

doi: 10.17586/2226-1494-2023-23-6-1171-1177

УДК 004.258

Алгоритм контроллера нечеткой логики для размещения файлов в системе хранения данных

Татьяна Михайловна Татарникова¹✉, Евгений Дмитриевич Архипцев²

¹ Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, 197002, Российская Федерация

^{1,2} Санкт-Петербургский электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), Санкт-Петербург, 197022, Российская Федерация

¹ tm-tatarn@yandex.ru✉, <http://orcid.org/0000-0002-6419-0072>

² lokargenia@gmail.com, <https://orcid.org/0009-0006-3001-953X>

Аннотация

Введение. Обсуждается задача организации многоуровневого хранения данных. Информация со временем теряет свою актуальность и стоимость ее хранения на носителях высокой доступности, например твердотельных накопителях, становится нецелесообразной. До сих пор размещение новых файлов в системе хранения данных решается горизонтально — без учета многоуровневой организации системы. Миграция файлов между уровнями системы хранения данных выполняется с течением времени, когда накапливается статистика по частоте их запроса. Все файлы обладают метаданными, такими как тип, размер, дата создания и другими, из которых можно выделить некоторую информацию об их важности. Это позволяет реализовать распределение данных по уровням системы хранения на ее входе. **Метод.** Предложена современная система хранения данных, представленная четырьмя уровнями. Первый уровень Hi-End предназначается для хранения критически важных данных с требованиями максимальной скорости доступа и надежности. Второй уровень Upper Mid-Range предназначается для корпоративных приложений, требующих высокой скорости доступа. Третий уровень Mid-Range предлагается использовать для организации файловых хранилищ, а четвертый — Entry Level — для создания резервных копий и архивов. В предложенном алгоритме размещения файлов по уровням системы хранения данных учитываются метрики, указывающие на требования к хранению и выбору уровня системы хранения данных, соответствующего предъявляемым требованиям. К таким метрикам относятся степень доступности (скорость предоставления информации), важность (стоимость потери данных в случае аппаратного и программного сбоя), срок хранения и частота запроса. Метрики выделяются из метаданных сохраняемых файлов. **Основные результаты.** Предложено новое решение, основанное на функциях контроллера нечеткой логики. Алгоритм его работы может быть интегрирован в систему хранения данных до процесса записи нового файла. Алгоритм включает три основных шага. На первом шаге выполняется анализ файловых метрик для формирования соответствующих им входных нечетких множеств. На втором шаге для формирования итогового нечеткого множества применяется логическая модель. На завершающем этапе происходит получение результата нечеткого вывода и размещение файла на соответствующий уровень системы хранения данных. Приведен пример работы контроллера для файлов с разными значениями метрических характеристик. **Обсуждение.** Контроллер нечеткой логики может быть интегрирован в работу многоуровневой системы хранения данных.

Ключевые слова

большие данные, многоуровневая система хранения данных, требования к хранению данных, метрики файлов, контроллер нечеткой логики, рациональное хранение

Ссылка для цитирования: Татарникова Т.М., Архипцев Е.Д. Алгоритм контроллера нечеткой логики для размещения файлов в системе хранения данных // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 6. С. 1171–1177. doi: 10.17586/2226-1494-2023-23-6-1171-1177

Fuzzy logic controller algorithm for placing files in a data storage system

Tatiana M. Tatarnikova¹✉, Evgeny D. Arkhiptsev²

¹ Saint Petersburg State University of Aerospace Instrumentation, Saint Petersburg, 197002, Russian Federation

^{1,2} Saint Petersburg Electrotechnical University “LETI”, Saint Petersburg, 197022, Russian Federation

¹ tm-tatarn@yandex.ru✉, <http://orcid.org/0000-0002-6419-0072>

² lokargenia@gmail.com, <https://orcid.org/0009-0006-3001-953X>

Abstract

The problem of organizing multi-level data storage is discussed. Information loses its relevance over time and the cost of storing it on highly available media, such as solid-state drives, becomes impractical. Until now, the placement of new files in the data storage system is decided horizontally — without taking into account the multi-level organization of the system. The file migration between storage system tiers occurs over time as statistics on the frequency of requested files are accumulated. All files have metadata, such as type, size, creation date and others, from which some information about the importance of the information can be extracted and then the distribution by levels of the data storage system can be implemented at the system input. A modern data storage system represented by four levels is proposed. The first Hi-End level is intended for storing critical data with the requirements of maximum access speed and reliability. The second level, Upper Mid-Range, is intended for enterprise applications that require high access speeds. The third level, Mid-Range, is proposed to be used for organizing file storage, and the fourth, Entry Level, is proposed to be used for creating backup copies and archives. The proposed algorithm for arranging files across tiers of a data storage system takes into account metrics indicating storage requirements and selecting a level of a data storage system that meets the requirements. These metrics include availability (speed of information delivery), importance (cost of data loss due to hardware and software failures), retention period, and request frequency. Metrics are extracted from the metadata of saved files. A new solution based on the functions of a fuzzy logic controller is proposed. Its operation algorithm can be integrated into the data storage system before the process of writing a new file. The algorithm includes three main steps. At the first step, file metrics are analyzed to form the corresponding input fuzzy sets. At the second step, a logical model is used to form the final fuzzy set. At the final stage, the fuzzy output result is obtained and the file is placed at the appropriate level of the data storage system. An example of how the controller works for files with different values of metric characteristics is given. A fuzzy logic controller can be integrated into the operation of a multi-level data storage system.

