме совокупности веб-серверов, вычислительная мощность которых формируется на основе арендованной инфраструктуры IaaS. Арендатор создает собственный парк виртуальных серверов (виртуальный дата-центр) и разворачивает коммерческую систему для обслуживания пользователей, получая определенную прибыль. Предложенная игровая модель позволяет сбалансировать расходы на аренду и прибыль, получаемую от обслуживания пользователей. Для решения этой задачи большое значение имеет установление значений коэффициентов K_1 и K_2. Наблюдение за работой системы позволяет решить задачу. Что касается значений множеств D_i, t_i, зависящих от задач пользователей, то встроенные средства мониторинга вычислительного процесса, имеющиеся в операционных системах, позволяют найти простое решение.

Таблица 1. Матрица игры

		Стратегии клиентов P			
		P1	P2	P3	P4
Стратегия СРВ		0,5	1,25	2,5	4,5
R1	1	2	1,25	7,5	17,5
R2	2	6	3	2,5	12,2
R3	3	10	7	2	7,5
R4	4	14	11	6	2,5
R5	5	18	15	10	2

С использованием средств мониторинга и установленных значений коэффициентов K_1 и K_2 определяется текущее значение штрафа (цены игры) $S_э$, характерного для эффективной работы арендуемой системы. В процессе функционирования в зависимости от сложившейся ситуации (интенсивности потока запросов) средства мониторинга СРВ получают текущее значение штрафа, которое может отличаться от $S_э$. В этом случае необходимо уточнить множество возможных стратегий $P=\{P_1, P_2, \ldots, P_k\}$ и $R=\{R_1, R_2, \ldots, R_n\}$ и вновь решить задачу определения требуемой производительности СРВ. В идеале можно построить адаптивную СРВ на основе средств мониторинга и предложенной игровой задачи.

Таблица 2. Решение задачи

За дефицит производительности	5
За избыточную производительность	3

	Переменные		**Вероятности**	**Цена игры V**
	x_1	0	p_1 0	7,5
	x_2	0,048602	p_2 0,36451	
	x_3	0,036129	p_3 0,27097	
	x_4	0,048602	p_4 0,36451	
	x_5	0	p_5 0	
Целевая функция		0,133333	1	

Ограничения		
1	≤	1
0,7	≤	1
0,394408	≤	1
1	≤	1

ЛИТЕРАТУРА

1. Облачные сервисы 2013. Cnews-аналитика. – URL: http://www.cnews.ru/reviews/new/oblachnye_ servisy_2013/
2. **Батаев А. В.** Анализ использования облачных сервисов в банковском секторе // Молодой ученый. 2015. № 5. С. 234–240. URL: https://moluch.ru/archive/85/15818/
3. **Батаев А. В.** Перспективы внедрения облачных технологий в банковском секторе России // Научно-технические ведомости СПбГПУ. Экономические науки № 2 (192). 2014. С. 156–164.
4. **Монахов Д.Н., Монахов Н. В., Прончев Г. Б., Кузьменков Д. А.** Облачные технологии. – М.: Издательство МГУ им. М. В. Ломоносова, 2013.
5. **Риз Дж.** Облачные вычисления. – БХВ-Петербург, 2011. 288 с.
6. **Гордюшин А.В., Лебедева С. В.** Облачные технологии: технология создания «облака» // Вестник молодых ученых Санкт-Петербургского государственного университета технологии и дизайна. 2014. № 3. С. 53–57.
7. **Романова И.** Облачные технологии и их применение // Молодой ученый. 2016. № 17.1. С. 109–112. URL: https://moluch.ru/archive/121/33593/
8. **Макаров Д.В., Романчук В. А.** Облачные SaaS, IaaS, PaaS системы для искусственного интеллекта // Современная техника и технологии. 2015. № 5 URL: http://technology.snauka.ru/2015/05/6731
9. **Круликовский А.П., Тупота Е. С.** Инструментарий для управления «облачными» технологиями // В сб.: Актуальные проблемы и перспективы развития экономики Труды Юбилейной XV международной научно-практической конференции. Крымский федеральный университет им. В. И. Вернадского. 2016. С. 218–219.
10. **Кондратьев А.А., Тищенко И. П., Фраленко В. П.** Разработка распределенной системы защиты облачных вычислений // Программные системы: теория и приложения. 2011. № 4(8). С. 61–70.
11. **Гатиятуллин Т.Р., Сухова А. Р.** Проблемы безопасности в облачных технологиях // Проблемы развития современной науки // Сб. ст. Международной научно-практической конференции / Отв. ред. Сукиасян А. А.. 2015. С. 44–46.
12. **Enrico Bini, Giorgio C. Buttazzo and Giuseppe M. Buttazzo.** Rate Monotonic Analysis: the Hyperbolic Bound // IEEE Transactions on Computers. 2003 52: 933–942.
13. **Liu C.L., Layland J. W.** Scheduling Algorithms foR Multiprogramming in a Hard Real-Time Environment. J. ACM, 1973–20, pp. 46–61.
14. **Оскар Ланге.** Оптимальные решения. – М.: Прогресс, 1967. 286 с.
15. **Романьков В. А.**Введение в Теорию Игр: уч. пособ. – М.: РГГУ, 2014. 699 с.
16. **Захаров А. В.** Теория игр в общественных науках: учебник для вузов / Нац. исслед. ун-т «Высшая школа экономики». – М.: Изд. дом Высшей школы экономики, 2015. 304 с.