Инженерный вестник Дона, №8 (2019)
ivdon.ru/ru/magazine/archive/n8y2019/6137

Вероятностный подход к оценке отказоустойчивости различных моделей распределенного хранения данных

А.С. Назаров, М.А. Дерябин, М.Г. Бабенко, Е.О. Тарасенко

Северо-Кавказский федеральный университет, Ставрополь

Аннотация: Статья посвящена разработке вероятностного подхода к оценке отказоустойчивости основных моделей распределенного хранения данных. В работе приводится анализ причин сбоев в распределенных системах хранения и характеристика количественных показателей отказоустойчивости системы и ее компонентов. Так же приводится сравнительный анализ распределенных систем хранения данных, использующих репликацию, с системами, основанными на алгоритмах отказоустойчивого разделения данных.
Ключевые слова: распределенная система хранения данных, сбои, отказоустойчивость, избыточность, репликация, отказоустойчивое разделение данных, избыточная система остаточных классов.

Введение

Вопрос хранения данных является важнейшим вопросом компьютерной техники со времен создания первых компьютеров. С развитием вычислительной техники требовались все более емкие, производительные и надежные способы хранения данных, а исследования в этой области никогда не прекращались.

Объемы информации растут настолько стремительно, что единичные носители информации уже давно не справляются с нагрузкой. Каждую неделю Facebook требует дополнительных 60 терабайт (2^{40} байт) памяти только для новых фотографий [1]. Пользователи YouTube загружают более 400 часов видео каждую минуту, и каждый день требуется 1 петабайт (2^{50} байт) нового дополнительного дискового пространства [2, 3]. К 2025 году, по прогнозам аналитиков компании IDC (International Data Corporation), человечество сформирует 175 зеттабайтов (2^{70} байт) информации.

Необходимы масштабируемые решения, обладающие высокой отказоустойчивостью и производительностью. Для этих целей используются распределенные системы хранения данных (СХД), которые представляют

собой комплекс специализированного оборудования и программных средств, предназначенный для хранения больших объемов данных и доступа к ним.

Развитие СХД является актуальным вопросом в современной технике, связанным с тем фактом, что объемы хранимой и обрабатываемой информации постоянно возрастают, что приводит к увеличению потребности и стоимости хранения информации. Организации и ученые по всему миру стремятся не только развивать инфраструктуру хранения данных, но и исследовать возможности повышения эффективности СХД [4, 5]: снижения энергопотребления, расходов на сервис, общей стоимости владения и закупки систем резервного копирования и хранения с одной стороны и повышения отказоустойчивости, масштабируемости и производительности таких систем с другой.

Как было сказано выше, СХД зачастую испытывает высокую нагрузку, что влечет необходимость учета ряда ключевых требований к системам такого рода. В большинстве СХД предусмотрено полное или частичное резервирование всех компонент, от блоков питания до самих устройств хранения. Несмотря на значительные усилия, как в промышленности, так и науке, высокая отказоустойчивость остается серьезной проблемой в управлении крупномасштабными IT-системами, а предотвращение катастроф и стоимость устранения текущих поломок составляют значительную часть общей стоимости эксплуатации. Из-за увеличения количества серверов в кластерах сохранение высоких уровней отказоустойчивости и доступности является растущей проблемой для многих сайтов, высокопроизводительных вычислительных систем и провайдеров интернет-услуг. Отказоустойчивость систем хранения особенно важна по нескольким причинам. Во-первых, отказ хранилища может не только привести к временной недоступности данных, но в худшем случае это может привести к потере данных. Во-вторых, технологические тенденции и увеличение объема рынка могут в

совокупности привести к тому, что сбои системы хранения в будущем будут происходить чаще [6]. Наконец, размер современных систем хранения, крупномасштабных IT-установок вырос до беспрецедентного масштаба с тысячами устройств хранения данных, что приводит к тому, что сбои компонентов становятся нормой, а не исключением [7].

1 Анализ причин методов борьбы со сбоями в распределенных системах хранения данных

Распределенное хранилище данных представляет собой систему с большим количеством запоминающих устройств, соединенных сетью, и инфраструктурой, обеспечивающей их функционирование. Увеличение размеров современных распределенных систем хранения, содержащих тысячи устройств хранения данных, приводит к увеличению частоты сбоев ее компонентов [8].

Основными причинами сбоев данных в распределенных системах хранения являются аппаратные, программные, сетевые сбои и сбои питания [9]. Vishwanath и Nagappan проанализировали надежность оборудования для крупной инфраструктуры распределенной системы хранения [8]: 78% всех сбоев сервера были связаны с жесткими дисками, 5% – с RAID-контроллерами (Rapid Array of Inexpensive Disk – RAID), 3% – с памятью, остальные 14% обусловлены другими факторами. Диски являются центральным элементом распределенных систем хранения [10] и наиболее распространенным компонентом отказа [11]. Например, в 2009 году Facebook временно потерял более 10% сохраненных фотографий из-за сбоя жестких дисков [12]. При отказе диска, хранимые на нем данные становятся недоступны, что неприемлемо в современных условиях.

Как показано на рисунке 1, сбои могут привести к временной или постоянной недоступности данных. Недоступность данных из-за перебоев в работе сети, сбоя узла (машины), перебоя в питании или автоматического

процесса восстановления является временной и не приводит к постоянной потере данных [9]. Недоступность данных из-за сбоя жесткого диска или повреждения данных приводит к постоянной потере данных.

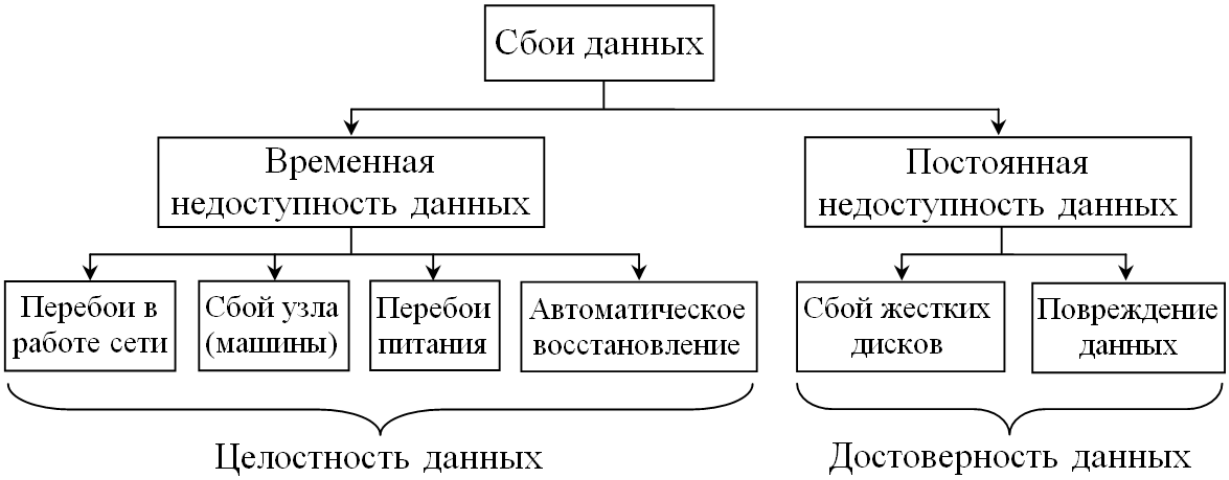


Рис. 1 – Сбои данных в распределенных системах хранения

В случае сбоев, приводящих к временной недоступности данных, недоступные данные могут быть рассмотрены как потерянные. Тогда для восстановления доступа к ним можно применить методы борьбы со сбоями, приводящими к постоянной недоступности данных. Таким образом, последствия любых сбоев, независимо от проблем их вызвавших, могут быть устранены с помощью различных методов введения избыточности данных [8], при этом реализация блоков прямого и обратного преобразования данных зависит от выбранных методов обеспечения отказоустойчивости. Коды стирания, репликация и отказоустойчивое разделение данных (Resilient Distributed Dataset – RDD) [13] являются наиболее важными методами обеспечения отказоустойчивости распределенных систем хранения данных.