Keywords

big data, multi-level data storage system, data storage requirements, file metrics, fuzzy logic controller, rational storage

For citation: Tatarnikova T.M., Arkhiptsev E.D. Fuzzy logic controller algorithm for placing files in a data storage system. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 6, pp. 1171–1177 (in Russian). doi: 10.17586/2226-1494-2023-23-6-1171-1177

Введение

Проблема больших данных в основном обсуждается в контексте их обработки специальными автоматизированными инструментами для получения статистики, выполнения анализа, прогнозов и принятия решений [1–3]. Тем не менее, физические процессы хранения больших данных остаются важными и также требуют поиска инструментов автоматизации в условиях ограниченности физических ресурсов [4, 5]. В частности, организация рационального хранения массивов неоднородных данных является одной из таких приоритетных задач.

Большие данные, как известно, хранятся в облаке. Рынок облачных вычислений в России растет на 30–40 % в год, при этом наибольшим спросом пользуются сервисы «приложение как услуга» (SaaS) и «инфраструктура как услуга» (IaaS). Причина — устаревание информационных систем компаний, которые в целях оптимизации аппаратно-программных ресурсов обращаются к облачным решениям [6, 7].

Инфраструктуру облачного хранилища образуют системы хранения данных (СХД). СХД отличается сложной архитектурой, возможностью объединять носители разной физической природы в сеть передачи данных¹, наличием отдельного программного обеспечения для

управления СХД² технологиями резервного копирования³, сжатия и виртуализации [8].

В свою очередь управление СХД подразумевает не только управление процессами записи-чтения, но и объемами данных — организацию рационального хранения, основанного на важности данных, стоимости их хранения и требований к срокам хранения [9]. Требованиями законов и норм Российской Федерации жестко закреплены сроки хранения информации практически во всех важнейших сферах: труд и занятость населения, информация и информатизация, техническое регулирование, образовательная, научная и культурная деятельность, медицина и фармацевтика. Также очевидна тенденция к переводу архивных документов в цифровой вид — бессрочное хранение. Согласно Федеральному закону № 374-ФЗ⁴, операторы связи

Режим доступа: <https://www.itu.int/rec/T-REC-Y.3510-201305-S> (дата обращения: 08.10.2023).

² Recommendation Y.3501: Cloud computing framework and high-level requirements. Geneva: ITU-T, 2013 [Электронный ресурс]. Режим доступа: <https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11917> (дата обращения: 08.10.2023).

³ Recommendation Y.3520: Cloud computing framework for end-to-end resource management. Geneva: ITU-T, 2015 [Электронный ресурс]. Режим доступа: <https://www.itu.int/rec/T-REC-Y.3520-201509-I> (дата обращения: 08.10.2023).

⁴ Федеральный закон № 374-ФЗ от 23.11.2020 г. О внесении изменений в части первую и вторую Налогового кодекса

¹ Recommendation Y.3510: Cloud computing infrastructure requirements. Geneva: ITU-T, 2013. [Электронный ресурс].

обязаны хранить на территории Российской Федерации информацию о фактах приема, передачи, доставки и/или обработки голосовой информации, текстовых сообщений, изображений, звуков, видео- или иных сообщений пользователей услугами связи — в течение трех лет с момента окончания осуществления таких действий. Таким образом, хранение автоматически включается в тариф на оказание услуг связи.

Несмотря на то, что СХД строятся по иерархическому принципу, автоматизация распределения данных по уровням СХД остается частично решенной: имеет место только горизонтальное размещение, а миграция данных между уровнями хранилища реализуется администратором СХД. Так, в [2–5] рассмотрены новые технологии распределенного хранения данных, суть которых сводится к обеспечению приемлемой пропускной способности СХД через кластеризацию и надежности хранения через избыточную репликацию данных. Анализ работ [4–10] показал наличие противоречия: с одной стороны, для СХД по-прежнему обсуждаются традиционные вычислительные характеристики, оставляя без внимания стоимость хранения, а с другой стороны, очевидно, что хранение разных данных требует разной оплаты предоставления услуги. Но последнее в настоящий момент решается через миграцию файлов с течением времени, когда накапливается статистика по частоте запрашиваемых данных.

