

ОПРЕДЕЛЕНИЕ РАЗМЕРА КЛАСТЕРА
И ЧИСЛА РЕПЛИК ВЫСОКОНАГРУЖЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Т. М. ТАТАРНИКОВА^{1*}, Е. Д. АРХИПЦЕВ², Н. С. КАРМАНОВСКИЙ³

¹ Санкт-Петербургский государственный университет аэрокосмического приборостроения,
Санкт-Петербург, Россия
*tm-tatarn@yandex.ru

² Санкт-Петербургский государственный электротехнический университет
„ЛЭТИ“ им. В.И. Ульянова (Ленина), Санкт-Петербург, Россия

³ Университет ИТМО, Санкт-Петербург, Россия

Аннотация. Проанализирована задача обеспечения качества обслуживания клиентов высоконагруженных информационных систем. Рассмотрены решения, предназначенные для обеспечения масштабирования таких систем. Предложена процедура определения необходимого числа узлов вычислительного кластера, основанная на теории очередей. Приведены результаты вычислительного эксперимента по выбору размера кластера, пропускная способность которого обеспечивает среднее время выполнения запроса пользователя, значение которого ниже установленного допустимого. Предложена модель определения числа реплик узлов, при котором при заданной интенсивности поступления запросов на обновление можно избежать критического уровня конфликта „запись—запись“.

Ключевые слова: высоконагруженная информационная система, качество обслуживания клиента, масштабирование, репликация, математическое моделирование

Ссылка для цитирования: Татарникова Т. М., Архипцев Е. Д., Кармановский Н. С. Определение размера кластера и числа реплик высоконагруженных информационных систем // Изв. вузов. Приборостроение. 2023. Т. 66, № 8. С. 646—651. DOI: 10.17586/0021-3454-2023-66-8-646-651.

DETERMINING THE CLUSTER SIZE
AND THE NUMBER OF REPLICAS OF HIGHLY LOADED
INFORMATION SYSTEMS

T. M. Tatarnikova^{1*}, E. D. Arkhitektsev², N. S. Karmanovskiy³

¹ St. Petersburg State University of Aerospace Instrumentation, St. Petersburg, Russia
*tm-tatarn@yandex.ru

² St. Petersburg Electrotechnical University, St. Petersburg, Russia

³ ITMO University, St. Petersburg, Russia

Abstract. The problem of ensuring the quality of customer service for highly loaded information systems is discussed. Existing solutions for scaling highly loaded information systems are considered. A procedure for determining the required number of nodes in a computing cluster based on the queuing theory is proposed. Results of a computational experiment on choosing the size of a computational cluster to ensure the average time for a user request executing to be below the established allowable time, are presented. A model for determining the number of replicas is proposed, in which, at a given intensity of incoming update requests, it is possible to avoid the critical level of the write-write conflict.

Keywords: high-load information system, customer service quality, scaling, replication, mathematical modeling

For citation: Tatarnikova T. M., Arkhiptsev E. D., Karmanovskiy N. S. Determining the cluster size and the number of replicas of highly loaded information systems. *Journal of Instrument Engineering*. 2023. Vol. 66, N 8. P. 646—651 (in Russian). DOI: 10.17586/0021-3454-2023-66-8-646-651.

Введение. Задача обеспечения качества обслуживания клиентов высоконагруженных информационных систем (ИС) — получения запрашиваемых данных за допустимое время, называемое „временем терпеливости“ пользователя при непрерывной работе ИС, и безостановочного доступа других клиентов — остается актуальной [1, 2].

Высокая нагрузка порождается большим количеством активных пользователей, объемом данных и сложными алгоритмами их обработки, а также фоновыми процессами.

Таким образом, обеспечение устойчивости системы к высокой нагрузке предполагает комплексное решение: реализацию алгоритмов обработки больших объемов данных, выбор оборудования, настройку операционной системы и программного обеспечения, использование механизмов масштабирования [3, 4].

В настоящей статье предлагаются решения (модели и методика), позволяющие масштабировать ИС с целью обеспечения качества обслуживания клиентов.

Способы масштабирования информационных систем. Существует два способа масштабирования ИС: репликация и фрагментация [5].

Репликация подразумевает копирование одних и тех же баз данных или их частей (реплик) на нескольких узлах вычислительного кластера. Количество реплик называется коэффициентом репликации. С целью ускорения доступа часто запрашиваемые в одном запросе данные размещаются на одном узле [6].