Наиболее распространенными методами введения избыточности являются репликация [14] и коды стирания [15]. Репликация – это простой механизм резервирования данных. Одни и те же данные копируются и

хранятся в нескольких местах в системе хранения. Если запрошенные данные недоступны на одном диске, они подаются со следующего доступного диска [16]. Коды стирания являются более сложным механизмом введения избыточности данных. Наряду с исходными данными создаются и сохраняются данные о четности, так что если запрошенные данные недоступны, их можно восстановить из данных о четности. Расходы на хранение для кодов стирания намного меньше, чем для репликации, следовательно, это уменьшает потребность в оборудовании для хранения данных и обеспечивает значительную экономию затрат и энергии в центрах обработки данных [15]. Однако восстановление данных после сбоя связано с высокой стоимостью реконструкции и трафиком сети.

Повышение отказоустойчивости и снижение эксплуатационных расходов является основной причиной, по которой пользователи распределенных систем хранения заинтересованы в переходе к кодам стирания или отказоустойчивому разделению данных. Как было отмечено выше, использование для повышения отказоустойчивости репликации сопряжено с большой избыточностью, а использование кодов стирания связано с высокой стоимостью реконструкции и трафиком сети после сбоя, поэтому наиболее перспективным направлением является использование алгоритмов отказоустойчивого разделения данных, среди которых выделяется система остаточных классов.

Под системой остаточных классов [17] понимается непозиционная система счисления, в которой каждое число A представляется в виде набора из k остатков от деления α_i этого числа на числа p_i, входящие в набор модулей:

$$A = (\alpha_1, \alpha_2, ..., \alpha_k),\ \alpha_i = A \bmod p_i,\ i = 1,2,...,k.$$

Набор оснований $\{p_1, p_2, ..., p_k\}$ определяет конкретную СОК. Согласно Китайской теореме об остатках (КТО) [18], такое представление

для любого числа A из промежутка $[0,P_k)$, где $P_k = p_1 \cdot p_2 \cdot ... \cdot p_k$, уникально лишь в случае, если все p_i попарно взаимно просты, то есть НОД$(p_i, p_j) = 1$ для всех $i \neq j$, $i, j = 1,2,...,k$. Число P_k принято называть рабочим диапазоном представления чисел в СОК.

Система остаточных классов служит основой для кодов исправления ошибок. Добавив к системе оснований $\{p_1, p_2, ..., p_k\}$ избыточные модули $p_{k+1}, p_{k+2}, ..., p_n$ и расширив представление числа $A \in [0,P_k)$ остатками от деления на новые модули $\alpha_{k+1}, \alpha_{k+2}, ..., \alpha_n$, получаем избыточную СОК (ИСОК), которая приобретает новые свойства. Так, если $p_i < p_j$ для всех p_i, $i = 1,2,...,k$, и p_j, $j = k+1, k+2, ..., n$, то потеря любых $n-k$ остатков не нарушает возможности восстановить исходное число A, что аналогично идее кодов стирания и позволяет добиться доступности данных в условиях распределенного хранения. Отметим, что, во-первых, основания в полной системе оснований $p_1, p_2, .., p_n$ должны быть попарно взаимно-простыми и, во-вторых, A должно принадлежать промежутку $[0,P_k)$ и не принадлежать промежутку $[P_k, P)$, $P = P_k \cdot p_{k+1} \cdot p_{k+2} \cdot \ldots \cdot p_n$ есть полный диапазон ИСОК.

Важнейшей особенностью ИСОК является возможность контроля целостности информации [19]. При соблюдении описанных выше условий, накладываемых на полный диапазон ИСОК, можно обнаружить наличие искажений в остатках числа A, представленного в СОК [18, 19]. Для этого необходимо восстановить значение A по полной системе оснований ИСОК. Если при этом полученная величина $A^* \in [0,P_k)$, то полученное значение можно считать верным и полагать, что $A = A^*$. Иначе, если $A^* \geq P_k$, то в одном или нескольких основаниях произошло искажение. При этом, различные алгоритмы [20, 21] позволяют локализовать искажение, если оно произошло не более чем в $\lfloor (n-k)/2 \rfloor$ остатках.

2 Разработка вероятностного подхода к оценке отказоустойчивости распределенных систем хранения данных

В общем смысле отказоустойчивость выражается в способности технической системы сохранять свою работоспособность после отказа одного или нескольких ее составных компонентов, а значит напрямую зависит от их надежности.

Количественные показатели надежности определяются путем расчетов, проведением испытаний, статистической обработкой данных эксплуатации и математическим моделированием. При этом, расчеты показателей надежности должны производиться на этапе проектирования системы с целью прогнозирования ожидаемой надежности, что позволяет выбрать наиболее подходящий вариант архитектуры системы и подобрать методы обеспечения надежности.

Для сложных систем, к которым безусловно относятся распределенные системы хранения данных, и их составных частей, которые не могут быть отремонтированы и просто заменяются новыми, таких как жесткие диски, применяется термин «средняя наработка на отказ» (Mean Time To Failure, MTTF), которое определяется как среднее время, которое проработает устройство до того момента, как произойдет отказ. Другими словами, средняя наработка на отказ – это среднее время от начала одного сбоя до начала другого. Однако современные производители жестких дисков определяют надежность своих продуктов в виде двух связанных показателей: годовой коэффициент отказов (Annualized Failure Rate – AFR), который представляет собой процент дисковых накопителей в совокупности, потерпевших неудачу в тесте, масштабированный до оценки за год; и уже рассмотренная ранее «средняя наработка на отказ» (MTTF).

Для оценки отказоустойчивости распределенной системы хранения данных удобнее использовать показатель AFR, который по сути отражает

вероятность сбоя ее каждого из ее компонентов в течение года. Соотношение между AFR и MTTF (в часах) определяется следующим выражением [22]:

$$\mathrm{AFR} = 1 - e^{-8766/\mathrm{MTTF}}, \quad (1)$$

где 8766 – количество часов в году.

Как уже отмечалось, жесткие диски являются основным компонентом распределенных систем хранения данных и наиболее частой причиной отказов, поэтому отказоустойчивость распределенных систем хранения во многом зависит от надежности жестких дисков, на которых она строится.

Например, общей спецификацией для дисков PATA и SATA может быть MTTF 300 000 часов, что дает приблизительный теоретический годовой процент отказов равный 2.92%, то есть 2.92% вероятности выхода из строя данного диска в течение года использования. AFR будет увеличиваться к концу и после окончания срока службы устройства или компонента. Исследование Google, проведенное в 2007 году, показало, что фактические значения AFR для отдельных накопителей, основанные на большой выборке дисков, варьировались от 1.7% для накопителей первого года до более 8.6% для накопителей трехлетнего возраста [16]. Исследование CMU (Carnegie Mellon University) 2007 показало, что среднее значение AFR составляет 3% за 1-5 лет на основе журналов замены для большой выборки накопителей [23]. Рассмотрим подход к оценке отказоустойчивости распределенных систем хранения данных на основе показателя AFR накопителей, на которых строится система.

Для оценки отказоустойчивости систем используется схожий по смыслу с параметром AFR показатель вероятности отказа при запросе – PFD (Probability of Failure on Demand) [24]. PFD равен средней вероятности того, что система не выполнит свою функцию по запросу. Способ расчета данной характеристики зависит от функций, выполняемых системой и ее структуры. В случае распределенной системы хранения данных – PFD отражает

среднюю вероятность того, что при запросе доступа к файлу, файл не будет восстановлен или будет восстановлен, но данные будут повреждены.