В настоящей работе предлагается алгоритм распределения данных в СХД с многоуровневой организацией еще до процесса записи нового файла. Алгоритм основан на работе контроллера нечеткой логики.

Материалы и методы

Процесс старения (потери актуальности) данных оценивается частотой обращений к ним во времени и описывается гауссовской кривой: количество обращений к свежим данным мало, по мере распространения сведений о новых данных число обращений возрастает, затем данные теряют свою актуальность, и число обращений к ним падает, и к архивированным данным количество обращений существенно меньше пикового [11]. Это свойство используется в создании многоуровневых СХД. Выделяют четыре уровня СХД (рис. 1):

- уровень 1 (Hi-End) образуют SSD (Solid-State Drive) твердотельные накопители и соответственно массив дисков из них. Хранение данных характеризуется высокими стоимостью, надежностью, функциональностью и масштабируемостью. Здесь размещаются критически важные данные с требованиями максимальной (для онлайн-решений) производительности и надежности;
- уровень 2 (Upper Mid-Range) образуют массивы дисков, обмен данными с которыми осуществляется по высокопроизводительному протоколу SAS (Serial Attached SCSI) или FC (Fibre Channel). Хранение характеризуется от средней до высокой стоимости, высокой надежностью, достаточной функцио-

нальностью. На этом уровне размещаются важные приложения, например, корпоративный почтовый сервис и виртуальные машины, для которых важна высокая производительность;

- уровень 3 (Mid-Range) образуют массивы дисков, подключенных по интерфейсу SATA (Serial Advanced Technology Attachment) или NL-SAS (Near Line Serial Attached SCSI). Хранение характеризуется средней стоимостью и ограниченной функциональностью. Фактически, на данном уровне образуются файловые хранилища, для которых важны большие объемы и энергоэффективность;
- уровень 4 (Entry Level) образуют ленточные СХД, предназначенные для создания резервных копий и архивов: отдельные накопители, автозагрузчики, ленточные библиотеки. Хранение характеризуется низкой стоимостью и ограниченной функциональностью.

Таким образом, перед размещением данных каждой единице информации, подлежащей обработке и хранению, будем присваивать определенные метрики, устанавливающие требования к хранению [12–14]. В их числе степень доступности (скорость предоставления информации), важность (стоимость потери данных в случае аппаратного и программного сбоя), срок хранения и частота запроса. Метрики также возможно выделить из метаданных сохраняемых файлов. Одним из известных стандартов описания метаданных является Dublin Core Metadata Element Set (DCMES) — набор элементов метаданных для описания цифровых или физических ресурсов¹.

Множество файловых метрик будем подавать на вход контроллера нечеткой логики до самого процесса записи нового файла в СХД.

Структуру любого контроллера нечеткой логики составляют (рис. 2): блок фаззификации, блок функции принадлежности, лингвистическая модель, модуль нечеткого вывода и блок дефаззификации [15].

Фаззификация — процесс применения функции принадлежности для интерпретации точных значений переменных x_1, x_2, \dots, x_n в нечеткие множества z_1, z_2, \dots, z_n лингвистической переменной.

Нечеткое множество определяется через базовую шкалу B переменной x и функцию принадлежности $\mu(x), x \in B$, принимающую значения на интервале $[0, 1]$.

Лингвистическая переменная — набор характеристик некоторого свойства — термов. Например, лингвистическая переменная «Важность данных» задается нечетким множеством {«низкая», «средняя», «высокая»}, в котором каждый терм имеет следующий диапазон значений:

- «низкая» — стоимость потери данных в случае аппаратного и программного сбоя не превысит 30 % активов компании;
- «средняя» — не превысит 80 % активов компании;
- «высокая» — превысит 80 % активов компании.

Российской Федерации и отдельные законодательные акты Российской Федерации.

¹ ГОСТ Р 7.0.10–2019 (ИСО 15836-1:2017) Набор элементов метаданных «Дублинское ядро». Введен 01.01.2020. Национальный стандарт Российской Федерации. 11 с.