Риск несогласованности данных является основным недостатком репликации — некоторые пользователи могут получить устаревшие данные в силу того, что обновления на репликах данных выполняются с определенной задержкой, называемой окном несогласованности.

Фрагментация подразумевает, что разные части базы данных размещаются на множестве узлов одного кластера. Желательно так сгруппировать данные, чтобы один пользователь в основном получал данные с одного узла. В базах данных типа NoSQL для группировки применяется естественная единица хранения данных — агрегат, объединяющий данные, которые, как правило, запрашиваются одновременно.

Репликация и фрагментация являются ортогональными методами: можно использовать любой из них или оба вместе. Репликация бывает двух видов: по одноранговой схеме или по схеме „ведущий—ведомый“ [7].

При одноранговой репликации все узлы кластера имеют одинаковый вес и могут выполнять операции записи. Однако, когда выполняется запись в два разных места, есть риск, что два человека попробуют обновить одну и ту же запись в один и тот же момент времени. Таким образом, возникает конфликт „запись—запись“. Несогласованность результатов чтения тоже приводит к проблемам, но они являются преодолимыми, а несогласованность результатов записи необратима.

При распределении данных по схеме „ведущий—ведомый“ происходит репликация данных по многим узлам кластера. Один из узлов назначается ведущим — доверенным источником. Остальные узлы являются ведомыми. Процесс репликации синхронизирует ведомые узлы с ведущим.

Репликация „ведущий—ведомый“ — это решение для баз данных с интенсивным выполнением операции чтения и неудачное для баз данных с интенсивным трафиком записи: если часто обновлять данные на ведущем узле, то вероятность того, что пользователи будут получать несогласованные данные, растет с увеличением числа ведомых узлов [8].

Преимуществом репликации „ведущий—ведомый“ является отказоустойчивость чтения: если на ведущем узле произойдет отказ, ведомые узлы смогут по-прежнему обрабатывать запросы на чтение. Сбой ведущего узла делает невозможной запись данных до тех пор, пока его работа не будет восстановлена. И наличие реплик узлов ускоряет процесс восстановления ведущего узла после его сбоя.

Различают строгую и итоговую согласованность: строгая гарантирует, что данные вернутся неискаженными и не устареют; итоговая не может гарантировать, что данные вернутся полностью неискаженными, но со временем данные обновятся на всех репликах.

Уточним, что количество узлов в кластере и коэффициент репликации — это разные параметры ИС. Например, кластер может иметь 100 узлов при коэффициенте репликации, равном 3. При распределенном хранении базы данных необходимо найти баланс (золотую середину) между этими параметрами. С одной стороны, время отклика на запрос пользователя не должно быть выше допустимого времени $T_{\text{доп}}$, которое определяет гарантированное время терпеливости клиента и соответственно качество функционирования ИС [9]. С другой стороны, окно несогласованности должно гарантировать, что конфликт чтения данных устраним за ограниченное время [10].

Определение размера кластера. Представим кластер замкнутой системой массового обслуживания, в которой N — число узлов кластера. Узлы задаются временем обслуживания $T_{i\text{сер}}, i=\overline{1, N}$. Переход запроса клиента распределенной базы данных задается матрицей вероятностей p_{ij} . Необходимо найти оптимальное количество узлов кластера, такое чтобы запрос клиента выполнялся за допустимое время $T_{\text{доп}}$.

Для решения поставленной задачи предлагается следующая рекуррентная процедура:

$$\bar{T}_i(J) = T_{i\text{сер}} \left(1 + \left(\frac{\bar{L}_i(J-1)}{N} \right) \right), \quad i = \overline{1, N}; \quad (1)$$

$$\bar{T}(J) = \sum_{i=1}^N e_i \bar{T}_i(J); \quad (2)$$

$$\bar{\Lambda}(J) = \frac{J}{\bar{T}(J)}; \quad (3)$$

$$\bar{L}_i(J) = \bar{\Lambda}(J) e_i \bar{T}_i(J), \quad (4)$$

где $\bar{T}_i(J)$ — среднее время пребывания заявки в i -м узле при наличии в сети J заявок; $\bar{T}(J)$ — среднее время пребывания заявки в кластере при наличии в сети J заявок; $\bar{\Lambda}(J)$ — пропускная способность кластера при наличии в нем J заявок; вектор $e = [e_i]_{i=\overline{1, N}}$ является решением системы линейных уравнений

$$e_i = \sum_{j=1}^N e_j p_{ij}, \quad (5)$$

которая определяет стационарное распределение цепи Маркова, управляющей переходами заявок с матрицей вероятностей переходов $p_{ij}, i, j = \overline{1, N}$.