Повышение отказоустойчивости влечет за собой один из вариантов резервирования и, следовательно, приводит к избыточности системы. Согласно ГОСТ 27.002-2015, избыточность системы может быть выражена значением кратности резервирования, которое представляет собой отношение числа резервных элементов к числу основных элементов:

$$\text{Redundancy} = \frac{I_{\text{Redundant}}}{I_{\text{Useful}}}. \quad (2)$$

Данный показатель очень важен в процессе проектирования системы, так как помимо обеспечения высокого уровня отказоустойчивости системы, проектируемая система должна отвечать требованиям экономической целесообразоности и эффективности.

2.1 Оценка отказоустойчивости распределенных систем хранения данных использующих репликацию

Рассмотрим модель распределенного хранения данных с резервированием, используемую в современных системах хранения данных. При таком способе хранения файл разрезается на части (Chunks) одинакового размера, затем создается несколько копий каждой части. Далее одинаковые копии записываются на разные физические постоянные запоминающие устройства (ПЗУ) – жесткие диски (HDD – Hard Disk Drive), чтобы при поломке одного из дисков, была возможность использовать другой, для восстановления файла. Количество копий каждой части называется фактором репликации (Replication Factor). Фактор репликации является настраиваемым параметром, определяемым для каждого файла. Все метаданные о расположении копий, работающих и отказавших жестких дисках, хранятся на главном сервере – центральном узле (Single Master), который координирует доступ к частям восстанавливаемого файла.

Использование показателя AFR жестких дисков, на которых построена распределенная система хранения данных, позволяет применить вероятностный подход для расчета аналогичного показателя PFD для распределенной системы хранения данных.

Для расчета показателя PFD – вероятности отказа системы при запросе в течение года (т.е. вероятности, что пользователь не сможет восстановить данные в какой-либо момент в течение года) используем рассмотренный выше показатель AFR – вероятность выхода из строя одного жесткого диска в течение года, и введем следующие обозначения: *RF* (Replication Factor) – фактор репликации, *CC* (Chunks Count) – количество частей, получившееся после разрезания файла. При необходимости расчета вероятности отказа в течение другого временного промежутка эксплуатации T_0, цифра 8766 часов в формуле (1) заменяется на T_0. Отметим, что наличие доступа к данным, хранящимся на жестком диске, не гарантирует их восстановления, так как данные могут быть повреждены, при этом жесткий диск будет функционировать в обычном режиме. Так же данные могут быть искажены в процессе передачи по каналам связи. В связи с этим необходимо ввести дополнительный параметр *er* – вероятность искажения данных. Показатель *er* – вероятность искажения данных, зависит от условий эксплуатации и устанавливается статистически для каждой конкретной распределенной системы хранения данных. На рисунке 2 представлена обобщенная схема распределенного хранения данных, соответствующая введенным обозначениям.

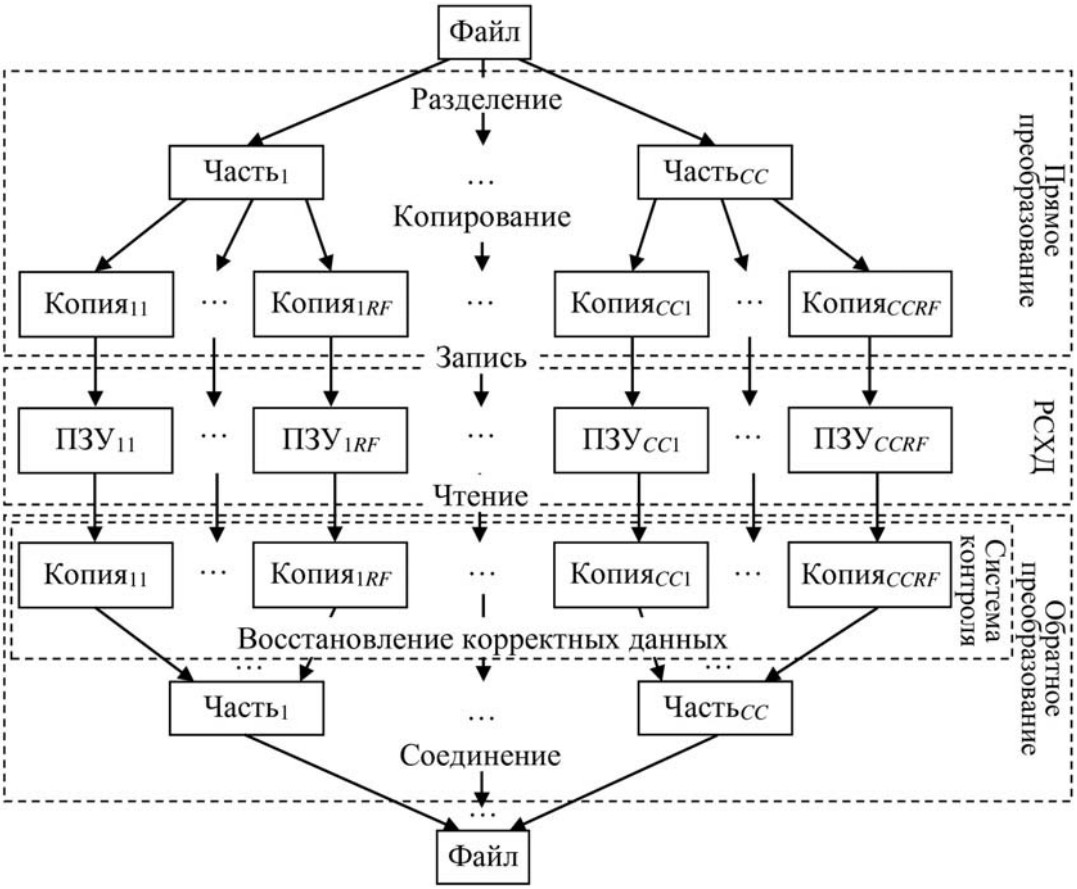


Рис. 2 – Обобщенная схема распределенного хранения данных с использованием резервирования

Обозначим $\overline{\text{PFD}}$ вероятность безотказной работы в течение года, тогда:

$$\overline{\text{PFD}} = 1 - \text{PFD} . \quad (3)$$

Пользователь сможет восстановить файл в том случае, если удастся восстановить каждую из частей файла. Обозначим вероятность того, что одна из частей будет восстановлена $\overline{\text{PFD}}_{CC}$. Тогда, учитывая, что части имеют одинаковый размер и создается одинаковое количество копий каждой из них, вероятность $\overline{\text{PFD}}$ того, что файл будет восстановлен, равна вероятности того, что одновременно будут восстановлены все части:

$$\overline{\text{PFD}} = \overline{\text{PFD}}_{CC}^{CC} . \quad (4)$$

Если имеется RF копий одной части, то для ее восстановления достаточно наличие доступа хотя бы к одному из жестких дисков, содержащих копию данной части при условии, что данные не повреждены. Таким образом, вероятность $\overline{\text{PFD}}_{CC}$ восстановления одной части равна сумме вероятностей того, что имеется доступ ровно к 1, 2, …, RF любым жестким дискам, содержащим копии этой части при условии, что данные на большинстве дисков в каждом из случаев корректные.

$$\overline{\text{PFD}}_{CC} = C_{RF}^{1}\overline{\text{PFD}}_1 + C_{RF}^{2}\overline{\text{PFD}}_2 + ... + C_{RF}^{RF}\overline{\text{PFD}}_{RF} = \sum_{j=1}^{RF}\left(C_{RF}^{j}\overline{\text{PFD}}_j\right), \quad (5)$$

где $\overline{\text{PFD}}_j$ – вероятность, что часть будет восстановлена при наличии доступа ровно к j жестким дискам, содержащим ее копии, с учетом вероятности их искажения.