Уровень	Класс данных	Носители
1 Hi-End	Критические приложения	SSD
2 Upper Mid-Range	Важные приложения, виртуализация	FC/SAS
3 Mid-Range	Инфраструктурные сервисы, файловые хранилища	SATA/NL-SAS
4 Entry Level	Резервные копии, архив	Ленточные СХД

Рис. 1. Иерархическая модель системы хранения данных

Fig. 1. Hierarchical model of the storage system

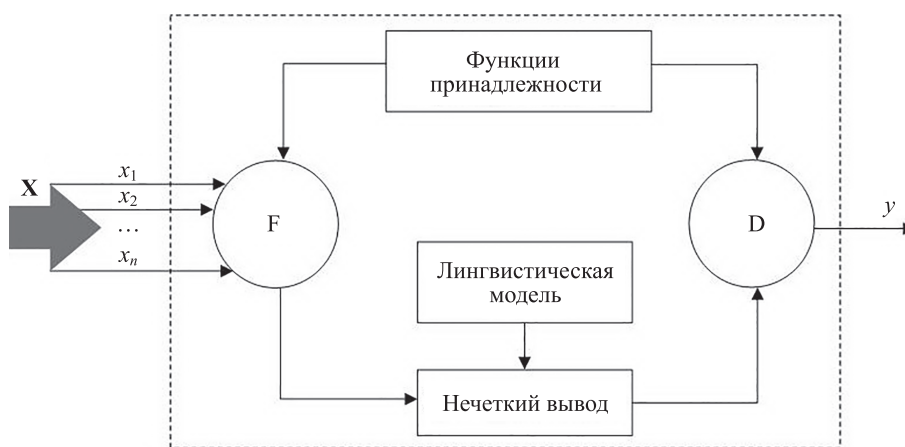


Рис. 2. Структура контроллера нечеткой логики.

F — блок фаззификации; D — блок дефаззификации; X — множество входных переменных; y — результат дефаззификации

Fig. 2. The structure of the fuzzy logic controller.

F — fuzzification block; D — defuzzification block; X — set of input variables; y — result of defuzzification

Функция принадлежности — субъективная степень уверенности эксперта в том, что данное конкретное значение базовой шкалы соответствует определяемому нечеткому множеству [16].

Лингвистическая модель — совокупность нечетких предикатных правил вида [17]

Rule: If x is $\mu(x)$, Then y is $\mu(y)$,

где x — имя входной переменной; y — имя переменной вывода; μ — заданная функция принадлежности; If <...> — предпосылка правила; Then <...> — заключение правила.

Для каждого правила Rule_i , $i = \overline{1, n}$ вычисляется значение истинности α_i , $i = \overline{1, n}$ предпосылки и применяется к заключению, что приводит к формированию нечетких подмножеств для всех термов результирующей лингвистической переменной $Y = \{y_1, y_2, \dots, y_k\}$ [18]. Формирование i -го нечеткого подмножества y_i , $i = \overline{1, k}$ выполняется применением операции нечеткой логики «И», согласно которой функция принадлежности выходной переменной μ_i «отсекается» по высоте, соответствующей значению α_i (рис. 3).

Далее все нечеткие подмножества y_i , $i = \overline{1, k}$ лингвистических переменных объединяются вместе, чтобы формировать итоговое нечеткое множество выходной переменной Y . Такое объединение называется композицией и для ее реализации используется операция нечеткой логики «ИЛИ».

Дефаззификация — процесс обратный интерпретации — четкого значения выходной переменной, например, определение центра тяжести итогового множества Y или центроида площади:

$$y = \frac{\int_{i=\min}^{i=\max} y_i \mu(y_i) dy_i}{\int_{i=\min}^{i=\max} \mu(y_i) dy_i}, \quad (1)$$

где y_i — результат выходной лингвистической переменной Y ; $\mu(y_i)$ — функция принадлежности нечеткого подмножества y_i после композиции; \min и \max — левая и правая границы интервала носителя нечеткого множества выходной переменной.

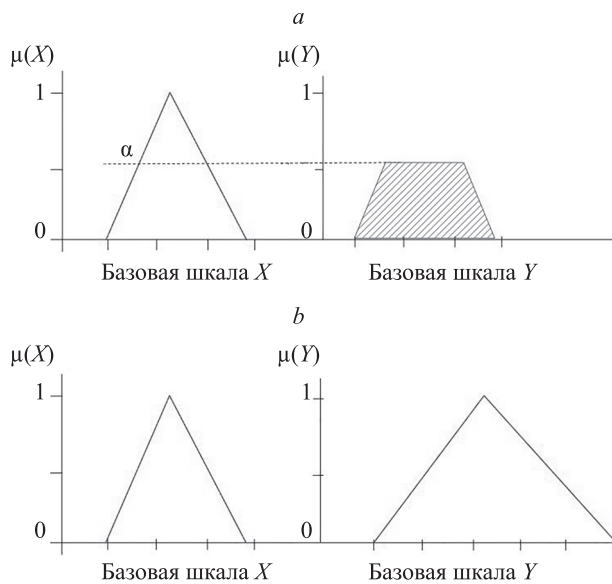


Рис. 3. Формирование лингвистической переменной нечеткого вывода: применением операции нечеткой логики И (a); применением операции нечеткой логики ИЛИ (b)
 Fig. 3. Formation of a linguistic variable of fuzzy inference: using the fuzzy logic operation AND (a); using the fuzzy logic operation OR (b)