Система (5) решается при дополнительном ограничении $\sum_{i=1}^N e_i = 1$. Решение (1)—(5) начинается с $\bar{L}_i(0) = 0$, для $i = \overline{1, N}$.

Если выполняется условие:

$$\frac{\Lambda(j-1)}{\Lambda(j)} \geq \varepsilon, \quad 0,9 < \varepsilon < 1, \quad (6)$$

то система вошла в состояние насыщения — найдено максимальное число заявок, при котором система находится в стационарном режиме.

Если выполняется условие:

$$T \leq T_{\text{доп}}, \tag{7}$$

то оптимальная структура системы — число узлов кластера — найдена, если не выполняется, следует увеличить число каналов (узлов) на единицу и повторить процедуру (1)—(5) для новой структуры. Структурная оптимизация повторяется до выполнения условия (7).

Наращивание вычислительной мощности кластера через увеличение узлов — не единственный способ выбора оптимальной структуры кластера. Воспользовавшись процедурой (1)—(5), можно подобрать характеристики узлов, на которых строится кластер. На практике это означает замену узлов новыми, более производительными.

На рис. 1 приведена зависимость времени обработки запроса клиента от размера кластера (числа узлов N). Если принять $T_{\text{доп}} = 0,33$ с, то из рис. 1 видно, что при $N = 3$ среднее время обработки запроса пользователя составит 0,5 с, что не соответствует $T_{\text{доп}}$. Увеличение числа узлов кластера до 10 позволяет сократить время отклика на запрос пользователя и выполнить условие (7). Вычисления производились при $T_{\text{исер}} = 0,3$ с, $i=1, N$.

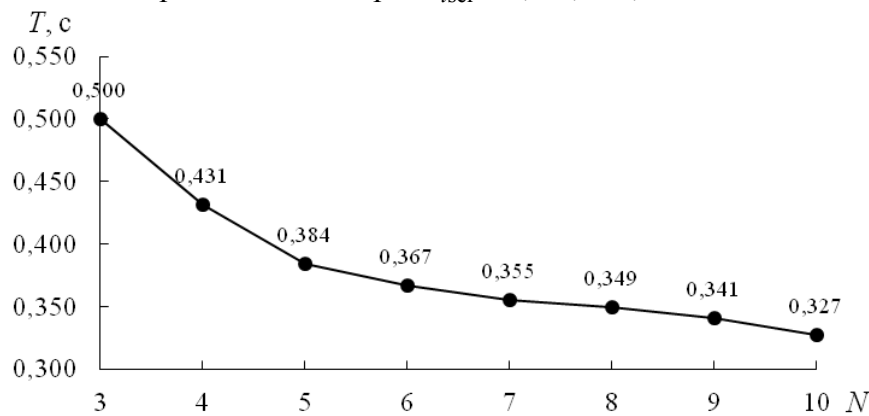


Рис. 1

На рис. 2 представлена динамика вхождения системы в состояние насыщения при $N = 3$ и $T_{\text{исер}} = 0,3$ с, $i=1,3$. Для других значений N динамика процесса насыщения системы аналогична приведенной на рис. 2.

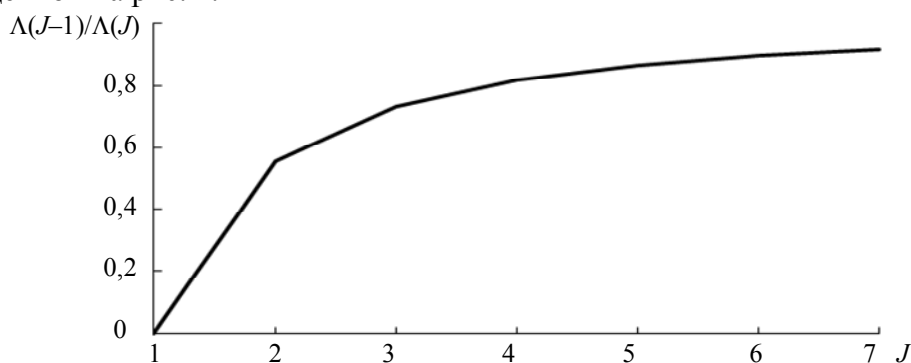


Рис. 2

Определение числа реплик. Для случайно поступающих запросов на запись вероятность пересечения двух или более запросов равна

$$P_w = 1 - e^{-\lambda T_w}, \tag{8}$$

где λ — реальная интенсивность поступления запросов на обновление данных в интервале времени T_w .