При резервировании, корректность данных устанавливается по мажоритарному принципу, т.е. большинством. Например, если имеется доступ к трем жестким дискам и на двух из них хранятся одинаковые копии, а копия на третьем диске отличается, то делаем вывод, что копия на третьем диске повреждена и используем корректную копию с первого или второго жесткого диска. Если, например, имеется доступ к двум жестким дискам и копия, хранящаяся на одном из них, отличается от копии, хранящейся на другом, то делаем вывод, что данные не могут быть восстановлены корректно. Обобщая все вышесказанное, данные должны быть одновременно доступны и корректны, тогда вероятность $\overline{\text{PFD}}_j$ восстановления одной части при наличии доступа ровно к j жестким дискам, содержащим ее копии, равна условной вероятности:

$$\overline{\text{PFD}}_j = \overline{\text{PFD}}_{j_A} \cdot \overline{\text{PFD}}_{j_{A|R}}, \quad (6)$$

где $\overline{\text{PFD}}_{j_A}$ – вероятность того, что имеется доступ ровно к j жестким дискам, содержащим копии восстанавливаемой части, $\overline{\text{PFD}}_{j_{A|R}}$ – вероятность того, что

часть, восстановленная по данным с j жестких дисков, содержащих копии восстанавливаемой части, будет корректна. Вероятность является условной, так как вероятность восстановления корректных данных устанавливается исходя из того, что доступны ровно j из RF жестких дисков, содержащих копии восстанавливаемой части. Иными словами, нет необходимости учитывать корректность копий, к которым нет доступа. Тогда вероятность того, что доступны ровно j жестких дисков, содержащих копии восстанавливаемой части равна:

$$\overline{\mathrm{PFD}}_{j_A} = (1-\mathrm{AFR})^j \,\mathrm{AFR}^{RF-j}, \tag{7}$$

а соответствующая ей вероятность корректного восстановления части равна:

$$\overline{\mathrm{PFD}}_{j_{A|R}} = \sum_{i=\left\lfloor \frac{j}{2} \right\rfloor+1}^{j} C_j^i (1-er)^i \, er^{j-i}. \tag{8}$$

Подставляя (7) и (8) в (6), а (6) в (5), получим формулу для вычисления $\overline{\mathrm{PFD}}_{CC}$ – вероятности восстановления одной части файла:

$$\overline{\mathrm{PFD}}_{CC} = \sum_{j=1}^{RF} \left(C_{RF}^j \cdot (1-\mathrm{AFR})^j \,\mathrm{AFR}^{RF-j} \cdot \sum_{i=\left\lfloor \frac{j}{2} \right\rfloor+1}^{j} C_j^i (1-er)^i \, er^{j-i} \right). \tag{9}$$

Подставляя (9) в (4), получим формулу для вычисления $\overline{\mathrm{PFD}}$ – вероятности восстановления файла:

$$\overline{\mathrm{PFD}} = \left(\sum_{j=1}^{RF} \left(C_{RF}^j \cdot (1-\mathrm{AFR})^j \,\mathrm{AFR}^{RF-j} \cdot \sum_{i=\left\lfloor \frac{j}{2} \right\rfloor+1}^{j} C_j^i (1-er)^i \, er^{j-i} \right) \right)^{CC}. \tag{10}$$

И окончательно, с четом (3):

$$\mathrm{PFD} = 1 - \left(\sum_{j=1}^{RF} \left(C_{RF}^j \cdot (1-\mathrm{AFR})^j \,\mathrm{AFR}^{RF-j} \cdot \sum_{i=\left\lfloor \frac{j}{2} \right\rfloor+1}^{j} C_j^i (1-er)^i \, er^{j-i} \right) \right)^{CC}. \tag{11}$$

Для распределенной системы хранения избыточность основных компонент напрямую зависит от количества формируемых избыточных данных. Преобразуем формулу (2) для вычисления избыточности следующим образом:

$$\text{Redundancy} = \frac{I_{\text{Full}} - I_{\text{Useful}}}{I_{\text{Useful}}} = \frac{I_{\text{Full}}}{I_{\text{Useful}}} - 1.$$

где I_{Full} – полный объем данных, I_{Useful} – полезный объем данных.

Для удобства использования коэффициент избыточности может быть выражен в процентах. Тогда в терминах, введенных для распределенной системы хранения данных с репликацией, избыточность может быть найдена по формуле:

$$\text{Redundancy} = (RF - 1) \cdot 100\%. \quad (12)$$

Даже незначительное снижение вероятности безотказной работы жестких дисков, может привести к существенному снижению вероятности восстановления информации, хранящейся в распределенной системе хранения данных. Выходом в подобной ситуации может стать увеличение фактора репликации, т.е. количества копий каждой из частей файла, либо замена ненадежных запоминающих устройств.

2.2 Оценка отказоустойчивости распределенных систем хранения данных на основе избыточной системы остаточных классов

Модель распределенного хранения данных с использованием разделения данных, так же, как и модель с использованием репликации, предполагает разрезание файла на части одинакового размера. Принципиальное отличие схем с репликацией от схем с разделением данных заключается в реализации блоков прямого и обратного преобразования. При использовании разделения данных создаются не копии каждой части, а формируются n подчастей, таким образом, что, имея доступ к k подчастям

($k < n$), пользователь может восстановить всю часть (Chunk). Алгоритмы формирования подчастей могут различаться в зависимости от используемой схемы разделения данных (коды стирания [16], алгоритм разделения данных Рабина [25], избыточная система остаточных классов [26]). Оценим отказоустойчивость распределенной системы хранения на основе избыточной системы остаточных классов, отметим при этом, что предлагаемый подход справедлив для любых алгоритмов разделения данных.

При использовании разделения данных на основе избыточной системы остаточных классов (ИСОК) формируются n остатков, таким образом, что, имея доступ к k остаткам ($k < n$), пользователь может восстановить всю часть (Chunk). Далее n остатков одной части записываются на разные физические постоянные запоминающие устройства (ПЗУ) – жесткие диски (HDD – Hard Disk Drive), чтобы при поломке одного из дисков, была возможность использовать другой, для восстановления данных. Все метаданные о расположении остатков, работающих и отказавших жестких дисках, как и в случае с репликацией хранятся на центральном узле (Single Master), который координирует доступ к остаткам восстанавливаемого файла. Общее количество остатков n и количество остатков достаточное для восстановления k, может быть различным в зависимости от требований к конкретной системе хранения данных. Схема разделения данных обозначается (k, n)-ИСОК. Размер каждого остатка при этом в k раз меньше размера части, тогда как при репликации каждая копия имеет такой же размер, как и сама часть, что позволяет либо существенно снизить избыточность без ущерба для надежности, сократив тем самым эксплуатационные затраты распределенной системы хранения данных, либо существенно повысить отказоустойчивость при таком же уровне избыточности, как в случае с использованием репликации.

Аналогично тому, как это было сделано в пункте 2.1, рассчитаем показатель PFD – вероятности отказа системы на основе ИСОК при запросе в течение года. Как и в случае с репликацией, вероятности отказа может быть рассчитана для любого другого временного промежутка эксплуатации.

Для расчета показателя PFD – вероятности отказа системы при запросе в течение года (т.е. вероятности, что пользователь не сможет восстановить данные в какой-либо момент в течение года), используем рассмотренный выше показатель AFR – вероятность выхода из строя одного жесткого диска в течение года, и введем следующие обозначения: n – общее количество остатков, формируемое для каждой части (Chunk), k – количество остатков, достаточное для восстановления части (Chunk), CC (Chunks Count) – количество частей, получившееся после разрезания файла. Так же как и в случае с репликацией, наличие доступа к данным, хранящимся на жестком диске, не гарантирует их восстановления, так как данные могут быть повреждены, поэтому необходимо ввести дополнительный параметр er – вероятность искажения данных. На рисунке 3 представлена обобщенная схема распределенного хранения с использованием ИСОК, соответствующая введенным обозначениям.