Результаты исследования

В определении уровня СХД для размещения нового файла участвуют следующие метрики:

- степень доступности, обозначим как нечеткое множество z_1 ; $z_1 = \{\text{«низкая»}, \text{«средняя»}, \text{«высокая»}\}$;
- важность данных $z_2 = \{\text{«низкая»}, \text{«средняя»}, \text{«высокая»}\}$;

- срок хранения $z_3 = \{\text{«малый»}, \text{«средний»}, \text{«большой»}\}$;
- частота запросов данных $z_4 = \{\text{«низкая»}, \text{«средняя»}, \text{«высокая»}\}$;

Введем конечное число функций принадлежности (рис. 4).

Формирование функций принадлежности для фаззификации элементов множеств $\{z_i\}_{i=1,4}$ позволяет осуществить переход к соответствующим нечетким множествам

$$\tilde{z}_i = \mu(z_i),$$

где \tilde{z}_i — нечеткие числа в интервале $[0, 1]$, соответствующие оценкам i -й метрики j -го файла.

Логическую модель зададим правилом Мамдани:

$$\mu(L) = \max_{k=1,n} \{\min[\mu_1(z_1)_k, \mu_2(z_2)_k, \mu_3(z_3)_k, \mu_4(z_4)_k]\},$$

где n — количество правил логической модели, $n = 3^4 = 81$ — четыре лингвистические переменные имеют по три терма.

Введем лингвистическую переменную L «уровень СХД», которая определяет результат контроллера нечеткой логики. $L = \{\text{«Hi-End»}, \text{«Upper Mid-Range»}, \text{«Mid-Range»}, \text{«Entry Level»}\}$. Значение L вычислим в результате дефаззификации выходного нечеткого множества методом центроида площади (1).

Выбор правила Мамдани из других возможных обоснован равнозначностью влияния метрических характеристик файлов на определение уровня СХД при записи нового файла.

Пусть, для 10 файлов контроллер нечеткой логики ищет оптимальный уровень СХД. В результате ме-

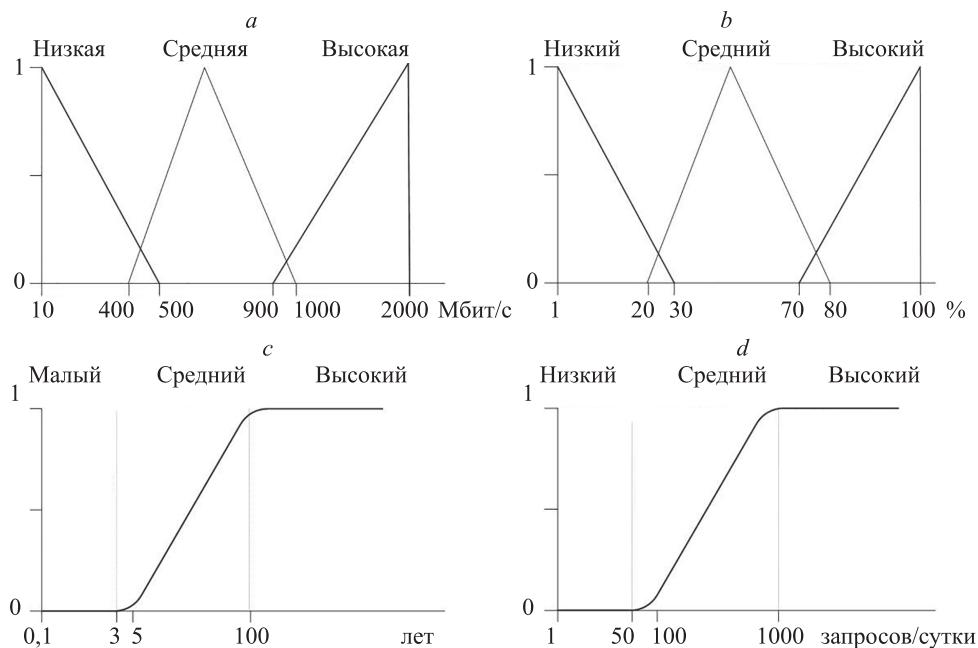


Рис. 4. Функции принадлежности метрических характеристик файлов: степень доступности (μ_1) (a); важность данных (μ_2) (b); срок хранения (μ_3) (c); частота запросов данных (μ_4) (d)