Определим максимальное количество узлов, участвующих в записи, R — число реплик узлов, при котором при заданной интенсивности поступления запросов на обновление можно избежать критического уровня конфликта „запись—запись“:

$$R\lambda T_w = \frac{1}{e}, \quad (9)$$

слева в (9) приведена максимальная производительность ВЦ, тогда

$$R = (e\lambda T_w)^{-1}. \quad (10)$$

На рис. 3 приведены результаты выбора числа реплик при разной интенсивности запросов на запись (обновление) данных $\lambda \in [0,05; 0,5]$ и размера окна несогласованности T_w .

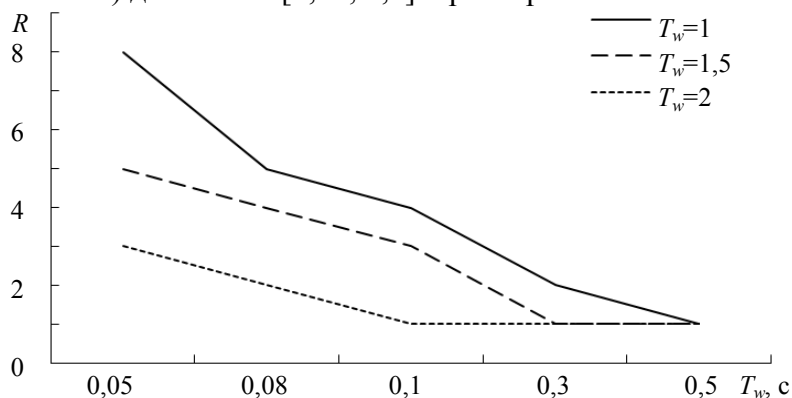


Рис. 3

Таким образом, для снижения риска формирования некорректных очередей запросов можно настраивать частоту обновления данных или при необходимости соблюдения интенсивности обновления данных — оценить число реплик, позволяющих не выходить за пределы окна несогласованности.

Заключение. Рассмотрены решения, предназначенные для масштабирования высоконагруженных информационных систем.

Предложена рекуррентная процедура выбора оптимальной структуры кластера для решения задачи фрагментации. Вычислительный эксперимент демонстрирует наращивание размера вычислительного кластера, пропускная способность которого обеспечивает среднее время выполнения запроса пользователя ниже установленного допустимого.

Предложена модель определения числа реплик, при котором для заданной интенсивности поступления запросов на обновление возможно избежать критического уровня конфликта „запись—запись“. Построена зависимость числа реплик от времени выполнения запроса при разных размерах окна несогласованности.

СПИСОК ЛИТЕРАТУРЫ

1. Тузов А. В. Исследование возможности использования линейной регрессии для предсказания расхода памяти в высоконагруженной информационной системе // Вестн. ЮУрГУ. Компьютерные технологии, управление, радиоэлектроника. 2018. Т. 18, № 3. С. 5—11.
2. Bogatyrev V. A., Derkach A. N., Aleksankov S. M. The model of reliability of dublicated real-time computers for cyber-physical systems // Studies in Systems, Decision and Control. 2020. Vol. 260. P. 11—21.
3. Амиров С. Н. Особенности разработки высоконагруженных систем // International Journal of Open Information Technologies. 2020. Vol. 8, N 8. P. 32—47.
4. Городничев М. Г., Кочупалов А. Е. Исследование методов межпроцессного взаимодействия в информационной системе с горизонтальным взаимодействием // Вестн. Евразийской науки. 2018. Т. 10, № 4. С. 1—32.