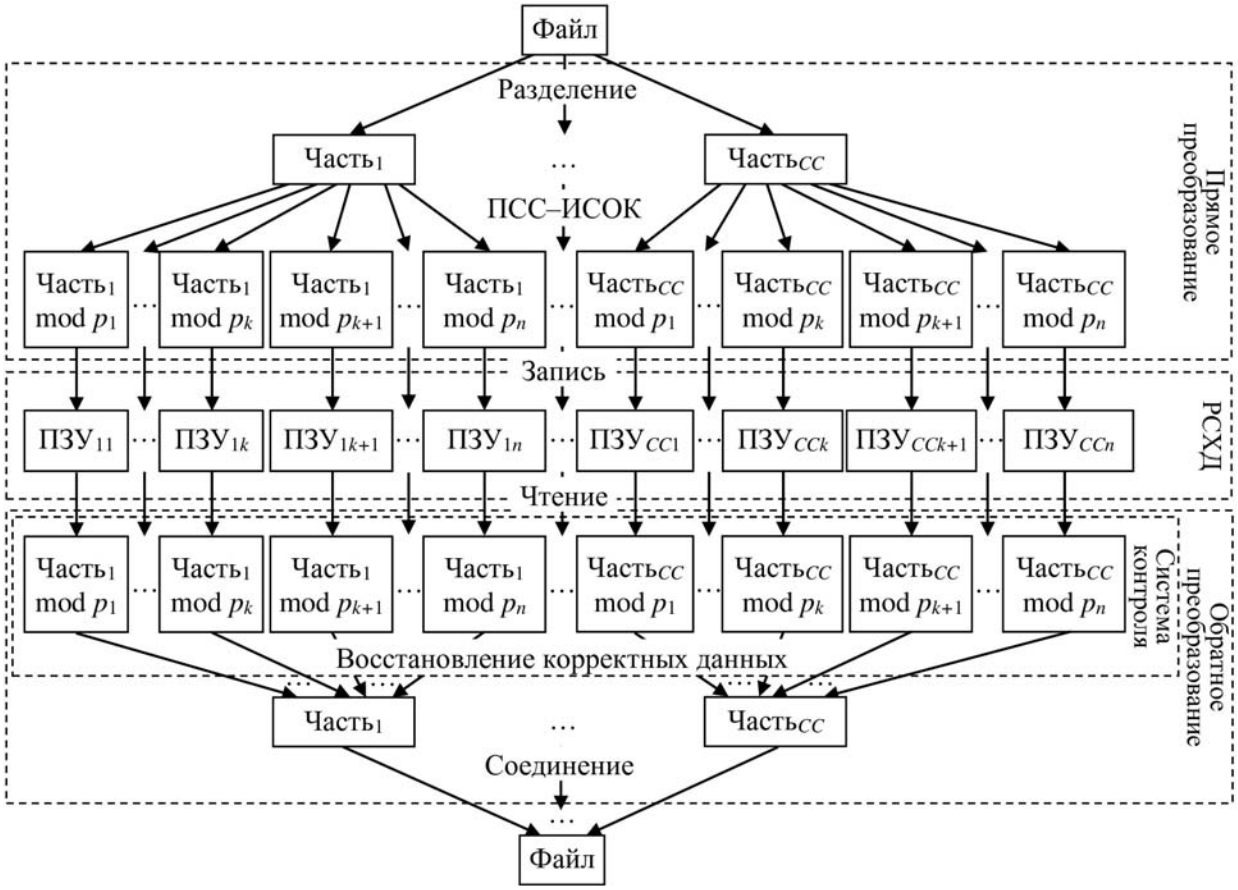


Рис. 3 – Обобщенная схема распределенного хранения данных с использованием избыточной системы остаточных классов

По аналогии с пунктом 2.1, обозначим $\overline{\text{PFD}}$ вероятность безотказной работы в течение года, формула (3). Как и в случае с репликацией, пользователь сможет восстановить файл в том случае, если сможет восстановить все части, тогда вероятность $\overline{\text{PFD}}$ того, что файл будет восстановлен, вычисляется по формуле (4).

Если для повышения отказоустойчивости используется (k, n)-схема разделения данных, то для восстановления одной части достаточно наличие доступа хотя бы к k жестким дискам, содержащим остатки данной части при условии, что данные не повреждены. Таким образом, вероятность $\overline{\text{PFD}}_{CC}$ восстановления одной части равна сумме вероятностей того, что имеется доступ ровно к k, $k+1$, $k+2$, …, n жестким дискам, содержащим остатки

этой части при условии, что количество дисков, содержащих искаженные данные, не превышает кратности ошибок, которые можно исправить с помощью доступных остатков (пункт 1).

$$\overline{\mathrm{PFD}}_{CC} = C_n^k \overline{\mathrm{PFD}}_k + C_n^{k+1} \overline{\mathrm{PFD}}_{k+1} + \ldots + C_n^n \overline{\mathrm{PFD}}_n = \sum_{j=k}^{n} \left(C_n^j \overline{\mathrm{PFD}}_j \right), \qquad (13)$$

где $\overline{\mathrm{PFD}}_j$ – вероятность, что часть будет восстановлена при наличии доступа ровно к j жестким дискам, содержащим ее остатки, с учетом вероятности их искажения.

При использовании (k, n)-ИСОК восстановленные данные будут корректны, если искажены не более $\lfloor (n-k)/2 \rfloor$ остатков [19]. Например, если используется $(2, 6)$-схема разделения данных и имеется доступ ко всем 6 жестким дискам, то данные будут восстановлены корректно при наличии искажений не более чем в двух остатках. Если, например, доступны только 4 или 5 жестких дисков из 6, то данные будут восстановлены корректно при наличии искажений не более чем в одном остатке. Тогда, как и в случае с репликацией, вероятность $\overline{\mathrm{PFD}}_j$ восстановления одной части при наличии доступа ровно к j жестким дискам, содержащим ее остатки, равна условной вероятности:

$$\overline{\mathrm{PFD}}_j = \overline{\mathrm{PFD}}_{j_A} \cdot \overline{\mathrm{PFD}}_{j_{A|R}}, \qquad (14)$$

где $\overline{\mathrm{PFD}}_{j_A}$ – вероятность того, что имеется доступ ровно к j жестким дискам, содержащим остатки восстанавливаемой части, $\overline{\mathrm{PFD}}_{j_{A|R}}$ – вероятность того, что часть, восстановленная по данным с j жестких дисков, содержащих остатки восстанавливаемой части, будет корректна. Тогда вероятность того, что доступны ровно j жестких дисков, содержащих остатки восстанавливаемой части равна:

$$\overline{\mathrm{PFD}}_{j_A} = (1 - \mathrm{AFR})^j \, \mathrm{AFR}^{n-j}. \qquad (15)$$

При условии, что доступны только j из n дисков, содержащих остатки восстанавливаемой части, (k, n)-ИСОК становится (k, j)-ИСОК, и ее корректирующая способность уменьшается до величины $\lfloor (j-k)/2 \rfloor$, тогда вероятность корректного восстановления части равна:

$$\overline{\mathrm{PFD}}_{j_{A|R}} = \sum_{i=j-\left\lfloor \frac{j-k}{2} \right\rfloor}^{j} C_j^i (1-er)^i \, er^{j-i}. \qquad (16)$$

Подставляя (15) и (16) в (14), а (14) в (13), получим формулу для вычисления $\overline{\mathrm{PFD}}_{CC}$ – вероятности восстановления одной части файла:

$$\overline{\mathrm{PFD}}_{CC} = \sum_{j=k}^{n} \left(C_n^j \cdot (1-\mathrm{AFR})^j \, \mathrm{AFR}^{n-j} \cdot \sum_{i=j-\left\lfloor \frac{j-k}{2} \right\rfloor}^{j} C_j^i (1-er)^i \, er^{j-i} \right), \qquad (17)$$

Подставляя (17) в (4), получим формулу для вычисления $\overline{\mathrm{PFD}}$ – вероятности восстановления файла:

$$\overline{\mathrm{PFD}} = \left(\sum_{j=k}^{n} \left(C_n^j \cdot (1-\mathrm{AFR})^j \, \mathrm{AFR}^{n-j} \cdot \sum_{i=j-\left\lfloor \frac{j-k}{2} \right\rfloor}^{j} C_j^i (1-er)^i \, er^{j-i} \right) \right)^{CC}. \qquad (18)$$

И окончательно, с четом (3):

$$\mathrm{PFD} = 1 - \left(\sum_{j=k}^{n} \left(C_n^j \cdot (1-\mathrm{AFR})^j \, \mathrm{AFR}^{n-j} \cdot \sum_{i=j-\left\lfloor \frac{j-k}{2} \right\rfloor}^{j} C_j^i (1-er)^i \, er^{j-i} \right) \right)^{CC}. \qquad (19)$$

Как уже отмечалось, в общем случае избыточность может быть рассчитана по формуле (2), которая может быть записана в следующем виде:

$$\mathrm{Redundancy} = \frac{I_{\mathrm{Full}}}{I_{\mathrm{Useful}}} - 1.$$

где I_{Full} – полный объем данных, I_{Useful} – полезный объем данных.