Fig. 4. Membership functions of metric characteristics of files: degree of accessibility (μ_1) (a); data importance (μ_2) (b); storage period (μ_3) (c); frequency of data requests (μ_4) (d)

Таблица 1. Значения метрических характеристик файлов

Table 1. Values of metric characteristics of files

Номер файла	z_1	z_2	z_3	z_4
1	0,9	0,8	1	0,9
2	0,6	0,7	0,8	0,8
3	0,7	0,6	0,7	0,4
4	0,9	0,8	0,9	0,1
5	0,8	0,7	0,6	0,7
6	0,9	0,3	0,4	0,2
7	0,4	0,5	0,2	0,7
8	0,2	0,8	1	0,9
9	0,3	0,1	0,3	0,3
10	0,6	0,4	0,4	0,7

три файла соответствуют данным, приведенным в табл. 1.

Результаты нечеткого вывода для размещения файлов в СХД приведены в табл. 2.

С течением времени значения метрик файлов будут меняться согласно гауссовской кривой в связи с потерей актуальности данных. Исходя из этого, работа контроллера для уже размещенных в СХД файлов должна периодически запускаться в сроки, установленные администратором СХД. Таким образом, будет осуществляться миграция данных по уровням СХД.

Заключение

Предложен новый алгоритм размещения файлов по уровням системы хранения данных в соответствии с метриками, указывающими на требования к хранению. Доказано использование в качестве метрик: степень доступности запрашиваемых данных (скорость ее предоставления); важность сохраняемых данных (стоимость

Таблица 2. Результаты фаззификации

Table 2. Fuzzification results

Номер файла	Центроид	Уровень хранения
1	0,9	Hi-End
2	0,725	Upper Mid-Range
3	0,6	Upper Mid-Range
4	0,675	Upper Mid-Range
5	0,7	Upper Mid-Range
6	0,45	Mid-Range
7	0,425	Mid-Range
8	0,5	Mid-Range
9	0,25	Entry Level
10	0,525	Upper Mid-Range

их потери в случае аппаратного и программного сбоя); запрашиваемый срок хранения данных и частота запроса одних и тех же данных. Метрики также возможно выделять из метаданных сохраняемых файлов.

Алгоритм размещения файлов по уровням системы хранения данных фактически является контроллером нечеткой логики, правила работы которого позволяют принимать решение о размещении файла на соответствующий уровень системы хранения данных. Номер уровня системы хранения данных для конкретного файла вычисляется на основе комплексного показателя значений метрик файлов.

Показан пример работы контроллера нечеткой логики по размещению файлов по уровням системы хранения данных с раскрытием содержания структурных элементов контроллера: нечетких множеств, функций принадлежности и логической модели.

Алгоритм работы контроллера нечеткой логики может быть интегрирован в работу системы хранения данных до процесса записи нового файла.

Литература

1. Боков С.И., Чупринов А.А. О роли обеспечения системы управления цифровой экономикой России на основе организации единого информационного пространства // *Наноиндустрия*. 2019. Т. 12. № 8(9). С. 135–139. <https://doi.org/10.22184/NanoRus.2019.12.89.135.139>
2. Challawala S., Lakhatariya J., Mehta C., Patel K. *MySQL 8 for Big Data: Effective Data Processing with MySQL 8, Hadoop, NoSQL APIs, and Other Big Data Tools*. Packt Publishing, 2017. 226 p.
3. Lakshmanan V., Tigani J. *Google BigQuery: The Definitive Guide: Data Warehousing, Analytics, and Machine Learning at Scale*. O'Reilly Media, 2019. 475 p.
4. Сивов В.В., Богатырев В.А. Отказоустойчивый кластер хранения данных для аналитических запросов в банковской сфере // *Advanced Engineering Research (Rostov-on-Don)*. 2023. Т. 23. № 1. С. 76–84. <https://doi.org/10.23947/2687-1653-2023-23-1-76-84>
5. Татарникова Т.М., Архипцев Е.Д. Определение числа реплик распределенного хранения больших данных // *Международная конференция по мягким вычислениям и измерениям*. 2023. Т. 1. С. 305–308.
6. Шевцов В.Ю., Абрамов В.С. Анализ современных систем хранения данных // *НБИ технологии*. 2019. Т. 13. № 1. С. 25–30. <https://doi.org/https://doi.org/10.15688/NBIT.jvolsu.2019.1.5>