5. Подольный В. П. Архитектура высоконагруженных систем. Системы сбора информации, распределенные системы управления, системы реального времени. М.: ООО „САМ Полиграфист“, 2022. 160 с.
6. Цвящченко Е. В. Анализ согласованности базы данных NoSQL на этапе проектирования информационной системы // Информационные системы и технологии. 2016. № 1. С. 74—83.
7. Богатырев В. А., Лисичкин Д. Э. Оптимизация периодичности инициализации контроля на основе дублированных вычислений // Программные продукты и системы. 2019. № 2. С. 214—220.
8. Sovetov B. Ya., Tatarnikova T. M., Poymanova E. D. Storage scaling management model // Information and Control Systems. 2020. № 5(108). С. 43—49. DOI: 10.31799/1684-8853-2020-5-43-49.
9. Татарникова Т. М., Вольский А. В. Оценка вероятностно-временных характеристик сетевых узлов с дифференциацией трафика // Информационно-управляющие системы. 2018. № 3(94). С. 54—60. DOI: 10.15217/issn1684-8853.2018.3.54.
10. Татарникова Т. М., Елизаров М. А. Процедура разрешения коллизий в RFID-системе // Изв. вузов. Приборостроение. 2017. Т. 60, № 2. С. 150—157. DOI: 10.17586/0021-3454-2017-60-2-150-157.

Сведения об авторах

- Татьяна Михайловна Татарникова** — д-р техн. наук, профессор; Санкт-Петербургский государственный университет аэрокосмического приборостроения, Институт информационных технологий и программирования; директор института; E-mail: tm-tatarn@yandex.ru
- Евгений Дмитриевич Архипцев** — аспирант; Санкт-Петербургский государственный электротехнический университет „ЛЭТИ“ им. В.И. Ульянова (Ленина); кафедра информационных систем; E-mail: lokargenia@gmail.com
- Николай Сергеевич Кармановский** — канд. техн. наук; Университет ИТМО, факультет безопасности информационных технологий; E-mail: karmanov50@mail.ru

Поступила в редакцию 23.03.23; одобрена после рецензирования 06.04.23; принята к публикации 22.06.23.

REFERENCES

1. Tuzov A.V. *Bulletin of The South Ural State University. Series: Computer Technologies, Automatic Control, Radio Electronics*, 2018, no. 3(18), pp. 5–11. (in Russ.)
2. Bogatyrev V.A., Derkach A.N., Aleksankov S.M. *Studies in Systems, Decision and Control*, 2020, vol. 260, pp. 11–21.
3. Amirov S.N. *International Journal of Open Information Technologies*, 2020, no. 8(8), pp. 32–47. (in Russ.)
4. Gorodnichev M.G., Kochupalov A.E. *The Eurasian Scientific Journal*, 2018, no. 4(10), pp. 1–32. (in Russ.)
5. Podolny V.P. *Arkhitektura vysokonagruzhennykh sistem. Sistemy sbora informatsii, raspredelennyye sistemy upravleniya, sistemy real'nogo vremeni* (Architecture of Highly Loaded Systems. Information Collection Systems, Distributed Control Systems, Real-Time Systems), Moscow, 2022, 160 p. (in Russ.)
6. Cvyashhenko E.V. *Information Systems and Technologies*, 2016, no. 1, pp. 74–83. (in Russ.)
7. Bogatyrev V.A., Lisichkin D.E. *Software & Systems*, 2019, no. 2, pp. 214–220. (in Russ.)
8. Sovetov B.Ya., Tatarnikova T.M., Poymanova E.D. *Information and Control Systems*, 2020, no. 5(108), pp. 43–49, DOI: 10.31799/1684-8853-2020-5-43-49.
9. Tatarnikova T.M., Volskiy A.V. *Information and Control Systems*, 2018, no. 3(94), pp. 54–60, DOI: 10.15217/issn1684-8853.2018.3.54. (in Russ.)
10. Tatarnikova T.M., Elizarov M.A. *Journal of Instrument Engineering*, 2017, no. 2(60), pp. 150–157, DOI: 10.17586/0021-3454-2017-60-2-150-157. (in Russ.)

Data on authors

- Tatiana M. Tatarnikova** — Dr. Sci., Professor; St. Petersburg State University of Aerospace Instrumentation, Institute of Information Technologies and Programming; Director of the Institute; E-mail: tm-tatarn@yandex.ru
- Evgeny D. Arkhiptsev** — Post-Graduate Student; St. Petersburg Electrotechnical University, Department of Information Systems; E-mail: lokargenia@gmail.com
- Nikalay S. Karmanovskiy** — PhD; ITMO University, Faculty of Secure Information Technologies; E-mail: karmanov50@mail.ru

Received 23.03.23; approved after reviewing 06.04.23; accepted for publication 22.06.23.