Для удобства использования коэффициент избыточности может быть выражен в процентах. Тогда в терминах, введенных для распределенной системы хранения данных с использованием схем разделения данных, избыточность может быть найдена по формуле:

$$\text{Redundancy} \approx \left(\frac{n}{k} - 1 \right) \cdot 100\%. \quad (20)$$

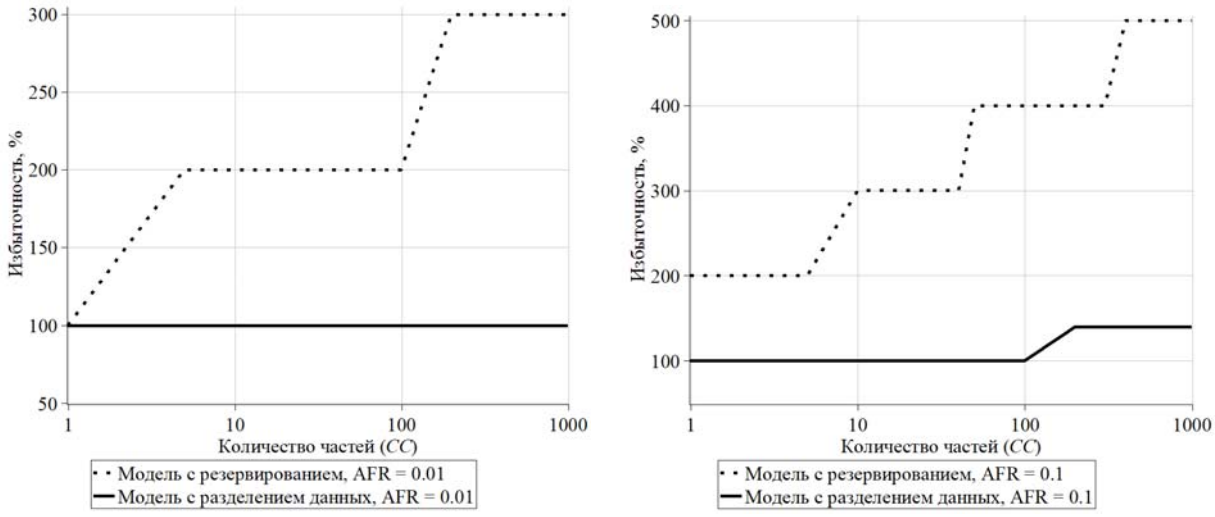
Замечание. Знак «≈» в формуле (20) объясняется тем, что для некоторых схем разделения данных невозможно добиться того, чтобы суммарная разрядность k подчастей, минимально необходимых для восстановления части, была равна разрядности самой части. Примером может служить избыточная система остаточных классов, имеющая ограничения, накладываемые на ее модули. Учитывая, что модули избыточной системы остаточных классов должны быть взаимнопростыми, а избыточные модули должны быть больше каждого из рабочих, практически невозможно подобрать их таким образом, чтобы они точно перекрывали рабочий диапазон. Модули избыточной системы остаточных классов всегда будут перекрывать диапазон больше необходимого. Однако, за счет подбора модулей всегда можно минимизировать превышение необходимого диапазона и формула (20) может быть использована для расчета избыточности и в таких случаях.

3 Сравнительный анализ моделей распределенных систем хранения данных

В пункте 2 был предложен вероятностный подход к оценке отказоустойчивости распределенных систем хранения данных, основанных на репликации и использовании схем разделения данных. С одной стороны, использование схем разделения данных позволяет повысить отказоустойчивость распределенных систем хранения данных по сравнению с репликацией при одинаковом уровне избыточности, с другой стороны,

позволяет снизить избыточность при одинаковом уровне отказоустойчивости.

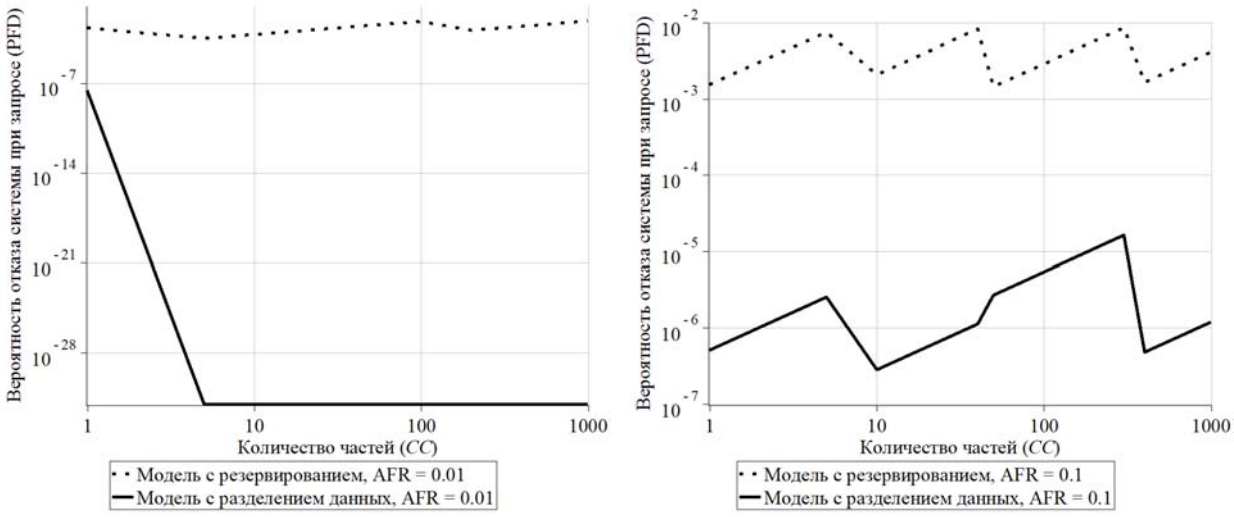
Если приоритетом при проектировании распределенной системы хранения данных является снижение эксплуатационных затрат, то схемы разделения данных подбираются таким образом, чтобы максимально снизить избыточность с сохранением приемлемого уровня отказоустойчивости. На рисунке 4 приведен график зависимости избыточности данных при использовании репликации и схем разделения данных в зависимости от размера хранимых файлов и годового показателя AFR жестких дисков, на которых строится распределенная система хранения данных. Вероятность искажения данных для обоих случаев равна $er = 0.001$. Фактор репликации и схема разделения данных подобраны таким образом, чтобы обеспечить приемлемую отказоустойчивость распределенной системы хранения данных: $PFD \leq 10^{-2}$.