References

1. Bokov S.I., Chuprinov A.A. On the role of digital economy of Russia management system based on organizing a unified information space. *Nanoindustry*, 2019, vol. 12, no. 8(9), pp. 135–139. (in Russian). <https://doi.org/10.22184/NanoRus.2019.12.89.135.139>
2. Challawala S., Lakhatariya J., Mehta C., Patel K. *MySQL 8 for Big Data: Effective Data Processing with MySQL 8, Hadoop, NoSQL APIs, and Other Big Data Tools*. Packt Publishing, 2017, 226 p.
3. Lakshmanan V., Tigani J. *Google BigQuery: The Definitive Guide: Data Warehousing, Analytics, and Machine Learning at Scale*. O'Reilly Media, 2019, 475 p.
4. Sivov V.V., Bogatyrev V.A. Data warehouse failover cluster for analytical queries in banking. *Advanced Engineering Research (Rostov-on-Don)*, 2023, vol. 23, no. 1, pp. 76–84. <https://doi.org/10.23947/2687-1653-2023-23-1-76-84>
5. Tatarnikova T.M., Arkhptsev E.D. Determine the number of distributed big data storage replicas. *International Conference on Soft Computing and Measurements*, 2023, vol. 1, pp. 305–308. (in Russian)
6. Shevtsov V.Yu., Abramov E.S. The analysis of modern data storage systems. *NBIT technologies*, 2019, vol. 13, no. 1, pp. 25–30. (in Russian). <https://doi.org/https://doi.org/10.15688/NBIT.jvolsu.2019.1.5>

7. Татарникова Т.М., Пойманова Е.Д. Методика дифференцированного наращивания емкости системы хранения данных с многоуровневой структурой // Научно-технический вестник информационных технологий, механики и оптики. 2020. Т. 20. № 1. С. 66–73. <https://doi.org/10.17586/2226-1494-2020-20-1-66-73>
8. *Information Storage and Management* / 2nd ed. New Jersey: John Wiley & Sons Inc., 2016. 544 p.
9. Попов С.Г., Фридман В.С. Обзор методов динамического распределения данных в распределенных системах управления базами данных // Информатика, телекоммуникации и управление. 2018. Т. 11. № 4. С. 82–107. <https://doi.org/10.18721/JCSTCS.11407>
10. Полищук Ю.В. Взаимосвязь ценности и старения информации при управлении большими техническими системами // Информационные технологии. 2019. Т. 25. № 6. С. 381–384. <https://doi.org/10.17587/it.25.381-384>
11. Kish L.B., Granqvist C.G. Does information have mass? // *Proceedings of the IEEE*. 2013. V. 101. N 9. P. 1895–1899. <https://doi.org/10.1109/jproc.2013.2273720>
12. Svetov B.Ya., Tatarnikova T.M., Poymanova E.D. Storage scaling management model // Информационно-управляющие системы. 2020. № 5. С. 43–49. <https://doi.org/10.31799/1684-8853-2020-5-43-49>
13. Богатырев В.А., Богатырев С.В., Богатырев А.В. Оценка готовности компьютерной системы к своевременному обслуживанию запросов при его совмещении с информационным восстановлением памяти после отказов // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23. № 3(145). С. 608–617. <https://doi.org/10.17586/2226-1494-2023-23-3-608-617>
14. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. Reliability and timeliness of servicing requests in infocommunication systems, taking into account the physical and information recovery of redundant storage devices // *Proc. of the 2022 International Conference on Information, Control, and Communication Technologies (ICCT)*. 2022. P. 1–4. <https://doi.org/10.1109/icct56057.2022.9976800>
15. Шилкина С.В., Фокина Е.Н. Контроллер нечеткой логики в управлении технологическими процессами // Вестник Сибирского государственного автомобильно-дорожного университета. 2018. Т. 15. № 1. С. 106–114. <https://doi.org/10.26518/2071-7296-2018-1-106-114>
16. Савченко Д.В., Резникова К.М., Смышляева А.А. Нечеткая логика и нечеткие информационные технологии // Интернет-журнал «Отходы и ресурсы». 2021. Т. 8. № 1 [Электронный ресурс]. URL: <https://resources.today/PDF/10ECOR121.pdf> (дата обращения: 08.10.2023). <https://doi.org/10.15862/10ECOR121>
17. Киселева Э.А., Краева А.А., Савинова Ю.С. Обзор нечеткой логики в управлении // *Международный журнал прикладных наук и технологий Integral*. 2019. № 3. С. 59.
18. Веретёхин А.В., Ячменева В.М. Оценка эколого-экономической безопасности промышленного предприятия с использованием инструментария нечеткой логики // Научно-технические ведомости СПбГПУ. Экономические науки. 2017. Т. 10. № 3. С. 140–157. <https://doi.org/10.18721/JE.10313>
7. Tatarnikova T.M., Poymanova E.D. Differentiated capacity extension method for system of data storage with multilevel structure. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2020, vol. 20, no. 1, pp. 66–73. (in Russian). <https://doi.org/10.17586/2226-1494-2020-20-1-66-73>
8. *Information Storage and Management*. 2nd ed. New Jersey, John Wiley & Sons Inc., 2016, 544 p.
9. Popov S.G., Fridman V.S. Review of methods for dynamic distribution of data in distributed database management systems. *Computing, Telecommunications and Control*, 2018, vol. 11, no. 4, pp. 82–107. (in Russian). <https://doi.org/10.18721/JCSTCS.11407>
10. Polishuk Yu.V. Relations between significance and aging of information when control big technical systems. *Information Technology*, 2019, vol. 25, no. 6, pp. 381–384. (in Russian). <https://doi.org/10.17587/it.25.381-384>
11. Kish L.B., Granqvist C.G. Does information have mass? *Proceedings of the IEEE*, 2013, vol. 101, no. 9, pp. 1895–1899. <https://doi.org/10.1109/jproc.2013.2273720>
12. Svetov B.Ya., Tatarnikova T.M., Poymanova E.D. Storage scaling management model. *Information and Control Systems*, 2020, no. 5, pp. 43–49. (in Russian). <https://doi.org/10.31799/1684-8853-2020-5-43-49>
13. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. Assessment of the readiness of a computer system for timely servicing of requests when combined with information recovery of memory after failures. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 3(145), pp. 608–617. (in Russian). <https://doi.org/10.17586/2226-1494-2023-23-3-608-617>
14. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. Reliability and timeliness of servicing requests in infocommunication systems, taking into account the physical and information recovery of redundant storage devices. *Proc. of the 2022 International Conference on Information, Control, and Communication Technologies (ICCT)*, 2022, pp. 1–4. <https://doi.org/10.1109/icct56057.2022.9976800>
15. Shilkina S.V., Fokina E.N. The controller of fuzzy logic in the management of technological processes. *The Russian Automobile and Highway Industry Journal*, 2018, vol. 15, no. 1, pp. 106–114. (in Russian). <https://doi.org/10.26518/2071-7296-2018-1-106-114>
16. Savchenko D.V., Reznikova K.M., Smyshlyaeva A.A. Fuzzy logic and fuzzy information technology. *Russian Journal of Resources, Conservation and Recycling*, 2021, vol. 8, no. 1. Available at: <https://resources.today/PDF/10ECOR121.pdf> (accessed: 08.10.2023). (in Russian). <https://doi.org/10.15862/10ECOR121>
17. Kiseleva E.A., Kraeva A.A., Savinova Iu.S. Overview of fuzzy logic in management. *International journal of applied sciences and technology Integral*, 2019, no. 3, pp. 59. (in Russian)
18. Veretyokhin A.V., Yachmenova V.M. Evaluation of ecological and economic safety of an industrial enterprise using fuzzy logic tools. *St. Petersburg State Polytechnical University Journal. Economics*, 2017, vol. 10, no. 3, pp. 140–157. (in Russian). <https://doi.org/10.18721/JE.10313>

Авторы

Татарникова Татьяна Михайловна — доктор технических наук, профессор, профессор, Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, 197002, Российская Федерация; профессор, Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), Санкт-Петербург, 197022, Российская Федерация, [sc 36715607400](mailto:tm-tatarn@yandex.ru), <http://orcid.org/0000-0002-6419-0072>, tm-tatarn@yandex.ru

Архипцев Евгений Дмитриевич — аспирант, Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), Санкт-Петербург, 197022, Российская Федерация, <https://orcid.org/0009-0006-3001-953X>, lokargenia@gmail.com

Authors

Tatiana M. Tatarnikova — D.Sc., Full Professor, Saint Petersburg State University of Aerospace Instrumentation, Saint Petersburg, 197002, Russian Federation; Professor, Saint Petersburg Electrotechnical University “LETI”, Saint Petersburg, 197022, Russian Federation, [sc 36715607400](mailto:tm-tatarn@yandex.ru), <http://orcid.org/0000-0002-6419-0072>, tm-tatarn@yandex.ru

Evgeny D. Arkhipev — PhD Student, Saint Petersburg Electrotechnical University “LETI”, Saint Petersburg, 197022, Russian Federation, <https://orcid.org/0009-0006-3001-953X>, lokargenia@gmail.com

Статья поступила в редакцию 09.10.2023
Одобрена после рецензирования 06.11.2023
Принята к печати 28.11.2023

Received 09.10.2023
Approved after reviewing 06.11.2023
Accepted 28.11.2023



Работа доступна по лицензии
Creative Commons
«Attribution-NonCommercial»