а *б*

Рис. 4 – Избыточность распределенных систем хранения данных с использованием репликации и схем разделения данных при $er = 0.001$, $PFD \leq 10^{-2}$: (*а*) AFR = 0.01; (*б*) AFR = 0.1

Если приоритетом при проектировании распределенной системы хранения данных является повышение отказоустойчивости, то схемы разделения данных подбираются таким образом, чтобы повысить отказоустойчивость с сохранением того же уровня избыточности, что и при репликации. На рисунке 5 приведен график зависимости показателя PFD распределенных систем хранения данных с использованием репликации и схем разделения данных в зависимости от размера хранимых файлов и годового показателя AFR жестких дисков, на которых строится система. Вероятность искажения данных в обоих случаях равна $er = 0.001$. Фактор репликации подобран таким образом, чтобы обеспечить приемлемую отказоустойчивость распределенной системы хранения данных: $PFD \le 10^{-2}$. Схема разделения данных подбирается так, чтобы ее избыточность была равна избыточности при репликации, обеспечивающей необходимый уровень отказоустойчивости.



а *б*

Рис. 5 – Отказоустойчивость распределенных систем хранения данных с использованием репликации и схем разделения данных при $er = 0.001$, $PFD \le 10^{-2}$: (*а*) AFR = 0.01; (*б*) AFR = 0.1

Полученные в ходе сравнительного анализа данные показали, что использование алгоритмов разделения данных для снижения избыточности позволяет в среднем в 3.36 раза уменьшить избыточность систем, построенных на жестких дисках с высоким годовым показателем вероятности выхода из строя AFR = 0.1 и в среднем в 2.38 раза уменьшить избыточность систем, построенных на жестких дисках с низким годовым показателем вероятности выхода из строя AFR = 0.01 по сравнению с репликацией при одинаковом уровне отказоустойчивости сравниваемых распределенных систем хранения данных (рис. 4). При увеличении размеров хранимого файла, выигрыш в избыточности при использовании схем разделения данных по сравнению с резервированием возрастает. Отметим, что даже снижение избыточности в два раза позволяет существенно сократить эксплуатационные затраты в случае крупномасштабных распределенных систем хранения.

Использование алгоритмов разделения данных для повышения отказоустойчивости распределенных систем хранения данных позволяет снизить вероятность отказа системы при запросе (PFD) в течение года в среднем на 3 порядка, с $3.6 \cdot 10^{-3}$ до $3 \cdot 10^{-6}$ для систем, построенных на жестких дисках с высоким годовым показателем вероятности выхода из строя AFR = 0.1 по сравнению с репликацией. При использовании схем разделения данных в распределенных системах хранения, построенных на жестких дисках с годовым показателем вероятности выхода из строя AFR = 0.01 и ниже, вероятность отказа системы при запросе (PFD) в течение года стремится к нулю при одинаковом уровне избыточности сравниваемых распределенных систем хранения данных (рис. 5).

Предложенные методы расчета отказоустойчивости и избыточности могут быть использованы для расчета параметров распределенной сети хранения данных, исходя из требований, предъявляемых к ее

отказоустойчивости, общего объема памяти и объема хранимых полезных данных.

Заключение

В работе рассмотрены структуры распределенного хранения данных на основе репликации и алгоритмов отказоустойчивого разделения данных. Разработан вероятностный подход к оценке параметров различных моделей распределенного хранения данных, позволяющий на этапе проектирования получить оценку отказоустойчивости распределенной системы хранения данных, исходя из надежности ее компонентов и предполагаемых условий эксплуатации, с учетом ее архитектурных особенностей. Установлено, что использование алгоритмов отказоустойчивого разделения данных позволяет многократно повысить отказоустойчивость и/или снизить избыточность распределенных систем хранения данных.

Благодарности

Данная работа была выполнена при поддержке грантов Президента Российской Федерации №МК-6294.2018.9 и №МК-341.2019.9, грантов РФФИ №18-07-00109, №19-07-00130, стипендии Президента Российской Федерации СП-1685.2019.5.

Литература

1. Beaver D., Kumar S., Li H. [et al.] Finding a Needle in Haystack: Facebook's Photo Storage // Operating Systems Design and Implementation (OSDI): Proceedings of USENIX Symposium. Vancouver, British Columbia, Canada: ACM Press, 2010. Vol. 10. No. 2010. pp. 1-8.

2. Abu-Libdeh H., Princehouse L., Weatherspoon H. RACS: A case for cloud storage diversity // Cloud Computing: Proceedings of the 1st ACM Symposium, SoCC'10. Indianapolis, Indiana, USA: ACM Press, 2010. pp. 229-239.

3. Nachiappan R., Javadi R., Calheiros R.N. [et al.] Cloud storage reliability for Big Data applications: A state of the art survey // Journal of Network and Computer Applications. 2017. Vol. 97. pp. 35-47.

4. Система хранения данных. TAdviser. URL: tadviser.ru/index.php/Статья:Система_хранения_данных.

5. Пономарев В.А. Математические модели производительности, надежности и стоимости функционирования системы хранения дедуплицированных данных на SSD-дисках // Инженерный вестник Дона. 2019. №6. URL: ivdon.ru/ru/magazine/archive/n6y2019/6012.

6. Prabhakaran V., Bairavasundaram L.N., Agrawal N. [et al.] IRON file systems // Operating systems principles: Proceedings of the 20th ACM symposium, SOSP'05. New York City, New York, USA: ACM Press, 2005. Vol. 39. No. 5. pp. 206-220.

7. Ghemawat S., Gobioff H., Leung S.-T. The Google file system // Operating Systems Principles: Proceedings of the 19th ACM Symposium, SOSP'03. Bolton Landing, New York, USA: ACM Press, 2003. Vol. 37. No. 5. pp. 29-43.

8. Tchernykh A.N., Chervyakov N.I., Nazarov A.S. [et al.] An Approach for Mitigating Cloud Computing Uncertainty by Modular Data Encryption // Engineering and Telecommunication (En&T-2016): Proceedings of the III International Conference. Moscow, Russia: MIPT, 2016. pp. 62-64.

9. Rajasekharan A. Data Reliability in Highly Fault-tolerant Cloud Systems // Seagate Point of View. 2014. URL: seagate.com/files/www-content/_shared/_masters/category-info/data-reliability-fault-tolerant-cloud-pv0031-1-1410-us.pdf.

10. Brewer E., Ying L., Greenfield L. [et al.] Disks for Data Centers // Write Paper from Google. 2016. pp. 1-16.

11. Hughes G.F., Murray J.F., Kreutz-Delgado K. [et al.] Improved Disk-Drive Failure Warnings // IEEE Transactions on Reliability. 2002. Vol. 51. No. 3. pp. 350-357.

12. Gunawi H.S., Do T., Joshi P. [et al.] Data Reliability in Highly Fault-tolerant Cloud Systems. Hot Dep // Usenix: The Advanced Computing System Association. 2010. URL: usenix.org/events/hotdep10/tech/full_papers/Gunawi.pdf.

13. Zaharia M. Chowdhury M., Franklin M.J. [et al.] Spark: Cluster computing with working sets // Hot topics in cloud computing: Proceedings of the 2nd USENIX Conference, Hot Cloud'10. Boston, Massachusetts, USA: ACM Press, 2010. pp. 10-10.

14. Li W., Yang Y., Yuan D. A Novel Cost-Effective Dynamic Data Replication Strategy for Reliability in Cloud Data Centres // Dependable, Autonomic and Secure Computing: Proceedings of the 2011 IEEE Ninth International Conference. Sydney, Australia: IEEE, 2011. pp. 496-502.

15. Huang C., Simitci H., Xu Y. [et al.] Erasure coding in windows azure storage // Annual Technical Conference: Proceedings of USENIX Conference. Boston, Massachusetts, USA: USENIX Association, 2012. pp. 15-26.

16. Plank, J.S. Erasure codes for storage systems: A brief primer // USENIX Magazine. 2013. Vol. 38. pp. 44-51.

17. Эрдниева Н.С. Использование системы остаточных классов для маломощных приложений цифровой обработки сигналов // Инженерный вестник Дона. 2013. №2. URL: ivdon.ru/ru/magazine/archive/n2y2013/1621

18. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. М.: Советское радио, 1968. – 440 с.

19. Ding C., Pei D., Salomaa A. Chinese remainder theorem: applications in computing, coding, cryptography. Singapore: World Scientific, 1996. 214 p.

20. Goh V.T., Siddiqi M.U. Multiple error detection and correction based on redundant residue number systems // IEEE Transactions on Communications. 2008. Vol. 56. No. 3. pp. 325-330.

21. Червяков Н.И., Шапошников А.В., Рядов С.А. Модулярные параллельные вычислительные структуры нейропроцессорных систем. М.: Физматлит, 2002. 288 с.

22. Singleton, R. Maximum distance q-nary codes // IEEE Transactions on Information Theory. 1964. Vol. 10. No. 2. pp. 116-118.

23. Schroeder B., Gibson G.A. Understanding disk failure rates: What does an MTTF of 1,000,000 hours mean to you? // ACM Transactions on Storage (TOS). 2007. Vol. 3. № 3. pp. 8.

24. Braband J., VomHövel R., Schäbe H. Probability of failure on demand: The why and the how // Computer Safety, Reliability, and Security: Proceedings of International Conference. Berlin, Germany: Springer, 2009. pp. 46-54.

25. Rabin M.O. Efficient dispersal of information for security, load balancing, and fault tolerance // Journal of the ACM. 1989. Vol. 36. № 2. pp. 335-348.

26. Червяков Н.И., Бережной В.В., Назаров А.С. [и др.] Метод анализа корректирующих способностей кода системы остаточных классов // Сборник тезисов Международной конференции по мягким вычислениям и измерениям. С.-Петербург: Изд-во Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» им. В.И. Ульянова (Ленина), 2018. С. 372-375.

References

1. Beaver D., Kumar S., Li H. [et al.] Finding a Needle in Haystack: Facebook's Photo Storage. Operating Systems Design and Implementation (OSDI): Proceedings of USENIX Symposium. Vancouver, British Columbia, Canada: ACM Press, 2010. Vol. 10. No. 2010. pp. 1-8.

2. Abu-Libdeh H., Princehouse L., Weatherspoon H. RACS: A case for cloud storage diversity. Cloud Computing: Proceedings of the 1st ACM Symposium, SoCC'10. Indianapolis, Indiana, USA: ACM Press, 2010. pp. 229-239.

3. Nachiappan R., Javadi R., Calheiros R.N. [et al.] Cloud storage reliability for Big Data applications: A state of the art survey. Journal of Network and Computer Applications. 2017. Vol. 97. pp. 35-47.

4. Sistema khraneniya dannykh. TAdviser. URL: tadviser.ru/index.php/Stat'ya:Sistema_khraneniya_dannykh.

5. Ponomarev V.A. Inzhenernyj vestnik Dona. 2019. №6. URL: ivdon.ru/ru/magazine/archive/n6y2019/6012.

6. Prabhakaran V., Bairavasundaram L.N., Agrawal N. [et al.] IRON file systems. Operating systems principles: Proceedings of the 20th ACM symposium, SOSP'05. New York City, New York, USA: ACM Press, 2005. Vol. 39. No. 5. pp. 206-220.

7. Ghemawat S., Gobioff H., Leung S.-T. The Google file system. Operating Systems Principles: Proceedings of the 19th ACM Symposium, SOSP'03. Bolton Landing, New York, USA: ACM Press, 2003. Vol. 37. No. 5. pp. 29-43.

8. Tchernykh A.N., Chervyakov N.I., Nazarov A.S. [et al.] An Approach for Mitigating Cloud Computing Uncertainty by Modular Data Encryption. Engineering and Telecommunication (En&T-2016): Proceedings of the III International Conference. Moscow, Russia: MIPT, 2016. pp. 62-64.

9. Rajasekharan A. Data Reliability in Highly Fault-tolerant Cloud Systems. Seagate Point of View. 2014. URL: seagate.com/files/www-content/_shared/_masters/category-info/data-reliability-fault-tolerant-cloud-pv0031-1-1410-us.pdf.

10. Brewer E., Ying L., Greenfield L. [et al.] Disks for Data Centers. Write Paper from Google. 2016. pp. 1-16.

11. Hughes G.F., Murray J.F., Kreutz-Delgado K. [et al.] Improved Disk-Drive Failure Warnings. IEEE Transactions on Reliability. 2002. Vol. 51. No. 3. pp. 350-357.

12. Gunawi H.S., Do T., Joshi P. [et al.] Data Reliability in Highly Fault-tolerant Cloud Systems. Hot Dep. Usenix: The Advanced Computing System Association. 2010. URL: usenix.org/events/hotdep10/tech/full_papers/Gunawi.pdf.

13. Zaharia M. Chowdhury M., Franklin M.J. [et al.] Spark: Cluster computing with working sets. Hot topics in cloud computing: Proceedings of the 2nd USENIX Conference, Hot Cloud'10. Boston, Massachusetts, USA: ACM Press, 2010. pp. 10-10.

14. Li W., Yang Y., Yuan D. A Novel Cost-Effective Dynamic Data Replication Strategy for Reliability in Cloud Data Centres. Dependable, Autonomic and Secure Computing: Proceedings of the 2011 IEEE Ninth International Conference. Sydney, Australia: IEEE, 2011. pp. 496-502.

15. Huang C., Simitci H., Xu Y. [et al.] Erasure coding in windows azure storage. Annual Technical Conference: Proceedings of USENIX Conference. Boston, Massachusetts, USA: USENIX Association, 2012. pp. 15-26.

16. Plank, J.S. Erasure codes for storage systems: A brief primer. USENIX Magazine. 2013. Vol. 38. pp. 44-51.

17. Erdnieva N.S. Ispol'zovanie sistemy ostatochnykh klassov dlya malomoshchnykh prilozheniy tsifrovoy obrabotki signalov. Inzhenernyy vestnik Dona. 2013. №2. URL: ivdon.ru/ru/magazine/archive/n2y2013/1621

18. Akushskiy I.Ya., Yuditskiy D.I. Mashinnaya arifmetika v ostatochnykh klassakh [Machine arithmetic in residual classes]. M.: Sovetskoe radio, 1968. 440 p.

19. Ding C., Pei D., Salomaa A. Chinese remainder theorem: applications in computing, coding, cryptography. Singapore: World Scientific, 1996. 214 p.

20. Goh V.T., Siddiqi M.U. Multiple error detection and correction based on redundant residue number systems. IEEE Transactions on Communications. 2008. Vol. 56. No. 3. pp. 325-330.

21. Chervyakov N.I., Shaposhnikov A.V., Ryadnov S.A. Modulyarnye parallel'nye vychislitel'nye struktury neyroprotsessornykh sistem [Modular parallel computing structures of neuroprocessor systems]. M.: Fizmatlit, 2002. 288 p.

22. Singleton, R. Maximum distance q-nary codes. IEEE Transactions on Information Theory. 1964. Vol. 10. № 2. pp. 116-118.

23. Schroeder B., Gibson G.A. Understanding disk failure rates: What does an MTTF of 1,000,000 hours mean to you? ACM Transactions on Storage (TOS). 2007. Vol. 3. № 3. pp. 8.

24. Braband J., VomHövel R., Schäbe H. Probability of failure on demand: The why and the how. Computer Safety, Reliability, and Security: Proceedings of International Conference. Berlin, Germany: Springer, 2009. pp. 46-54.

25. Rabin M.O. Efficient dispersal of information for security, load balancing, and fault tolerance. Journal of the ACM. 1989. Vol. 36. No. 2. pp. 335-348.

26. Chervyakov N.I., Berezhnoy V.V., Nazarov A.S. Sbornik tezisov Mezhdunarodnoj konferencii po myagkim vychisleniyam i izmereniyam. S.-Peterburg: Izd-vo Sankt-Peterburgskogo gosudarstvennogo elektrotekhnicheskogo universiteta «LETI» im. V.I. Ul'yanova (Lenina), 2018. pp. 372-375.